



HAL
open science

Actes des 11es Journées d'Intelligence Artificielle Fondamentale

Tiago de Lima, Bruno Zanuttini

► **To cite this version:**

Tiago de Lima, Bruno Zanuttini. Actes des 11es Journées d'Intelligence Artificielle Fondamentale. Plate-Forme Intelligence Artificielle, Association Française pour l'Intelligence Artificielle, 2017. hal-04594043

HAL Id: hal-04594043

<https://ut3-toulouseinp.hal.science/hal-04594043v1>

Submitted on 30 May 2024

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

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



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0
International License

Journées d'Intelligence Artificielle Fondamentale

Caen, du 3 au 5 juillet 2017

dans le cadre de la Plate-Forme de l'Intelligence Artificielle



Table des matières

P. Balbiani, J.-F. Condotta.

Un modèle pour raisonner sur les relations entre régions indéterminées

K. Belahcene, N. Maudet, C. Labreuche, W. Ouerdane, V. Mousseau.

An efficient SAT formulation for learning multicriteria non-compensatory sorting models

F. Belardinelli, R. Condurache, C. Dima, W. Jamroga, A.V. Jones.

Bisimulations for Verifying Strategic Abilities with an Application to ThreeBallot

S. Benferhat, A. Levray, K. Tabia.

Approximation de l'inférence MAP via les transformations probabilistes-possibilistes

F. Berreby, G. Bourgne, J.-G. Ganascia.

Cadre Déclaratif Modulaire pour Représenter et Appliquer des Principes Éthiques

P. Besnard, T. Guyet, V. Masson.

Admissible generalizations of examples as rules

P. Besnard, S. Doutre, D. Longin.

Le système Sesame : Spécifier des sémantiques argumentatives (démonstration)

E. Bonnet, S. Gaspers, A. Lambilliotte, S. Rümmele, A. Saffidine.

The Parameterized Complexity of Positional Games

E. Bonzon, J. Delobelle, S. Konieczny, N. Maudet.

Une Sémantique Graduée Paramétrique pour la Persuasion

N. Boudjani, A. Gouaich, S. Kaci.

Jeu d'apprentissage basé sur le débat pour la construction de preuves mathématiques

Q. Cohen-Solal, M. Bouzid, A. Niveau.

Séquences temporelles d'informations qualitatives : raisonner sur la topologie de régions mobiles de taille constante

M. Couceiro, N. Hug, H. Prade, G. Richard.

Fonctions préservées par analogie : une méthode pour étendre des données booléennes

V. Delcroix.

Réseau bayésien d'un critère de décision dans un problème de décision répété et multicritères

T. Dong, C. Le Duc, M. Lamolle.

Tableau-based Revision for Expressive Description Logics

S. Doutre, J.-G. Mailly.

Dynamique des systèmes d'argumentation : changement de sémantique et forçage d'extension

T. Khaled, B. Benhamou, P. Siegel.

Vers une nouvelle méthode de calcul de modèles stables et extensions en programmation logique

S. Konieczny, R. Pino Perez.

Sur la contraction itérée

J.-M. Lagniez, D. Le Berre, T. de Lima, V. Montmirail.

Un raccourci récursif pour CEGAR : Application au problème de satisfiabilité en logique modale K

J.-B. Lamy.

Les oiseaux picorant artificiels : une nouvelle méta-heuristique inspirée du comportement des pigeons

C. Leturc, G. Bonnet.

Une logique modale normale de la confiance

P. Siegel, A. Doncescu, V. Risch, S. Sené.

Représentation des systèmes dynamiques discrets booléens en logique des hypothèses

A. Wilczynski, L. Gourvès, J. Lesca.

Allocation d'objets par des échanges le long d'un réseau social

O. Gasquet, A. Herzig, D. Longin, F. Maris, M. Valais.

TouIST again... (formalisez et résolvez facilement des problèmes avec des solveurs SAT, SMT et QBF)

Un modèle pour raisonner sur les relations entre régions indéterminées

Philippe Balbiani¹Jean-François Condotta²¹Institut de recherche en informatique de Toulouse²Centre de recherche en informatique de Lens

Philippe.Balbiani@irit.fr condotta@cril.univ-artois.fr

Résumé

Lorsque nous raisonnons sur les relations entre régions dans l'espace, les frontières des entités dont nous nous occupons peuvent dans certains cas ne pas être connues avec certitude. Nous parlons alors de régions indéterminées, i.e. nous parlons de couples (a_1, a_2) de régions tels que $a_1 \subseteq a_2$. Les relations primitives entre les régions indéterminées (a_1, a_2) et (b_1, b_2) sont déterminées par les différentes possibilités qu'il y a de situer dans l'espace les régions a_1, a_2, b_1 et b_2 tout en préservant les contraintes $a_1 \subseteq a_2$ et $b_1 \subseteq b_2$. Elles sont au nombre de 46. Dans cet article, nous analysons les propriétés algébriques de ces relations primitives. Nous analysons également la complexité de la résolution de contraintes qualitatives qu'elles permettent d'exprimer.

Abstract

When we reason about the relationships between regions in space, the boundaries of the entities we deal with may in some cases not be known with certainty [7]. We are then talking about indeterminate regions, i.e. we are talking about couples (a_1, a_2) of regions such that $a_1 \subseteq a_2$. The primitive relations between the indeterminate regions (a_1, a_2) and (b_1, b_2) are determined by the different possibilities of locating the regions a_1, a_2, b_1 and b_2 while preserving the constraints $a_1 \subseteq a_2$ and $b_1 \subseteq b_2$. There are 46 of them. In this article, we analyze the algebraic properties of these primitive relations. We also analyze the complexity of the resolution of qualitative constraints that they allow one to express.

1 Introduction

Le paysagiste concevant un jardin d'agrément, l'architecte planifiant la construction d'un bâtiment, le décorateur préparant la scène d'un théâtre, tous doivent trouver une solution à un même problème : exprimer les rapports entre certaines régions de l'espace et agencer ces régions de manière à satisfaire les rapports exprimés. Quelles régions de

l'espace et quelles relations primitives entre elles doivent-ils considérer ? De nombreuses raisons les convaincront de considérer que les régions de l'espace qui les intéressent sont les fermés réguliers de l'espace des réels. C'est que, dans tout espace topologique, les fermés réguliers constituent une algèbre de Boole. Voilà pour le choix des entités primitives. Mais quid des relations primitives entre ces régions ? La première des relations qui s'impose à l'esprit est la relation d'inclusion : le parterre est situé au centre du jardin, la salle de réunion est logée au cœur du bâtiment, les tableaux sont accrochés au milieu du mur de la scène de théâtre.

La relation d'inclusion ne saurait résumer à elle seule les rapports entre fermés réguliers dans un espace topologique. Il faut parfois distinguer la relation de chevauchement — qui est vérifiée entre deux régions lorsque les ouverts qu'elles définissent ont une intersection non-vide — et la relation de contact — qui est vérifiée entre deux régions lorsqu'elles ont une intersection non-vide. La relation de contact permet, avec la relation d'égalité, de définir un jeu de 8 relations binaires constituant une partition de l'ensemble des relations possibles entre les fermés réguliers non-vides des espaces topologiques. Ces 8 relations sont les relations de l'algèbre qualitative *RCC8* et ont connu un destin plutôt singulier [13, 19, 24, 25, 26]. *RCC8* possède de nombreuses variantes (*RCC5* [15, 26], *BRCC8* [30], *Cc* et *L^c* [16, 17, 18, 28], etc) et a été appliquée à de nombreux domaines (conception architecturale [14], navigation de robot [12, 13], systèmes d'informations géographiques [9, 10], etc).

Une variante de *RCC8* à propos de laquelle peu de choses ont été écrites est la variante *Egg-Yolk* [11]. Elle a été également considérée, sous des formes diverses, par [4, 5, 8, 27, 31]. Dans cette variante, il est admis que les frontières des régions dont nous nous occupons ne peuvent pas être connues avec certitude. Aussi, les entités primiti-

ves de la variante Egg-Yolk sont-elles des couples (a_1, a_2) de régions *RCC5* tels que $a_1 \subseteq a_2$: les éléments de a_1 sont les points de l'espace qui appartiennent certainement à l'entité et les éléments de a_2 sont les points de l'espace qui appartiennent peut-être à l'entité. Les relations primitives entre des entités (a_1, a_2) et (b_1, b_2) de ce type sont déterminées par les différentes possibilités qu'il y a de situer dans l'espace les régions *RCC5* a_1, a_2, b_1 et b_2 tout en préservant les contraintes $a_1 \subseteq a_2$ et $b_1 \subseteq b_2$. Elles sont au nombre de 46. Dans cet article, nous analysons les propriétés algébriques de ces relations primitives. Nous analysons également la complexité de la résolution de contraintes qualitatives qu'elles permettent d'exprimer.

2 Algèbre des régions indéterminées

Soit X un ensemble non-vide.

2.1 Opérations sur les régions indéterminées

Une région indéterminée de X est un couple (a_1, a_2) de parties de X tel que $a_1 \subseteq a_2$: les éléments de a_1 sont les points de X qui appartiennent certainement à l'entité représentée et les éléments de a_2 sont les points de X qui appartiennent peut-être à l'entité représentée. Nous dirons qu'une région indéterminée (a_1, a_2) de X est précise ssi $a_1 = a_2$. Soit $IR^2(X)$ l'ensemble des régions indéterminées de X . Nous définissons sur $IR^2(X)$ les opérations u_X (arité 0), \downarrow_X (arité 1), \uparrow_X (arité 1), $*_X$ (arité 1), $+_X$ (arité 2) et \cdot_X (arité 2) de la façon suivante :

- $u_X = (\emptyset, X)$,
- $(a_1, a_2)^{\downarrow_X} = (a_1, a_1)$,
- $(a_1, a_2)^{\uparrow_X} = (a_2, a_2)$,
- $(a_1, a_2)^{*X} = (X \setminus a_2, X \setminus a_1)$,
- $(a_1, a_2) +_X (b_1, b_2) = (a_1 \cup b_1, a_2 \cup b_2)$,
- $(a_1, a_2) \cdot_X (b_1, b_2) = (a_1 \cap b_1, a_2 \cap b_2)$.

u_X représente la région indéterminée à propos de laquelle nous ne savons rien : aucun point de X ne lui appartient certainement et tout point de X lui appartient peut-être. Pour toute région indéterminée (a_1, a_2) de X , $(a_1, a_2)^{\downarrow_X}$ représente la région précise constituée des points de X appartenant certainement à (a_1, a_2) et $(a_1, a_2)^{\uparrow_X}$ représente la région précise constituée des points de X appartenant peut-être à (a_1, a_2) . Concernant l'opération $*_X$, sa définition répond au principe de dualité entre les adverbes "certainement" et "peut-être" : un point de X appartient certainement au complément de la région indéterminée (a_1, a_2) de X lorsqu'il n'appartient pas peut-être à (a_1, a_2) et un point de X appartient peut-être au complément de la région indéterminée (a_1, a_2) de X lorsqu'il n'appartient certainement pas à (a_1, a_2) . Enfin, pour ce qui est des opérations $+_X$ et \cdot_X , des commentaires semblables peuvent être effectués.

2.2 Algèbres concrètes

Les opérations que nous venons de définir vérifient bien-sûr un certain nombre de propriétés, mais lesquelles ? Quel type d'algèbre est la structure algébrique concrète $(IR^2(X), u_X, \downarrow_X, \uparrow_X, *_X, +_X, \cdot_X)$? Tout d'abord, puisque X est non-vide, alors

$$— u_X^{\downarrow_X} \neq u_X^{\uparrow_X}.$$

Soit $0_X = u_X^{\downarrow_X}$ et $1_X = u_X^{\uparrow_X}$. Bien entendu, $0_X = (\emptyset, \emptyset)$ et $1_X = (X, X)$. Remarquons au passage que si les régions indéterminées (a_1, a_2) et (b_1, b_2) de X sont précises alors les régions $(a_1, a_2)^{*X}$, $(a_1, a_2) +_X (b_1, b_2)$ et $(a_1, a_2) \cdot_X (b_1, b_2)$ sont précises aussi. Pour tout $(a_1, a_2) \in IR^2(X)$, soit $(a_1, a_2)^- = (a_1, a_2)^{\downarrow_X}$ et $(a_1, a_2)^+ = (a_1, a_2)^{\uparrow_X}$. Clairement, pour tout $(a_1, a_2), (b_1, b_2) \in IR^2(X)$,

- $(a_1, a_2)^{\downarrow_X \downarrow_X} = (a_1, a_2)^{\downarrow_X}$ et $(a_1, a_2)^{\downarrow_X \uparrow_X} = (a_1, a_2)^{\downarrow_X}$,
- $(a_1, a_2)^{\uparrow_X \downarrow_X} = (a_1, a_2)^{\uparrow_X}$ et $(a_1, a_2)^{\uparrow_X \uparrow_X} = (a_1, a_2)^{\uparrow_X}$,
- $(a_1, a_2)^{*X \downarrow_X} = (a_1, a_2)^{\uparrow_X *X}$ et $(a_1, a_2)^{*X \uparrow_X} = (a_1, a_2)^{\downarrow_X *X}$,
- $((a_1, a_2) +_X (b_1, b_2))^{\downarrow_X} = (a_1, a_2)^{\downarrow_X} +_X (b_1, b_2)^{\downarrow_X}$ et $((a_1, a_2) +_X (b_1, b_2))^{\uparrow_X} = (a_1, a_2)^{\uparrow_X} +_X (b_1, b_2)^{\uparrow_X}$,
- $((a_1, a_2) \cdot_X (b_1, b_2))^{\downarrow_X} = (a_1, a_2)^{\downarrow_X} \cdot_X (b_1, b_2)^{\downarrow_X}$ et $((a_1, a_2) \cdot_X (b_1, b_2))^{\uparrow_X} = (a_1, a_2)^{\uparrow_X} \cdot_X (b_1, b_2)^{\uparrow_X}$.

De plus, $(IR^2(X), +_X, \cdot_X)$ est un treillis distributif tel que pour tout $(a_1, a_2) \in IR^2(X)$,

- $(a_1, a_2) +_X 0_X = (a_1, a_2)$ et
- $(a_1, a_2) \cdot_X 1_X = (a_1, a_2)$.

Egalement, pour tout $(a_1, a_2) \in IR^2(X)$,

- $(a_1, a_2)^{\downarrow_X} +_X (a_1, a_2)^{*X \uparrow_X} = 1_X$ et $(a_1, a_2)^{\uparrow_X} +_X (a_1, a_2)^{*X \downarrow_X} = 1_X$,
- $(a_1, a_2)^{\downarrow_X} \cdot_X (a_1, a_2)^{*X \uparrow_X} = 0_X$ et $(a_1, a_2)^{\uparrow_X} \cdot_X (a_1, a_2)^{*X \downarrow_X} = 0_X$.

D'autre part, pour tout $(a_1, a_2) \in IR^2(X)$,

- $(a_1, a_2)^{\downarrow_X} +_X (a_1, a_2)^{\uparrow_X} = (a_1, a_2)^{\uparrow_X}$ et $(a_1, a_2)^{\downarrow_X} \cdot_X (a_1, a_2)^{\uparrow_X} = (a_1, a_2)^{\downarrow_X}$.

Finalement, pour tout $(a_1, a_2), (b_1, b_2) \in IR^2(X)$,

- si $(a_1, a_2)^{\downarrow_X} = (b_1, b_2)^{\downarrow_X}$ et $(a_1, a_2)^{\uparrow_X} = (b_1, b_2)^{\uparrow_X}$ alors $(a_1, a_2) = (b_1, b_2)$.

2.3 Algèbres abstraites

Soit $(A, u, \downarrow, \uparrow, *, +, \cdot)$ une structure algébrique abstraite de type $(0, 1, 1, 1, 2, 2)$. Nous dirons que $(A, u, \downarrow, \uparrow, *, +, \cdot)$ est non-dégénérée ssi

$$— u^{\downarrow} \neq u^{\uparrow}.$$

Soit $0 = u^{\downarrow}$ et $1 = u^{\uparrow}$. Pour tout $a \in A$, soit $a^- = a^{\downarrow}$ et $a^+ = a^{\uparrow}$. Nous dirons que $(A, u, \downarrow, \uparrow, *, +, \cdot)$ est fermée ssi pour tout $a, b \in A$,

$$— a^{\downarrow} = a^{\downarrow} \text{ et } a^{\downarrow \uparrow} = a^{\downarrow},$$

- $a^{\downarrow} = a^{\uparrow}$ et $a^{\uparrow\uparrow} = a^{\downarrow}$,
- $a^{*\downarrow} = a^{\uparrow*}$ et $a^{*\uparrow} = a^{\downarrow*}$,
- $(a + b)^{\downarrow} = a^{\downarrow} + b^{\downarrow}$ et $(a + b)^{\uparrow} = a^{\uparrow} + b^{\uparrow}$,
- $(a \cdot b)^{\downarrow} = a^{\downarrow} \cdot b^{\downarrow}$ et $(a \cdot b)^{\uparrow} = a^{\uparrow} \cdot b^{\uparrow}$.

Nous dirons que $(A, u, \downarrow, \uparrow, *, +, \cdot)$ est standard ssi $(A, +, \cdot)$ est un treillis distributif tel que pour tout $a \in A$,

- $a + 0 = a$ et
- $a \cdot 1 = a$.

Nous dirons que $(A, u, \downarrow, \uparrow, *, +, \cdot)$ est parfaite ssi pour tout $a \in A$,

- $a^{\downarrow} + a^{*\uparrow} = 1$ et $a^{\uparrow} + a^{*\downarrow} = 1$,
- $a^{\downarrow} \cdot a^{*\uparrow} = 0$ et $a^{\uparrow} \cdot a^{*\downarrow} = 0$.

Nous dirons que $(A, u, \downarrow, \uparrow, *, +, \cdot)$ est monotone ssi pour tout $a \in A$,

- $a^{\downarrow} + a^{\uparrow} = a^{\uparrow}$ et $a^{\downarrow} \cdot a^{\uparrow} = a^{\downarrow}$.

Nous dirons que $(A, u, \downarrow, \uparrow, *, +, \cdot)$ est adéquate ssi pour tout $a, b \in A$,

- si $a^{\downarrow} = b^{\downarrow}$ et $a^{\uparrow} = b^{\uparrow}$ alors $a = b$.

Nous dirons qu'une structure algébrique de type $(0, 1, 1, 1, 2, 2)$ est normale ssi elle est non-dégénérée, fermée, standard, parfaite, monotone et adéquate. De tout ce qui précède, il s'ensuit que pour tout ensemble non-vidé X , $(\mathcal{IR}^2(X), u_X, \downarrow_X, \uparrow_X, *_X, +_X, \cdot_X)$ est une structure algébrique normale de type $(0, 1, 1, 1, 2, 2)$.

2.4 Plongement injectif

Soit $(A, u, \downarrow, \uparrow, *, +, \cdot)$ une structure algébrique normale de type $(0, 1, 1, 1, 2, 2)$. Par conséquent, $(A, u, \downarrow, \uparrow, *, +, \cdot)$ est non-dégénérée, fermée, standard, parfaite, monotone et adéquate. Nous allons construire un ensemble X non-vidé tel que $(A, u, \downarrow, \uparrow, *, +, \cdot)$ peut être injectivement plongée dans $(\mathcal{IR}^2(X), u_X, \downarrow_X, \uparrow_X, *_X, +_X, \cdot_X)$. Pour tout $a \in A$, soit $a^- = a^{\downarrow}$ et $a^+ = a^{\uparrow}$. Soit $B = \{a^\alpha : a \in A \text{ et } \alpha \in \{-, +\}\}$. Soit 0_B (arité 0), 1_B (arité 0), $*_B$ (arité 1), $+_B$ (arité 2) et \cdot_B (arité 2) les opérations sur B définies de la façon suivante :

- $0_B = u^{\downarrow}$,
- $1_B = u^{\uparrow}$,
- $a^{*B} = a^*$,
- $a +_B b = a + b$,
- $a \cdot_B b = a \cdot b$.

Puisque $(A, u, \downarrow, \uparrow, *, +, \cdot)$ est non-dégénérée, fermée, standard et parfaite, alors $(B, 0_B, 1_B, *_B, +_B, \cdot_B)$ est une algèbre de Boole non-dégénérée. Par conséquent, pour tout $a, b \in B$, $a +_B b = b$ ssi $a \cdot_B b = a$. Nous définissons sur B la relation binaire \leq_B de la façon suivante :

- $a \leq_B b$ ssi $a +_B b = b$ et $a \cdot_B b = a$.

Puisque $(A, u, \downarrow, \uparrow, *, +, \cdot)$ est monotone, alors

Lemme 1 Pour tout $a \in A$, $a^- \leq_B a^+$.

Puisque $(A, u, \downarrow, \uparrow, *, +, \cdot)$ est adéquate, alors

Lemme 2 Pour tout $a, b \in A$, si $a^- = b^-$ et $a^+ = b^+$ alors $a = b$.

Puisque $(B, 0_B, 1_B, *_B, +_B, \cdot_B)$ est une algèbre de Boole non-dégénérée, alors \leq_B est une relation d'ordre sur B telle que pour tout $a, b \in B$, $a +_B b = \sup_{\leq_B} \{a, b\}$ et $a \cdot_B b = \inf_{\leq_B} \{a, b\}$. Nous dirons qu'une partie F de B est un B -filtre ssi pour tout $a, b \in B$,

- $1_B \in F$,
- si $a \in F$ et $a \leq_B b$ alors $b \in F$,
- si $a \in F$ et $b \in F$ alors $a \cdot_B b \in F$.

Nous dirons qu'un B -filtre F est propre ssi $0_B \notin F$. Nous dirons qu'un B -filtre F propre est premier ssi pour tout $a, b \in B$, si $a +_B b \in F$ alors $a \in F$, ou $b \in F$. Pour tout $a \in B$, soit $[a]_B = \{b \in B : a \leq_B b\}$. Il est bien connu que pour tout $a \in B$, $[a]_B$ est un B -filtre et si $a \neq 0_B$ alors $[a]_B$ est propre. De plus, pour tout $a \in B$ et pour tout B -filtre propre F , si $a \notin F$ alors il existe un B -filtre premier G tel que $a \notin G$ et $F \subseteq G$. Par conséquent, pour tout $a, b \in B$, si $a \not\leq_B b$ alors il existe un B -filtre premier F tel que $a \in F$ et $b \notin F$. Clairement,

Lemme 3 Pour tout B -filtre premier F , $0_B \notin F$ et $1_B \in F$.

Soit X_B l'ensemble de tous les B -filtres premiers. De tout ce qui précède, il s'ensuit que X_B est non-vidé. Pour tout $a \in A$ et pour tout $\alpha \in \{-, +\}$, soit $h^\alpha(a) = \{F \in X_B : a^\alpha \in F\}$. Clairement, pour tout $a \in A$ et pour tout B -filtre premier F , $a^{\downarrow\downarrow} \in F$ ssi $a^{\downarrow} \in F$, $a^{\uparrow\uparrow} \in F$ ssi $a^{\uparrow} \in F$, $a^{\downarrow\uparrow} \in F$ ssi $a^{\downarrow} \in F$ et $a^{\uparrow} \in F$ et $a^{\uparrow\downarrow} \in F$ ssi $a^{\uparrow} \in F$. Par conséquent,

Lemme 4 Pour tout $a \in A$, $h^-(a^{\downarrow}) = h^-(a)$, $h^+(a^{\downarrow}) = h^-(a)$, $h^-(a^{\uparrow}) = h^+(a)$ et $h^+(a^{\uparrow}) = h^+(a)$.

De plus, pour tout $a \in A$ et pour tout B -filtre premier F , $a^{*\downarrow} \in F$ ssi $a^{\uparrow} \notin F$ et $a^{*\uparrow} \in F$ ssi $a^{\downarrow} \notin F$. Par conséquent,

Lemme 5 Pour tout $a \in A$, $h^-(a^*) = X_B \setminus h^+(a)$ et $h^+(a^*) = X_B \setminus h^-(a)$.

Egalement, pour tout $a, b \in A$ et pour tout B -filtre premier F , $(a + b)^{\downarrow} \in F$ ssi $a^{\downarrow} \in F$, ou $b^{\downarrow} \in F$ et $(a + b)^{\uparrow} \in F$ ssi $a^{\uparrow} \in F$, ou $b^{\uparrow} \in F$. Par conséquent,

Lemme 6 Pour tout $a, b \in A$, $h^-(a + b) = h^-(a) \cup h^-(b)$ et $h^+(a + b) = h^+(a) \cup h^+(b)$.

Finalement, pour tout $a, b \in A$ et pour tout B -filtre premier F , $(a \cdot b)^{\downarrow} \in F$ ssi $a^{\downarrow} \in F$ et $b^{\downarrow} \in F$ et $(a \cdot b)^{\uparrow} \in F$ ssi $a^{\uparrow} \in F$ et $b^{\uparrow} \in F$. Par conséquent,

Lemme 7 Pour tout $a, b \in A$, $h^-(a \cdot b) = h^-(a) \cap h^-(b)$ et $h^+(a \cdot b) = h^+(a) \cap h^+(b)$.

Pour tout $a \in A$, soit $h(a) = (h^-(a), h^+(a))$.

Lemme 8 Pour tout $a \in A$, $h(a)$ est une région non-déterminée sur X_B .

Démonstration : Soit $a \in A$. Supposons que $h(a)$ n'est pas une région non-déterminée sur X_B . Par conséquent, $h^-(a) \not\subseteq h^+(a)$. Soit $F \in X_B$ tel que $F \in h^-(a)$ et $F \notin h^+(a)$. Par conséquent, $a^- \in F$ et $a^+ \notin F$. Puisque F est un B -filtre, alors $a^- \not\subseteq_B a^+$: contradiction avec le lemme 1. \dashv

Lemme 9 Pour tout $a, b \in A$, si $h(a) = h(b)$ alors $a = b$.

Démonstration : Soit $a, b \in A$. Supposons que $h(a) = h(b)$ et $a \neq b$. Par conséquent, $h^-(a) = h^-(b)$ et $h^+(a) = h^+(b)$. De plus, par le lemme 2, $a^- \neq b^-$, ou $a^+ \neq b^+$.

Cas $a^- \neq b^-$. Puisque \leq_B est une relation d'ordre sur B , alors $a^- \not\subseteq_B b^-$, ou $b^- \not\subseteq_B a^-$. Sans perte de généralité, supposons que $a^- \not\subseteq_B b^-$. Soit F est B -filtre premier tel que $a^- \in F$ et $b^- \notin F$. Par conséquent, $F \in h^-(a)$ et $F \notin h^-(b)$. Par conséquent, $h^-(a) \neq h^-(b)$: contradiction.

Cas $a^+ \neq b^+$. Semblable au cas précédent. \dashv

Lemme 10 1. $h(u) = u_{X_B}$.

2. Pour tout $a \in A$, $h(a^\downarrow) = h(a)^{\downarrow_{X_B}}$ et $h(a^\uparrow) = h(a)^{\uparrow_{X_B}}$.

3. Pour tout $a \in A$, $h(a^*) = h(a)^{*_{X_B}}$.

4. Pour tout $a, b \in A$, $h(a + b) = h(a) +_{X_B} h(b)$.

5. Pour tout $a, b \in A$, $h(a \cdot b) = h(a) \cdot_{X_B} h(b)$.

Démonstration : (1) Conséquence directe du lemme 3.

(2) Conséquence directe du lemme 4.

(3) Conséquence directe du lemme 5.

(4) Conséquence directe du lemme 6.

(5) Conséquence directe du lemme 7. \dashv

De tout ce qui précède, il s'ensuit que

Proposition 1 h est un plongement injectif de $(A, u, \downarrow, \uparrow, *, +, \cdot)$ dans $(\mathcal{IR}^2(X_B), u_{X_B}, \downarrow_{X_B}, \uparrow_{X_B}, *_{X_B}, +_{X_B}, \cdot_{X_B})$.

3 Algèbre des relations primitives

Soit X un ensemble non-vidé.

3.1 Régions qualitatives

Une région qualitative de X est une partie non-vidé a de X . Concernant les positions relatives de deux régions qualitatives, il paraît raisonnable de s'arrêter aux relations DR, PO, PP, PPI et EQ définies de la façon suivante :

- $DR(a, b)$ ssi $a \cap b = \emptyset$,
- $PO(a, b)$ ssi $a \cap b \neq \emptyset$, $a \not\subseteq b$ et $a \not\supseteq b$,
- $PP(a, b)$ ssi $a \subseteq b$ et $a \not\supseteq b$,

— $PPI(a, b)$ ssi $a \not\subseteq b$ et $a \supseteq b$,

— $EQ(a, b)$ ssi $a = b$.

Nous obtenons donc un jeu de 5 relations primitives, DR, PO, PP, PPI et EQ , parmi lesquelles EQ est la relation identité. Ce sont les 5 relations primitives de $RCC5$ [15, 26]. Soit \mathcal{A}_5 l'ensemble de ces 5 relations primitives. Quelles opérations peuvent être définies sur les éléments de \mathcal{A}_5 ? Soit $A \in \mathcal{A}_5$. Il existe une seule relation primitive $B \in \mathcal{A}_5$, dénotée A^{-1} et appelée transposition de A , pour laquelle il existe des régions qualitatives a et b telles que $B(a, b)$ et $A(b, a)$. Soit $A, B \in \mathcal{A}_5$. Il existe un plus petit ensemble $R \subseteq \mathcal{A}_5$ de relations primitives, dénoté $A \circ B$ et appelé composition de A et B , tel que pour tout $C \in \mathcal{A}_5$, s'il existe des régions qualitatives a, b et c telles que $C(a, b)$, $A(a, c)$ et $B(c, b)$ alors $C \in R$. Une partie de \mathcal{A}_5 est appelée relation complexe de \mathcal{A}_5 .

3.2 Régions indéterminées qualitatives

D'une certaine façon, la variante Egg-Yolk que nous allons étudier constitue une intégration étroite de $RCC5$ à la Wölfl et Westphal [29]. Des régions qualitatives a_1 et a_2 telles que $PP(a_1, a_2)$ définissent une région indéterminée qualitative dénotée (a_1, a_2) . Une relation atomique potentielle entre régions indéterminées qualitatives est une matrice $A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$ dont les coefficients sont dans $\{DR, PO, PP, PPI, EQ\}$. Nous dirons que la relation atomique potentielle A entre régions indéterminées qualitatives est consistante ssi il existe des régions indéterminées qualitatives (a_1, a_2) et (b_1, b_2) telles que pour tout $i, j \in \{1, 2\}$, $A_{ij}(a_i, b_j)$. 46 relations atomiques potentielles entre régions indéterminées qualitatives sont consistantes —

$$\begin{pmatrix} DR & DR \\ DR & DR \end{pmatrix}, \begin{pmatrix} DR & DR \\ DR & PO \end{pmatrix}, \begin{pmatrix} DR & DR \\ DR & PP \end{pmatrix}, \begin{pmatrix} DR & DR \\ DR & PPI \end{pmatrix}, \begin{pmatrix} DR & DR \\ DR & EQ \end{pmatrix}, \begin{pmatrix} DR & PO \\ DR & PO \end{pmatrix}, \begin{pmatrix} DR & PO \\ DR & PP \end{pmatrix}, \begin{pmatrix} DR & PO \\ DR & PPI \end{pmatrix}, \begin{pmatrix} DR & PO \\ DR & EQ \end{pmatrix}, \begin{pmatrix} DR & PP \\ DR & PP \end{pmatrix}, \begin{pmatrix} DR & PP \\ DR & PPI \end{pmatrix}, \begin{pmatrix} DR & PP \\ DR & EQ \end{pmatrix}, \begin{pmatrix} DR & PPI \\ DR & PPI \end{pmatrix}, \begin{pmatrix} DR & PPI \\ DR & EQ \end{pmatrix}, \begin{pmatrix} DR & EQ \\ DR & EQ \end{pmatrix}, \begin{pmatrix} PO & PO \\ PO & PO \end{pmatrix}, \begin{pmatrix} PO & PO \\ PO & PP \end{pmatrix}, \begin{pmatrix} PO & PO \\ PO & PPI \end{pmatrix}, \begin{pmatrix} PO & PO \\ PO & EQ \end{pmatrix}, \begin{pmatrix} PO & PP \\ PO & PP \end{pmatrix}, \begin{pmatrix} PO & PP \\ PO & PPI \end{pmatrix}, \begin{pmatrix} PO & PP \\ PO & EQ \end{pmatrix}, \begin{pmatrix} PO & PPI \\ PO & PPI \end{pmatrix}, \begin{pmatrix} PO & PPI \\ PO & EQ \end{pmatrix}, \begin{pmatrix} PP & PP \\ PP & PP \end{pmatrix}, \begin{pmatrix} PP & PP \\ PP & PPI \end{pmatrix}, \begin{pmatrix} PP & PP \\ PP & EQ \end{pmatrix}, \begin{pmatrix} PPI & PPI \\ PPI & PPI \end{pmatrix}, \begin{pmatrix} PPI & PPI \\ PPI & EQ \end{pmatrix}, \begin{pmatrix} EQ & EQ \\ EQ & EQ \end{pmatrix}, \begin{pmatrix} EQ & EQ \\ EQ & PPI \end{pmatrix}, \begin{pmatrix} EQ & EQ \\ EQ & EQ \end{pmatrix}$$

$\left(\begin{array}{cc} PP & PP \\ PPI & PP \end{array} \right), \left(\begin{array}{cc} PP & PP \\ PPI & PPI \end{array} \right), \left(\begin{array}{cc} PP & PP \\ PPI & EQ \end{array} \right),$
 $\left(\begin{array}{cc} PP & PP \\ EQ & PP \end{array} \right), \left(\begin{array}{cc} PPI & PO \\ PPI & PO \end{array} \right), \left(\begin{array}{cc} PPI & PO \\ PPI & PPI \end{array} \right),$
 $\left(\begin{array}{cc} PPI & PP \\ PPI & PO \end{array} \right), \left(\begin{array}{cc} PPI & PP \\ PPI & PP \end{array} \right), \left(\begin{array}{cc} PPI & PP \\ PPI & PPI \end{array} \right),$
 $\left(\begin{array}{cc} PPI & PP \\ PPI & EQ \end{array} \right), \left(\begin{array}{cc} PPI & PPI \\ PPI & PPI \end{array} \right), \left(\begin{array}{cc} PPI & EQ \\ PPI & PPI \end{array} \right),$
 $\left(\begin{array}{cc} EQ & PP \\ PPI & PO \end{array} \right), \left(\begin{array}{cc} EQ & PP \\ PPI & PP \end{array} \right), \left(\begin{array}{cc} EQ & PP \\ PPI & PPI \end{array} \right)$ et
 $\left(\begin{array}{cc} EQ & PP \\ PPI & EQ \end{array} \right)$ — parmi lesquelles $\left(\begin{array}{cc} EQ & PP \\ PPI & EQ \end{array} \right)$ est
la relation identité. Soit \mathcal{A}_{46} l'ensemble des relations atomiques potentielles consistantes. Soit $A \in \mathcal{A}_{46}$.

Proposition 2 Si B est une relation atomique potentielle telle que pour tout $i, j \in \{1, 2\}$, $B_{ij} = A_{ji}^{-1}$ alors B est consistante.

Démonstration : Supposons que B soit une relation atomique potentielle telle que pour tout $i, j \in \{1, 2\}$, $B_{ij} = A_{ji}^{-1}$. Puisque $A \in \mathcal{A}_{46}$, alors A est consistante. Par conséquent, il existe des régions indéterminées qualitatives (a_1, a_2) et (b_1, b_2) telles que pour tout $i, j \in \{1, 2\}$, $A_{ji}(b_j, a_i)$. Par conséquent, pour tout $i, j \in \{1, 2\}$, $A_{ji}^{-1}(a_i, b_j)$. Puisque pour tout $i, j \in \{1, 2\}$, $B_{ij} = A_{ji}^{-1}$, alors pour tout $i, j \in \{1, 2\}$, $B_{ij}(a_i, b_j)$. Par conséquent, B est consistante. \dashv

Comme le lecteur peut lui-même le constater, il existe exactement une relation atomique potentielle consistante $B \in \mathcal{A}_{46}$, dénotée A^{-1} et appelée transposition de A , telle que pour tout $i, j \in \{1, 2\}$, $B_{ij} = A_{ji}^{-1}$.

Proposition 3 A^{-1} est l'unique relation atomique potentielle consistante B pour laquelle il existe des régions indéterminées qualitatives a, b telles que $B(a, b)$ et $A(b, a)$.

Démonstration : Puisque $A \in \mathcal{A}_{46}$, alors A est consistante. Par conséquent, il existe des régions indéterminées qualitatives (a_1, a_2) et (b_1, b_2) telles que pour tout $i, j \in \{1, 2\}$, $A_{ji}(b_j, a_i)$. Par conséquent, pour tout $i, j \in \{1, 2\}$, $A_{ji}^{-1}(a_i, b_j)$. Puisque pour tout $i, j \in \{1, 2\}$, $B_{ij} = A_{ji}^{-1}$, alors pour tout $i, j \in \{1, 2\}$, $B_{ij}(a_i, b_j)$. Pour ce qui est de l'unicité, supposons que $B' \in \mathcal{A}_{46}$ soit telle qu'il existe des régions indéterminées qualitatives a' et b' telles que $B'(a', b')$ et $A(b', a')$ et $B'' \in \mathcal{A}_{46}$ soit telle qu'il existe des régions indéterminées qualitatives a'' et b'' telles que $B''(a'', b'')$ et $A(b'', a'')$. Par conséquent, pour tout $i, j \in \{1, 2\}$, $B'_{ij}(a'_i, b'_j)$ et $A_{ji}(b'_j, a'_i)$ et pour tout $i, j \in \{1, 2\}$, $B''_{ij}(a''_i, b''_j)$ et $A_{ji}(b''_j, a''_i)$. Par conséquent, pour tout $i, j \in \{1, 2\}$, $B'_{ij}(a'_i, b'_j)$ et $A_{ji}^{-1}(a'_i, b'_j)$ et pour tout $i, j \in \{1, 2\}$, $B''_{ij}(a''_i, b''_j)$ et $A_{ji}^{-1}(a''_i, b''_j)$. Par conséquent, pour tout $i, j \in \{1, 2\}$, $B'_{ij} = A_{ji}^{-1}$ et pour tout $i, j \in \{1, 2\}$, $B''_{ij} = A_{ji}^{-1}$. Par conséquent, pour tout $i, j \in \{1, 2\}$, $B'_{ij} = B''_{ij}$.

Par conséquent, $B' = B''$. \dashv

Soit $A, B \in \mathcal{A}_{46}$.

Proposition 4 Si C est une relation atomique potentielle telle que pour tout $i, j \in \{1, 2\}$, $C_{i,1} \in C_{i,2} \circ PPI$, $C_{i,2} \in C_{i,1} \circ PP$, $C_{1,j} \in PP \circ C_{2,j}$ et $C_{2,j} \in PPI \circ C_{1,j}$ et pour tout $i, j, k \in \{1, 2\}$, $C_{ij} \in A_{ik} \circ B_{kj}$ alors C est consistante.

Démonstration : Supposons que C soit une relation atomique potentielle telle que pour tout $i, j \in \{1, 2\}$, $C_{i,1} \in C_{i,2} \circ PPI$, $C_{i,2} \in C_{i,1} \circ PP$, $C_{1,j} \in PP \circ C_{2,j}$ et $C_{2,j} \in PPI \circ C_{1,j}$ et pour tout $i, j, k \in \{1, 2\}$, $C_{ij} \in A_{ik} \circ B_{kj}$. Dans ce cas, le réseau de contraintes RCC5 suivants (à 6 variables $x_1, x_2, y_1, y_2, z_1, z_2$) est chemin-consistant : $x_i A_{i,k} y_k, y_k B_{k,j} z_j, x_i C_{i,j} z_j, x_1 PP x_2, y_1 PP y_2, z_1 PP z_2$. Puisqu'il est atomique, alors il est aussi consistant. Par conséquent, la relation atomique potentielle C est consistante. \dashv

Comme le lecteur peut lui-même le constater, il existe une plus petit ensemble $R \subseteq \mathcal{A}_{46}$ de relations primitives, dénoté $A \circ B$ et appelé composition de A et B , tel que pour toute relation atomique potentielle C entre régions indéterminées qualitatives, si pour tout $i, j, k \in \{1, 2\}$, $C_{ij} \in A_{ik} \circ B_{kj}$ alors $C \in R$.

Proposition 5 $A \circ B$ le plus petit ensemble $R \subseteq \mathcal{A}_{46}$ de relations primitives tel que pour toute relation atomique potentielle C entre régions indéterminées qualitatives, s'il existe des régions indéterminées qualitatives a, b et c telles que $C(a, b)$, $A(a, c)$ et $B(c, b)$ alors $C \in R$.

Démonstration : Soit C une relation atomique potentielle entre régions indéterminées qualitatives pour laquelle il existe des régions indéterminées qualitatives a, b et c telles que $C(a, b)$, $A(a, c)$ et $B(c, b)$. Par conséquent, pour tout $i, j, k \in \{1, 2\}$, $C_{ij}(a_i, b_j)$, $A_{ik}(a_i, c_k)$ et $B_{kj}(c_k, b_j)$. Par conséquent, pour tout $i, j, k \in \{1, 2\}$, $C_{ij} \in A_{ik} \circ B_{kj}$. \dashv

Une partie de \mathcal{A}_{46} est appelée relation complexe de \mathcal{A}_{46} .

3.3 Représentations faibles

Soit \mathcal{R}_5 l'ensemble des relations complexes de \mathcal{A}_5 . Bien entendu, $\text{Card}(\mathcal{R}_5) = 2^5$. Pour tout $R, S \in \mathcal{R}_5$, nous définissons

- $0 = \emptyset$,
- $-R = \mathcal{A}_5 \setminus R$,
- $R \sqcup S = R \cup S$,
- $id = EQ$,
- $R^{-1} = \{A^{-1} : A \in R\}$,
- $R \circ S = \sqcup \{A \circ B : A \in R \text{ et } B \in S\}$.

Proposition 6 $(\mathcal{R}_5, 0, -, \sqcup, id, ^{-1}, \circ)$ est une algèbre de relations faiblement représentable, c'est-à-dire qu'il existe un ensemble U non-vide et une fonction $\varphi : \mathcal{R}_5 \rightarrow 2^{U \times U}$ tels que $\varphi(0) = \emptyset$ et pour tout $R, S \in \mathcal{R}_5$, $\varphi(-R) = 2^{U \times U} \setminus \varphi(R)$, $\varphi(R \sqcup S) = \varphi(R) \cup \varphi(S)$, $\varphi(id) = Id_U$, $\varphi(R^{-1}) = \varphi(R)^{-1}$ et $\varphi(R \circ S) \supseteq \varphi(R) \circ \varphi(S)$.

Démonstration : Soit $U = \mathbb{Z}$. Pour tout $R \in \mathcal{R}_5$, soit $\varphi(R)$ l'ensemble des couples (a, b) de régions qualitatives de \mathbb{Z} pour lesquels il existe $A \in R$ tel que $A(a, b)$. Comme le lecteur peut lui-même le constater, (U, φ) est une représentation faible de $(\mathcal{R}_5, 0, -, \sqcup, id, ^{-1}, \circ)$ telle que $\varphi(PP) \neq \emptyset$. \dashv

Soit \mathcal{R}_{46} l'ensemble des relations complexes de \mathcal{A}_{46} . Bien entendu, $Card(\mathcal{R}_{46}) = 2^{46}$. Pour tout $R, S \in \mathcal{R}_{46}$, nous définissons

- $0 = \emptyset$,
- $-R = \mathcal{A}_{46} \setminus R$,
- $R \sqcup S = R \cup S$,
- $id = \begin{pmatrix} EQ & PP \\ PPI & EQ \end{pmatrix}$,
- $R^{-1} = \{A^{-1} : A \in R\}$,
- $R \circ S = \sqcup \{A \circ B : A \in R \text{ et } B \in S\}$.

Proposition 7 $(\mathcal{R}_{46}, 0, -, \sqcup, id, ^{-1}, \circ)$ est une algèbre de relations faiblement représentable.

Démonstration : Soit (U, φ) une représentation faible de $(\mathcal{R}_5, 0, -, \sqcup, id, ^{-1}, \circ)$ telle que $\varphi(PP) \neq \emptyset$. Soit (U', φ') où $U' = \varphi(PP)$ et pour toute relation atomique potentielle consistante A , $\varphi'(A) = \{(u_1, u_2), (v_1, v_2)\} : \text{pour tout } i, j \in \{1, 2\}, (u_i, v_j) \in \varphi(A_{i,j})\}$. Alors (U', φ') est une interprétation faible de $(\mathcal{R}_{46}, 0, -, \sqcup, id, ^{-1}, \circ)$. \dashv

Voir [20] pour une étude détaillée de représentations faibles d'algèbres de relations qualitatives.

4 Réseaux de contraintes

Nous nous attaquons maintenant au problème de la résolution de réseaux de contraintes qualitatives du calcul Egg-Yolk.

4.1 Réseaux de contraintes qualitatives

Un réseau de contraintes qualitatives (RCQ) du calcul Egg-Yolk (respectivement, du calcul RCC5) est un couple $\mathcal{N} = (V, C)$ constitué d'un ensemble non-vide V de variables représentant des entités spatiales et d'une fonction C de $V \times V$ vers \mathcal{R}_{46} (respectivement, \mathcal{R}_5) définissant des agencements possibles entre les entités spatiales considérées. Nous dirons qu'un RCQ $\mathcal{N} = (V, C)$ est non-vide (respectivement, atomique) ssi pour tout $v, v' \in V$, $Card(C(v, v')) \geq 1$ (respectivement, $Card(C(v, v')) = 1$).

4.2 RCQ cohérents

Etant donné un RCQ $\mathcal{N} = (V, C)$ du calcul Egg-Yolk (respectivement, du calcul RCC5), nous dirons que \mathcal{N} est cohérent ssi il existe un ensemble non-vide X et une application α associant à chaque $v \in V$ une région indéterminée $\alpha(v)$ de X (respectivement, une région $\alpha(v)$ de X) telle que pour tout $v, v' \in V$, il existe $A \in C(v, v')$ telle que $A(\alpha(v), \alpha(v'))$. Une telle application sera appelée solution de \mathcal{N} . Un scénario S de \mathcal{N} sur l'ensemble de variables $V' \subseteq V$ est un RCQ atomique (V', C') tel que $C'(v, v') \subseteq C(v, v')$ pour tout $v, v' \in V'$. Le problème de la cohérence des réseaux de contraintes qualitatives du calcul Egg-Yolk (respectivement, du calcul RCC5) est le suivant :

— étant donné un RCQ $\mathcal{N} = (V, C)$ du calcul Egg-Yolk (respectivement, du calcul RCC5), déterminer si \mathcal{N} est cohérent ou non.

Montrons que le problème de la cohérence du calcul Egg-Yolk est, comme le problème de la cohérence de RCC5 [26], un problème NP-complet.

Proposition 8 Le problème de la cohérence des réseaux de contraintes qualitatives atomiques entre régions indéterminées est dans P .

Démonstration : Il suffit de remarquer que le problème de la cohérence des réseaux de contraintes qualitatives atomiques entre régions indéterminées qualitatives est polynomialement réductible au problème de la cohérence des réseaux de contraintes qualitatives atomiques de RCC5. Ce dernier étant dans P [15, 26], il s'ensuit que le problème de la cohérence des réseaux de contraintes qualitatives atomiques entre régions indéterminées est dans P . \dashv

Proposition 9 Le problème de la cohérence des réseaux de contraintes qualitatives entre régions indéterminées est dans NP .

Démonstration : Il suffit de remarquer que l'algorithme consistant à effectuer l'opération de choix suivante et à vérifier si le réseau de contraintes atomiques résultant est cohérent constitue une procédure non-déterministe permettant de décider le problème de la cohérence des réseaux de contraintes qualitatives entre régions indéterminées :

- pour tout $i, j \in \{1, \dots, n\}$ faire
 - choisir A dans $C(i, j)$; $C(i, j) := \{A\}$.

\dashv

Proposition 10 Le problème de la cohérence des réseaux de contraintes qualitatives entre régions indéterminées est NP -difficile.

Démonstration : Il suffit de remarquer que le problème de la cohérence des réseaux de contraintes qualitatives entre

régions est polynomialement réductible au problème de la cohérence des réseaux de contraintes qualitatives entre régions indéterminées. \dashv

4.3 Fermeture algébrique des RCQ

Nous employons le terme de sous-classe de relations pour désigner un ensemble de relations fermé pour les opérations $^{-1}$, \cap et \circ . Dans la suite de cette section, nous allons caractériser des sous-classes de relations de \mathcal{R}_{46} pour lesquels le problème de la cohérence des RCQ entre régions indéterminées qualitatives est un problème polynomial. Nous montrerons en particulier que pour certaines sous-classes de relations de \mathcal{R}_{46} la méthode polynomiale du calcul de la fermeture algébrique d'un RCQ est une méthode complète pour le problème de la cohérence. Avant de poursuivre, rappelons qu'étant donné un RCQ $\mathcal{N} = (V, C)$, \mathcal{N} est dit algébriquement clos [22] ssi pour tout $v, v', v'' \in V$, $C(v, v'') \subseteq C(v, v') \circ C(v', v'')$. Dans la littérature, les termes de RCQ chemin-cohérents, ou encore faiblement chemin-cohérents sont également utilisés. Voir [1] pour l'utilisation d'autres formes de cohérence. Etant donné un RCQ $\mathcal{N} = (V, C)$, la méthode du calcul de la fermeture algébrique consiste à calculer le plus large sous-réseau de \mathcal{N} algébriquement clos en itérant l'opération de triangulation $C(v, v') \leftarrow C(v, v') \cap (C(v, v'') \circ C(v'', v'))$ pour tout triplet de variables $v, v', v'' \in V$ jusqu'à ce qu'un point fixe soit atteint. Cette méthode peut être implantée en $O(\text{Card}(V)^3)$ [6].

4.4 Approximation des RCQ

Un RCQ entre régions indéterminées peut être approximé par un réseau de contraintes de $RCC5$ en projetant ses contraintes sur les régions composant les régions indéterminées. Formellement, ce RCQ de $RCC5$ est défini de la manière suivante. Soit $\mathcal{N} = (V = \{v_1, \dots, v_n\}, C)$, avec $n \leq 1$, un RCQ non-vide du calcul Egg-Yolk. $RCC5(\mathcal{N})$ est le réseau de contraintes qualitatives de $RCC5$ (V', C') défini par :

- pour chaque variable $v_i \in V$, avec $i \in \{1, \dots, n\}$ est introduit deux variables v_i^1 et v_i^2 dans V' correspondant aux deux régions de la région indéterminée représentée par v_i ;
- pour tout v_i^j, v_k^l avec $i, k \in \{1, \dots, n\}$ et $j, l \in \{1, 2\}$, $C'(v_i^j, v_k^l) = \{A_{jl} : A \in C(v_i, v_k)\}$.

Notons que dans le cas général, une solution du RCQ $RCC5(\mathcal{N})$ n'est pas solution du réseau de contraintes qualitatives du calcul Egg-Yolk \mathcal{N} contrairement à des cas particuliers que nous caractériserons par la suite.

Lemme 11 *Soit \mathcal{N} un RCQ entre régions indéterminées. Si \mathcal{N} est non-vide et algébriquement clos alors $RCC5(\mathcal{N})$*

est un réseau de contraintes qualitatives de $RCC5$ non-vide et algébriquement clos.

Démonstration : Supposons que $\mathcal{N} = (V, C)$ est non-vide et algébriquement clos où $V = \{v_1, \dots, v_n\}$. Par définition du RCQ $RCC5(\mathcal{N}) = (V', C')$, il est immédiat que $RCC5(\mathcal{N})$ est non-vide. Montrons que $RCC5(\mathcal{N})$ est algébriquement clos. Soit $V' = \{v_1^1, v_1^2, \dots, v_n^1, v_n^2\}$. Considérons trois variables $v_i^j, v_k^l, v_m^o \in V'$ et une relation atomique $A' \in C'(v_i^j, v_m^o)$. Montrons que $A' \in C'(v_i^j, v_k^l) \circ C'(v_k^l, v_m^o)$. Du fait que $A' \in C'(v_i^j, v_m^o)$, par définition de C' nous savons qu'il existe $A \in C(v_i, v_m)$ telle que $A' = A_{jo}$. D'autre part, comme \mathcal{N} est algébriquement clos, il existe deux relations atomiques $B \in C(v_i, v_k)$ et $C \in C(v_k, v_m)$ telles que $A \in B \circ C$. Puisque $B \in C(v_i, v_k)$ et $C \in C(v_k, v_m)$, nous avons $B_{jl} \in C'(v_i^j, v_k^l)$ et $C_{lo} \in C'(v_k^l, v_m^o)$. D'autre part, comme $A \in B \circ C$, nous pouvons affirmer que $A_{jo} \in B_{jl} \circ C_{lo}$. De tout ceci nous pouvons en déduire que $A' \in C'(v_i^j, v_k^l) \circ C'(v_k^l, v_m^o)$. Ainsi, nous pouvons conclure que $RCC5(\mathcal{N})$ est algébriquement clos. \dashv

Dans la suite, étant donné une relation $R \in \mathcal{R}_{46}$ et deux entiers $i, j \in \{1, 2\}$, R_{ij} dénotera la relation de $RCC5$ définie par $R_{ij} = \{A_{ij} : A \in R\}$. Les ensembles de relations des régions indéterminées que nous allons considérer par la suite seront construites à partir d'ensemble de relations de $RCC5$ de la manière suivante. Soit $E \subseteq \mathcal{R}_5$ un ensemble de relations des régions. E^\times est le sous-ensemble de relations de \mathcal{R}_{46} contenant l'ensemble des relations $R \in \mathcal{R}_{46}$ satisfaisant les propriétés suivantes : (1) pour tout $i, j \in \{1, 2\}$, $R_{ij} \in E$ et (2) pour toute relation atomique $A \in \mathcal{A}_{46}$ telle que $A_{ij} \in R_{ij}$ pour tout $i, j \in \{1, 2\}$, $A \in R$. Une sous-classe est dite traitable lorsque le problème de la cohérence des RCQ définis à partir des relations de cette sous-classe est un problème polynomial. Parmi les sous-classes traitables de $RCC5$, nous considérerons \mathcal{S}_5 , la plus petite sous-classe (pour l'inclusion) de $RCC5$ contenant l'ensemble des relations singletons, \mathcal{D}_5^{14} et \mathcal{D}_5^{20} les deux sous-classes étudiées dans [23] et \mathcal{H}_5 [26] la sous-classe des relations de Horn de $RCC5$ connu pour être l'unique sous-classe maximale traitable pour le problème de la cohérence incluant l'ensemble des 5 relations singletons de $RCC5$. Voir [15] pour une classification complète des classes traitables de $RCC5$. Les sous-classes \mathcal{S}_5 , \mathcal{D}_5^{14} , \mathcal{D}_5^{20} et \mathcal{H}_5 contiennent respectivement 12, 14, 20 et 27 relations sans compter la relation vide. D'autre part, nous avons $\mathcal{S}_5 \subset \mathcal{D}_5^{14} \subset \mathcal{H}_5$ et $\mathcal{S}_5 \subset \mathcal{D}_5^{20} \subset \mathcal{H}_5$. Par programme, nous pouvons déterminer que les ensembles \mathcal{S}_5^\times , $(\mathcal{D}_5^{14})^\times$, $(\mathcal{D}_5^{20})^\times$ et \mathcal{H}_5^\times contiennent respectivement 638, 826, 2560 et 6543 relations sans compter la relation vide. D'autre part, des inclusions précédentes nous avons : $\mathcal{S}_5^\times \subset (\mathcal{D}_5^{14})^\times \subset \mathcal{H}_5^\times$ et $\mathcal{S}_5^\times \subset (\mathcal{D}_5^{20})^\times \subset \mathcal{H}_5^\times$.

Exemple 1 *A titre d'illustration considérons la relation $R \in \mathcal{R}_{46}$ définie par :*

$R = \left(\begin{array}{cc} DR & PP \\ PO & PO \end{array} \right), \left(\begin{array}{cc} DR & PP \\ PPI & PO \end{array} \right), \left(\begin{array}{cc} PO & PP \\ PO & PO \end{array} \right), \left(\begin{array}{cc} PO & PP \\ PPI & PO \end{array} \right),$
 $\left(\begin{array}{cc} PPI & PP \\ PPI & PO \end{array} \right)$. R est une relation appartenant à \mathcal{S}_5^X . Nous pouvons en effet vérifier que $R_{11} = \{DR, PO, PPI\}$, $R_{12} = \{PP\}$, $R_{21} = \{PO, PPI\}$, $R_{22} = \{PO\}$ sont des relations de \mathcal{S}_5 et que pour toute relation atomique $A \in \mathcal{A}_{46}$ telle que $A_{ij} \in R_{ij}$ pour tout $i, j \in \{1, 2\}$ nous avons $A \in R$.

Etant donnée $E \subseteq \mathcal{R}_5$ un ensemble de relations de $RCC5$ contenant la relation universelle (la relation contenant les 5 relations de base) et une relation $R \in \mathcal{R}_5$, nous noterons par $E(R)$ la plus petite relation (pour l'inclusion) de E incluant R . D'autre part, pour toute relation $R \in \mathcal{R}_{46}$, $E(R)$ dénotera la relation de \mathcal{R}_{46} définie par $E(R) = \{A \in \mathcal{A}_{46} \text{ tel que } A_{ij} \in E(R_{ij}) \text{ pour tout } i, j \in \{1, 2\}\}$. Nous pouvons montrer la propriété suivante :

Lemme 12 Soit $E \subseteq \mathcal{R}_5$ un ensemble de relations des régions contenant la relation universelle. Pour tout $R \in \mathcal{R}_{46}$, $R \in E^X$ ssi $E(R) = R$.

4.5 Sous-classes traitables

Etant donné $\mathcal{N} = (V, C)$ un réseau de contraintes qualitatives de $RCC5$ (respectivement, de l'agèbre des régions indéterminées), nous dénoterons par $E(\mathcal{N})$ le réseau de contraintes (E, C') de $RCC5$ (respectivement, de l'agèbre des régions indéterminées) défini par $C'(i, j) = E(C(i, j))$ pour tout $i, j \in \{1, \dots, n\}$. D'autre part, nous serons par la suite concerné par la propriété de faible globale cohérence des RCQ qui se défini de la manière suivante [23] : un RCQ $\mathcal{N} = (V, C)$ est faiblement globalement cohérent ssi tout scénario cohérent de \mathcal{N} sur $V' \subseteq V$ peut être étendu en un scénario cohérent de \mathcal{N} sur V .

Lemme 13 Soit $E \subseteq \mathcal{R}_5$ une sous-classe de $RCC5$ contenant la relation universelle. Si tout RCQ défini sur E algébriquement clos et non vide est un RCQ faiblement globalement cohérent alors E^X est une sous-classe de \mathcal{R}_{46} .

Démonstration : Montrons que E^X est fermé pour les opérations $^{-1}$, \cap et \circ .

- Fermeture pour $^{-1}$. Soit $R \in E^X$ et considérons la relation $T = R^{-1}$. Clairement, pour tout $i, j \in \{1, 2\}$ nous avons $T_{ij} = (R^{-1})_{ij} = (R_{ji})^{-1}$. Puisque $R_{ij} \in E$ et que E est un ensemble fermé pour $^{-1}$, nous pouvons affirmer que $(R_{ji})^{-1} \in E$ pour tout $i, j \in E$. Ainsi, $T_{ij} \in E(T_{ij})$ pour tout $i, j \in E$. Par conséquent, nous pouvons affirmer que $T = E(T)$ et que donc $T = R^{-1} \in E^X$.
- Fermeture pour \cap . Soient $R, S \in E^X$ et considérons la relation $T = R \cap S$. Montrons que $E(T) = T$. Par définition de $E(T)$ il est clair que $T \subseteq E(T)$. Montrons que $E(T) \subseteq T$. Soit A une relation atomique de \mathcal{A}_{46} telle que $A \in E(T)$. Pour tout $i, j \in \{1, 2\}$, nous avons

$A_{ij} \in E(T_{ij})$. Ainsi, $A_{ij} \in E((R \cap S)_{ij}) \subseteq E(R_{ij} \cap S_{ij}) \subseteq E(R_{ij}) \cap E(S_{ij})$ pour tout $i, j \in \{1, 2\}$. Nous savons que $E(R_{ij}) = R_{ij}$ et $E(S_{ij}) = S_{ij}$. Ainsi, $A_{ij} \in R_{ij} \cap S_{ij}$ pour tout $i, j \in \{1, 2\}$. Il en résulte que $A_{ij} \in R_{ij}$ et $A_{ij} \in S_{ij}$ pour tout $i, j \in \{1, 2\}$. Ainsi, $A \in R$ et $A \in S$. Nous pouvons en conclure que $A \in (R \cap S)$.

- Fermeture pour \circ . Soient $R, S \in E^X$ et considérons la relation $T = R \circ S$. Montrons que $E(T) \subseteq R \circ S$. Considérons le réseau de contraintes qualitatives du calcul Egg-Yolk $\mathcal{N} = (V = \{v_1, v_2, v_3\}, C)$ avec $C(v_1, v_2) = R$, $C(v_2, v_3) = S$ et $C(v_1, v_3) = T$. D'après le lemme 11 nous avons $\mathcal{N}' = RCC5(\mathcal{N})$ qui est un réseau de contraintes de $RCC5$ non-vide et algébriquement clos. De cela, nous pouvons montrer que $\mathcal{N}'' = E(\mathcal{N}')$ est également un réseau de contraintes de $RCC5$ non-vide et algébriquement clos. De plus, notons que toutes les contraintes de \mathcal{N}'' sont définis par une relation appartenant à E . Du fait de la propriété posée par hypothèse sur l'ensemble E , nous pouvons affirmer que \mathcal{N}'' est un RCQ faiblement globalement cohérent et non-vide. Soit A une relation atomique de \mathcal{A}_{46} appartenant à $E(T)$. Du fait que \mathcal{N}'' est un RCQ faiblement globalement cohérent, nous pouvons affirmer que le réseau cohérent de contraintes atomiques défini sur l'ensemble de variables $\{v_1^1, v_1^2, v_3^1, v_3^2\}$ et par les contraintes $C(v_1^i, v_3^j) = A_{ij}$ pour tout $i, j \in \{1, 2\}$, peut être étendu en un scénario cohérent de \mathcal{N}'' . Ainsi, nous pouvons affirmer que $A \in E(R) \circ E(S)$. Du fait que $E(R) = R$ et $E(S) = S$, nous pouvons en déduire que $A \in R \circ S$. Par conséquent, $E(T) \subseteq T$. Il s'ensuit que $E(T) = T$. Nous pouvons en conclure que $T = R \circ S$ est une relation de E^X .

+

Long et Li [23] ont montré que tout réseau de contraintes qualitatives de $RCC5$ algébriquement-clos et défini par des relations de \mathcal{D}_5^{14} ou des relations de \mathcal{D}_5^{20} est un RCQ faiblement globalement cohérent (contrairement aux RCQ définis sur \mathcal{H}_5). A partir de cela et du lemme 13 nous pouvons en déduire la propriété suivante :

Proposition 11 Les ensembles \mathcal{S}_5^X , $(\mathcal{D}_5^{14})^X$, $(\mathcal{D}_5^{20})^X$ sont des sous-classes du calcul Egg-Yolk.

En examinant la preuve du lemme 13 nous pouvons remarquer qu'une propriété suffisante pour E^X de \mathcal{R}_{46} (avec $E \subseteq \mathcal{R}_5$) soit clos pour les opérations $^{-1}$ et \cap est que E soit également clos pour $^{-1}$ et \cap . Ainsi, nous pouvons affirmer que l'ensemble \mathcal{H}_5^X est clos pour les opérations $^{-1}$ et \cap . Néanmoins, comme le montre l'exemple suivant, \mathcal{H}_5^X n'est pas une sous-classe du fait que cet ensemble n'est pas fermé pour l'opération de composition.

Exemple 2 Considérons les relations $R, S, T \in \mathcal{R}_{46}$ définies par :

$$R = \left\{ \begin{pmatrix} EQ & PP \\ PPI & PPI \end{pmatrix} \right\}, S = \left\{ \begin{pmatrix} DR & DR \\ DR & DR \end{pmatrix}, \begin{pmatrix} DR & PP \\ PPI & EQ \end{pmatrix} \right\},$$

$$T = \left\{ \begin{pmatrix} DR & DR \\ DR & DR \end{pmatrix}, \begin{pmatrix} DR & DR \\ DR & PO \end{pmatrix}, \begin{pmatrix} DR & DR \\ PO & PO \end{pmatrix}, \begin{pmatrix} DR & DR \\ PPI & PO \end{pmatrix}, \right.$$

$$\left. \begin{pmatrix} DR & DR \\ PPI & PPI \end{pmatrix}, \begin{pmatrix} DR & PP \\ PPI & PPI \end{pmatrix} \right\}. R \text{ et } S \text{ sont deux relations appartenant à } \mathcal{H}_5^\times, \text{ tandis que } T = R \circ S \text{ n'est pas une relation de } \mathcal{H}_5^\times.$$

Lemme 14 Soit $E \subseteq \mathcal{R}_5$ un ensemble de relations de RCC5. Si tout RCQ défini sur E non vide et algébriquement clos est cohérent alors tout RCQ \mathcal{N} défini sur E^\times non vide et algébriquement clos est cohérent.

Démonstration : Soit un RCQ \mathcal{N} défini sur E^\times non vide et algébriquement clos. Soit $\mathcal{N}' = \text{RCC5}(\mathcal{N})$. Nous pouvons montrer que \mathcal{N}' est non-vidé, algébriquement clos et défini sur E . Ainsi, \mathcal{N}' est cohérent. D'autre part, nous pouvons montrer que tout solution de \mathcal{N}' est solution de \mathcal{N} . Il en résulte que \mathcal{N} est cohérent. \dashv

Lemme 15 Soit $E \subseteq \mathcal{R}_5$ une sous-classe de relations de RCC5. Si tout RCQ défini sur E non vide et algébriquement clos est faiblement globalement cohérent alors la méthode de la fermeture algébrique est complète pour le problème de la cohérence de tout RCQ entre régions indéterminées défini sur E^\times .

Démonstration : Soit $\mathcal{N} = (V, C)$ un réseau de contraintes qualitatives des régions indéterminées défini sur E^\times . La méthode de la fermeture algébrique consiste à itérer l'opération de triangulation $C(v, v') \leftarrow C(v, v') \cap (C(v, v'' \circ C(v'', v')))$ pour tout triplet de variables v, v', v'' jusqu'à ce qu'un point fixe soit atteint ou que le réseau de contraintes ne soit plus non-vidé. Dans ce deuxième cas, nous pouvons affirmer que le réseau de contraintes initial est non cohérent. Dans le premier cas, nous pouvons affirmer que le RCQ obtenu est équivalent au réseau initial, non-vidé, défini par des relations de E^\times (puisque E^\times est une sous-classe) et est algébriquement clos. Nous pouvons en conclure que ce réseau de contraintes, ainsi que le réseau initial, admettent une solution et sont donc cohérents. \dashv

A partir de ce qui précède, nous pouvons en déduire la propriété suivante :

Proposition 12 La méthode de la fermeture algébrique est complète pour le problème de la cohérence de tout RCQ défini sur $\mathcal{S}_5^\times, (\mathcal{D}_5^{14})^\times$ ou encore $(\mathcal{D}_5^{20})^\times$.

5 Conclusion

Beaucoup reste à faire.

Les parties maximales de l'algèbre qualitative RCC5 pour lesquelles le problème de la consistance de réseaux

de contraintes est dans P ont été répertoriées avec précision [15]. Dans le cadre du raisonnement sur les relations entre régions indéterminées, pouvons-nous caractériser les parties maximales de l'ensemble de nos 46 relations primitives pour lesquelles le problème de la consistance de réseaux de contraintes est dans P ?

Si nous nous intéressons maintenant à des régions (a_1, a_2) où a_1 et a_2 sont des fermés réguliers de l'espace des réels tels que $a_1 \subseteq a_2$, les relations primitives entre des entités (a_1, a_2) et (b_1, b_2) de ce type sont déterminées par les différentes possibilités qu'il y a de situer dans l'espace les fermés réguliers a_1, a_2, b_1 et b_2 tout en préservant les contraintes $a_1 \subseteq a_2$ et $b_1 \subseteq b_2$. Quelles sont ces relations primitives ? Quid du problème de la consistance de réseaux de contraintes exprimés dans ce cadre ?

Remerciements

Nous tenons à remercier les rapporteurs auxquels le comité scientifique de JIAF 2017 a fait appel et qui ont contribué à la maturation du travail que nous présentons aujourd'hui.

Références

- [1] Amaneddine, N., Condotta, J.-F., Sioutis, M. : *Efficient approach to solve the minimal labeling problem of temporal and spatial qualitative constraints*. In : Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence. AAAI Press (2013) 696–702.
- [2] Balbiani, P., Gencer, Ç. : *Admissibility and unifiability in contact logics*. In : Logic, Language, and Computation. Springer (2015) 44–60.
- [3] Balbiani, P., Tinchev, T., Vakarelov, D. : *Modal logics for region-based theories of space*. Fundamenta Informaticæ **81** (2007) 29–82.
- [4] Beaubouef, T., Petry, F. : *Vagueness in spatial data : rough set and egg-yolk approaches*. In : Engineering of Intelligent Systems. Springer-Verlag (2001) 367–373.
- [5] Beaubouef, T., Petry, F., Ladner, R. : *Spatial data methods and vague regions : a rough set approach*. Applied Soft Computing **7** (2007) 425–440.
- [6] van Beek, P. : *Approximation Algorithms for Temporal Reasoning*. In : the Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI'89), 1291–1296.
- [7] Bennett, B. : *What is a forest ? On the vagueness of certain geographic concepts*. Topoi **20** (2001) 189–201.
- [8] Clementini, E., Di Felice, P. : *Approximate topological relations*. International Journal of Approximate Reasoning **16** (1997) 173–204.

- [9] Cohn, A., Bennett, B., Gooday, J., Gotts, N. : *Representing and reasoning with qualitative spatial relations about regions*. In : *Spatial and Temporal Reasoning*. Kluwer Academic Publishers (1997) 97–134.
- [10] Cohn, A., Bennett, B., Gooday, J., Gotts, N. : *Qualitative spatial representation and reasoning with the Region Connection Calculus*. *GeoInformatica* **1** (1997) 275–316.
- [11] Cohn, A., Gotts, N. : *The ‘Egg-Yolk’ representation of regions with indeterminate boundaries*. In : *Geographic Objects with Indeterminate Boundaries*. Taylor & Francis (1996) 171–188.
- [12] Cohn, A., Hazarika, S. : *Qualitative spatial representation and reasoning : an overview*. *Fundamenta Informaticæ* **46** (2001) 1–29.
- [13] Cohn, A., Renz, J. : *Qualitative spatial representation and reasoning*. In : *Handbook of Knowledge Representation*. Elsevier (2008) 551–596.
- [14] Hois, J., Bhatt, M., Kutz, O. : *Modular ontologies for architectural design*. In : *Formal Ontologies meet Industry*. IOS Press (2009) 66–77.
- [15] Jonsson, P., Drakengren, T. : *A complete classification of tractability in RCC-5*. *Journal of Artificial Intelligence research* **6** (1997) 211–221.
- [16] Kontchakov, R., Nenov, Y., Pratt-Hartmann, I., Zakharyashev, M. : *Topological logics with connectedness over Euclidean spaces*. *ACM Transactions on Computational Logic* **14** (2013) DOI : 10.1145/2480759.2480765.
- [17] Kontchakov, R., Pratt-Hartmann, I., Wolter, F., Zakharyashev, M. : *Spatial logics with connectedness predicates*. *Logical Methods in Computer Science* **6** (2010) 1–43.
- [18] Kontchakov, R., Pratt-Hartmann, I., Zakharyashev, M. : *Interpreting topological logics over Euclidean spaces*. In : *Proceedings of the Twelfth International Conference on the Principles of Knowledge Representation and Reasoning*. AAAI Press (2010) 534–544.
- [19] Li, S., Ying, M. : *Region Connection Calculus : its model and composition table*. *Artificial Intelligence* **145** (2003) 121–146.
- [20] Ligozat, G. : *Simple models for simple calculi*. In : *Spatial Information Theory*. Springer-Verlag (1999) 173–188.
- [21] Ligozat, G., Mitra, D., Condotta, J.-F. : *Spatial and temporal reasoning : beyond Allen’s calculus*. *AI Communications* **17** (2004) 223–233.
- [22] Ligozat, G., Renz, J. : *What is a qualitative calculus ? A general framework*. In : *PRICAI 2004 : Trends in Artificial Intelligence*. Springer-Verlag (2004) 53–64.
- [23] Long, L., Li, S. : *On Distributive Subalgebras of Qualitative Spatial and Temporal Calculi*. In : *the Proceedings of the 12th International Conference Spatial Information Theory (COSIT’15)*, LNCS 9368, 354–374.
- [24] Randell, D., Cui, Z., Cohn, A. : *A spatial logic based on regions and connection*. In : *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning*. Morgan Kaufman (1992) 165–176.
- [25] Renz, J. : *Qualitative Spatial Reasoning with Topological Information*. Springer (2002).
- [26] Renz, J., Nebel, B. : *On the complexity of qualitative spatial reasoning : a maximal tractable fragment of the Region Connection Calculus*. *Artificial Intelligence* **108** (1999) 69–123.
- [27] Roy, A., Stell, J. : *Spatial relations between indeterminate regions*. *International Journal of Approximate Reasoning* **27** (2001) 205–234.
- [28] Tinchev, T., Vakarelov, D. : *Logics of space with connectedness predicates : complete axiomatizations*. In : *Advances in Modal Logic*. College Publications (2010) 434–453.
- [29] Wölfl, S., Westphal, M. : *On combinations of binary qualitative constraint calculi*. In : *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence*. AAAI Press (2009) 967–972.
- [30] Wolter, F., Zakharyashev, M. : *Spatio-temporal representation and reasoning based on RCC-8*. In : *Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning*. Morgan Kaufmann (2000) 3–14.
- [31] Worboys, M., Clementini, E. : *Integration of imperfect spatial information*. *Journal of Visual Languages and Computing* **12** (2001) 61–80.

Une formulation SAT pour l'apprentissage de modèles de classement multicritères non-compensatoires

K. Belahcène¹ C. Labreuche² N. Maudet³ V. Mousseau¹ W. Ouerdane¹

¹ LGI, CentraleSupélec, Université Paris-Saclay, Châtenay-Malabry, France

² Thales Research & Technology, 91767 Palaiseau Cedex, France

³ Sorbonne Universités, UPMC Univ Paris 06, CNRS, LIP6 UMR 7606, 75005 Paris
nom.prenom@{centralesupelec.fr, thalesgroup.com, lip6.fr}

Résumé

La littérature en aide à la décision multicritère (MCDA) offre une variété de méthodes pour affecter des alternatives, évaluées sur de multiples attributs, à des catégories ordonnées. Les modèles de tri non-compensatoire (NCS) affectent des alternatives à des catégories en les comparant à des profils délimitant ces catégories. Plusieurs travaux ont proposé des approches, basées sur un ensemble d'apprentissage, pour ajuster les paramètres d'un modèle NCS. Les approches exactes, basées sur la programmation linéaire mixte, garantissent que l'ensemble d'apprentissage soit restauré, mais peuvent difficilement prendre en charge de larges ensembles de données. Les approches heuristiques quant à elles peuvent prendre en charge de larges ensembles d'apprentissage mais n'offrent pas de garantie sur le modèle inféré. Dans ce papier, nous proposons une formulation alternative pour ajuster le modèle NCS. Cette formulation, basée sur un problème SAT, garantie de trouver le modèle le plus consistant avec l'ensemble d'apprentissage, et est computationnellement plus efficace que les approches MIP existantes.

1 Introduction

L'aide à la décision multicritères (MCDA) a pour objectif d'aider un décideur (DM) à prendre des décisions concernant des options évaluées selon différents points de vue, représentés formellement par des fonctions monotones appelées *critères*. Dans cet article, nous nous intéressons à un problème de classement ordinal, où les options doivent être affectées à des catégories, prédéfinies, et ordonnées entre elles.

Nous abordons donc un problème de classement ordinal avec un modèle non-compensatoire (NCS, cf. [1, 2]) dans lequel les catégories sont définies par un profil de limite inférieure. Ces profils spécifient les évaluations minimales

sur chaque critère pour qu'un objet soit affecté dans une catégorie au-dessus du profil considéré. En fait, une option n'a pas besoin d'être aussi bonne que le profil sur tous les critères. Un cas particulier de NCS consiste à considérer que la famille de coalitions suffisantes de critères peut être définie à l'aide de poids additifs sur les critères et d'un seuil. Ce modèle est connu sous le nom de MR-Sort (voir e.g. [4, 6]). En fait, MR-Sort et NCS sont tous deux des cas particuliers du modèle Electre Tri, une méthode pour affecter des options dans des catégories ordonnées basée sur une relation de surclassement (voir [5], pp. 389-401 or [3], pp. 381-385).

L'apprentissage de tels modèles (plus précisément, des profils définissant les catégories et des coalitions de critères suffisantes pour y être affectées) sur la base d'exemples de classement, peut être réalisée à l'aide de programmation mathématique, et certaines formulations MIP (Mixed Integer Programming) ont été proposées. Toutefois, la résolution s'avère difficile, et n'est pas envisageable pour des jeux de données issues de situations réelles (voir [4]). Dans ce cas, des approches heuristiques ont été proposées, mais dont l'optimalité ne peut plus être garantie (voir [6, 7]). Nous proposons une solution alternative, basée sur une reformulation comme un problème de satisfiabilité. Tout en garantissant l'optimalité, les résultats expérimentaux s'avèrent très prometteurs, même sur des instances de taille significative.

L'article est organisé de la manière suivante. En section 2, nous présentons les notions et concepts de base permettant d'appréhender le problème d'apprentissage d'un modèle de tri non-compensatoire. En section 3, nous présentons notre formulation SAT pour inférer un modèle NCS sur la base d'exemples d'apprentissage. Nous évaluons ensuite la pertinence et l'intérêt de cette formulation en Section 4. Nous concluons en mentionnant des perspectives intéressantes.

2 Positionnement du problème

Nous détaillons dans cette section les notions permettant de formuler le problème d'apprentissage des paramètres d'un modèle de tri non-compensatoire. En Section 2.1, nous formulons le modèle NCS et définissons ses paramètres. En Section 2.2, nous spécifions les entrées et sorties du problème d'apprentissage.

2.1 Classement non-compensatoire

Définition 1 (critères, profils, coalitions) \mathcal{N} désigne un ensemble fini de critères, chaque critère $i \in \mathcal{N}$ étant une fonction qui à une alternative x associe une valeur dans un ensemble ordonné (\mathbb{X}_i, \leq_i) , et la relation \leq_i pouvant être interprétée comme "moins bon que". Les alternatives sont donc décrites par un $|\mathcal{N}|$ -uplet de valeurs multi-critères appelé profil, et on note $\mathbb{X} = \prod_{i \in \mathcal{N}} \mathbb{X}_i$ l'ensemble de ces profils. Par analogie avec un système de vote où les critères seraient les votants, Une partie de \mathcal{N} est appelée une coalition de critères.

Dans une optique non-compensatoire, lorsque l'on est amené à comparer des alternatives, on s'intéresse à la nature des points de vue suivant lesquels l'une des alternatives peut apparaître comme étant la meilleure, sans se préoccuper de l'intensité de la préférence suivant l'un ou l'autre de ces points de vue. On définit ainsi la fonction

$$O_{\mathcal{N}} : \mathbb{X} \times \mathbb{X} \rightarrow \mathcal{P}(\mathcal{N}) \\ (x, y) \mapsto \{i \in \mathcal{N} : x_i \geq_i y_i\}$$

Lorsque $O_{\mathcal{N}}(x, y) = \mathcal{N}$, l'alternative x est au moins aussi bonne que l'alternative y sur l'ensemble des critères : on dit que x domine faiblement y au sens de Pareto. La dominance faible définit un ordre partiel sur l'ensemble des alternatives. On peut étendre cette relation d'ordre en se fondant sur le principe "comparer, puis agréger" et en élargissant l'ensemble des coalitions de critères que l'on estime suffisantes pour juger qu'une alternative est meilleure.

Définition 2 (ensemble de coalitions suffisantes) Un ensemble convenable de coalitions suffisantes de critères \mathcal{T} est une partie de $\mathcal{P}(\mathcal{N})$ close supérieurement pour l'inclusion :

$$\forall T \in \mathcal{T}, \forall S \in \mathcal{P}(\mathcal{N}), S \supseteq T \Rightarrow S \in \mathcal{T}$$

Un problème de tri ordinal consiste à affecter une alternative, décrite par un profil $x \in \mathbb{X}$, à une catégorie parmi un ensemble fini et ordonné $C_1 < \dots < C_p$. Dans une approche inductive, fondée sur les principes, on s'attache à définir une fonction de tri entre les ensembles \mathbb{X} et $C_1 < \dots < C_p$, vérifiant certaines propriétés, telles que :

- Pareto : être croissante entre \mathbb{X} , ordonné par la dominance, et $C_1 < \dots < C_p$;

- non-compensation : interdire qu'une "très bonne" valeur sur un critère ait davantage d'influence qu'une valeur "suffisante".

Le modèle de tri non compensatoire NCS fondé sur des profils proposé par [1, 2] vérifie ces deux propriétés.

Définition 3 (jeu de profils) un jeu de profils convenable b est un $(p-1)$ -uplet de profils frontières vérifiant la condition

$$\forall k < k' \in \llbracket 1, p-1 \rrbracket, \forall i \in \mathcal{N}, b_i^k \leq_i b_i^{k'}$$

Par convention, un tel jeu sera complété par deux profils : b_0 , représentant une alternative anti-idéale, dominée par toutes ; et b_p représentant une alternative idéale, qui domine toutes les autres.

Un jeu de profils convenable b définit une suite $(b^k)_{0 \leq k \leq p}$ croissante dans l'ensemble des profils \mathbb{X} ordonné par la dominance faible. Sur chaque critère $i \in \mathcal{N}$, la suite $(b_i^k)_{0 \leq k \leq p}$ est croissante dans l'ensemble \mathbb{X}_i ordonné par \leq_i , et permet de définir une suite d'intervalles emboîtés $\{x_i \in \mathbb{X}_i : x_i \geq_i b_i^k\}_{0 \leq k \leq p}$ définissant les valeurs acceptables au niveau k pour le critère i .

Définition 4 (tri non-compensatoire NCS) Soient un ensemble de critères \mathcal{N} , et un ensemble fini et totalement ordonné de catégories $C_1 < \dots < C_p$. Pour tout couple (b, \mathcal{T}) , où b désigne un jeu de profils convenable et \mathcal{T} un ensemble convenable de coalitions suffisantes, la procédure de tri $NCS_{b, \mathcal{T}}$ associe à toute alternative décrite par le profil $x \in \mathbb{X}$ la catégorie C_k telle que $O_{\mathcal{N}}(x, b^k) \in \mathcal{T}$ et $O_{\mathcal{N}}(x, b^{k+1}) \notin \mathcal{T}$

Remarque 1 Les modèles de tri non-compensatoire $NCS_{b, \mathcal{T}}$ où l'ensemble des coalitions suffisantes contiendrait la coalition vide, ou ne contiendrait pas la coalition unanime, sont dégénérés car ils ne sont pas surjectifs sur l'ensemble des catégories $C_1 < \dots < C_p$.

Exemple 1 Thierry est journaliste et prépare un comparatif de modèles automobiles pour un numéro spécial. Il considère un ensemble M de modèles populaires, qu'il souhaite ranger de manière à présenter un sous-ensemble de modèles "sélectionnés par la rédaction".

Afin d'établir cette sélection, Thierry a décidé de ne retenir que 4 critères : le coût (mesuré en euros), l'accélération (mesurée par le temps en secondes pour passer de 0 à 100km.h⁻¹, de préférence le plus faible possible), le freinage et la tenue de route (mesurés sur des échelles qualitatives, de préférence les plus élevés possibles). L'ensemble \mathbb{X} des 14 modèles est décrit par la table de performance donnée par la Table 1.

Thierry souhaite affecter à chaque modèle une catégorie parmi C_{\star} (Sans intérêt) $<$ $C_{\star\star}$ (Sélectionné) $<$ $C_{\star\star\star}$ (Recommandé). Il emploie pour ce faire un modèle de tri

M	coût	accélération ^o	freinage	tenue de route
m ₁	16 973	29	2.66	2.5
m ₂	18 342	30.7	2.33	3
m ₃	15 335	30.2	2	2.5
m ₄	18 971	28	2.33	2
m ₅	17 537	28.3	2.33	2.75
m ₆	15 131	29.7	1.66	1.75

TABLE 1 – Table de performance

M	coût	accélération ^o	freinage	tenue de route
***	≤ 15 500	≤ 28.8	≥ 2.5	≥ 2.6
b**	15 500	28.8	2.5	2.6
**	≤ 17 250	≤ 30	≥ 2.2	≥ 1.9
b*	17 250	30	2.2	1.9
*	> 17 250	> 30	< 2.2	< 1.9

TABLE 2 – Profils frontières et catégories par critère

non-compensatoire $NCS_{b,\tau}$ dont les valeurs des profils sont données par la Table 2.

En prenant en considération ces seuils, Thierry obtient une nouvelle table de performance, où chaque critère est évalué sur l'échelle $\star < \star\star < \star\star\star$:

M	coût	accélération ^o	freinage	tenue de route
m ₁	**	**	***	**
m ₂	*	*	**	***
m ₃	***	*	*	**
m ₄	*	***	**	**
m ₅	*	***	**	***
m ₆	***	**	*	*

TABLE 3 – Appréciation des performances

Ces appréciations sont ensuite agrégées en appliquant les règles suivantes :

- une alternative est ** ou *** si elle est au moins ** pour le coût ou l'accélération, et au moins ** pour le freinage ou la tenue de route ;
- une alternative est *** si elle est *** pour le coût ou l'accélération, et *** pour le freinage ou la tenue de route.

Pour être jugé globalement **, se révéler *** sur un critère ne confère pas d'avantage particulier, ainsi qu'il sied à une procédure non-compensatoire. Les différentes coalitions de critères (qui forment un hypercube de dimension 4) sont représentées sur la figure 2.

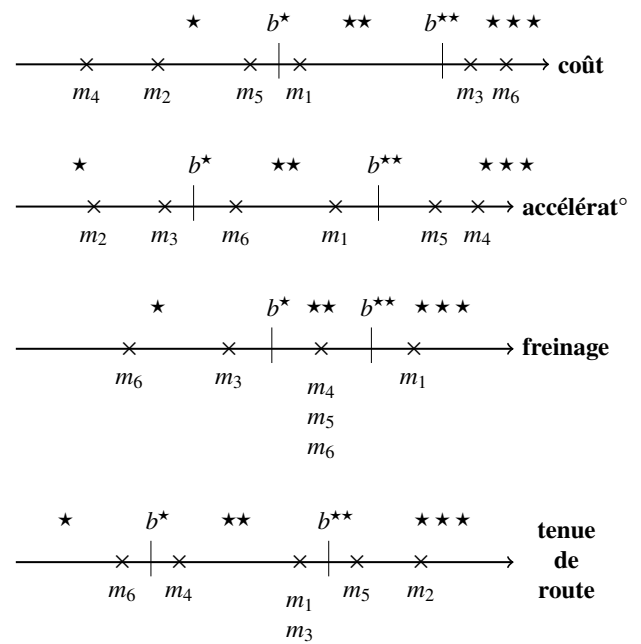


FIGURE 1 – Représentation des performances en fonction des limites des catégories

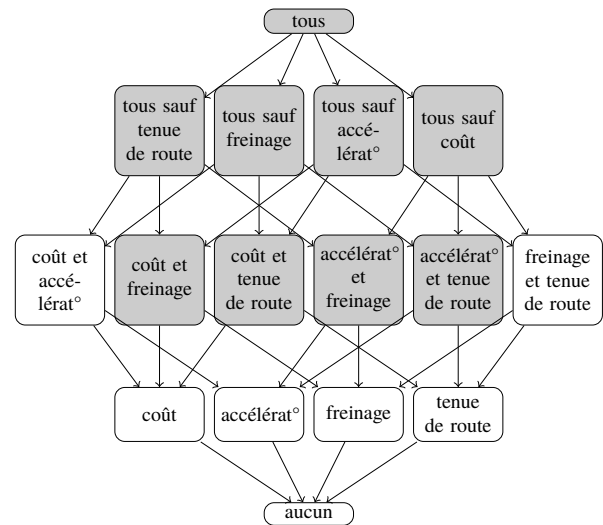


FIGURE 2 – Les coalitions de critères en gris sont suffisantes, les autres insuffisantes. Les flèches dénotent la force relative des coalitions qui diffèrent d'un seul critère, en pointant vers la plus faible.

Enfin, le modèle fournit le tri suivant :

véhicule	catégorie
m_1	★★
m_2	★
m_3	★★
m_4	★★
m_5	★★★
m_6	★

TABLE 4 – Catégories assignées par le modèle

2.2 Le paradigme de la désagregation

Dans une situation de décision donnée, en faisant l'hypothèse que les principes caractérisant le tri non-compensatoire structurent les préférences du décideur, on peut s'interroger sur la valeur des paramètres permettant de décrire au mieux le point de vue du décideur. Une option consiste à simplement lui demander de décrire du mieux qu'il peut les seuils séparant les catégories sur chacun des critères, et d'énumérer les coalitions suffisantes minimales. Afin d'obtenir ces informations de la manière rapide et fiable, autant que faire se peut, un analyste pourrait employer la stratégie fondée sur le modèle décrite dans [8]. Elle permet d'obtenir ces paramètres en se restreignant à ne demander que des informations préférentielles holistiques - est-ce qu'une alternative (fictive) peut être affectée une certaine catégorie? - et de construire le questionnaire le plus court.

Nous optons pour une démarche plus indirecte, proche du paradigme de l'apprentissage machine, où un ensemble d'affectation de référence est donné, dont on suppose qu'il décrit le point de vue du décideur. L'objectif est de prolonger ces affectations avec un modèle non-compensatoire. Dans ce contexte, nous désignons par une affectation, une application A d'un sous ensemble d'alternatives de références $\mathbb{X}^* \subset \mathbb{X}$ vers un ensemble ordonné de catégories $C_1 < \dots < C_p$. En outre, nous faisons l'hypothèse que les préférences recueillies ne sont pas perturbées par un quelconque bruit. Ces alternatives de référence font apparaître des valeurs d'un intérêt particulier sur chaque critère $i \in \mathcal{N}$, $\mathbb{X}_i^* := \bigcup_{x \in \mathbb{X}^*} x_i$. Nous recherchons alors un jeu de paramètres (b, \mathcal{T}) convenable (conformément aux définitions 2 et 3), tel que le modèle $NCS_{b, \mathcal{T}}$ associe chaque alternative de référence $x \in \mathbb{X}^*$ à la catégorie qui lui est assignée $A(x)$.

3 Une formulation SAT pour NCS

Un problème de satisfiabilité booléenne consiste en un ensemble de variables booléennes V et une proposition logique en ces variables $f : \{0, 1\}^V \rightarrow \{0, 1\}$. Une solution v^* est un antécédent de 1 par la proposition f . Un problème est dit *satisfiable* s'il admet au moins une solution, *insat-*

satisfiable dans le cas contraire. Sans perte de généralité, on peut supposer que la proposition f est écrite sous forme normale conjonctive $f = \bigwedge_{c \in \mathcal{C}} c$, où chaque *clause* $c \in \mathcal{C}$ est elle-même une disjonction en les variables ou leur négation $\forall c \in \mathcal{C}, c = \bigvee_{v \in \mathcal{C}^+} v \vee \bigvee_{v \in \mathcal{C}^-} \neg v$. Ainsi, une solution satisfait au moins une condition (positive ou négative) de chaque clause.

Les problèmes présentés ci-après font un usage immo-déré de clauses où toutes les variables apparaissent négativement, sauf une : $a \vee \neg b_1 \vee \dots \vee \neg b_n$, qui représente l'implication logique $(b_1 \wedge \dots \wedge b_n) \Rightarrow a$.

Lemme 1 Soit $A : \mathbb{X}^* \rightarrow C_1 < \dots < C_p$ une affectation prolongée par un modèle NCS paramétré par le jeu de profils b et un ensemble de coalitions suffisantes \mathcal{T} . Si $B \subseteq \mathcal{N}$ est une coalition de critère telle qu'il existe une alternative de référence $x \in \mathbb{X}^*$ qui surpasse la frontière supérieure de sa catégorie $b^{A(x)+1}$ sur chacun des critères de B , alors cette coalition n'est pas suffisante.

$$\forall B \subseteq \mathcal{N}, [\exists x \in \mathbb{X}^* : \forall i \in B, x_i \geq b_i^{A(x)+1}] \Rightarrow B \notin \mathcal{T}$$

Démonstration 1 Soit A une affectation, (b, \mathcal{T}) un couple de paramètres convenable pour le tri non-compensatoire, B une coalition de critères et x une alternative satisfaisant la prémisse, et supposons que B soit suffisante (i.e. appartenante à \mathcal{T}). L'alternative x serait au moins aussi bonne que le profil $b^{A(x)+1}$ pour une coalition suffisante de critère, et serait donc affecté à une catégorie strictement meilleure que $A(x)$. En particulier, le modèle $NCS_{b, \mathcal{T}}$ ne prolongerait pas l'affectation A .

Lemme 2 Soit $A : \mathbb{X}^* \rightarrow C_1 < \dots < C_p$ une affectation prolongée par un modèle NCS paramétré par le jeu de profils b et un ensemble de coalitions suffisantes \mathcal{T} . Si $B \subseteq \mathcal{N}$ est une coalition de critère telle qu'il existe une alternative de référence $x \in \mathbb{X}^*$ surpassée par la frontière inférieure de sa catégorie $b^{A(x)}$ sur chacun des critères de B , alors la coalition complémentaire $\mathcal{N} \setminus B$ est suffisante.

$$\forall B \subseteq \mathcal{N}, [\exists x \in \mathbb{X}^* : \forall i \in B, x_i < b_i^{A(x)}] \Rightarrow (\mathcal{N} \setminus B) \in \mathcal{T}$$

Démonstration 2 Soit A une affectation, (b, \mathcal{T}) un couple de paramètres convenable pour le tri non-compensatoire, B une coalition de critères et x une alternative satisfaisant la prémisse, et supposons que $\mathcal{N} \setminus B$ soit insuffisante (i.e. n'appartienne pas à \mathcal{T}). L'ensemble des critères pour lesquels l'alternative x serait au moins aussi bonne que le profil $b^{A(x)}$ est un sous-ensemble de $\mathcal{N} \setminus B$, et serait considéré insuffisant. Par conséquent, x se verrait affectée à une catégorie strictement moins bonne que $A(x)$ et, en particulier, le modèle $NCS_{b, \mathcal{T}}$ ne prolongerait pas l'affectation A .

Définition 5 (formulation NCS-SAT) Soient \mathcal{N} un ensemble de critères, $C_1 < \dots < C_p$ un ensemble ordonné de

catégories, \mathbb{X}^* un ensemble d'alternatives de référence. À toute affectation $A : \mathbb{X}^* \rightarrow C_1 < \dots < C_p$ on associe un problème de satisfiabilité booléenne noté $NCS - SAT(A)$. Ce problème porte sur les deux familles de variables :

- y_B indicées par une coalition de critères $B \subseteq N$;
- $z_{i,h,k}$ triplement indicées par un critère $i \in N$, une frontière entre catégories $1 \leq h \leq p-1$ et une valeur $k \in \mathbb{X}_i^*$ prise par une alternative de référence sur le critère i .

Il est décrit sous forme normale conjonctive par la conjonction des 5 familles de clauses suivantes.

1. Pour tout critère $i \in N$, pour toute frontière entre catégories adjacentes $1 \leq h \leq p-1$, pour toute paire de valeurs ordonnées $k < k' \in \mathbb{X}_i^*$:

$$z_{i,h,k'} \vee \neg z_{i,h,k} \quad (1)$$

2. Pour tout critère $i \in N$, pour toute paire ordonnée de frontières $1 \leq h < h' \leq p-1$, for toute valeur $k \in \mathbb{X}_i^*$:

$$z_{i,h,k} \vee \neg z_{i,h',k} \quad (2)$$

3. Pour toute paire ordonnée de coalitions $B \subset B' \subseteq N$:

$$z_{B'} \vee \neg z_B \quad (3)$$

4. Pour toute coalition $B \subseteq N$, pour toute frontière $1 \leq h \leq p-1$, pour toute alternative de référence immédiatement au-delà de cette frontière $x \in \mathbb{X}^*$: $A(x) = C_{h+1}$:

$$\left(\bigvee_{i \in B} z_{i,h,x_i} \right) \vee y_{N \setminus B} \quad (4)$$

5. Pour toute coalition $B \subseteq N$, pour toute frontière $2 \leq h \leq p$, pour toute alternative de référence immédiatement en-deçà de cette frontière $x \in \mathbb{X}^*$: $A(x) = C_h$:

$$\left(\bigvee_{i \in B} \neg z_{i,h,x_i} \right) \vee \neg y_B \quad (5)$$

Théorème 1 Soient une affectation $A : \mathbb{X}^* \rightarrow C_1 < \dots < C_p$ surjective, un ensemble de coalitions suffisantes \mathcal{T} et un jeu de profils b convenables pour le tri non-compensatoire. Si le modèle $NCS_{b,\mathcal{T}}$ prolonge l'affectation A , alors le couple de multiplats à composantes booléennes :

- y_B^* indicé par une coalition de critères $B \subseteq N$ et défini par $y_B = 1 \iff B \in \mathcal{T}$;
- $z_{i,h,k}^*$ triplement indicé par un critère $i \in N$, une frontière entre catégories $1 \leq h \leq p-1$ et une valeur $k \in \mathbb{X}_i^*$ prise par une alternative de référence sur le critère i , et définie par $z_{i,h,k} = 1 \iff k \geq b_i^h$;

forme une solution du problème de satisfiabilité booléenne $NCS - SAT(A)$.

Démonstration 3 Si b définit un jeu de profils convenable pour le tri non-compensatoire, alors les clauses (1) sont vérifiées car si $k < k'$ et k est au-delà du seuil, il en va de même pour k' . Les clauses (2) requièrent que, si une valeur donnée se situe au delà d'un certain seuil $b_i^{h'}$, alors elle se situe aussi au-delà des seuils inférieurs b_i^h pour $h < h'$, ce qui traduit exactement la définition 3 des profils convenables pour le tri ordinal. Les clauses (3) requièrent que, si une coalition est considérée suffisante, alors il en va de même de toute coalition qui la contient, ce qui correspond à la définition 2 d'un ensemble de coalitions suffisantes convenable pour le tri ordinal. Si le modèle $NCS_{b,\mathcal{T}}$ prolonge l'affectation A , alors les clauses (5) sont vérifiées - sinon, d'après le lemme 1, une des alternatives de référence $u \in \mathbb{X}^*$ affectée à la catégorie C_{h-1} surclasserait le profil b^h sur une coalition suffisante de critères. De même, les clauses (4) sont vérifiées, sinon, d'après le lemme 2, une alternative de référence $a \in \mathbb{X}^*$ affectée à la catégorie C_h ne surclasserait pas le profil b^h , puisque l'ensemble des critères sur lesquels a est au moins aussi bon que b^h serait inclus dans une coalition insuffisante.

Théorème 2 Soit une affectation $A : \mathbb{X}^* \rightarrow C_1 < \dots < C_p$ surjective. Si le couple de multiplats à composantes booléennes :

- y_B^* indicé par une coalition de critères $B \subseteq N$
- $z_{i,h,k}^*$ triplement indicé par un critère $i \in N$, une frontière entre catégories $1 \leq h \leq p-1$ et une valeur $k \in \mathbb{X}_i^*$ prise par une alternative de référence sur le critère i ,

forme une solution du problème de satisfiabilité booléenne $NCS - SAT(A)$, alors tout jeu de profils b tel que, pour tout critère $i \in N$ et pour toute frontière $1 \leq h \leq p-1$, pour toute valeur $k \in \mathbb{X}_i^*$ $z_{i,h,k}^* = 0 \implies k <_i b_i^h$ et $z_{i,h,k}^* = 1 \implies k \geq_i b_i^h$ et l'ensemble de coalitions de critères $\mathcal{T} := \{B \subseteq N : y_B^* = 1\}$ constituent un couple de paramètres convenables pour le tri non-compensatoire et définissent un modèle $NCS_{b,\mathcal{T}}$ qui prolonge l'affectation A .

Démonstration 4 En supposant qu'il en existe, soient y^*, z^* un couple de multiplats solution du problème $NCS - SAT(A)$, et b, \mathcal{T} un jeu de profils et un ensemble de coalitions répondant aux conditions du théorème 2.

Les clauses(2) et (3) garantissent que le jeu de profils b et l'ensemble de coalitions suffisantes \mathcal{T} conviennent pour le tri non compensatoire.

Afin de montrer que le modèle $NCS_{b,\mathcal{T}}$ prolonge l'affectation A , il reste à prouver que dans ce modèle, toute alternative de référence surclasse le profil décrivant la limite inférieure de la catégorie à laquelle est affectée sur une coalition suffisante de critères, mais pas celui décrivant la limite supérieure. Soit $x \in \mathbb{X}^*$ une telle alternative de référence.

- Concernant la limite inférieure : si $A(x) = C_1$, il n'y a rien à démontrer; sinon, on considère l'ensemble

$B := \mathcal{N} \setminus O_{\mathcal{N}}(x, b^{A(x)-1})$ qui, par construction, est exactement l'ensemble $\{i \in \mathcal{N} : z_{i, A(x)-1, x_i}^* = 0\}$. Pour la coalition B , la frontière indiquée par $(A(x)-1)$ et l'alternative de référence x , la clause (4) s'écrit $(\bigvee_{i \in B} z_{i, h, x_i}^*) \vee y_{\mathcal{N} \setminus B}^* = 1$. Le premier terme étant nul, on en déduit que $y_{\mathcal{N} \setminus B}^* = 1$, i.e. $O_{\mathcal{N}}(x, b^{A(x)}) \in \mathcal{T}$.

- Concernant la limite supérieure : si $A(x) = C_p$, il n'y a rien à démontrer; sinon, pour toute coalition $B \in \mathcal{T}$ (B n'est pas vide car A est surjective), la clause (5) concernant la coalition B , la frontière indiquée par $A(x)$ et l'alternative de référence x impose $(\bigvee_{i \in B} \neg z_{i, h, x_i}^*) \vee \neg y_B^* = 1$, avec $y_B^* = 1$, donc l'une au moins des valeurs booléennes $(z_{i, h, x_i}^*)_{i \in B}$ est nulle, et pour ce critère $i_0 \in B$ la valeur x_{i_0} est strictement inférieure au seuil $b_{i_0}^h$, ce qui assure que x ne surpasse pas b^h sur la totalité des critères de la coalition B .

3.1 Discussion

Nous proposons de décrire et d'expliquer les différentes clauses présentées dans le Théorème 1 :

1. *Échelles ascendantes* - si $k < k' \in \mathbb{X}_i^*$ et k est au dessus d'un certain seuil b_i^h , alors k' aussi. Il est nécessaire et suffisant de considérer les clauses où k et k' sont des valeurs consécutives de \mathbb{X}_i^* .
2. *Une hiérarchie de profils* - si $1 \leq h < h' \leq p-1$ et $k \in \mathbb{X}_i^*$ est au dessus du seuil $b_i^{h'}$, alors il est également au dessus de b_i^h . Il est nécessaire et suffisant de considérer les clauses où $h' = h+1$.
3. *Force des coalitions* - si une coalition $B \subseteq \mathcal{N}$ est suffisante, alors toute coalition $B' \supseteq B$ contenant B est aussi suffisante. Il est nécessaire et suffisant de considérer les clauses où la coalition B' contient exactement un critère de plus que B , correspondant aux arêtes représentées sur la figure 1.
4. *Les alternatives sont surclassées par la limite supérieure de leur classe* - fondée sur le lemme 1, cette règle permet d'inférer les coalitions insuffisantes correspondant aux points forts de l'alternative.
5. *Les alternatives surclassent la limite inférieure de leur classe* - fondée sur le lemme 2, cette règle permet d'inférer les coalitions suffisantes complémentaires aux points faibles de l'alternative.

4 Implémentation et mise en œuvre

Dans cette section, nous étudions la performance intrinsèque de la formulation proposée dans la section 3. Nous avons implémenté l'algorithme 1, en utilisant un solveur

Algorithm 1: SAT-BASED NCS DISAGGREGATION

Data: an assignment $A : \mathbb{X}^* \rightarrow C_1 < \dots < C_p$
Result: correct NCS profiles $b \in (\prod_{i \in \mathcal{N}} \mathbb{X}_i^*)^{p-1}$ and a correct NCS set of sufficient coalitions $\mathcal{T} \in \mathcal{P}(\mathcal{N})$ such that the NCS model with parameters b and \mathcal{T} extends the assignment A , or None if there is no such model.

- 1 Formulate the SAT problem with variables and clauses described in Theorem 2
 - 2 Solve it
 - 3 **if** the problem is unsat **then**
 - 4 return None
 - 5 **else**
 - 6 compute b and \mathcal{T} as detailed in Theorem 2 and return them
-

SAT, afin de résoudre les instances du problème d'apprentissage d'un modèle NCS étant donné l'affectation d'un ensemble d'alternatives de référence. Dans ce qui suit, nous commençons par décrire notre protocole expérimental, en précisant quelques détails techniques sur l'implémentation. Nous exposons ensuite les résultats de l'étude expérimentale concernant le temps de calcul de notre programme, en considérant l'influence de la taille de l'ensemble d'apprentissage, le nombre de critères et le nombre de catégories.

4.1 Protocole expérimental et détails d'implémentation

L'algorithme que nous testons prend en entrée l'affectation d'un ensemble d'alternatives \mathbb{X}^* , où chaque alternative est décrite par un vecteur de performances sur l'ensemble des critères \mathcal{N} , dans un ensemble de catégories $C_1 < \dots < C_p$. Au cœur de cet algorithme, nous résolvons un problème de satisfiabilité booléenne connu pour être NP-complet mais ayant reçu une grande attention en termes d'outils de résolution performants. Si nous ne pouvons pas nous attendre à ce que ces solveurs fonctionnent rapidement dans le pire des cas - à moins que $P = NP$ - leur performance doit être mesurée dans la pratique, en résolvant des instances du problème.

Génération du jeu de données.

Dans le cadre de ce travail, nous considérons comme données d'entrée pour l'algorithme des jeux de données générés aléatoirement et sans bruit (consistants). D'une part, l'algorithme que nous proposons n'est pas encore équipé pour prendre en charge des données bruitées, ce qui ne nous permet pas de l'alimenter avec des données décrivant des préférences réellement observées (de tel jeux de données sont utilisés afin d'évaluer les performances des dispositifs d'apprentissage automatique des préférences). D'autre

part, l'utilisation de données d'entrée complètement aléatoires et non structurées n'a pas de sens dans le contexte de la décision algorithmique. Afin d'assurer que les données préférentielles que nous utilisons font sens, nous mettons en œuvre un modèle de décision pour les générer, et en particulier un modèle compatible avec l'hypothèse non compensatoire que nous postulons. Plus précisément, nous utilisons un modèle MR Sort pour la génération de l'ensemble d'apprentissage, un modèle qui fait l'hypothèse additionnelle que les coalitions suffisantes ont une structure additive, i.e. chaque critère $i \in \mathcal{N}$ à un pouvoir de vote $w_i \in]0, +\infty[$, et une coalition $B \subseteq \mathcal{N}$ est considérée suffisante si et seulement si son pouvoir global de vote $\sum_{i \in B} w_i$ dépasse un certain seuil de majorité qualifiée $\lambda \in]0, +\infty[$.

Lors de la génération d'un jeu de données, les paramètres considérés sont : le nombre de critères $|\mathcal{N}|$, le nombre de catégories p , et le nombre d'alternatives $|\mathbb{X}^*|$. Nous considérons que tous les critères sont continus dans l'intervalle $[0, 1]$, qui est computationnellement plus contraignant pour notre algorithme que le cas où un critère est mesuré sur une échelle comportant un ensemble fini de valeurs.

Pour l'ensemble des critères, nous générons un ensemble de profils ascendants en échantillonnant uniformément $p-1$ nombres dans l'intervalle $[0, 1]$ et en les triant par ordre croissant. Nous générons les poids en échantillonnant $|\mathcal{N}|-1$ nombres dans l'intervalle $[0, 1]$, en les triant, et en les utilisant comme la somme cumulée des poids. λ est aléatoirement choisi avec une probabilité uniforme dans l'intervalle $]0.5, 1[$. Finalement, nous échantillonnons uniformément $|\mathbb{X}^*| |\mathcal{N}|$ -uplets dans l'intervalle $[0, 1]^{\mathcal{N}}$, définissant la table de performance des alternatives de référence, et nous les affectons dans des catégories $C_1 < \dots < C_p$ selon le modèle MR Sort \mathcal{M}^0 avec les paramètres générés, à savoir : les profils, les poids et le seuil de majorité.

Résolution du problème SAT.

Une fois les données générées, nous traduisons, suivant l'algorithme 1, l'affectation en un problème binaire de satisfaction décrit par un ensemble de variables et de clauses, comme décrit par la définition 5. Ce problème de satisfaction binaire est écrit dans un fichier au format DIMACS et passé en ligne de commande au solveur SAT - CryptoMiniSat 5.0.1 - gagnant de la compétition SAT 2016 et publié sous la licence MIT. Si le solveur trouve une solution, alors elle est convertie en paramètres pour le modèle NCS, tel que décrit par le théorème 2. Le modèle NCS \mathcal{M}^c fourni par le programme est alors validé avec les données d'entrée - est ce que le modèle prolonge l'affectation donnée ?

Conformément au théorème 2, nous nous attendons à ce que le solveur parvienne toujours à identifier une solution, puisque le modèle MR Sort \mathcal{M}^0 permettant de générer l'affectation fournit une instance particulière de modèle NCS prolongeant l'affectation.

Évaluation de la capacité du modèle inféré à restaurer le modèle original.

Dans le but d'apprécier la "proximité" entre le modèle calculé \mathcal{M}^c et celui à partir duquel les exemples d'affectation ont été générés \mathcal{M}^0 , nous avons procédé comme suit : nous avons échantillonné un large ensemble de n profils de performances dans $\mathbb{X} = [0, 1]^{\mathcal{N}}$ et nous avons calculé l'affectation de ces performances selon le modèle original \mathcal{M}^0 et le modèle \mathcal{M}^c retourné par l'algorithme 1. Sur cette base, nous avons calculé *err - rate*, la proportion d'"erreurs", i.e. les profils qui ne sont pas affectés à la même catégorie par les deux modèles. Pour obtenir un échantillon raisonnable pour \mathbb{X} , nous avons fait varier la taille de l'échantillon de $\mathbb{X} = [0, 1]^{\mathcal{N}}$ selon le nombre de critères $|\mathcal{N}| : n = \text{Max}(\text{Min}(4^{|\mathcal{N}|}, 3 \cdot 10^5), 10^4)$.

4.2 Résultats

Nous avons exécuté le protocole d'expérimentation, décrit précédemment, en faisant varier les différentes valeurs des paramètres gouvernant la taille du problème à résoudre. Nous avons fixé les valeurs suivantes :

- le nombre de critères $|\mathcal{N}| = 5, 7, 9, 11$;
- le nombre de catégories $p = 2, 3$;
- le nombre d'alternatives $|\mathbb{X}^*| = 25, 50, 100, 200, 400$.

Pour chaque valeur du triplet des paramètres, nous avons échantillonné 100 modèles MR Sort, et enregistré le temps de calcul (t) et le taux d'erreur (*err - rate*) sur un ordinateur portable avec le système d'exploitation Windows 7 (64 bit) et embarquant un processeur Intel Core i7-4600 cadencé à 2.1 GHz et 8 GB de RAM.

Résultats en termes de temps de calcul.

Les premiers résultats concernent le temps de calcul. Ce temps varie entre le dixième de seconde et environ une centaine de secondes, couvrant ainsi trois ordres de grandeur. Les figures 3 et 4 décrivent l'évolution du temps de calcul (représentées en utilisant une échelle logarithmique) en fonction de la taille de l'ensemble d'apprentissage (échelle logarithmique) et le nombre de critères (échelle proportionnelle), respectivement.

D'emblée, la figure 3 et la table 5 indiquent que nous sommes environ 100 fois plus rapide que l'implémentation faisant appel à une formulation en programmation linéaire mixte (MIP), décrite dans [4]. Par exemple, si nous considérons 11 critères, 3 catégories, et 100 alternatives de référence, l'exécution de l'algorithme 1 demande en moyenne 10 secondes, tandis que ce même budget temporel ne permet de traiter des problèmes avec 4 critères, 2 catégories et 20 alternatives de référence par MIP, d'après [4]. En outre, nous soulignons que le temps de calcul dans notre cas inclut à la fois la formulation du problème SAT et sa résolution (voir Algorithme 1)

Reference alternatives	#crit #categ.	11		9		7		5		5 (MIP, [4])	
		3	2	3	2	3	2	3	2	3	2
25		2.10^0	2.10^0	6.10^{-1}	4.10^{-1}	2.10^{-1}	1.10^{-1}	7.10^{-2}	5.10^{-2}	2.10^0	5.10^{-1}
50		5.10^0	4.10^0	1.10^0	9.10^{-1}	3.10^{-1}	2.10^{-1}	1.10^{-1}	7.10^{-2}	8.10^0	2.10^0
100		1.10^1	7.10^0	2.10^0	6.10^{-1}	4.10^{-1}	4.10^{-1}	1.10^{-1}	1.10^{-1}	2.10^1	4.10^1
200		2.10^1	1.10^1	5.10^0	4.10^0	1.10^0	8.10^{-1}	3.10^{-1}	2.10^{-1}	–	–
400		4.10^1	3.10^1	1.10^1	7.10^0	2.10^0	2.10^0	6.10^{-1}	4.10^{-1}	–	–

TABLE 5 – Temps de calcul (secondes)

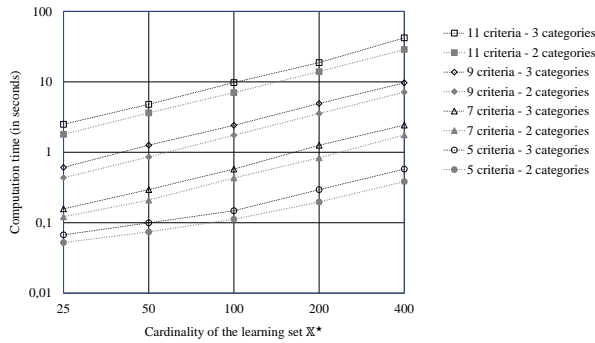


FIGURE 3 – Temps de calcul en fonction de la taille de l'ensemble d'apprentissage

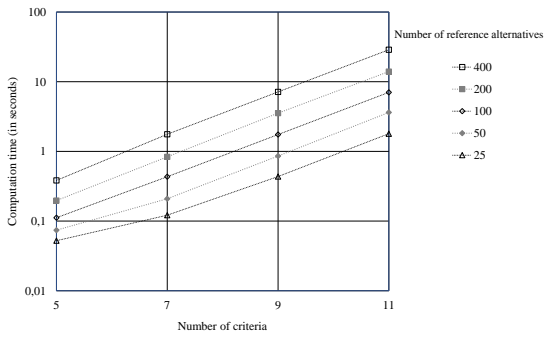


FIGURE 4 – Temps de calcul en fonction du nombre de critères

Ensuite, l'allure des différentes représentations graphiques semble indiquer une dépendance affine entre, d'une part, $\log t$ et $\log |\mathbb{X}^*|$, et, d'autre part $\log t$ et $\log |\mathcal{N}|$. Plus précisément, les coefficients permettant un ajustement affine semblent compatibles avec une loi

$$t \propto |\mathbb{X}^*| \times 2^{|\mathcal{N}|},$$

où le temps de calcul est proportionnel au nombre d'alternatives de référence et au nombre de coalitions de critères (ce

qui correspond à l'ordre de grandeur du nombre de clauses dans le problème NCS-SAT).

En guise de synthèse, nous proposons une formulation permettant de prévoir le temps de calcul nécessaire pour résoudre une quelconque instance : compter 10 secondes pour 11 critères, 3 catégories et 100 alternatives de référence ; puis doubler pour chaque critère supplémentaire, ou lorsque le nombre d'exemples d'apprentissage double.

Résultats sur la capacité du modèle inféré à restaurer le modèle original.

Les tables 7 (pour 2 catégories) et 6 (pour 3 catégories) indiquent le taux d'erreur en fonction du nombre de critères et la taille de l'ensemble d'apprentissage. Ces résultats sont résumés dans les figures 5 et 6. Sans surprise, les résultats obéissent aux principes selon lesquels un modèle avec un plus grand degré de liberté est plus difficile à déterminer - ainsi les taux d'erreur augmentent avec le nombre de critères ou le nombre de catégories. Inversement, la taille de l'ensemble d'alternatives de références, où l'on s'assure que les modèles \mathcal{M}^0 et \mathcal{M}^c coïncident, représente l'effort fourni - en termes de collecte d'information - pour aligner les deux modèles. En toute logique, on observe que le taux d'erreur diminue avec la taille de l'ensemble d'apprentissage.

Reference alternatives	5 crit.	7 crit.	9 crit.	11 crit.	5 (MIP, [4])
25	17%	20%	25%	25%	22%
50	10%	15%	20%	21%	15%
100	5%	8%	12%	18%	7%
200	2%	4%	8%	12%	–
400	1%	2%	4%	6%	–

TABLE 6 – Taux d'erreur – 3 catégories

La comparaison entre la première et la dernière colonne des tables 7 ou 6 révèle que la performance de l'algorithme 1, basé sur la formulation SAT qui sélectionne aléatoirement un modèle NCS parmi l'ensemble des modèles compatibles, n'est pas mauvaise et plutôt similaire à la performance de l'implémentation MIP proposée par [4], qui sélectionne le

MR Sort compatible en optimisant une certaine fonction d'erreur.

5 Conclusion et perspectives

Reference alternatives	5 crit.	7 crit.	9 crit.	11 crit.	5 (MIP, [4])
25	10%	11%	13%	11%	12%
50	5%	9%	10%	11%	6%
100	3%	5%	9%	10%	3%
200	2%	3%	5%	6%	–
400	1%	2%	3%	5%	–

TABLE 7 – Taux d'erreur –2 catégories

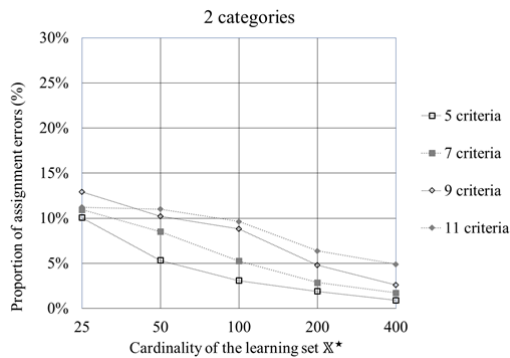


FIGURE 5 – Erreurs d'affectation en fonction de la taille de l'ensemble d'apprentissage – 2 catégories

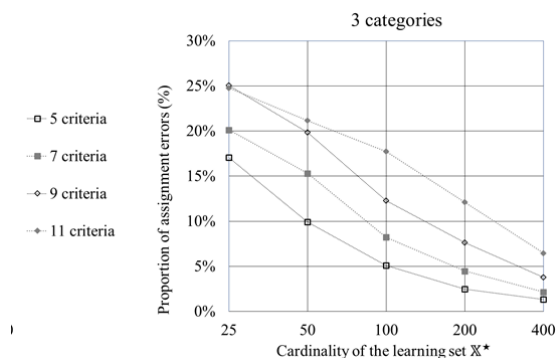


FIGURE 6 – Erreurs d'affectation en fonction de la taille de l'ensemble d'apprentissage –3 catégories

Dans ce papier, nous avons considéré le modèle de tri multicritère non-compensatoire [1, 2], et nous avons proposé une nouvelle formulation comme un problème de satisfiabilité, permettant d'inférer ce modèle à partir d'un ensemble d'apprentissage fourni par un décideur. La question de l'apprentissage de ce modèle a déjà été abordée dans la littérature et résolue par diverses solutions, en particulier la programmation mathématique en variables mixtes [4], ainsi qu'une heuristique spécifique [6, 7]. En raison du temps de calcul élevé, la formulation exacte ne peut être appliquée qu'à des ensembles d'apprentissage de taille limitée. Les heuristiques quant à elles peuvent traiter de larges ensembles de données, mais ne garantissent pas de trouver le modèle compatible avec l'ensemble d'apprentissage. Notre nouvel algorithme fournit une telle garantie. Nous avons implémenté et testé notre formulation, celle-ci surpasse les approches exactes en termes de temps de calcul (réduction du temps de calcul par un facteur 100).

Par ailleurs, cette bonne performance en termes de calcul aurait pu être contrebalancée par des capacités limitées du modèle inféré dans le cadre de la généralisation. En effet, l'approche MIP se préoccupe de trouver le modèle le plus représentatif parmi l'ensemble des modèles compatibles (à travers l'utilisation d'une fonction objectif), tandis que notre approche SAT renvoie le premier modèle compatible trouvé. Nos expérimentations ont montré que les approches MIP et SAT ont des performances similaires en termes de généralisation. De ce fait, nous croyons que cet algorithme apporte une forte valeur ajoutée aux techniques d'apprentissage des modèles de tri non-compensatoire basées sur des ensembles d'apprentissage, et particulièrement lorsque ces ensembles sont de grande taille.

Notre travail laisse des questions ouvertes et offre également de nouvelles perspectives. Premièrement, l'algorithme proposé est bien adapté pour être intégré dans un processus interactif. Ce dernier permettra au décideur d'élucider interactivement le modèle de tri non-compensatoire sur la base d'un ensemble d'apprentissage qui est défini de manière incrémentale (voire d'autres informations préférentielles). L'exploration de l'ensemble des modèles possibles (ALL-SAT) mérite d'être étudiée. Deuxièmement, ce travail considère que l'ensemble d'apprentissage est consistant (sans bruit), et une extension pour inclure l'inconsistance peut être envisagée. L'étude d'une formulation MAX-SAT pourrait être intéressante dans ce cadre. Enfin, il serait pertinent de considérer une telle formulation dans la problématique de construction d'explications pour des recommandations répondant à des problèmes de tri multi-critères.

Références

- [1] D. Bouyssou and T. Marchant. An axiomatic approach to noncompensatory sorting methods in MCDM, I: The case of two categories. *European Journal of Operational Research*, 178(1) :217—245, 2007.
- [2] D. Bouyssou and T. Marchant. An axiomatic approach to noncompensatory sorting methods in MCDM, II : More than two categories. *European Journal of Operational Research*, 178(1) :246—276, 2007.
- [3] D. Bouyssou, T. Marchant, M. Pirlot, A. Tsoukiàs, and Ph. Vincke. Evaluation and decision models with multiple criteria : Stepping stones for the analyst. *International Series in Operations Research and Management Science*, Volume 86. Springer, Boston, 1st edition, 2006.
- [4] A. Leroy, V. Mousseau, and M. Pirlot. Learning the parameters of a multiple criteria sorting method. In R. Brafman, F. Roberts, and A. Tsoukiàs, editors, *Algorithmic Decision Theory*, volume 6992 of *Lecture Notes in Artificial Intelligence*, pages 219—233. Springer, 2011.
- [5] B. Roy and D. Bouyssou. *Aide multicritère à la décision : méthodes et cas*. Economica, Paris, 1993.
- [6] O. Sobrie, V. Mousseau, and M. Pirlot. Learning a majority rule model from large sets of assignment examples. In P. Perny, M. Pirlot, and A. Tsoukiàs, editors, *Algorithmic Decision Theory*, pages 336—350, Brussels, Belgium, 2013. Springer.
- [7] O. Sobrie, V. Mousseau and M. Pirlot. Learning the Parameters of a Non Compensatory Sorting Model. In *Algorithmic Decision Theory, ADT 2015*, pages 153—170, 2015.
- [8] K. Belahcène, V. Mousseau, M. Pirlot and O. Sobrie. Preference Elicitation and Learning in a Multiple Criteria Decision Aid perspective. Research report 2017-02, LGI CentraleSupélec, <http://www.lgi.ecp.fr/Biblio/PDF/CR-LGI-2017-02.pdf>, 2017.

Bisimulations for Verifying Strategic Abilities with an Application to ThreeBallot

F. Belardinelli¹ R. Condurache² C. Dima² W. Jamroga³ A. V. Jones⁴

¹ IBISC, Université d'Evry & IRIT Toulouse, France

² LACL, Université Paris-Est Créteil, France

³ Institute of Computer Science, Polish Academy of Sciences

⁴ Vector Software, Inc., London, United Kingdom

préciser les mails

Résumé

Dans cet article nous proposons une notion de bisimulation alternée des capacités stratégiques avec information imparfaite. La bisimulation préserve les formules d'ATL pour les variantes *objective* et *subjective* de la sémantique basée sur états avec information imparfaite, qui est couramment utilisé dans la modélisation et la vérification des systèmes multi-agents. De plus, nous appliquons ce résultat théorique à la vérification de la résistance à la coercition dans le système de vote dit three-ballot, un protocole qui n'utilise pas la cryptographie. En particulier, nous démontrons que des simplifications naturelles du modèle initial du protocole sont en fait des bisimulations du modèle original, et donc elles satisfont les mêmes propriétés dans ATL, y compris la résistance à la coercition. Ces simplifications permettent la terminaison de la vérification par l'outil *model checker* MCMAS sur des modèles avec un plus grand nombre d'électeurs et de candidats par rapport au modèle initial. Cet article a été accepté à AAMAS2017.

Abstract

We propose a notion of alternating bisimulation for strategic abilities under imperfect information. The bisimulation preserves formulas of ATL for both the *objective* and *subjective* variants of the state-based semantics with imperfect information, which are commonly used in the modeling and verification of multi-agent systems. Furthermore, we apply the theoretical result to the verification of coercion-resistance in the three-ballot voting system, a voting protocol that does not use cryptography. In particular, we show that natural simplifications of an initial model of the protocol are in fact bisimulations of the original model, and therefore satisfy the same ATL properties, including coercion-resistance. These simplifications allow the model-checking tool MCMAS to terminate on models with a larger number of voters and candidates, compared with the ini-

tial model. This paper has been accepted for presentation at AAMAS2017.

1 Introduction

The realm of formal languages for expressing strategic abilities of rational agents has witnessed a steady growth in recent years [8, 9, 24]. Among the most significant contributions we mention alternating-time temporal logic [3], strategy logic [13, 33], coalition logic [38]. These languages include modal operators, indexed to coalitions $A \subseteq Ag$ of agents, to express that the agents in A have a strategy to enforce a certain outcome, regardless of the behavior of the agents in $Ag \setminus A$. These syntactical features allow us to express winning conditions in multi-player games, notions of equilibrium (e.g. Nash), strategy-proofness [13, 34].

However, if these logics for strategies are to be applied to the specification and verification of multi-agent systems [22, 28, 31], they need to be coupled with efficient model checking techniques. Unfortunately, while in contexts of perfect information we benefit from tractable algorithms for model checking [3], the situation is rather different once we consider imperfect information. In contexts of imperfect information the complexity of the verification task ranges between Δ_2^P -completeness to undecidability, depending on whether we allow for perfect recall [20, 26]. In this setting it is crucial to develop complementary model checking techniques, in order to make the problem amenable.

In this line of research abstractions have proved to be a valuable tool for efficient verification [14, 15]. In this approach the concrete system S to be verified is abstracted into a "simpler" model S^A , which typically contains "less" transitions and therefore is "easier" to check in principle.

Then, the verification result is transferred from the abstract S^A to the concrete S by virtue of some preservation result. Normally, preservation is guaranteed by proving that the abstract S^A is *(bi)similar* to S . (Bi)simulations are a powerful tool to analyze the expressiveness of modal languages, starting with van Benthem’s result on modal logic as the bisimulation-invariant fragment of first-order logic [6]. However, (bi)simulations are a lot less understood in logics for strategies, where they have been studied mostly for contexts of perfect information [4, 23, 2].

In this paper we advance the state-of-the-art by introducing (bi)simulations for alternating-time temporal logic (ATL) under imperfect information. We prove that these (bi)simulations preserve the interpretation of formulas in ATL, when interpreted with imperfect information and imperfect recall, for both the objective and subjective semantics [8, 9]. Most interestingly for MAS verification, we apply these (bi)simulations to the abstraction of a class of electronic voting protocols without encryption.

Electronic voting has increasingly been considered as a robust alternative to paper-based voting due to a number of advantages it offers : accessibility, availability, voter turnout, less expensive and easier to use than paper voting, faster and more accurate ballot counting and results. However, electronic voting poses a number of challenges, some of which are common also to paper voting, but in a more technological setting : resistance and resilience to coercion and other types of fraud, secrecy, anonymity, verifiability, democracy (the right to vote at most once), accountability. Other issues are specific to electronic voting : access to internet, privatization, as well as public understanding and trust [41].

An increasing amount of research has focused recently on the verification of many of these properties for various types of voting protocols [5, 16]. The frameworks used for modeling and verifying security properties of voting protocols include, to mention only a few, process calculi such as the *applied π -calculus* or CSP [18, 25, 43], rewriting-based approaches [11, 19, 7], approaches based on flat transition systems etc.

Here we develop a verification procedure for voting protocols that is based on a multi-agent logics approach. The main advantage of an approach based on multi-agent logics is the provision of a unified specification language for a variety of properties. A simple example is the variety of english statements of (non-probabilistic) coercion resistance that is around in the literature, which are usually implemented as behavioral equivalence properties involving some process algebraic model of the system [16]. However such approaches do not make it clear what is the system model and what is the property to be verified on the system. Multi-agent logics allow a clear separation of these two, as well as a wider variety of properties, involving the existence of attacker strategies. Our results, while only preliminary and

addressing a simplified version of the Three Ballot protocol [40], allow the verification of systems with an increasing number of voters and candidates when compared with the approach based on process calculi from [35, 36].

Scheme of the Paper. In Section 2 we introduce the syntax and semantics of ATL interpreted under imperfect information and imperfect recall. In Section 3 we define (bi)simulation relations in this setting and prove that they preserve the interpretation of formulas in ATL. Then, in Section 4 we present the three-ballot voting protocol and formalize it as a game structure. In particular, we provide two abstractions of the three-ballot voting protocol and show that all systems are indeed bisimilar. Finally, in Section 5 we evaluate the gains in verification time and resources of model checking these abstractions in comparison to the original model. We conclude in Section 6 by discussing related works and by pointing to future directions of research.

This paper has been accepted for presentation at the 16th International Conference on Autonomous Agents and Multi-agent Systems (AAMAS2017).

2 The Formal Setting

In this section we introduce the syntax of ATL and its semantics defined on concurrent games structures with imperfect information. The following definitions and notation are taken from [20]. Concurrent game structures have been introduced in [3] in a perfect information setting. Here we consider their version for contexts of imperfect information [27].

Definition 1 A concurrent game structure with imperfect information, or *iCGS*, is a tuple $\mathcal{G} = \langle Ag, AP, S, s_0, \{\sim_i\}_{i \in Ag}, Act, d, \rightarrow, \pi \rangle$, where

- Ag is a nonempty and finite set of agents. Subsets $A \subseteq Ag$ of agents are called groups.
- S is a non-empty set of states and $s_0 \in S$ is the initial state of \mathcal{G} .
- For each agent $i \in Ag$, \sim_i is an equivalence relation on S , called the indistinguishability relation for i .
- Act is a finite non-empty set of actions. A tuple $\vec{a} = (a_i)_{i \in Ag} \in Act^{Ag}$ is called a joint action.
- $d : Ag \times S \rightarrow (2^{Act} \setminus \{\emptyset\})$ is the protocol function. For every $i \in Ag$, $d(i)$ returns the set of actions available to agent i at each state. Protocol d satisfies the property that, for all states $s, s' \in S$ and any agent i , $s \sim_i s'$ implies $d(i, s) = d(i, s')$, that is, the same actions are available to agent i in indistinguishable states.
- $\rightarrow \subseteq S \times Act^{Ag} \times S$ is the transition relation such that, for every state $s \in S$ and joint action $\vec{a} \in Act^{Ag}$. We write $s \xrightarrow{\vec{a}} r$ for $(s, \vec{a}, r) \in \rightarrow$. Moreover, $s \xrightarrow{\vec{a}} r$ only if $a_i \in d(i, s)$ for every agent $i \in Ag$.

— AP is a set of atomic propositions and $\pi : S \rightarrow 2^{AP}$ is the state-labeling function.

By Def. 1 in a given state s , each agent $i \in Ag$ can perform the enabled actions in $d(i, s)$. A joint action \vec{a} fires a transition from state s to some state s' only if each a_i is enabled for agent i in s . Further, each agent i is equipped with an indistinguishability relation \sim_i , with $s \sim_i s'$ meaning that i cannot tell state s from state s' , i.e., agent i possesses the same information in the two states. In particular, the same actions are enabled in indistinguishable states.

Given an iCGS \mathcal{G} as above, a *run* is a finite or infinite sequence $\lambda = s_0 \vec{a}_0 s_1 \dots$ in $((S \cdot Act^{Ag})^* \cdot S) \cup (S \cdot Act^{Ag})^\omega$ such that for every $j \geq 0$, $s_j \xrightarrow{\vec{a}_j} s_{j+1}$. Given a run $\lambda = s_0 \vec{a}_0 s_1 \dots$ and $j \geq 0$, $\lambda[j]$ denotes the $j+1$ -th state s_j in the sequence. For a group $A \subseteq Ag$ of agents, a *joint A-action* denotes a tuple $\vec{a}_A = (a_i)_{i \in A} \in Act^A$ of actions, one for each agent in A . For groups $A \subseteq B \subseteq Ag$ of agents, a joint A-action \vec{a}_A is *extended* by a joint B-action \vec{b}_B , denoted $\vec{a}_A \sqsubseteq \vec{b}_B$, if for every $i \in A$, $a_i = b_i$. Also, a joint A-action \vec{a}_A is *enabled* at state $s \in S$ if for each agent $i \in A$, $(a_A)_i \in d(i, s)$.

We now introduce a notion of strategy adapted to iCGS with imperfect information [27].

Definition 2 A (uniform) strategy for an agent $i \in Ag$ is a function $\sigma : S \rightarrow Act$ that is compatible with d and \sim_i , i.e.,

- for every state $s \in S$, $\sigma(s) \in d(i, s)$;
- for all states $s, r \in S$, $s \sim_i r$ implies $\sigma(s) = \sigma(r)$.

By Def. 2 a strategy in an iCGS has to be uniform in the sense that in indistinguishable states it must return the same action. Such strategies are also known as *observational* in the literature on game theory. Note that in this paper we use memoryless strategies, whereby only the current state determines the action to perform. This choice is dictated by the application in hand, namely voting protocols, in which each agent's memory is encoded in the agent's state¹. Perfect recall strategies with imperfect information can be defined similarly, as memoryless strategies on tree unfoldings of iCGS. We leave this extension for future work.

A strategy for a group A of agents is a family $\sigma_A = \{\sigma_a \mid a \in A\}$ of strategies, one for each agent in A . Given groups $A \subseteq B \subseteq Ag$, a strategy σ_A for group A , a state $s \in S$, and a joint B-action $\vec{b}_B \in Act^B$ that is enabled at s , we say that \vec{b}_B is *compatible* with σ_A (in s) whenever $\sigma_A(s) \sqsubseteq \vec{b}_B$. For states $s, r \in S$ and strategy σ_A , we denote $s \xrightarrow{\sigma_A(s)} r$ if $s \xrightarrow{\vec{a}} r$ for some joint action $\vec{a} \in Act^{Ag}$ that is compatible with σ_A .

We define two notions of *outcomes* of strategy σ_A at state s , corresponding to the *objective* and *subjective* interpretation of ATL operators. Fix a state s and a strategy σ_A for group A .

1. Therefore memoryless strategies already encode the agent's memory of all her past observations.

1. The set of *objective outcomes* of σ_A at s is defined as $out_{obj}^{\mathcal{G}}(s, \sigma_A) = \{\lambda \in Run(\mathcal{G}) \mid \forall j \geq 0, \lambda[j] \xrightarrow{\sigma_A(\lambda[j])} \lambda[j+1]\}$.
2. The set of *subjective outcomes* of σ_A at s is defined as $out_{subj}^{\mathcal{G}}(s, \sigma_A) = \bigcup_{i \in A, s' \sim_i s} out_{obj}^{\mathcal{G}}(s', \sigma_A)$.

Definition 3 The set of ATL formulas φ is defined by the following BNF :

$$\varphi ::= p \mid \neg\varphi \mid \varphi \rightarrow \varphi \mid \langle\langle A \rangle\rangle X\varphi \mid \langle\langle A \rangle\rangle \varphi U \varphi \mid \langle\langle A \rangle\rangle \varphi R \varphi$$

where $p \in AP$ and $A \subseteq Ag$.

The ATL operator $\langle\langle A \rangle\rangle$ intuitively means that ‘the agents in group A have a (collective) strategy to achieve ...’, where the goals are LTL formulas built by using operators ‘next’ X , ‘until’ U , and ‘release’ R . Note that the ‘release’ operator R cannot be defined in ATL with imperfect information by using ‘until’ U and ‘globally’ G , as it is the case in perfect information contexts [30], so we include it for completeness. We define A -formulas as the formulas in ATL in which A is the only group appearing in ATL modalities.

Traditionally, ATL under imperfect information has been given either state- or history-based semantics, and several variations have been considered on the interpretation of strategy operators. Here we present both the objective and subjective variants of the state-based semantics with imperfect information and imperfect recall.

Definition 4 Given an iCGS \mathcal{G} , an ATL formula φ , the subjective (resp. objective) semantics of φ at state s , denoted $(\mathcal{G}, s) \models_x \varphi$ for $x = subj$ (resp. $x = obj$), is defined recursively as follows :

$$\begin{aligned} (\mathcal{G}, s) \models_x p & \quad \text{iff } p \in \pi(s) \\ (\mathcal{G}, s) \models_x \neg\varphi & \quad \text{iff } (\mathcal{G}, s) \not\models_x \varphi \\ (\mathcal{G}, s) \models_x \varphi \wedge \varphi' & \quad \text{iff } (\mathcal{G}, s) \models_x \varphi \text{ and } (\mathcal{G}, s) \models_x \varphi' \\ (\mathcal{G}, s) \models_x \langle\langle A \rangle\rangle X\varphi & \quad \text{iff } \exists \sigma_A \forall \lambda \in out_x^{\mathcal{G}}(s, \sigma_A), (\mathcal{G}, \lambda[1]) \models_x \varphi \\ (\mathcal{G}, s) \models_x \langle\langle A \rangle\rangle \varphi U \varphi' & \quad \text{iff } \exists \sigma_A \forall \lambda \in out_x^{\mathcal{G}}(s, \sigma_A), \exists j \geq 0 \text{ with} \\ & \quad (\mathcal{G}, \lambda[j]) \models_x \varphi' \text{ and } \forall 0 \leq k < j, (\mathcal{G}, \lambda[k]) \models_x \varphi \\ (\mathcal{G}, s) \models_x \langle\langle A \rangle\rangle \varphi R \varphi' & \quad \text{iff } \exists \sigma_A \forall \lambda \in out_x^{\mathcal{G}}(s, \sigma_A), \text{ either } \forall j \geq 0, \\ & \quad (\mathcal{G}, \lambda[j]) \models_x \varphi, \text{ or } \exists k \geq 0 \text{ with } (\mathcal{G}, \lambda[k]) \models_x \varphi' \\ & \quad \text{and } \forall 0 \leq l \leq k, (\mathcal{G}, \lambda[l]) \models_x \varphi \end{aligned}$$

Remark 5 The knowledge operator K_i can be appended to the syntax of ATL with the following semantics :

$$(\mathcal{G}, s) \models_x K_i \varphi \text{ iff } \forall s' \in S, s' \sim_i s \text{ implies } (\mathcal{G}, s') \models_x \varphi$$

By considering the subjective interpretation of ATL, this operator can be derived : $(\mathcal{G}, s) \models_{subj} K_i \varphi$ iff $(\mathcal{G}, s) \models_{subj} \langle\langle i \rangle\rangle \varphi U \varphi$. There exists no such definition for the knowledge operator in ATL with the objective semantics.

3 Simulations and Bisimulations

In this section we define simulation and bisimulation relations on iCGS with imperfect information and perfect recall. The main result we prove is that bisimulations preserve the interpretation of formulas in ATL. We start by introducing relevant notions that will be used in the rest of the paper.

A *partial strategy* for agent $i \in Ag$ is a partial function $\sigma : S \rightarrow Act$ such that for each $s_1, s_2 \in S$, if $s_1 \sim_i s_2$ then $\sigma(s_1) = \sigma(s_2)$. We denote the domain of the partial strategy σ as $dom(\sigma)$. Given a group $A \subseteq Ag$, a *partial strategy profile* for $A \subseteq Ag$ is a tuple $(\sigma_i)_{i \in A}$ of partial strategies, one for each agent $i \in A$. The set of partial strategy profiles for A is denoted $PStr_A$. Given a set $U \subseteq S$ of states and a group $A \subseteq Ag$, we denote $PStr_A(U)$ the set of partial strategies whose domain is U :

$$PStr_A(U) = \{(\sigma_i)_{i \in A} \in PStr_A \mid dom(\sigma_i) = U \text{ for all } i \in A\}$$

Given a group $A \subseteq Ag$ of agents, the *collective knowledge relation* \sim_A^E is defined as $\bigcup_{i \in A} \sim_i$, while the *common knowledge relation* \sim_A^C is the transitive closure $(\bigcup_{i \in A} \sim_i)^+$ of \sim_A^E . Then, $E_A^G(q) = \{q' \in S \mid q' \sim_A^E q\}$ and $C_A^G(q) = \{q' \in S \mid q' \sim_A^C q\}$ are respectively the *collective* and *common knowledge neighbourhoods* of state q for group A in the iCGS \mathcal{G} .

Definition 6 (Simulation) Given two iCGS $\mathcal{G} = \langle Ag, AP, S, s_0, \{\sim_i\}_{i \in Ag}, Act, d, \rightarrow, \pi \rangle$ and $\mathcal{G}' = \langle Ag, AP, S', s'_0, \{\sim'_i\}_{i \in Ag}, Act', d', \rightarrow', \pi' \rangle$ sharing the set of agents Ag and the set of atoms AP , and a group $A \subseteq Ag$ of agents, a relation $\Rightarrow_A \subseteq S \times S'$ is a simulation for A iff $q \Rightarrow_A q'$ implies that

1. $\pi(q) = \pi'(q')$;
2. For every $i \in A$ and $r' \in S'$, if $q' \sim'_i r'$ then for some $r \in S$ we have that $q \sim_i r$ and $r \Rightarrow_A r'$.
3. By denoting $C_A(q) = C_A^G(q)$ and $C'_A(q') = C_A^{G'}(q')$, there exists a mapping $ST = ST_{C_A(q), C'_A(q')}$ with $ST : PStr_A(C_A(q)) \rightarrow PStr_A(C'_A(q'))$ such that for any two states $r \in C_A(q)$, $r' \in C'_A(q')$, if $r \Rightarrow_A r'$ then the following two properties hold:
 - (a) For every partial strategy $\sigma_A \in PStr_A(C_A(q))$ and state $s' \in S'$, if $r' \xrightarrow{ST(\sigma_A)(r')} s'$ then there exists some state s such that $r \xrightarrow{\sigma_A(r)} s$ and $s \Rightarrow_A s'$.
 - (b) $ST_{C_A(q), C'_A(q')} = ST_{C_A(r), C'_A(r')}$.

A relation \Leftrightarrow_A is a *bisimulation* iff both \Rightarrow_A and $\Rightarrow_A^{-1} = \{(q', q) \mid q \Rightarrow_A q'\}$ are simulations.

Intuitively, by Def. 6 state q' *simulates* q , i.e., $q \Rightarrow_A q'$ implies that (1) q and q' agree on the interpretation of atoms; (2) q simulates the epistemic transitions from q' ;

BALLOT		BALLOT		BALLOT	
Alex Jones	○	Alex Jones	○	Alex Jones	●
Bob Smith	●	Bob Smith	●	Bob Smith	○
Carol Wu	○	Carol Wu	●	Carol Wu	○
3147524		7523416		5530219	

FIGURE 1 – Three-ballot showing a vote for Bob Smith

and (3) for every partial strategy σ_A , defined on the common knowledge neighborhood $C_A(q)$, we are able to find some partial strategy $ST(\sigma_A)$ (the same for all states in $C_A(q)$) such that the transition relations $\xrightarrow{ST(\sigma_A)}$ and $\xrightarrow{\sigma_A}$ commute with the simulation relation \Rightarrow_A .

Remark 7 The problem of checking for the existence of a bisimulation between two iCGS, for some set of agents A is in PSPACE.

In order to prove that bisimilar states satisfy the same formulas in ATL, we prove the following auxiliary result.

Proposition 8 If $q \Rightarrow_A q'$ then for every uniform strategy σ_A , there exists a uniform strategy σ'_A such that

- (*) For every run $\lambda' \in out_x^{G'}(q', \sigma'_A)$, for $x \in \{subj, obj\}$, there exists an infinite run $\lambda \in out_x^G(q, \sigma_A)$ such that $\lambda(i) \Rightarrow_A \lambda'(i)$ for every $i \geq 0$.

By using Proposition 8 we are finally able to prove the main preservation result of this paper.

Theorem 9 Given two iCGS \mathcal{G} and \mathcal{G}' and states $q \in S$, $q' \in S'$, suppose that $q \Leftrightarrow_A q'$. Then for every A -formula φ ,

$$(\mathcal{G}, q) \models \varphi \text{ if and only if } (\mathcal{G}', q') \models \varphi$$

By Theorem 9 we obtain that bisimilar states preserve the interpretation of ATL formulas. More precisely, if states q and q' are A -bisimilar then they satisfy the same A -formulas.

4 Three-Ballot Voting Protocol

ThreeBallot [40, 39] is a voting protocol that strives to achieve some desirable properties, such as anonymity and verifiability of voting, without the use of cryptography. The protocol proceeds as follows. Each voter identifies herself at the poll site, and gets a paper “multi-ballot” to vote with. The multi-ballot consists of three vertical ballots – identical except for ID numbers at the bottom, see Figure 1 (presented after [40]). The voter fills in the multi-ballot, separates the three parts and casts them in the ballot box. To cast a vote for a candidate, one must mark exactly two (arbitrary) bubbles on the row of the candidate. To not vote for

a candidate, one must mark exactly one of the bubbles on the candidate's row (again, arbitrary one). In all the other cases the vote is invalid. The ballots are tallied by counting the number of bubbles marked for each candidate, and then subtracting the number of voters from the count.

While voting, the voter also receives a copy of one of her three ballots, and she can take it home. After the election closes, all the ballots are scanned and published on the web bulletin board. In consequence, the voter can check if her receipt matches a ballot listed on the bulletin board. If no ballot matches the receipt, the voter can file a protest.

Since ThreeBallot is not a cryptographic protocol, it does not heavily rely on computers and counting can be done directly. Moreover, voters have no responsibility to ensure the integrity of cryptographic keys, and the security process in their vote is essentially the same as with traditional ballots.

Properties. ThreeBallot was proposed to provide several properties that reduce the possibility of electoral fraud.

Anonymity (cf. e.g. [35]) requires that no agent should ever know how another voter voted, except in cases when it is inevitable, such as when all the voters voted for the same candidate. Anonymity is important because it limits the opportunities of coercion and vote-buying. *Coercion-resistance* requires that the voter cannot reveal the value of her vote beyond doubt, even if she fully cooperates with the coercer. As a consequence, the coercer has no way of deciding whether to execute his threat (or, dually, pay for the vote). A preliminary formalization of coercion-resistance and receipt-freeness in ATL has been presented in [44].

Finally, *end-to-end voter verifiability* [42, 41] provides a way to verify the outcome of the election by allowing voters to audit the information published by the system. Typically, the focus is on individual verifiability : each voter should be able check if her vote has been taken into account and has not been altered.

4.1 iCGS Model

We present here three iCGS models of the Three-Ballot Voting system. All these models have been specified in ISPL (Interpreted System Programming Language), the input language of MCMAS. Several aspects of the voting system have not been modeled : the ID of each ribbon, the copy of the ribbon which is given back to each voter after casting his/her ballot, the possibility for voters to verify the presence of the ribbon they are given back after voting. We model a single attacker who is also a voter and, as such, must obey the voting protocol and does not interact in any particular way with the other agents.

In the iCGS below, each agent is represented by means of its local variables and their evolution. The vote collector and bulletin board (BB) are modeled by the Environment agent (call it Env). This agent contains local variables mo-

deling the fact that the voting process is open and the values of ribbons on the BB. These variables are observable by all voters, including the attacker. Env also contains private variables used for collecting ribbons and disposes of the three actions $Act_e = \{stop, collect, nop\}$ for waiting closing elections, collecting votes and, finally, looping after the end of the publication of the BB.

Elections are closed immediately after the voting starts. This peculiarity of our models avoids us dealing with a vote collector which never stops the voting process, which may lead to the vacuous falsity of the formulas checked unless some fairness property is enforced – and, for the time being, fairness is not handled by our alternating bisimulation.

The agents representing voters have each a private variable representing their choice for a candidate. Then they share three "ballot" variables with Env. These variables represent the ribbons that are created by the "voting machine". Casting the vote is modeled by creating the three ribbons, compatible with the choice of each candidate. Votes are already cast in the initial state. Being visible by Env, the values of the three ribbons are copied by Env on the (variables represented on the) BB in a random order. Each agent has two actions : *vote*, by which the voter casts his/her vote, and *nop*, a non-voting or idle action. *vote* is enabled only in the initial state, *nop* is enabled everywhere. All agent variables are never modified during the voting process.

In the first model, denoted \mathcal{G}_{tot} , for each agent choice, all configurations of the three ribbons which are compatible with the agent's choice may occur. The communication between each agent and Env is entirely at Env's charge, who has direct access to agents' ribbons and copies them onto the BB. Copying is also done at random : Env chooses a non-copied ribbon from a voter who has cast his vote (boolean variables are defined to help Env identify these situations) and copies it onto a free position on the BB.

With the second model, denoted \mathcal{G}_{lex} , we model a voting machine which sorts, according to the lexicographic order, the three ribbons produced for the agent's choice, and places the largest one in the first "ballot" variable of the voter, the second largest in the second variable, and the smallest in the third variable. Hence, for each choice of an agent, there are still several configurations of ribbons that are produced, but we no longer produce all permutations of a configuration, but a single representative of that permutation.

Finally, we modify \mathcal{G}_{lex} into a third model, in which Env no longer copies ribbons on the BB, but rather counts the votes for each candidate by peeping at the "ballot" variables of each voter. This model is denoted \mathcal{G}_{count} .

Formally, in the case of \mathcal{G}_{tot} for n voters and nc candidates, each global state has the form $(vopen, pub, (ribb_\ell)_{1 \leq \ell \leq 3n})$,

$(ch_i, v_i)_{1 \leq i \leq n}, (s_{ij}, a_{ij})_{1 \leq i \leq n, 1 \leq j \leq 3}$ where :

1. The local state for voter i is
 $(vopen, pub, ribb_1, ribb_2, \dots, ribb_{3n}, v_i, s_{i1}, s_{i2}, s_{i3})$
2. Boolean $vopen$ holds true when the vote is opened and pub signals that all ribbons of agents that have voted are published on the BB.
3. Integer $1 \leq ch_i \leq nc$ specifies the choice of agent i .
4. Boolean v_i ($1 \leq i \leq n$) tells whether agent i has voted.
5. Integer variables s_{ij} ($1 \leq j \leq 3$) represent the "ballots" of voter i . They are shared between each agent and Env, who copies them onto the BB.
6. Integer variables $ribb_\ell$ ($1 \leq \ell \leq 3n$) represent the BB.
7. Booleans a_{ij} are used by Env for remembering which ballots s_{ij} have been copied on the BB.

Initial states are such that $vopen = true$, $v_i = false$ for all $i \leq n$, variables $ribb_\ell$ are undefined value \perp , $a_{ij} = false$ and, for variables s_{ij} we have the following rules modeling the creation of a triple of ribbons compatible with a choice of a candidate : for each voter i , let $b_{jk} = b'_{jk}$ be the bit representing the bubble on the line corresponding with candidate k of the j th ballot of i 's vote, as given by ch_i . A tuple $(b_{jk})_{1 \leq j \leq 3, 1 \leq k \leq nc}$ is compatible with choice ch_i if the following properties hold :

1. if $k = ch_i$ then $\exists p \leq 3$ s.t. $b_{pk} = 0$ and $\forall p' \neq p$, $b_{p'k} = 1$
2. if $k \neq ch_i$ then $\exists p \leq 3$ s.t. $b_{pk} = 1$ and $\forall p' \neq p$, $b_{p'k} = 0$

Denote $B(ch_i)$ the set of bit tuples $(b_{jk})_{1 \leq j \leq 3, 1 \leq k \leq nc}$ compatible with ch_i . Denote further by $R(ch_i)$ the transformation of these bit tuples into integer triples modeling the valid ballots compatible with the choice ch_i , $R(ch_i) = \{(st_j)_{1 \leq j \leq 3} \mid st_j = \sum_{1 \leq k \leq nc} b_{jk} \cdot 2^{k-1}, (b_{jk})_{1 \leq j \leq 3, 1 \leq k \leq nc} \in B(ch_i)\}$. (For instance, valid triples of integers compatible with a voting intention for candidate 2 and $nc = 2$ are all permutations of $(3, 2, 0)$ plus all permutations of $(2, 2, 1)$.)

Then $(s_{ij})_{1 \leq j \leq 3} \in R(ch_i)$ for each $1 \leq i \leq n, 1 \leq j \leq 3$.

Transitions are then of the form :

$(vopen, pub, (ribb_\ell)_{1 \leq \ell \leq 3n},$

$(ch_i, v_i)_{1 \leq i \leq n}, (s_{ij}, a_{ij})_{1 \leq i \leq n, 1 \leq j \leq 3}) \xrightarrow{(a_e, a_1, a_2, \dots, a_n)} (vopen', pub', (ribb'_\ell)_{1 \leq \ell \leq 3n}, (ch'_i, v'_i)_{1 \leq i \leq n}, (s'_{ij}, a'_{ij})_{1 \leq i \leq n, 1 \leq j \leq 3})$ with :

1. $vopen' = false$ if $(a_e = stop \text{ or } vopen = false)$ and $vopen' = true$ otherwise. Action $a_e = stop$ is the only available action for Env if $vopen = true$.
2. For $a_i = vote$, $v'_i = true$, and for $a_i = nop$, $v'_i = v_i$.
3. For $a_e = collect$ and $a_i = nop$ for all i we have the following rules :
 - (a) There exists some subset of pairs $A \subseteq \{1, \dots, n\} \times \{1, \dots, 3\}$ with $a'_{ij} = a_{ij} = true$ for all $(i, j) \in A$.
 - (b) There exists $(i_0, j_0) \notin A$ with $a'_{i_0, j_0} = true$, $a_{i_0, j_0} = false$ and for all $(i, j) \notin A \cup \{(i_0, j_0)\}$, $a'_{ij} = false$.

(c) There exists some $B \subseteq \{1, \dots, 3n\}$ with $card(B) = card(A)$, $ribb'_\ell = ribb_\ell$ for all $\ell \in B$.

(d) There exists some $k \notin B$, $1 \leq k \leq 3n$ with $ribb_k = \perp$, $ribb'_k = s_{i_0, j_0}$ and $ribb'_\ell = \perp$ for all $\ell \notin B \cup \{k\}$.

4. Action $a_e = nop$ can only be executed when, for each i , either all $a_{ij} = true$ or $v_i = false$, and its effect is to modify only $pub' = true$, all the other variables remaining unchanged.

In \mathcal{G}_{lex} , transitions are identical to the above, the only difference being in the initial states, more specifically in the configuration of variables s_{ij} . These are instantiated such that $(s_{ij})_{1 \leq j \leq 3} \in \{max(Perm((st_j)_{1 \leq j \leq 3})) \mid (st_j)_{1 \leq j \leq 3} \in R_{ch_i}\}$ for each $1 \leq i \leq n$, the maximum being considered under the lexicographic order and $Perm((st_j)_{1 \leq j \leq 3})$ stands for the set of all permutations of the tuple $(st_j)_{1 \leq j \leq 3}$.

Finally, the iCGS \mathcal{G}_{count} is similar with \mathcal{G}_{lex} but all variables $ribb_\ell$ are replaced with nc variables $(co_k)_{1 \leq k \leq nc}$. The local state for agent i is then $(vopen, pub, co_1, \dots, co_{nc}, v_i, s_{i1}, s_{i2}, s_{i3})$. The description of transitions is then the same, excepting the case for $a_e = collect$ and items 3.(c)-3.(d) above (defining the updates of variables $ribb_\ell$), which are replaced by the following :

- 3.(c') For each $1 \leq k \leq nc$, if $a'_{ij} \neq a_{ij}$ then $co'_k = co_k + b_{ijk}$, where b_{ijk} is the k -th least significant bit of s_{ij} , otherwise $co'_k = co_k$.

4.2 Bisimulations for \mathcal{G}_{tot} , \mathcal{G}_{lex} and \mathcal{G}_{count}

The three models defined in the previous section seem naturally related w.r.t. some properties – in particular those related with the attacker modifying the outcome of the vote or breaking the anonymity. The interest in simplifying the model is that checking the coercion resistance property can be done faster and with less memory on \mathcal{G}_{count} than on \mathcal{G}_{lex} , which, on its turn, requires less time and memory than \mathcal{G}_{tot} , as we will see in the last section. In this section we show that the three models are bisimilar for the attacker, for the set of atomic propositions that refer only to choices of the agents. The fact which formalizes the "natural relation" between them and allows us to check a coercion resistance property on the simplest one and then generalizing the results on the two others, in particular on the largest model. Note that this bisimulation works because the properties do not refer to the status of the BB. For instance, these bisimulations would not be useful for simplifying systems for verifiability [18].

Formally, for each choice for an agent i to vote for a candidate j , we utilize an atomic proposition $p_{ch_i=j}$, which holds true only in those states in which $ch_i = j$. Then if we denote the attacker $att = n$ and $AP = \{p_{ch_i=j} \mid 1 \leq i \leq n, 1 \leq j \leq nc\}$, the following relation is an $\{att\}$ -bisimulation over AP between \mathcal{G}_{tot} and \mathcal{G}_{lex} :
 $(vopen, pub, (ribb_\ell)_{1 \leq \ell \leq 3n}, (ch_i, v_i)_{1 \leq i \leq n},$

$(s_{ij}, a_{ij})_{1 \leq i \leq n, 1 \leq j \leq 3} \iff_{\{att\}}^1 (vopen', pub', (ribb'_\ell)_{1 \leq \ell \leq 3n}, (ch'_i, v'_i)_{1 \leq i \leq n}, (s'_{ij}, a'_{ij})_{1 \leq i \leq n, 1 \leq j \leq 3})$ iff the following hold :

1. $vopen = vopen', pub = pub', v_i = v'_i, ch_i = ch_i$ for all $1 \leq i \leq n$ and $ribb_\ell = ribb'_\ell$ for all $1 \leq \ell \leq 3n$.
2. For each $1 \leq i \leq n$, if we denote b_{jk} the k th least significant bit of s_{ij} and b'_{jk} the k th bit of s'_{ij} , then both $(b_{jk})_{1 \leq j \leq 3, 1 \leq k \leq nc}, (b'_{jk})_{1 \leq j \leq 3, 1 \leq k \leq nc} \in B(ch_i)$.
3. Denote ρ_i the S_3 -permutation of (s_{i1}, s_{i2}, s_{i3}) into $(s'_{i1}, s'_{i2}, s'_{i3})$, i.e. $s_{ij} = s'_{\rho_i(j)}$. Also when $s_{ij} = s'_{ij} = \perp$ we put $\rho_i = id_{\{1,2,3\}}$. Then $a_{ij} = a'_{\rho_i(j)}$ for all i, j .

Stated differently, the 3rd item above says that (b'_{jk}) is the largest, in lexicographic order, among all tuples in B_{ch_i} which are permutations of (b_{jk}) .

For \mathcal{G}_{lex} and \mathcal{G}_{count} , we may consider the following $\{att\}$ -bisimulation over AP :
 $(vopen, pub, (ribb_\ell)_{1 \leq \ell \leq 3n}, (ch_i, v_i)_{1 \leq i \leq n}, (s_{ij}, a_{ij})_{1 \leq i \leq n, 1 \leq j \leq 3}) \iff_{\{att\}}^2 (vopen', pub', (co_k)_{1 \leq k \leq nc}, (ch'_i, v'_i)_{1 \leq i \leq n}, (s'_{ij}, a'_{ij})_{1 \leq i \leq n, 1 \leq j \leq 3})$ where :

1. $vopen = vopen', pub = pub', v_i = v'_i, ch_i = ch_i, s_{ij} = s_{ij}$ and $a_{ij} = a'_{ij}$ for all $1 \leq i \leq n, 1 \leq j \leq 3$.
2. For each $1 \leq \ell \leq 3n$ and $1 \leq k \leq nc$, if we denote $b_{\ell k}$ the k th least significant bit on the ribbon $ribb_\ell$, then :

$$co_k = \sum \{b_{\ell k} \mid ribb_\ell \neq \perp, 1 \leq \ell \leq 3n\}$$

To prove that these relations are indeed alternating bisimulations, note that the condition 1 is trivially satisfied as whenever $q \iff_{\{att\}}^1 q'$ ($\iota = 1, 2$), we must have that $(ch_i = j) \in q$ iff $(ch_i = j) \in q'$.

To prove properties 2 and 2-dual for $\iff_{\{att\}}^1$, take states q, r in \mathcal{G}_{tot} and $r \in \mathcal{G}_{lex}$ with $q \iff_{\{att\}}^1 q', q \sim_{att} r$. Then

$$\begin{aligned} q &= (vopen, pub, (ribb_\ell)_{1 \leq \ell \leq 3n}, (ch_i, v_i)_{1 \leq i \leq n}, (s_{ij}, a_{ij})_{1 \leq i \leq n, 1 \leq j \leq 3}) \\ q' &= (vopen, pub, (ribb_\ell)_{1 \leq \ell \leq 3n}, (ch_i, v_i)_{1 \leq i \leq n}, (s'_{ij}, a'_{ij})_{1 \leq i \leq n, 1 \leq j \leq 3}) \\ r &= (vopen, pub, (ribb_\ell)_{1 \leq \ell \leq 3n}, (\overline{ch}_i, \overline{v}_i)_{1 \leq i \leq n}, (\overline{s}_{ij}, \overline{a}_{ij})_{1 \leq i \leq n, 1 \leq j \leq 3}) \end{aligned}$$

with q, q' related by the definition of $\iff_{\{att\}}^1$ above and $\overline{ch}_{att} = ch_{att}, \overline{s}_{att, j} = s'_{att, j}$ and $\overline{a}_{att, j} = a'_{att, j}$ for all $1 \leq j \leq 3$. Put then

$$\begin{aligned} r' &= (vopen, pub, (ribb_\ell)_{1 \leq \ell \leq 3n}, (ch'_i, v'_i)_{1 \leq i \leq n-1}, (\overline{ch}_{att}, \overline{v}_{att}), \\ &\quad (s'_{ij}, a'_{ij})_{1 \leq i \leq n-1, 1 \leq j \leq 3}, (\overline{s}_{att, j}, \overline{a}_{att, j})_{1 \leq j \leq 3}) \end{aligned}$$

and we get the desired result : $q' \iff_{\{att\}}^1 r'$ and $r \sim_{att} r'$. The mirror argument also works : for $q \iff_{\{att\}}^1 q'$ and $q' \sim_{att} r'$ we may choose $q \iff_{\{att\}}^1 r$ and $r \sim_{att} r'$.

Conditions 2 and 2-dual for $\iff_{\{att\}}^2$ can be proved similarly, by observing that, $q \iff_{\{att\}}^2 q'$ and $q \sim_{att} r$ with

$q, r \in \mathcal{G}_{lex}$ and $q' \in \mathcal{G}_{count}$, then :

$$\begin{aligned} q &= (vopen, pub, (ribb_\ell)_{1 \leq \ell \leq 3n}, (ch_i, v_i)_{1 \leq i \leq n}, (s_{ij}, a_{ij})_{1 \leq i \leq n, 1 \leq j \leq 3}) \\ q' &= (vopen, pub, (co_k)_{1 \leq k \leq nc}, (ch_i, v_i)_{1 \leq i \leq n}, (s'_{ij}, a'_{ij})_{1 \leq i \leq n, 1 \leq j \leq 3}) \\ r &= (vopen, pub, (ribb_\ell)_{1 \leq \ell \leq 3n}, (\overline{ch}_i, \overline{v}_i)_{1 \leq i \leq n}, (\overline{s}_{ij}, \overline{a}_{ij})_{1 \leq i \leq n, 1 \leq j \leq 3}) \end{aligned}$$

with the same relationship between variables of q and r ; and then, for

$$\begin{aligned} r' &= (vopen, pub, (co_k)_{1 \leq k \leq nc}, (ch'_i, v'_i)_{1 \leq i \leq n-1}, (\overline{ch}_{att}, \overline{v}_{att}), \\ &\quad (s'_{ij}, a'_{ij})_{1 \leq i \leq n-1, 1 \leq j \leq 3}, (\overline{s}_{att, j}, \overline{a}_{att, j})_{1 \leq j \leq 3}) \end{aligned}$$

we get $q' \iff_{\{att\}}^2 r'$ and $r \sim_{att} r'$.

Finally, for conditions 3 and 3-dual, note first that for any state $q \in \mathcal{G}_{tot}$ or $q \in \mathcal{G}_{lex}$, $C_{att}(q)$ is just the equivalence class of q w.r.t. \sim_{att} , that is, if $q = (vopen, pub, (ribb_\ell)_{1 \leq \ell \leq 3n}, (ch_i, v_i)_{1 \leq i \leq n}, (s_{ij}, a_{ij})_{1 \leq i \leq n, 1 \leq j \leq 3})$, then $C_{att}(q)$ is composed of all states with local state for att of the form $((ribb_\ell)_{1 \leq \ell \leq 3n}, ch_{att}, v_{att}, (s_{att, j}, a_{att, j})_{1 \leq j \leq 3})$. Similarly, for $q' \in \mathcal{G}_{count}$ with $q' = (vopen, pub, (co_k)_{1 \leq k \leq nc}, (ch'_i, v'_i)_{1 \leq i \leq n}, (s'_{ij}, a'_{ij})_{1 \leq i \leq n, 1 \leq j \leq 3})$, $C_{att}(q')$ is composed of all states with local state for att composed of $((co_k)_{1 \leq k \leq nc}, ch_{att}, v_{att}, (s_{att, j}, a_{att, j})_{1 \leq j \leq 3})$.

Note first that, in all three iCGS, on each neighborhood $C_{att}(q)$, only one or two partial strategies for att can be defined, depending whether the vote is open or not. Therefore, we may define the mapping $ST_{att}^{C_{att}^{G_{tot}}(\cdot), C_{att}^{G_{lex}}(\cdot)}$ to associate to each partial strategy prescribing nop to att in some $C_{att}^{G_{tot}}(q)$ the quasi-identical strategy prescribing the same action in $C_{att}^{G_{lex}}(q')$, and, similarly, to the partial strategy prescribing $vote$ to att in $C_{att}^{G_{tot}}(q)$ the strategy prescribing $vote$ in $C_{att}^{G_{lex}}(q')$. The dual mapping ST' is defined similarly, and the same definitions work for the bisimulation $\iff_{\{att\}}^2$. Note that these definitions already satisfy property 3.(b) of bisimulations.

To prove property 3.(a), consider first the strategy $vote_{att}$ prescribing $vote$ for att on $C_{att}^{G_{tot}}(q)$ and take some state r with $q \xrightarrow{vote_{att}} r'$. Since when voting is enabled, Environment does not collect votes, r' has the same BB as q' and all booleans a_{ij} are false. Therefore, we may choose the state r which has the same local variables as r' for all voters and the same BB as q , and get $q' \xrightarrow{ST(vote_{att})} r'$ and $r \iff_{\{att\}}^1 r'$. A similar proof works for $\iff_{\{att\}}^2$.

Consider now the strategy $none_{att}$ prescribing action nop for att on $C_{att}^{G_{tot}}(q)$ and take again some state r' with $q' \xrightarrow{nop_{att}} r'$. Note that this action is only enabled when the vote is closed. Then the only agent which executes a non-idle action on the above transition is Environment. By this transition, Environment copies one of the ribbons onto the BB. This transition can then be simulated in \mathcal{G}_{tot} by copying the same ribbon (but which might be stored at a different position in q than in q') onto the BB. A mirror argument can be used to prove 3.(a) for ST' .

Finally, for proving point 3.(a) for $q \stackrel{2}{\Leftrightarrow}_{\{att\}} q'$, note that the same considerations above apply for the case of a transition from a state q' in which the voting is open. For the case of states q, q' where the voting is closed and hence only strategy nop_{att} is available to the attacker, we note that the only action which is compatible with $q \xrightarrow{nop_{att}} r$ in both models is Environment collecting votes. This corresponds in \mathcal{G}_{count} with an action in which Environment counts votes, and hence we may find a state r which is an nop_{att} -compatible successor of q , which has the same local states for voters as r' and in which each counter co_k of r' keeps the sum of the bullets on k th line on the copied ribbons from r . This will ensure $r \stackrel{2}{\Leftrightarrow}_{\{att\}} r'$.

5 Experimental Results

In this section, we exhibit the improvements in running time when checking the same properties over the three bisimilar models. The three models are checked with growing number of voters and candidates. For our experiments, we have used the last version of MCMAS (1.2.2) [31]. Tests were made on a virtual machine running Ubuntu 16.04.1 LTS on a Dell PowerEdge R720 server with two Intel Xeon E5-2650 8 core processors at 2GHz, and 128 GB of RAM. The .ispl files containing the tested models of the voting system are available at [1].

The formulas that are verified on all these models represent a variant of coercion resistance [44]. They specify the fact that the attacker att has no strategy by which he could know how agent i has voted ($i \neq att$) :

$$\varphi_i = \langle\langle att \rangle\rangle F \left(pub \wedge \left(v_i \rightarrow \bigvee_{1 \leq j \leq nc} K_{att}(j = ch_i) \right) \right)$$

(Recall that, in our model the attacker is also a voter, which corresponds with situations in which a voter fully cooperates with the attacker).

MCMAS provides two options, `-atlk 2` or `-uniform`, for checking ATL formulas with uniform strategies, with some differences in the semantics of ATL formulas (`-uniform` is similar with “irrevocable strategies” of [2]). We observed that neither of these options were stable, and lead to a number of experiments ending with inconsistent results or MCMAS terminating abnormally. We refer the interested reader to [10].

We then checked the coercion resistance property with `-atlk 1` option, which utilizes ATL with perfect information. This is nevertheless consistent with our theoretical setting since all tests show that the formulas are false, and whenever a positive ATL formula is false under the perfect information semantics, it is also false under the imperfect information semantics, and hence preserved by alternating bisimulations.

For the *total model* \mathcal{G}_{tot} the only configurations for which MCMAS produces results in reasonable time are shown in

Table 1, which gives running times and state space (denoted $|S|$). For \mathcal{G}_{lex} , the state space is smaller and, therefore, the model with three voters and three candidates gives a also reasonable running time. For all the other cases, MCMAS outputs the result faster than for \mathcal{G}_{tot} . Statistics are given in Table 2. Finally, the models \mathcal{G}_{count} can be verified much faster, the number of reachable states decreasing substantially, allowing for verifying the formula for 4 voters and 3 candidates in 44 seconds. Statistics are given in Table 3. In all these tables, NA means a 2 hours timeout has been reached without obtaining any result.

		# voters		
		2v	3v	4v
# candid.	2c	0.93 s S = 3.49091e+06	7.765 s S = 1.46625e+10	NA
	3c	23.61 s S = 2.44048e+08	NA	NA

TABLE 1 – MCMAS statistics for \mathcal{G}_{tot}

		# voters		
		2v	3v	4v
# candid.	2c	0.38 s S = 196388	3.42 s S = 1.92068e+08	823.12 s S = 2.26211e+11
	3c	15.32 s S = 8.09895e+06	4807.79 s S = 1.03982e+11	NA

TABLE 2 – MCMAS statistics for \mathcal{G}_{lex}

		# voters			
		2v	3v	4v	5v
# candid.	2c	0.15 s S = 4406	0.72 s S = 39201	2.39 s S = 3.08043e+06	17.03 s S = 6.57133e+07
	3c	0.44 s S = 101993	4.29 s S = 3.81446e+06	44.18 s S = 2.17425e+09	NA

TABLE 3 – MCMAS statistics for \mathcal{G}_{count}

We also verified an anonymity property, specified in CTLK, with the same aim at showing the improvements obtained with bisimulations. Note that, for any group A , an A -bisimulation is also a bisimulation of the epistemic labeled transition systems, hence the two systems satisfy the same CTLK formulas. The CTLK formula that we tested is $\varphi_i^c = AG(not_same \rightarrow (\bigwedge_{1 \leq j \leq nc} \neg K_{att} p_{ch_i=j}))$ which utilizes an atomic proposition `not_same` which avoids unanimity. Note that `not_same` can be defined using only the atoms in AP .

6 Conclusions

In this paper we advanced the state-of-the-art in the model theory of the strategy logic ATL under imperfect information and imperfect recall. Specifically, we introduced a novel notion of (bi)simulation on iCGS that preserves the interpretation of ATL formulas (Theorem 9). Then, we applied this theoretical result to the verification of the Three-Ballot voting system, a relevant voting protocol without

		# voters		
		2v	3v	4v
# candid.	2c	0.776 s S = 3.72655e+06	6.531 s S = 1.46625e+10	NA
	3c	19.811 s S = 2.44048e+08	2628.61 s S = 1.69347e+13	NA

TABLE 4 – MCMAS statistics : \mathcal{G}_{rot} and CTLK formula

		# voters		
		2v	3v	4v
# candid.	2c	0.37 s S = 196388	3.035 s S = 1.92068e+08	NA
	3c	15.26 s S = 8.09895e+06	NA s	NA

TABLE 5 – MCMAS statistics : \mathcal{G}_{rex} and CTLK formula

		# voters			
		2v	3v	4v	5v
# candid.	2c	0.099 s S = 4406	0.553 s S = 39201	1.507 s S = 3.08043e+06	8.87 s S = 6.57133e+07
	3c	0.44 s S = 101993	4.29 s S = 3.81446e+06	26.078 s S = 2.17425e+09	NA

TABLE 6 – MCMAS statistics : \mathcal{G}_{count} and CTLK formula

cryptography. In particular, we model check the “simpler” bisimilar abstractions of the ThreeBallot system, and then transfer the result to the original model in virtue of Theorem 9. As reported in the experimental results, the gains in terms of both time and memory resources are significant.

The literature on both logics for strategies and the formal verification of voting protocols is extensive and rapidly growing. Hereafter we only consider the works most closely related to the present contribution.

Bisimulations for ATL. An in-depth study of model equivalences induced by various temporal logics appears in [23]. Bisimulations for ATL with perfect information have been introduced in [4]. Since then there have been various attempts to extend these to imperfect information contexts [2, 17]. In [17, 32] non-local model equivalences for ATL with imperfect information have been put forward. However, to our knowledge these works do not deal with the imperfect information/imperfect recall setting here considered, nor do they provide a local account of bisimulations.

Verification of Voting Protocols. The present contribution is inspired by recent works on the verification of voting protocols, mostly by using the π -calculus and CSP [18, 25, 43]. In [5] the authors define two semantic criteria for single transferable vote (STV) schemes, then show how bounded model-checking and SMT solvers can be used to check whether these criteria are met. In [35] anonymity properties of voting protocols are verified by using CSP. In particular, in [36] the authors construct CSP models of the ThreeBallot system and use them to produce an automated formal analysis of their anonymity properties. One issue we identify with this approach is that the system model and the property to be verified are not clearly distinguished. On the contrary, multi-agent logics allow a clear separation of the two, as well as a wider variety of properties, also involving

the existence of attacker strategies. Specifically, in our experiments we are able to model check ThreeBallot systems with 5 voters and 2 candidates, or 4 candidates and 3 voters, while in [36] results are provided for at most 3 voters and 2 candidates.

Future Work. We envisage several extensions of the present contribution. First, it is of interest to develop bisimulations for iCGS with perfect and bounded recall, as in many application domains agents do have some memory of past states and actions. Also for the verification of voting protocols, it is key to extend ATL with epistemic modalities to express naturally properties of anonymity and confidentiality. We remarked that individual knowledge is expressible in the subjective semantics. However, no such result holds for the objective interpretation, nor common knowledge happens to be definable. Finally, we aim at automating and implementing the procedures described in this paper in a model checking tool for the formal verification of (electronic) voting protocols.

Acknowledgements. F. Belardinelli acknowledges the support of the ANR JCJC Project SVEDaS (ANR-16-CE40-0021). W. Jamroga acknowledges the support of the National Centre for Research and Development (NCBR), Poland, under the project VoteVerif (POLLUX-IV/1/2016).

Références

- [1] *ISPL Files for ThreeBallot Voting Protocol*. <https://www.dropbox.com/sh/ferdoqe9hi4cmbx/AAA1hLy0grmCoBVqZf6wbKLWa?dl=0>. November 2016.
- [2] Ågotnes, T., V. Goranko et W. Jamroga: *Alternating-time Temporal Logics with Irrevocable Strategies*. Dans *Proceedings of TARK XI*, pages 15–24, 2007.
- [3] Alur, R., T. A. Henzinger et O. Kupferman: *Alternating-Time Temporal Logic*. *Journal of the ACM*, 49(5) :672–713, 2002.
- [4] Alur, Rajeev, Thomas A. Henzinger, Orna Kupferman et Moshe Y. Vardi: *Alternating refinement relations*. Dans *In Proceedings of the Ninth International Conference on Concurrency Theory (CONCUR’98), volume 1466 of LNCS*, pages 163–178. Springer-Verlag, 1998.
- [5] Beckert, Bernhard, Rajeev Goré, Carsten Schürmann, Thorsten Borner et Jian Wang: *Verifying Voting Schemes*. *J. Inf. Secur. Appl.*, 19(2) :115–129, avril 2014, ISSN 2214-2126. <http://dx.doi.org/10.1016/j.jisa.2014.04.005>.
- [6] Blackburn, P., M. de Rijke et Y. Venema: *Modal Logic*, tome 53 de *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2001.

- [7] Boureau, I., A. V. Jones et A. Lomuscio: *Automatic Verification of Epistemic Specifications Under Convergent Equational Theories*. Dans *Proceedings of the 11th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'12)*, pages 1141–1148. IFAAMAS, 2012.
- [8] Bulling, N., J. Dix et W. Jamroga: *Model Checking Logics of Strategic Ability : Complexity*. Dans *Specification and Verification of Multi-agent Systems*, pages 125–159. Springer, 2010.
- [9] Bulling, N. et W. Jamroga: *Comparing variants of strategic ability : how uncertainty and memory influence general properties of games*. *Autonomous Agents and Multi-Agent Systems*, 28(3) :474–518, 2014.
- [10] Busard, Simon, Charles Pecheur, Hongyang Qu et Franco Raimondi: *Reasoning about memoryless strategies under partial observability and unconditional fairness constraints*. *Information and Computation*, 242 :128–156, 2015.
- [11] Cervesato, I., N. A. Durgin, P. Lincoln, J. C. Mitchell et A. Scedrov: *A Meta-Notation for Protocol Analysis*. Dans *Proceedings of the 12th IEEE Computer Security Foundations Workshop (CSFW'99)*, pages 55–69. IEEE Computer Society, 1999.
- [12] Chadha, Rohit, Steve Kremer et Andre Scedrov: *Formal Analysis of Multiparty Contract Signing*. *J. Autom. Reasoning*, 36(1-2) :39–83, 2006. <http://dx.doi.org/10.1007/s10817-005-9019-5>.
- [13] Chatterjee, K., T. Henzinger et N. Piterman: *Strategy Logic*. Dans *Proceedings of the 18th International Conference on Concurrency Theory (CONCUR07)*, tome 4703, pages 59–73, 2007.
- [14] Clarke, E. M., O. Grumberg, S. Jha, Y. Lu et H. Veith: *Counterexample-Guided Abstraction Refinement*. Dans *Proceedings of the 12th International Conference on Computer Aided Verification (CAV00)*, tome 1855 de *Lecture Notes in Computer Science*, pages 154–169. Springer, 2000.
- [15] Clarke, E. M., O. Grumberg et D. Long: *Model Checking and Abstractions*. *ACM Transactions on Programming Languages and Systems*, 16(5) :1512–1542, 1994.
- [16] Cortier, Véronique: *Formal Verification of e-Voting : Solutions and Challenges*. *ACM SIGLOG News*, 2(1) :25–34, janvier 2015, ISSN 2372-3491. <http://doi.acm.org/10.1145/2728816.2728823>.
- [17] Dastani, M. et W. Jamroga: *Reasoning about Strategies of Multi-Agent Programs*. Dans *Proceedings of AAMAS2010*, pages 625–632, 2010.
- [18] Delaune, S., S. Kremer et M. Ryan: *Verifying Privacy-Type Properties of Electronic Voting Protocols*. *Journal of Computer Security*, 17(4) :435–487, 2009.
- [19] Denker, G. et J. K. Millen: *Modeling Group Communication Protocols Using Multiset Term Rewriting*. *Electr. Notes Theor. Comput. Sci.*, 71 :20–39, 2002.
- [20] Dima, Catalin et Ferucio Laurentiu Tiplea: *Model-checking ATL under Imperfect Information and Perfect Recall Semantics is Undecidable*. *CoRR*, abs/1102.4225, 2011. <http://arxiv.org/abs/1102.4225>.
- [21] Eijck, Jan van et Simona Orzan: *Epistemic Verification of Anonymity*. *Electr. Notes Theor. Comput. Sci.*, 168 :159–174, 2007. <http://dx.doi.org/10.1016/j.entcs.2006.08.026>.
- [22] Gammie, P. et R. van der Meyden: *MCK : Model Checking the Logic of Knowledge*. Dans *Proceedings of 16th International Conference on Computer Aided Verification (CAV04)*, tome 3114 de *Lecture Notes in Computer Science*, pages 479–483. Springer, 2004.
- [23] Goltz, Ursula, Ruurd Kuiper et Wojciech Penczek: *Propositional Temporal Logics and Equivalences*. Dans *Proceedings of CONCUR '92*, pages 222–236, 1992.
- [24] Goranko, V. et W. Jamroga: *Comparing Semantics for Logics of Multi-agent Systems*. *Synthese*, 139(2) :241–280, 2004.
- [25] Hoare, C. A. R.: *Communicating Sequential Processes*. *Commun. ACM*, 21(8) :666–677, 1978.
- [26] Jamroga, W. et J. Dix: *Model checking abilities under incomplete information is indeed δ_p^2 -complete*. Dans *Proceedings of the 4th European Workshop on Multi-Agent Systems EUMAS'06*, pages 14–15. Cite-seer, 2006.
- [27] Jamroga, W. et W. van der Hoek: *Agents that Know How to Play*. *Fundamenta Informaticae*, 62 :1–35, 2004.
- [28] Kacprzak, M., W. Nabialek, A. Niewiadomski, W. Penczek, A. Pólrola, M. Szreter, B. Woźna et A. Zbrzezny: *VerICS 2007 - a Model Checker for Knowledge and Real-Time*. *Fundamenta Informaticae*, 85(1) :313–328, 2008.
- [29] Küsters, Ralf, Tomasz Truderung et Andreas Vogt: *Verifiability, Privacy, and Coercion-Resistance : New Insights from a Case Study*. Dans *32nd IEEE Symposium on Security and Privacy, S&P 2011, 22-25 May 2011, Berkeley, California, USA*, pages 538–553. IEEE Computer Society, 2011. <http://dx.doi.org/10.1109/SP.2011.21>.

- [30] Laroussinie, François, Nicolas Markey et Ghasan Oreiby: *On the Expressiveness and Complexity of ATL*. Logical Methods in Computer Science, 4(2:7), mai 2008. <http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/LMO-lmcs08.pdf>.
- [31] Lomuscio, A., H. Qu et F. Raimondi: *MC-MAS : A Model Checker for the Verification of Multi-Agent Systems*. Software Tools for Technology Transfer, 2015. <http://dx.doi.org/10.1007/s10009-015-0378-x>, <http://dx.doi.org/10.1007/s10009-015-0378-x>.
- [32] Melissen, M.: *Game Theory and Logic for Non-repudiation Protocols and Attack Analysis*. Thèse de doctorat, University of Luxembourg, 2013.
- [33] Mogavero, F., A. Murano, G. Perelli et M. Y. Vardi: *Reasoning About Strategies : On the Model-Checking Problem*. ACM Transactions in Computational Logic, 15(4) :34 :1–34 :47, 2014. <http://doi.acm.org/10.1145/2631917>.
- [34] Mogavero, F., A. Murano et M. Vardi: *Reasoning About Strategies*. Dans *Proceedings of the 30th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS10)*, tome 8, pages 133–144. Schloss Dagstuhl, 2010.
- [35] Moran, Murat, James Heather et Steve Schneider: *Verifying anonymity in voting systems using CSP*. Formal Aspects of Computing, 26(1) :63–98, 2014, ISSN 1433-299X. <http://dx.doi.org/10.1007/s00165-012-0268-x>.
- [36] Moran, Murat, James Heather et Steve Schneider: *Automated anonymity verification of the ThreeBallot and VAV voting systems*. Software & Systems Modeling, 15(4) :1049–1062, 2016, ISSN 1619-1374. <http://dx.doi.org/10.1007/s10270-014-0445-x>.
- [37] Neuhäuffer, Martin R. et Joost-Pieter Katoen: *Bisimulation and Logical Preservation for Continuous-Time Markov Decision Processes*. Dans *CONCUR 2007 - Concurrency Theory, 18th International Conference, CONCUR 2007, Lisbon, Portugal, September 3-8, 2007, Proceedings*, tome 4703 de *Lecture Notes in Computer Science*, pages 412–427. Springer, 2007. http://dx.doi.org/10.1007/978-3-540-74407-8_28.
- [38] Pauly, M.: *A Modal Logic for Coalitional Power in Games*. Journal of Logic and Computation, 12(1) :149–166, 2002. <http://dx.doi.org/10.1093/logcom/12.1.149>.
- [39] R. Rivest, W. Smith: *Three Voting Protocols : Three-Ballot, VAV, and Twin*. Dans *Proceedings of USENIX/ACCURATE Electronic Voting Technology Workshop (EVT)*, 2007.
- [40] Rivest, R. L.: *The ThreeBallot Voting System*, October 2006. <http://theory.lcs.mit.edu/~rivest/Rivest-TheThreeBallotVotingSystem.pdf>.
- [41] Ryan, Peter A.: *The Computer Ate My Vote*. Dans Boca, Paul, Jonathan P. Bowen et Jawed Siddiqi (rédacteurs) : *Formal Methods : State of the Art and New Directions*, chapitre 5, pages 148–184. Springer Verlag, 2009.
- [42] Ryan, Peter Y. A., Steve A. Schneider et Vanessa Teague: *End-to-End Verifiability in Voting Systems, from Theory to Practice*. IEEE Security & Privacy, 13(3) :59–62, 2015.
- [43] Schneider, S. et A. Sidiropoulos: *CSP and Anonymity*. Dans *Proceedings of the 1996 European Symposium on Research in Computer Security (ESORICS'96)*, tome 1146 de *Lecture Notes in Computer Science*, pages 198–218. Springer-Verlag, 1996.
- [44] Tabatabaei, M., W. Jamroga et P.Y.A. Ryan: *Expressing Receipt-Freeness and Coercion-Resistance in Logics of Strategic Ability : Preliminary Attempt*. Dans *Proceedings of the 1st International Workshop on AI for Privacy and Security PrAISe 2016*, pages 1 :1–1 :8. ACM, 2016. <http://krak.ipipan.waw.pl/~wjamroga/papers/coercionInATL16praise.pdf>.

A Appendix

Proof of Proposition 8

We will inductively define a sequence of partial uniform strategy profiles $(\bar{\sigma}_A^n)_{n \in \mathbb{N}} \in PS tr_A$ with $\bar{\sigma}_A^n = (\bar{\sigma}_i^n)_{i \in A}$ and $dom(\bar{\sigma}_A^n) \subseteq dom(\bar{\sigma}_A^{n+1})$ for each $n \in \mathbb{N}$. These partial strategies will be constructed using the strategy σ_A and the mapping ST from point 3 in Def. 6 of simulation.

Define first the sequence $dom^n(\sigma_A, q)$, for $n \in \mathbb{N}$, of sets of \mathcal{G} -states such that $s \in dom^n(\sigma_A, q)$ if s can be reached in at most n steps from q by applying actions compatible with strategy σ_A :

$$dom^0(\sigma_A, q) = \emptyset, \quad dom^1(\sigma_A, q) = C_A(q)$$

$$dom^{n+1}(\sigma_A, q) = dom^n(\sigma_A, q) \cup \{r \mid \exists s \in dom^n(\sigma_A, q), s \xrightarrow{\sigma_A(s)} r\}$$

Also, denote σ_A^n the partial strategy resulting from restricting σ_A to $dom^n(\sigma_A, q)$ (and setting it as undefined on the complement of this set).

Further, consider some total order \lesssim on S . Then, $\min_{\lesssim} U$ is the minimum of U w.r.t. \lesssim . Similarly, we assume that S' is endowed with a total order \lesssim' .

The desired sequence of partial uniform strategy profiles $\bar{\sigma}_A^n$, for $n \geq 1$, is defined as $\bar{\sigma}_A^n(r)' = ST_{C_A(u), C_A(u)'}(\sigma_A^1)(r')$

and

$$\bar{\sigma}_A^{n+1}(r') = \begin{cases} \bar{\sigma}_A^n(r'), & \text{for } r' \in \text{dom}(\bar{\sigma}_A^n) \\ ST_{C_A(u), C'_A(u')}(\sigma_A^{n+1})(r'), & \text{for } C'_A(r') \cap \text{ran}(\bar{\sigma}_A^n) \neq \emptyset, \\ & r' \notin \text{dom}(\bar{\sigma}_A^n), u' = \min_{\leq'} C'_A(r') \cap \text{ran}(\bar{\sigma}_A^n) \\ & \text{and } u = \min_{\leq} \{v \in \text{dom}^n(\sigma_A, q) \mid v \Rightarrow_A u'\} \end{cases}$$

By induction on n we may easily observe that whenever $C'_A(r') \cap \text{dom}(\bar{\sigma}_A^n) \neq \emptyset$, then $C'_A(r') \subseteq \text{dom}(\bar{\sigma}_A^n)$.

We may also show that, whenever we take some $u' \in C'_A(r') \cap \text{ran}(\bar{\sigma}_A^n) \neq \emptyset$ with $r' \notin \text{dom}(\bar{\sigma}_A^n)$, we have $\{u \in \text{dom}^n(\sigma_A, q) \mid u \Rightarrow_A u'\} \neq \emptyset$. To see this, take some $u' \in C'_A(r') \cap \text{ran}(\bar{\sigma}_A^n)$, which implies that there exists $v' \in \text{dom}(\bar{\sigma}_A^n)$ with $v' \xrightarrow{\bar{\sigma}_A^n} u'$. But this means, by definition, that $\bar{\sigma}_A^n(v') = ST_{C_A(v), C'_A(v')}(\sigma_A^n)(v')$ for some $v \Rightarrow_A v'$ with $v \in \text{dom}(\sigma_A^{n-1})$. From property 3.a, this implies the existence of u with $u \Rightarrow_A u'$ and $v \xrightarrow{\sigma_A^{n-1}} u$, hence $v \in \text{dom}(\sigma_A^n)$.

We then prove by induction on n that $\bar{\sigma}_A^n$ is uniform. The case $n = 1$ is trivial. As regards the induction step, note first that, if $r'_1, r'_2 \in \text{dom}(\bar{\sigma}_A^{n+1})$, $C'_A(r'_1) = C'_A(r'_2)$ and if $r'_1 \in \text{dom}(\bar{\sigma}_A^n)$ then $r'_2 \in \text{dom}(\bar{\sigma}_A^{n+1})$ as well.

On the other hand, for $r'_1, r'_2 \in \text{dom}(\bar{\sigma}_A^{n+1}) \setminus \text{dom}(\bar{\sigma}_A^n)$, if $r'_1 \sim_i r'_2$ for some $i \in A$, we have that $C'_A(r'_1) = C'_A(r'_2)$. Therefore, if we take $r' = \min_{\leq'} C'_A(r'_1) \cap \text{ran}(\bar{\sigma}_A^n)$ and $r = \min_{\leq} \{u \in \text{dom}^n(\sigma_A, q) \mid u \Rightarrow_A r'\}$ we have, by definition, $\bar{\sigma}_A^{n+1}(r'_1) = ST_{C_A(r), C'_A(r)}(\sigma_A^{n+1})(r'_1)$ and $\bar{\sigma}_A^{n+1}(r'_2) = ST_{C_A(r), C'_A(r)}(\sigma_A^{n+1})(r'_2)$ and these values are identical, since $ST_{C_A(r), C'_A(r)}(\sigma_A)$ is a uniform strategy in \mathcal{G}' .

As a result, the ‘‘limit’’ partial strategy profile $\bar{\sigma}_A = \bigcup_{n \in \mathbb{N}} \bar{\sigma}_A^n$, defined by $\bar{\sigma}_A(q) = \bar{\sigma}_A^n(q)$ whenever $q \in \text{dom}(\bar{\sigma}_A^n)$, is clearly uniform and has $\text{dom}(\bar{\sigma}_A) = \text{ran}(\bar{\sigma}_A)$. We then only need to transform it into a (total) uniform strategy profile by imposing a fixed action $a_0 \in \text{Act}$ wherever $\bar{\sigma}_A^n$ was undefined, that is, defining the following uniform strategy profile σ'_A :

$$\sigma'_A(r') = \begin{cases} \bar{\sigma}_A(r') & \text{for } r' \in \text{dom}(\bar{\sigma}_A) \\ a_0 & \text{otherwise} \end{cases}$$

To prove property (*) for the objective semantics, consider a run $\lambda' \in \text{out}_{obj}^{\mathcal{G}'}(q', \sigma'_A)$ and set $\lambda[0] = q$. We then build inductively the run λ as follows: assume $\lambda[k]$ has been built, with $\lambda[j] \Rightarrow_A \lambda'[j]$ for all $j \leq k$. Then we apply point 3.a from Def. 6 of simulation to the pair $\lambda[k] \Rightarrow_A \lambda'(k)$ and by using the fact that $\lambda'(k+1) \in \sigma'_A(\lambda[k]) = ST_{C_A(v), C'_A(v')}(\sigma_A)(\lambda'(k))$, where $v' = \min_{\leq'} C'_A(\lambda'(k))$ and $v = \min_{\leq} \{u \in \text{dom}^n(\sigma_A, q) \mid u \Rightarrow_A \lambda'(k)\}$, to obtain the existence of a state $u \in \sigma_A(\lambda[k])$ such that $u \Rightarrow_A \lambda'(k+1)$. This is the state that we choose for $\lambda[k+1]$, i.e., $\lambda[k+1] := u$.

To prove property (*) for the subjective semantics, consider a run $\lambda' \in \text{out}_{subj}^{\mathcal{G}'}(q', \sigma'_A)$. Hence, $\lambda'(0) \in C'_A(q')$.

By applying point 2 of the definition of simulation (and a short induction on the length of the indistinguishability path connecting q' with $\lambda'(0)$), we obtain the existence of a state $r \in C_A(q)$ such that $r \Rightarrow_A \lambda'(0)$. Then set $\lambda[0] := r$. The rest of the construction of λ' is identical to the inductive case for the objective semantics, that is, for every k , we set $\lambda[k+1] := u$ where $u \in \sigma_A(\lambda[k])$ and $u \Rightarrow_A \lambda'(k+1)$.

The property (*) is then proved for both semantics.

Proof of Theorem 9

The proof is by induction on the structure of φ .

The case for propositional atoms is immediate as $(\mathcal{G}, q) \models p$ iff $p \in \pi(q)$, iff $p \in \pi'(q')$ by definition of bisimulation, iff $(\mathcal{G}', q') \models p$. The inductive case for propositional connectives is also straightforward.

For $\varphi = \langle\langle A \rangle\rangle \psi_1 U \psi_2$, $(\mathcal{G}, q) \models \varphi$ implies that for some strategy σ_A , for all $\lambda \in \text{out}_x^{\mathcal{G}}(q, \sigma_A)$, for some $j_1 \geq 0$, $(\mathcal{G}, \lambda[j_1]) \models \psi_2$ and for every $k < j_1$, $(\mathcal{G}, \lambda[k]) \models \psi_1$. Again by Proposition 8, there exists strategy σ'_A s.t. for all $\lambda' \in \text{out}_x^{\mathcal{G}'}(q', \sigma'_A)$, there exists $\lambda \in \text{out}_x^{\mathcal{G}}(q, \sigma_A)$ with $\lambda[j] \Leftrightarrow_A \lambda'[j]$ for all $j \geq 0$. By the induction hypothesis, we get that for all j , $(\mathcal{G}, \lambda[j]) \models \psi_1$ iff $(\mathcal{G}', \lambda'[j]) \models \psi_1$ and similarly for ψ_2 . Hence, $(\mathcal{G}', \lambda'[j_1]) \models \psi_2$ and $(\mathcal{G}', \lambda'[k]) \models \psi_1$ for all $k < j_1$, that is, $(\mathcal{G}', h') \models \varphi$.

The cases for $\varphi = \langle\langle A \rangle\rangle X \psi$ and $\varphi = \langle\langle A \rangle\rangle R \psi_2$ are proved similarly.

Approximation de l'inférence *MAP* via les transformations probabilistes-possibilistes

Salem Benferhat Amélie Levray Karim Tabia

CRIL, Univ. Artois - CNRS UMR 8188,
62300 Lens, France
{benferhat, levray, tabia}@cril.univ-artois.fr

Résumé

Dans ce papier, nous utilisons les modèles graphiques et les transformations probabilistes-possibilistes afin de donner une approximation de l'inférence *MAP* dans les réseaux crédaux. Dans un premier temps, nous présentons les transformations du cadre probabiliste imprécis vers le cadre possibiliste, nous étudions exactement deux transformations de la littérature que nous comparons expérimentalement. Ensuite, nous présentons trois critères de choix que nous utilisons dans l'expérience pour restreindre le domaine de réponse retourné. L'idée est de comparer notre approche, qui consiste à transformer un réseau crédal en un réseau possibiliste pour effectuer l'inférence *MAP*, avec les approches (exactes et approchées) existantes dans les réseaux crédaux. L'expérience conduite est la suite de précédents travaux mais de manière approfondie avec l'introduction des nouveaux critères de choix. La dernière partie du papier discute les résultats de cette expérimentation.

Abstract

This paper uses belief graphical models and probability-possibility transformations to give an approximation of *MAP* inference in credal networks. We first present transformations from credal to possibilistic network, we study exactly two different transformations from the literature to compare them during the experimentation. Then we present three criteria that we use in the process of our experiment. The idea is to compare our approach, which consists in transforming a credal network into a possibilistic one to perform *MAP* inference, with both exact and approximate *MAP* inference in credal networks. This experiment is the continuation of our previous work but in-depth and with the use of new criteria of choice. The last part of this paper discusses the results of the experiment.

1 Introduction

L'inférence *MAP* (Maximum A Posteriori) dans les réseaux Bayésiens est un problème de grand intérêt et est

étudié depuis des années [11, 12, 16, 17, 18]. De ce fait, il existe une variété de méthodes et d'algorithmes pour calculer et trouver la configuration ayant le degré de probabilité le plus élevé étant donné un ensemble de variables observées. Malgré tout, les réseaux Bayésiens, aussi connus qu'ils soient, peuvent parfois sembler inadaptés pour représenter des informations, telles que la connaissance d'un groupe d'experts, ou alors pour encoder des informations incomplètes ou mal connues. C'est pourquoi, nous avons besoin de cadre qui nous donne plus de flexibilité sur les paramètres du modèle. C'est dans cet esprit que les réseaux crédaux [4] ont été conçus. Ils généralisent les réseaux Bayésiens et offrent plus d'expressivité puisqu'ils représentent l'information à l'aide de probabilités imprécises. Le problème que l'on rencontre à vouloir raisonner avec un cadre aussi expressif est que cela implique une complexité de calcul plus élevée. Néanmoins, des méthodes et des algorithmes pour calculer des requêtes *MAP* existent et donnent des résultats plutôt concluants. Ces méthodes, cependant, ne sont pas très efficaces en termes de complexité de calcul et lorsqu'on fait face à des problèmes à taille réelle (avec beaucoup de variables). Le but de ce papier est de fournir une nouvelle méthode pour inférer dans les réseaux crédaux, plus précisément effectuer l'inférence *MAP*. Cette méthode est basée sur les transformations probabilistes-possibilistes. En cause : ces méthodes existent déjà et leurs propriétés ont déjà fait l'objet de travaux [2, 3, 6, 8, 10, 19, 20]. De plus, l'inférence *MAP* dans les réseaux possibilistes est aussi déjà connue, ce qui nous donne une bonne base pour ce travail. Ainsi, dans ce papier nous étudions deux transformations des probabilités imprécises vers les possibilistes (qui ont été adaptées aux modèles graphiques dans [3]) et effectuons une vaste expérimentation sur l'inférence *MAP*. Ce travail suit la ligne directe de nos précédents travaux [3], où nous avons analysé plusieurs problèmes concernant la commutativité des

transformations (des probabilités imprécises vers les possibilités), tels la marginalisation et le conditionnement, deux notions indispensables quand on s'intéresse aux requêtes *MAP*. Ce papier propose aussi un début d'expérimentation pour observer l'efficacité de l'inférence *MAP*. Ce papier¹ est organisé comme suit :

- La première partie rappelle les notions de base des modèles graphiques utilisés dans notre expérimentation.
- La seconde partie présente le problème de l'inférence *MAP* et ses difficultés.
- La troisième partie introduit notre approche, les transformations, et les critères.
- Pour enfin, terminer sur une analyse des résultats.

2 Bref rappel sur les réseaux crédaux et les réseaux possibilistes

Dans cette section, nous présentons les principaux modèles graphiques utilisés, les réseaux crédaux et les réseaux possibilistes. Mais d'abord, on introduit les réseaux Bayésiens.

2.1 Réseaux Bayésiens

Les réseaux Bayésiens (\mathcal{BN}) sont des modèles graphiques probabilistes [5] très connus, basés sur la théorie des probabilités, et spécifiés par deux composants :

- un *composant graphique* : un graphe acyclique dirigé (DAG) avec des nœuds représentant chaque variable A_i et les flèches encodant les relations de dépendance entre les variables.
- un *composant numérique* : où on associe à chaque variable A_i une table de probabilités locale $p(A_i|par(A_i))$ pour A_i dans le contexte de ses parents $par(A_i)$.

Cette représentation aux moyens de modèles graphiques nous permet d'encoder de manière compacte une distribution de probabilités sur un ensemble de variables. La distribution jointe encodée par le graphe \mathcal{BN} est calculée grâce à la règle de chaînage suivante :

$$P(A_1, \dots, A_n) = \prod_{i=1}^n P(A_i|par(A_i)). \quad (1)$$

2.2 Réseaux crédaux

Les réseaux crédaux sont aussi des modèles graphiques probabilistes, et de la même manière que les réseaux Bayésiens sont basés sur la théorie des probabilités, les réseaux crédaux sont eux basés sur la théorie des probabilités imprécises [21, 13]. Une notion clé de cette théorie est celle

d'ensemble crédal qui est souvent interprété comme un ensemble de croyances imprécises, dans le sens où la mesure de probabilités réelle se trouve dans l'ensemble mais il n'existe aucun moyen de la déterminer exactement due au manque de connaissance. Dans le but de caractériser un ensemble crédal, on peut utiliser un ensemble de points extrêmes, des intervalles de probabilités ou des contraintes linéaires. Dans notre cas, nous utilisons des distributions de probabilités à intervalles qui sont un moyen très naturel et commun pour définir l'imprécision et les informations malconnues. Dans une distribution de probabilités à intervalles IP , chaque interprétation $\omega_i \in \Omega$ est associée à un intervalle de probabilités $IP(\omega_i)=[l_i, u_i]$ où l_i (resp. u_i) dénote la borne inférieure (resp. borne supérieure) de la probabilité de ω_i . Notons qu'alors qu'une distribution de probabilités standard p induit un ordre complet sur l'ensemble des mondes possibles Ω , une distribution de probabilités à intervalles IP peut induire un ordre partiel puisque certaines interprétations peuvent être incomparables en cas d'intervalles qui se chevauchent. Définissons maintenant les modèles graphiques probabilistes basés sur les ensembles crédaux, autrement dit les réseaux crédaux [4, 15].

Définition 1 (Réseaux crédaux) *Un réseau crédal $CN = \langle G, K \rangle$ est un modèle graphique probabiliste où*

- $G = \langle V, E \rangle$ est un graphe acyclique dirigé, comme l'est un \mathcal{BN} , avec $V = \{A_1, \dots, A_n\}$ un ensemble de variables et E est un ensemble d'arcs.
- $K = \{IP_1, IP_2, \dots, IP_n\}$ est une collection de distributions de probabilités à intervalles, chaque IP_i est associée à une variable A_i dans le contexte de ses parents $par(A_i)$.

Notons qu'en pratique, dans les tables locales, on peut aussi spécifier un ensemble de points extrêmes à la place d'une distribution de probabilités à intervalles comme on le trouve dans le logiciel JavaBayes².

2.3 Réseaux possibilistes

Un réseau possibiliste $\mathcal{PN} = \langle G, \Theta \rangle$ est défini par :

- un *composant graphique* G qui est un graphe acyclique dirigé.
- un *composant numérique* Θ , un ensemble de tables de possibilités locales $\pi(A_i|par(A_i))$.

La sémantique associée à un réseau possibiliste est la théorie des possibilités où un degré représente l'incertitude et ce degré peut être interprété de plusieurs façons, par exemple, comme étant un degré de probabilités supérieures [9]. La théorie des possibilités est régie par plusieurs axiomes comme pour la théorie des probabilités. Ainsi, toutes les tables locales doivent satisfaire la condition de normalisation définie par : $\forall i = 1..n$, pour chaque contexte de parent $par(a_i)$, $\max_{a_i \in D_i} (\pi(a_i | par(a_i))) = 1$.

1. Ce papier est la version traduite d'un papier en cours de soumission.

2. <http://www.cs.cmu.edu/~javabayes/Home/>

Dans le cadre possibiliste, la distribution de possibilités jointe est factorisée en utilisant la contrepartie possibiliste de la règle de chaînage suivante :

$$\pi(a_1, a_2, \dots, a_n) = \otimes_{i=1}^n (\pi(a_i | \text{par}(a_i))) \quad (2)$$

où \otimes dénote l'opérateur du produit ou l'opérateur du min en fonction de l'interprétation que l'on fait de l'échelle possibiliste, soit quantitative (avec le produit) soit qualitative (avec le min). Dans ce travail, on ne s'intéresse qu'à la partie produit.

3 L'inférence MAP : aspects pratiques et calculatoires

Lorsqu'on s'intéresse au raisonnement dans les modèles graphiques, on est principalement intéressé par deux tâches d'inférence : trouver la configuration la plus probable (inférence MAP) et calculer des distributions marginales. Ces problèmes sont généralement calculatoirement très difficiles. Dans cette section, nous développons un peu plus la notion d'inférence dans les modèles graphiques.

3.1 Type de requêtes d'inférence

L'inférence dans les modèles graphiques probabilistes consiste, de manière générale, au calcul de la probabilité d'un événement. Dans les réseaux crédaux, cela revient à calculer la borne inférieure et la borne supérieure de la probabilité d'un événement. Soit $V = \{A_1 \dots A_n\}$ l'ensemble des variables du modèle. Soit $O \subseteq V$ l'ensemble des variables observées et soit une instance $o \in O$. Soit $Q \subseteq V$ l'ensemble des variables de requêtes et une instance $q \in Q$. Il existe trois principaux types de requêtes :

- Calcul de la probabilité d'un événement d'intérêt q (Pr) étant donnée une évidence o . Le problème de décision associé est si $P(q|o) \geq s \in [0, 1]$.
- Calcul de l'explication la plus plausible (MPE). Étant donnée une observation o sur plusieurs variables, l'objectif est de calculer l'instance la plus probable q de toutes les variables restantes (non observées) Q . Notons qu'ici, $O \cup Q = \mathcal{A}$ et $Q \cap O = \emptyset$. Le problème de décision correspondant est si $P(q, o) \geq s \in [0, 1]$.
- Calcul du maximum a posteriori (MAP). Étant donnée une observation o sur plusieurs variables O , l'objectif est de trouver l'instance la plus probable q de l'ensemble de variables de requêtes Q . Dans les requêtes MAP, $Q \cap O = \emptyset$. Notons que lorsque Q et O couvre l'ensemble des variables V , le problème MAP se réduit à trouver l'explication la plus plausible (MPE).

3.2 Complexité calculatoire

La table suivante (Table 1) liste les résultats de complexité des requêtes décrites ci dessus dans les réseaux

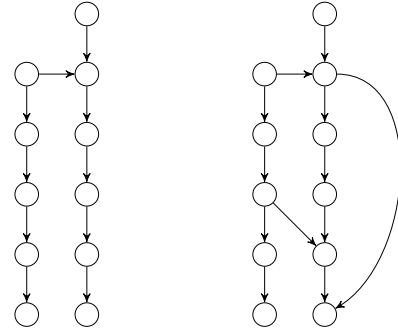


FIGURE 1 – A gauche : polytree, à droite : multiply-connected

Bayésiens et dans les réseaux crédaux [15]. On sépare ces réseaux en différentes catégories selon leur forme, ici nous nous intéressons à deux types de réseaux en particulier, les réseaux dits *polytree* et les réseaux dits *multiply-connected*. Un réseau *polytree* est une structure qui admet un seul chemin (indirect) entre deux nœuds du réseau et aussi dans lequel un nœud peut avoir plusieurs parents (dans le cas où un nœud n'admet qu'un parent au plus, on travaille alors avec des structures dites *tree*). Alors qu'une structure *multiply-connected* est une structure dans laquelle au moins une paire de nœuds admet plusieurs chemins (le graphe indirect associé comporte un cycle). Reportez-vous à la Figure 1 pour plus de détails.

En pratique, la taille des réseaux est souvent grande. Ce qui motive l'apparition de méthodes d'inférence approchée et dans ce papier, on vous propose une nouvelle méthode pour l'inférence MAP dans les réseaux crédaux. Et ce en transformant le réseau crédal CN en réseau possibiliste PN. Notons que la complexité de l'inférence dans les réseaux possibilistes est similaire à la complexité dans les réseaux Bayésiens.

Exemple 1 Voyons un exemple de requête MAP et un exemple de requête MPE dans un réseau possibiliste. En considérant le réseau possibiliste, sur l'ensemble des variables $V = \{A, B, C, D\}$, représenté par la Figure 2. Dans cet exemple, nous souhaitons calculer la requête MAP suivante : on cherche la configuration de D la plus plausible sachant que $A = F$. Pour répondre à ce genre de questions, il existe de nombreux algorithmes tels que l'élimination de variables ou l'arbre de jonction. Par ce biais, le calcul nous donne : $\pi(D = T | A = F) = 1$ et $\pi(D = F | A = F) = .6$, ainsi le résultat du maximum à posteriori de D sachant A est $D = T$. En comparaison, calculer la requête MPE pour l'observation $A = F$ revient à calculer le degré de possibilité le plus élevé pour l'ensemble des variables non-observées B, C, D . Alors, on calcule pour chaque configuration de A, B, C, D où $A = F$ le degré de possibilité, et la réponse à la requête, est l'instance avec le degré le plus

	Requêtes	Polytree	Bounded treewidth	Multiply-connected
Réseaux Bayésien	<i>Pr</i>	Polynomial	Polynomial	PP-Complet
	<i>MPE</i>	Polynomial	Polynomial	NP-Complet
	<i>MAP</i>	NP-Complet	NP-Complet	NP ^{PP} -Complet
Réseaux crédaux	<i>Pr</i>	NP-Complet	NP-Complet	NP ^{PP} -Complet
	<i>MPE</i>	Polynomial	Polynomial	NP-Complet
	<i>MAP</i>	Σ_2^P -Complet	Σ_2^P -Complet	NP ^{PP} -Difficile

TABLE 1 – Résumé de la complexité des requêtes dans les réseaux Bayésiens et les réseaux crédaux

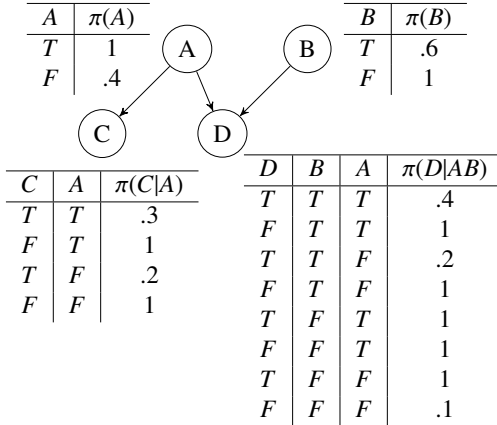


FIGURE 2 – Exemple d’un réseau possibiliste avec 4 variables booléennes

élevé, ici, $\pi(A = F, B = F, C = F, D = T) = .4$.

4 Une approche basée sur les transformations probabiliste-possibiliste

Plusieurs transformations probabiliste-possibiliste existent comme on le rappelle dans la sous-section 4.1. Des méthodes les généralisant aux probabilités imprécises ont été proposées par Masson et Denoeux [14] et plus encore dans [6]. Dans cette section, nous présentons deux de ces transformations.

4.1 Transformation probabilités \rightarrow possibilités

La théorie des probabilités et la théorie des possibilités sont deux cadres qui ont été étudiés de manière approfondie et plusieurs passerelles ont été construites entre les deux [10, 22]. Nous avons maintenant des transformations qui permettent de passer des probabilités vers les possibilités et inversement. Dubois et Prade ont, par exemple, proposé la transformation appelée **Optimal transformation (OT)**

[8] qui est définie par :

$$\pi(\omega_i) = \sum_{j|p(\omega_j) \leq p(\omega_i)} p(\omega_j) \quad (3)$$

Les transformations, de manière générale, sont tenues de satisfaire des principes de base afin de préserver le plus possible l’information. Dans ce sens, la transformation **OT** est considérée comme celle qui préserve le mieux l’information tout en satisfaisant ces principes³. Il existe d’autres travaux sur les transformations que l’on peut trouver dans [2, 3, 6, 8, 10, 19, 20].

4.2 Des probabilités imprécises vers les possibilités

La première transformation que nous étudions est celle de Masson et Denoeux dans [14], où les auteurs apprennent une distribution de possibilités à partir de données empiriques en transformant les intervalles de confiance en distributions de possibilités. Le premier point consiste à considérer une distribution de probabilités à intervalles comme moyen d’encoder l’ordre partiel \mathcal{M} sur Ω . Ainsi, soit \mathcal{M} représentant l’ordre partiel encodé par la distribution de probabilités à intervalles IP et soit \mathcal{C} l’ensemble des extensions linéaires (ordre total) qui sont compatibles avec l’ordre partiel \mathcal{M} , la transformation MD se déroule comme suit. Pour chaque extension linéaire $C_l \in \mathcal{C}$ et pour chaque interprétation $\omega_i \in \Omega$, on cherche la distribution de probabilités compatible qui donne le degré de possibilités le plus élevé, quand la transformation **OT** est utilisée pour obtenir la distribution de possibilités⁴.

$$\pi^{C_l}(\omega_i) = \max_{p_1 \dots p_n} \left(\sum_{p_j \leq p_i} p_j \right) \quad (4)$$

Ensuite, la distribution de possibilités finale est construite en prenant la valeur maximale, pour chaque interprétation, des distributions de possibilités π^{C_l} obtenues pour chaque extension linéaire. Cette transformation tente d’une part de préserver l’ordre des interprétations induit par IP et d’autre

3. Parmi ces principes se trouvent : la condition de consistance, la préservation de la préférence et le principe de maximum de spécificité.

4. En effet, la transformation MD se réduit à **OT** lorsqu’on considère des singletons à la place des intervalles de probabilités.

part de préserver le principe de dominance qui requiert que $\forall \phi \subseteq \Omega, P(\phi) \leq \Pi(\phi)$ (La probabilité est bornée par le degré de possibilités).

Il y a deux principaux inconvénients à cette transformation, le premier concerne la complexité de calcul. A savoir, le nombre d'extensions linéaires que l'on doit considérer est dans le pire des cas $N!$ (où N est le nombre de mondes possibles). En second, la distribution obtenue n'est pas nécessairement optimale, et pas, non plus, la plus spécifique dominant la distribution de probabilités à intervalles considérée. Du coup, nous avons besoin de considérer une autre approche pour transformer une distribution imprécise en une distribution de possibilités. Cette transformation se rapporte à la notion de distribution cumulative inférieures et supérieures. Dans notre précédent papier [3], nous avons déjà parlé de la transformation MD , nous avons aussi évoqué la distribution cumulative supérieure \bar{F} construite à partir d'une seule extension linéaire $C_i \in C$. Cette fois-ci, dans le but d'être le plus complet possible, l'on ne considère plus une seule distribution de possibilités mais deux, en ajoutant la distribution cumulative inférieure. Étant donné un ensemble d'intervalles de probabilités et une relation d'ordre \leq_{C_i} construite à partir d'une extension linéaire C_i entre les éléments ω_i , on peut facilement construire une p-box $[\underline{F}, \bar{F}]$. Étant donné les ensembles consécutifs $A_i = \{\omega_i, \forall \omega_i \in \Omega \text{ et t.q. } \omega_i \leq_{C_i} \omega_j \text{ ssi } i < j\}$, les distributions cumulatives inférieure et supérieure correspondants à Ω sont, respectivement :

$$\underline{F}(\omega_i) = \underline{P}(A_i) = \max\left(\sum_{\omega_j \in A_i} l_j, 1 - \sum_{\omega_j \notin A_i} u_j\right) \quad (5)$$

$$\bar{F}(\omega_i) = \bar{P}(A_i) = \min\left(\sum_{\omega_j \in A_i} u_j, 1 - \sum_{\omega_j \notin A_i} l_j\right) \quad (6)$$

Ces deux distributions obtenues sont deux distributions de possibilités qui encadrent (dans le sens dominant) les bornes de la distribution de probabilités imprécises. En dehors des considérations algorithmiques, l'utilisation des transformations par le biais de p-box [7] est intéressante malgré le fait qu'il n'existe pas de relation spécifique entre les deux (dans le sens où aucune ne généralise l'autre), puisqu'il existe des résultats entre les deux qui ont été démontrés [1]. Maintenant, regardons comment appliquer ces transformations dans le cadre des modèles graphiques.

4.3 Réseau crédal \rightarrow réseau possibiliste

Une méthode simple et directe pour transformer un réseau crédal en réseau possibiliste et qui nous donne l'avantage de préserver les relations d'indépendance entre les variables, est défini ainsi :

Définition 2 (Transformation graphique crédal-possibiliste) Soit CN un réseau crédal, $\mathcal{P}N_{CN}$ est un réseau possibiliste obtenu depuis CN et défini par :

- une composante graphique G qui est le même graphe que pour le réseau crédal ainsi $\mathcal{P}N_{CN}$ encode les mêmes relations d'indépendance que CN .
- un ensemble de tables de possibilités locales π_i qui sont le résultat de la transformation des ensembles crédaux locaux IP_i par TR , une transformation de probabilités imprécises vers les possibilités.

Exemple 2 Soit CN un réseau crédal représenté par la Figure 3 sur les deux variables binaires A et B . En utilisant la transformation MD , le réseau crédal CN de la Figure 3 sera transformé en un réseau possibiliste $\mathcal{P}N$ que l'on peut voir sur la Figure 4.

Concernant les principes respectés par ces transformations, nous avons étudiés plusieurs questions dans [3]. En effet, nous avons répondu à ces deux questions : i) Est-ce que la distribution $\pi_{\mathcal{P}N}$ domine IP_{CN} (la distribution jointe encodé par CN) ? et ii) Est-ce que l'ordre partiel des interprétations induit par IP_{CN} est préservé par la transformation TR ?

En réponse à la première question, pour les mondes élémentaires $\omega_i \in \Omega$, nous garantissons que la distribution de possibilités calculée domine bien les degrés de probabilités correspondant, dans le cas où le réseau crédal est un réseau Bayésien (à savoir, tous les intervalles sont des singletons). Concernant les événements arbitraires $\phi \subseteq \Omega$, la question reste ouverte. Si nous utilisons la transformation OT , la distribution de possibilités jointe obtenue ne garantit pas de dominer la distribution jointe de probabilités. En réponse à la seconde question, il n'existe aucune garantie que l'ordre des interprétations encodé par la distribution jointe sera le même après la transformation. Pour plus de détails, reportez-vous à [3]. Ces problèmes sont la raison pour laquelle nous ne pouvons proposer de méthode exacte d'inférence dans les réseaux crédaux. Mais notons que c'est l'absence de garantie sur la préservation de l'ordre qui rend notre méthode approchée.

5 Étude expérimentale

Dans cette section, nous donnons les résultats d'une expérience approfondie où l'on utilise de nouveaux critères pour le choix des résultats des requêtes MAP dans les réseaux crédaux.

5.1 Configuration de l'expérience et critères d'évaluation

Dans la précédente expérimentation effectuée dans [3], nous avons utilisé pour critère de choisir toutes interprétations qui n'étaient pas interval-dominées (voir Définition 3). Dans cette expérience, nous nous sommes rendus compte que les résultats n'étaient pas très convaincants.

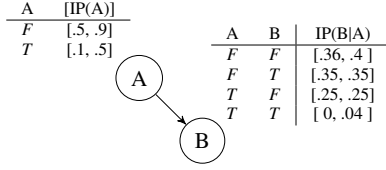


FIGURE 3 – Exemple d'un réseau crédal CN .

Une raison qui l'explique est le nombre d'interprétations qui est retourné et qui se trouve être trop grand, ce qui est dû en partie à la taille des réseaux. Dans cette nouvelle expérimentation que nous faisons, nous réduisons le nombre de résultats retournés en utilisant trois nouveaux critères. Rappelons, d'abord, la définition d'*Interval-dominance* qui se réfère aux instancés non-dominés des variables de requêtes.

Définition 3 (Interval-dominance) Une instance q_i des variables de requêtes Q domine une autre instance q_j si et seulement si $\underline{IP}(q_i|o) > \overline{IP}(q_j|o)$ où o est une instance de variables observées O .

Nous considérons maintenant trois nouveaux critères pour réduire le nombre de résultats. Il existe plusieurs raisons pour lesquelles on veut réduire ce nombre. Dans un premier cas, dans certaines applications où l'on utilise les requêtes *MAP*, on s'attend à une réponse limitée. Si le résultat contient toutes les interprétations disponibles, soit l'on se trouve dans un cas de totale ignorance, soit l'algorithme est juste mauvais. C'est pour cela qu'on introduit ces trois critères très connus *Maximax*, *Maximin*, et *Hurwicz*. Ces trois critères sont principalement utilisés dans la prise de décision sous incertitude depuis le début des années 1950. Ils sont, ci après, détaillés.

Le critère *Maximax* peut-être vu comme une approche optimiste. Il examine les gains maximaux des alternatives et choisit l'alternative qui a le meilleur gain. La définition 4, donne une définition formelle du critère *Maximax* dans le cadre des probabilités imprécises.

Définition 4 (Critère Maximax) Une instance q_i des variables de requêtes Q est un résultat de l'inférence *MAP* si et seulement si $\overline{IP}(q_i|o) \geq \max\{1 - \sum_{q_j \neq q_k} \underline{IP}(q_j|o), \forall q_k\}$, où o est un instance des variables observées O .

D'un autre côté, le critère *Maximin*, aussi connu sous le nom de *Wald's Maximin* criterion, est une approche pessimiste. Il suggère que le décideur examine seulement les gains minimums de chaque alternative et choisit l'alternative qui a le gain le moins mauvais. Dans notre cas, la définition de ce critère appliqué au contexte des probabilités imprécises est donnée par la suivante :

Définition 5 (Critère Maximin) Une instance q_i des variables de requêtes Q est un résultat de l'inférence *MAP* si et seulement si $\underline{IP}(q_i|o) \geq \max\{1 - \sum_{q_j \neq q_k} \overline{IP}(q_j|o), \forall q_k\}$, où o est un instance de variables observées O .

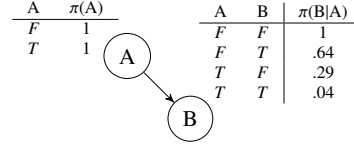


FIGURE 4 – Réseau possibiliste \mathcal{PN}_{CN} obtenu par transformation de CN (Fig. 3).

Réseaux	Topologie	#Nœuds	$ D_{A_i} $
<i>Alarm</i>	Multiply-connected	37	4
<i>Insurance</i>	Multiply-connected	27	5
<i>Poly</i>	Polytree	10	4
<i>Multi</i>	Multiply-connected	6	4

TABLE 2 – Description des réseaux crédaux utilisés.

Le dernier critère que nous passons en revue est le célèbre critère d'*Hurwicz*, suggéré par Leonid Hurwicz en 1951, qui sélectionne le gain minimum et le gain maximum pour chaque alternative donnée. Le critère d'*Hurwicz* tente de trouver un terrain d'entente entre les deux extrêmes, que représentent le critère optimiste et le pessimiste, en assignant une certaine proportion, α au critère optimiste et son contre poids $1 - \alpha$ au critère pessimiste. Cet index reflète l'attitude du décideur face au risque. Un décideur prudent assignera $\alpha = 1$ ce qui ramène le critère *Hurwicz* au critère *Maximin*. Un décideur aventureux mettra $\alpha = 0$ ce qui réduit le critère *Hurwicz* au critère *Maximax*.

La définition 6 définit formellement le critère d'*Hurwicz* avec les probabilités à intervalles en utilisant le coefficient α .

Définition 6 (Critère Hurwicz) Soit une instance q_i des variables de requêtes Q pour o une instance de variables observées de O , $a = \{\alpha * (1 - \sum_{q_j \neq q_k} \overline{IP}(q_j|o)) + (1 - \alpha) * (1 - \sum_{q_j \neq q_k} \underline{IP}(q_j|o)), \forall q_k\}$. Alors q_i est un résultat de l'inférence *MAP* si et seulement si $a = \max_{q_j} \{\alpha * (1 - \sum_{q_j \neq q_k} \overline{IP}(q_j|o)) + (1 - \alpha) * (1 - \sum_{q_j \neq q_k} \underline{IP}(q_j|o)), \forall q_k\}$.

Dans le cadre de notre expérimentation, nous avons choisi de poser $\alpha = 0.5$, ne donnant ni l'avantage à l'approche pessimiste ou optimiste.

Avant de présenter les résultats, nous donnons la description des benchmark utilisés dans l'expérimentation dans la table 2.

Présentons maintenant avec de plus amples détails l'étude expérimentale effectuée. En gardant en tête ce qui a été fait dans la précédente étude, nous poursuivons notre analyse plus loin, en incluant trois critères afin d'élaguer le nombre de résultats retourné par le réseau crédal. Nous utilisons aussi deux distributions de possibilités comme résultats à la transformation *CD*. En vérité, c'est une amélioration de la transformation *CD* utilisée dans [3], que l'on pouvait interpréter comme la distribution de possibilités cumulative supérieure, en ajoutant sa contre-partie, la dis-

tribution de possibilités cumulative inférieure. Ainsi, dans cet étude nous fournissons :

- Les taux de précision comparant l’algorithme exact développé dans JavaBayes avec :
 - l’algorithme d’inférence approchée utilisé dans le programme *GL2U*.
 - la précision fournie par l’inférence *MAP* dans les réseaux possibilistes obtenus par la transformations *MD*.
 - la précision fournie dans les réseaux possibilistes obtenues en utilisant les distributions de possibilités cumulatives inférieure et supérieure.
- Les taux d’inclusion de chaque méthode : on calcule le nombre d’instances de nos méthodes incluses dans les instances retournées par la méthode exacte.
- Les taux de précision et d’inclusion de chaque méthode en utilisant les trois critères sur les résultats de la méthode exacte (JavaBayes).
- Les taux de résultats : on compare le nombre de résultats retourné par l’algorithme avec le nombre de résultats possibles. L’idée est de mettre en lumière le niveau d’incertitude ou confusion retournée pour chaque algorithme.
- Tous ces résultats en variant le nombre de variables de requêtes de 1 à 5. Dans chacun des cas, on effectue l’expérience sur 200 fichiers.

5.2 Résultats

Les résultats sont donnés dans le graphique suivant (Figure 5). Afin de rester concis, nous n’y montrons que la comparaison entre *MD*, *CD* et *GL2U* sur un seul type de réseaux : les polyarbres et la comparaison des critères *Interval-dominance*, *Maximin* et *Hurwicz*. On a choisi d’oublier le critère *Maximin* puisque les résultats sont similaires avec ceux des critères *Maximax* et *Hurwicz*. Une information que l’on présente sur ce graphe, plus précisément sur le premier diagramme, est le nombre de fichiers auquel l’algorithme *GL2U* a pu répondre. En effet, nous déroulons tous les algorithmes sur approximativement 200 fichiers pour chaque nombre de variables de requêtes, mais dans certains cas, l’algorithme *GL2U* ne peut pas fournir de réponse. Du coup, le point sur le diagramme représente le taux de fichiers répondus. Par exemple, pour une seule variable de requête, la méthode *GL2U* n’a répondu qu’à 70% des 200 fichiers.

Dans un premier temps, notons que pour les autres types de fichiers, nous observons les mêmes résultats avec plus ou moins d’écart dans les taux de précision. Les résultats diffèrent de manière significative en fonction du nombre de variables de requêtes et du domaines de ces variables. En effet, si nous considérons trois variables de requêtes avec un domaine de cinq valeurs pour chaque variable, le nombre de configuration à prendre en compte peut aller jus-

qu’à 125, ce qui veut dire que lorsqu’on transforme des probabilités à intervalles vers les possibilités, nous devons considérer au plus 125 extensions linéaires. Ce qui résulte en une distribution de possibilités qui est bien plus imprécise (où il y a partout des 1). Et du coup, on observe moins de précision. Cependant, nous pouvons argumenter qu’en terme des réponses incluses, nous avons de bons résultats. De plus, dans la table 3 qui suit, nous montrons les résultats sur le nombre d’interprétations retourné pour l’inférence *MAP*. En fait, pour les critères *Interval-dominance* et *Maximax*, on fournit le taux d’interprétations répondues sur le nombre d’interprétations possible. Et ce, pour les algorithmes exact (de JavaBayes), et la même chose pour *MD*, *CD*, *GL2U*. Nous fournissons aussi, le ratio de réponses incluses, le nombre de réponses (des méthodes approchées) qui sont incluses dans l’ensemble des réponses retournées par l’algorithme exacte.

Dans la table 3, nous pouvons réellement voir trois type de résultats :

- i) Dans le cas du critère *Interval-dominance*, le nombre d’instances retournées en résultat de l’inférence *MAP* est élevé, ce qui révèle trop d’ignorance. Au contraire, lorsqu’on utilise le critère *Maximax*, on observe, avec un taux de 0.36, plus de précision et notons que dans ce cas, *CD* donne un nombre d’instances assez similaire.
- ii) Lorsqu’on compare le taux de réponses retournées pour la transformation *MD*, nous remarquons qu’en fait cette transformation nous donne le réseau possibiliste le plus précis en respect avec le réseau crédal. De plus, il montre de meilleurs résultats en terme de réponses incluses comparé aux deux autres approches.
- iii) La dernière information qui ressort de ces résultats repose sur l’attitude de l’algorithme *GL2U*, en effet il retourne, de manière générale, un plus grand nombre d’interprétations, même comparé à l’approche exacte. Et lorsqu’on considère plusieurs variables de requêtes, le nombre de réponses retournées grimpe rapidement pour atteindre tout l’ensemble des mondes possibles.

En se concentrant sur les résultats donnés par le graphique, nous observons, que même si l’algorithme approchée *GL2U* est plus précis, il ne tient pas la route lorsque le nombre de variables de requêtes atteint ou dépasse 3, ce qui est une mauvaise nouvelle sachant que dans ce genre d’applications, on s’attend à pouvoir travailler avec un grand nombre de variables. Cette information est représentée dans le graphique par les points (·) qui évalue le taux de fichiers où *GL2U* a pu répondre, et on peut s’apercevoir qu’au delà de trois variables de requêtes, le nombre de fichiers dégringole jusqu’à atteindre 0. Ainsi, la méthode que l’on propose, par transformation avec *MD* est une alternative plutôt intéressante à *GL2U* puisqu’elle nous donne les

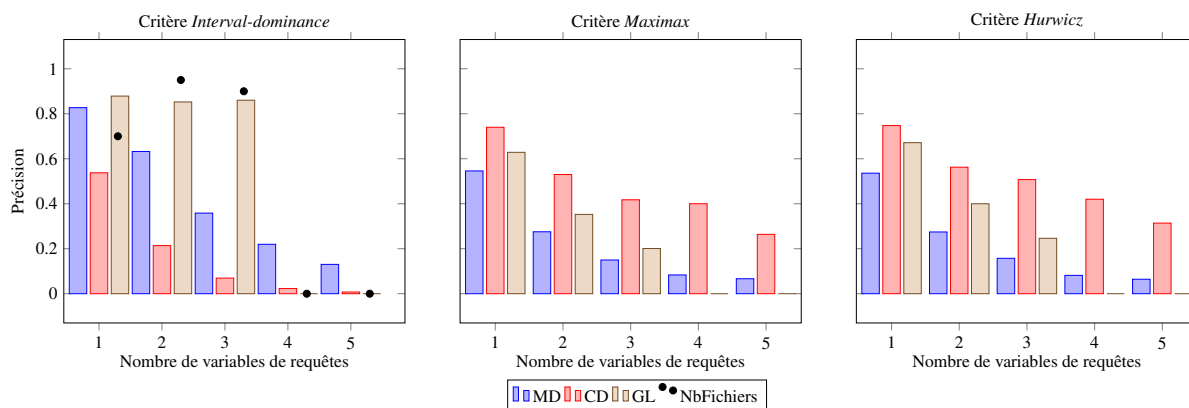


FIGURE 5 – Résultats sur les réseaux de type polyarbres

Critères	<i>MD</i>	<i>CD</i>	<i>GL2U</i>	
.794	.685	.36	.88	$\% rponses / D_Q $
<i>Interval-dominance</i>	.967	1	.891	% Inclusion
.36	.685	.36	.88	$\% rponses / D_Q $
<i>Maximax</i>	.546	.74	.629	% Inclusion

TABLE 3 – Ratio de réponses retournées sur l’ensemble possible des mondes disponibles VS Ratio de réponse incluses

second meilleurs résultats dans le cas du critère *Interval-dominance*. En ce qui concerne l’algorithme *CD*, nous reconnaissons la moindre efficacité qu’il présente en utilisant le critère *Interval-dominance*, et ce résultat s’explique par le nombre d’interprétations retournées par JavaBayes qui est trop important. Au contraire, nous observons l’effet inverse avec les trois autres critères, où *CD* nous donne les meilleurs résultats autant en termes de précision que d’inclusion des réponses.

Globalement, les trois algorithmes que l’on analyse ici, tendent à montrer le même comportement envers le nombre de variables de requêtes, ce qui insinue que les taux de précision baissent tous quand le nombre de variables augmente. Pourtant, on peut tout de même conclure que *GL2U* n’est définitivement pas à considérer puisque il atteint un timeout trop vite. Et que *MD* et *CD* nous offrent la possibilité de travailler l’inférence *MAP* sur plus de réseaux mais plus encore de travailler avec des réseaux plus grands et avec plus de variables de requêtes. Nous pouvons, en dernier lieu, inférer que l’utilisation du critère *Interval-dominance* n’est pas une bonne idée puisqu’il retourne un nombre d’interprétations trop important qui ne nous offrent ainsi pas les informations suffisantes pour prendre des décisions. En effet, avec un aussi grand nombre de réponses retournées, on ne peut rien faire dans des cas de classifications ou de recommandations. Enfin, en général, une approche optimiste (en utilisant le critère *Maximax*) est la meilleure variante des trois proposées.

6 Remarques finales

Dans ce travail, nous avons effectué une étude expérimentale sur les réseaux crédaux. Le but était d’améliorer le processus d’inférence *MAP*. La méthode que nous proposons consiste à transformer le réseau crédal en réseau possibiliste. Les avantages revendiqués sont l’amélioration de la complexité de calcul, l’utilisation de critères de choix qui nous permettent de restreindre le nombre de résultat retournés. Nous avons montré que, premièrement, utiliser l’algorithme approché (*GL2U*) sur les réseaux crédaux n’était pas calculatoirement intéressant étant donné la limite qu’il montre lors de plusieurs variables de requêtes. Ensuite, l’utilisation du critère *Maximax* améliore la qualité de l’algorithme *CD* de manière significative, et nous permet d’interroger de plus grands réseaux et en nombre plus important. En conclusion, l’algorithme *CD* représente une bonne alternative pour approximer l’inférence *MAP* dans les réseaux crédaux. Cette expérience, toutefois, n’est conduite que dans le cadre possibiliste quantitatif, nous allons par la suite expérimenter la même chose dans le cadre possibiliste qualitatif, et donner une comparaison des deux cadres (qualitatif vs. quantitatif).

Références

- [1] Baudrit, C et D Dubois: *Practical representations of incomplete probabilistic knowledge*. Compu-

- tational Statistics & Data Analysis, 51(1) :86–108, 2006. <https://doi.org/10.1016/j.csda.2006.02.009>.
- [2] Benferhat, S., A. Levray et K. Tabia: *On the Analysis of Probability-Possibility Transformations : Changing Operations and Graphical Models*. Dans *ECS-QARU 2015, Compiègne, France, July 15-17, 2015*.
- [3] Benferhat, S., A. Levray et K. Tabia: *Probability-Possibility Transformations : Application to Credal Networks*. Dans *SUM 2015, Québec City, QC, Canada, Sept 16-18, 2015. Proc.*, pages 203–219, 2015. http://dx.doi.org/10.1007/978-3-319-23540-0_14.
- [4] Cozman, F. G.: *Credal networks*. Artificial Intelligence, 120(2) :199 – 233, 2000. <http://www.sciencedirect.com/science/article/pii/S0004370200000291>.
- [5] Darwiche, A.: *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press, 2009, ISBN 978-0-521-88438-9.
- [6] Destercke, S., D. Dubois et E. Chojnacki: *Transforming Probability Intervals into Other Uncertainty Models*. Dans *EUSFLAT 2007 proc.*, tome 2, pages 367–373, Ostrava, Czech Republic, 2007.
- [7] Destercke, S, D Dubois et E Chojnacki: *Unifying practical uncertainty representations - I : Generalized p-boxes*. Int. J. Approx. Reasoning, 49(3) :649–663, 2008. <https://doi.org/10.1016/j.ijar.2008.07.003>.
- [8] Dubois, D., L. Foulloy, G. Mauris et H. Prade: *Probability-Possibility Transformations, Triangular Fuzzy Sets, and Probabilistic Inequalities*. Reliable Computing, pages 273–297, 2004.
- [9] Dubois, D et H Prade: *When upper probabilities are possibility measures*. Fuzzy Sets and Systems, 49(1) :65 – 74, 1992, ISSN 0165-0114. <http://www.sciencedirect.com/science/article/pii/016501149290110P>.
- [10] Klir, G. J. et J. F. Geer: *Information-Preserving Probability- Possibility Transformations : Recent Developments*. Dans *Fuzzy Logic*, pages 417–428. 1993.
- [11] Kwisthout, J.: *Treewidth and the Computational Complexity of MAP Approximations*. Dans *7th European Workshop, PGM 2014, Utrecht, The Netherlands, Sept. 17-19, 2014. Proc.*, pages 271–285, 2014. http://dx.doi.org/10.1007/978-3-319-11433-0_18.
- [12] Kwisthout, J., H.L. Bodlaender et L.C. van der Gaag: *The Necessity of Bounded Treewidth for Efficient Inference in Bayesian Networks*. Dans *ECAI 2010 - 19th, Lisbon, Portugal, August 16-20, 2010, Proc.*, pages 237–242, 2010. <http://www.booksonline.iospress.nl/Content/View.aspx?piid=17749>.
- [13] Levi, I.: *The enterprise of knowledge : an essay on knowledge, credal probability, and chance / Isaac Levi*. MIT Press Cambridge, Mass, 1980, ISBN 0262120828.
- [14] Masson, M. H. et T. Denoeux: *Inferring a possibility distribution from empirical data*. Fuzzy Sets and Systems, pages 319–340, 2006.
- [15] Mauá, D., C. P. de Campos, A. Benavoli et A. Antonucci: *Probabilistic Inference in Credal Networks : New Complexity Results*. J. Artif. Intell. Res. (JAIR), 50 :603–637, 2014.
- [16] Mauá, D.D, C.P. de Campos et F.G. Cozman: *The Complexity of MAP Inference in Bayesian Networks Specified Through Logical Languages*. Dans *Proc. of the 24th IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 889–895, 2015. <http://ijcai.org/Abstract/15/130>.
- [17] Pearl, J.: *Probabilistic reasoning in intelligent systems - networks of plausible inference*. Morgan Kaufmann series in representation and reasoning. Morgan Kaufmann, 1989.
- [18] Shimony, S.E. et C. Domshlak: *Complexity of probabilistic reasoning in directed-path singly-connected Bayes networks*. Artif. Intell., 151(1-2) :213–225, 2003. [http://dx.doi.org/10.1016/S0004-3702\(03\)00110-3](http://dx.doi.org/10.1016/S0004-3702(03)00110-3).
- [19] Slimen, Y. Ben, R. Ayachi et N. Ben Amor: *Probability-Possibility Transformation : - Application to Bayesian and Possibilistic Networks*. Dans *WILF13 2013, Genoa, Italy, Nov. 19-22, 2013.*, pages 122–130.
- [20] Sudkamp, T.: *On probability-possibility transformations*. Fuzzy Sets and Systems, pages 73–81, 1992.
- [21] Walley, P.: *Towards a unified theory of imprecise probability*. International Journal of Approximate Reasoning, 24(2–3) :125 – 148, 2000.
- [22] Zadeh, L. A.: *Fuzzy sets as a basis for a theory of possibility*. Fuzzy Sets and Systems, 100 :9–34, 1999.

Cadre Déclaratif Modulaire pour Représenter et Appliquer des Principes Éthiques

Fiona Berreby¹

Gauvain Bourgne¹

Jean-Gabriel Ganascia¹

¹ CNRS & Sorbonne Universités, UPMC Université Paris 06, LIP6 UMR 7606,
4 place Jussieu 75005 Paris, France

fiona.berreby@lip6.fr gauvain.bourgne@lip6.fr jean-gabriel.ganascia@lip6.fr

Abstract

Cet article examine l'utilisation de langages de haut niveau dans la conception d'agents autonomes éthiques. Il propose un cadre logique nouveau et modulaire pour représenter et raisonner sur une variété de théories éthiques, sur la base d'une version modifiée de l'Event Calculus, implémentée en Answer Set Programming. Le processus de prise de décision éthique est conçu comme une procédure en plusieurs étapes, capturée par quatre types de modèles interdépendants qui permettent à l'agent d'évaluer son environnement, de raisonner sur sa responsabilité et de faire des choix éthiquement informés. Notre ambition est double. Tout d'abord, elle est de permettre la représentation systématique d'un nombre illimité de processus de raisonnements éthiques, à travers un cadre adaptable et extensible en vertu de sa hiérarchisation et de sa syntaxe standardisée. Deuxièmement, elle est d'éviter l'écueil de nombreux travaux d'éthique computationnelle qui directement intègrent l'information morale dans l'engin de raisonnement général sans l'explicitement -alimentant ainsi les agents avec des réponses atomiques qui ne représentent pas la dynamique sous-jacente. Nous visons à déplacer de manière globale le fardeau du raisonnement moral du programmeur vers le programme lui-même.

1 Introduction

L'étude de la morale d'un point de vue computationnel a attiré l'intérêt croissant de chercheurs en intelligence artificielle[2]. En effet, l'autonomie croissante des agents artificiels et l'augmentation du nombre de tâches qui leur sont déléguées nous incitent à aborder leur capacité à traiter les restrictions et les préférences éthiques, que ce soit dans leur propre structure interne ou pour des interactions avec des utilisateurs humains. Des domaines aussi variés que la santé ou le transport posent des problèmes éthiques qui sont en ce sens particulièrement pressants, car ils peuvent

exiger des agents des prises de décisions dont les conséquences sont immédiates ou lourdes. L'éthique computationnelle peut aussi nous aider à mieux comprendre la morale et raisonner plus clairement sur les concepts éthiques qui sont employés dans des domaines philosophiques, juridiques et technologiques. Dans ce contexte, notre objectif est de fournir une architecture modulaire qui permette la représentation systématique et adaptable des principes éthiques. Pour ce faire, nous présentons un ensemble cohérent de modèles qui, ensemble, permettent à l'agent d'évaluer son environnement, d'intégrer des règles éthiques et déterminer à partir de la mise en œuvre de ces règles soit un plan d'action, soit une évaluation du comportement d'autres agents. Ceux-ci sont implémentés en Answer Set Programming¹, sur la base d'une version modifiée de l'Event Calculus. Ainsi, notre approche est une approche logique de l'éthique, qui existe en parallèle à d'autres telles que l'éthique par étude de cas ([3][2]), ou l'éthique par conception ([5]).

Nous avons choisi l'utilisation de la *logique non monotone* car son étude a été proposée comme moyen de gérer le genre de généralisations révocables qui caractérisent souvent le raisonnement de sens commun et qui sont mal capturées par les systèmes logiques classiques [17]. Le terme couvre une famille de cadres formels conçus pour appréhender le type d'inférence où aucune conclusion n'est définitive, mais reste ouverte à la modification à la lumière d'informations complémentaires. Ce type de raisonnement par défaut est constitutif du raisonnement éthique. Des facteurs tels que la présence d'options alternatives, de conséquences indirectes ou de circonstances atténuantes peuvent renverser le jugement éthique d'une action. En conséquence, les langages non monotones sont particulièrement adaptés à la modélisation du raisonnement éthique.

1. Pour une description de l'Answer Set Programming, voir [21].

L'article est structuré comme suit. Nous commençons par présenter les concepts philosophiques pertinents ainsi que les travaux connexes [Sect.2], puis nous présentons l'architecture du cadre [Sect.3]. Ensuite, nous définissons et discutons de chaque modèle [Sects.4-7], puis illustrons leur implémentation à l'aide d'un exemple jouet [Sect.8], et concluons [Sect.9].

2 Motivation

2.1 Théories Éthiques

L'étude de l'éthique est l'étude des croyances que les gens peuvent ou devraient avoir pour contrôler leur comportement. Une classification tripartite standard divise le champ entre la *méta-éthique*, qui concerne le statut ontologique des concepts éthiques, l'*éthique appliquée*, qui concerne l'application des règles morales à des environnements particuliers, et l'*éthique normative*, qui traite de la définition, de la comparaison et de l'explication de conceptions éthiques [10]. Le présent travail présente un intérêt pour l'éthique appliquée en ce sens qu'il présente un schéma de conception d'agents artificiels contraints par l'éthique qui peuvent agir dans une variété de domaines appliqués. Il présente aussi un intérêt pour l'éthique normative car son but est de modéliser les processus qui sous-tendent la prise de décision éthique normative, avec la possibilité de confronter des perspectives différentes. Il se concentre sur deux de ses principales branches : l'éthique conséquentialiste et l'éthique déontologique.

Le Bien et Le Juste

Les éthiques conséquentialistes s'articulent autour de l'idée que les actions doivent être évaluées en fonction de leur conséquences, et ne peuvent être justes ou injustes qu'en vertu de ce qu'elles produisent. Une action moralement juste est celle qui produit un bon, ou le meilleur, état de choses. Hors, afin de déterminer la justesse d'une action, les conséquentialistes doivent d'abord établir ce qui constitue un *bon* état de choses, c'est-à-dire déterminer ce qu'on appelle plus largement 'Le Bien' [1]. Cela leur permet ensuite d'affirmer que des actions font partie du 'Juste' dans la mesure où elles augmentent le Bien. Les théories conséquentialistes suivent donc, s'appuient et finalement dépassent, les théories du Bien. Les désaccords entre conséquentialistes sur ce qui constitue le Bien, ont engendré diverses traditions et doctrines conséquentialistes. L'*utilitarisme* voit le Bien résulter de la maximisation du bien-être collectif, l'*atruisme éthique* du bien-être des autres, l'*égoïsme éthique* de l'intérêt personnel, l'*utilitarisme des droits* du respect des droits individuels.

Les théories déontologiques (du grec *deon*, "devoir") prétendent que la valeur morale d'une action est déterminée (au moins en partie) par une caractéristique intrinsèque

de l'action. Habituellement, cette caractéristique est une obligation ou une interdiction. Par exemple, une règle déontologique peut indiquer que le mensonge est contraire à l'éthique, ce qui implique que tout énoncé qui contient un mensonge est interdit. Parce qu'une action est jugée juste ou non en fonction de sa conformité avec une norme ou un devoir, son évaluation éthique est au moins en partie indépendante de ses conséquences. Le Juste est ici prioritaire sur le Bien : une action peut être injuste pour le déontologue même si elle maximise le Bien, et juste même si elle le minimise. Les tentatives de définition du Bien seront désormais appelées *théories du Bien*, et les tentatives de définition du Juste, qu'elles soient conséquentialistes ou déontologiques, seront appelées *théories du Juste*.

2.2 Travaux Connexes

Un certain nombre de travaux ont proposé des modèles informatiques de théories éthiques, dont la déontologie basée sur les devoirs et les règles [3] [5] [26], la déontologie de commande divine [10], le conséquentialisme [16] [20], ou l'instanciation de normes [29]. Il existe aussi des approches de vérification formelle [13][14]. Cependant, certains de ces modèles tendent à intégrer directement l'information éthiques au sein du processus de prise de décision de l'agent, sans pour autant générer un raisonnement moral. Bien qu'ils réussissent à exécuter des implémentations directes de restrictions uniques, ils ne fournissent pas une représentation explicite des relations de causalité ou des processus de pensée éthique, limitant ainsi leur applicabilité et leur portée. Par exemple, en utilisant une logique prospective, Pereira et al. [26] modélisent une règle déontologique qui prohibe le meurtre intentionnel par la règle '*falsum* \leftarrow *intentionalKilling*.' Or ils déterminent si '*intentionalKilling*' vaut pour une action en indiquant atomiquement si cette action l'implique, utilisant des règles de la forme '*intentionalKilling* \leftarrow *end(A,iKill(Y))*.' où A est l'action évaluée. Le problème avec cette approche est que l'évaluation éthique est *indiquée* par des énoncés spécifiques à l'action, plutôt qu'*extraite* par une forme de compréhension de l'environnement et des règles éthiques en place. Il n'y a pas de représentation de la causalité, de sorte que l'action et ses conséquences ne sont pas liées dynamiquement ; leur relation est déclarée plutôt que déduite. Par conséquent, aucune notion de responsabilité éthique ne peut être élaborée sur cette base. En outre, les règles données manquant de puissance expressive, un nouveau programme est nécessaire pour modéliser chaque nouveau cas. Ces formalismes ne peuvent donc pas contraster différentes théories, ni expliciter leurs hypothèses.

Des travaux plus récents ont exploré de manière intégrale l'architecture des jugements éthiques [11] [9], représentant explicitement ces processus de raisonnement. Ce travail s'inscrit dans cette poursuite.

3 Schéma Structurel

3.1 Modèles et Modularité

La représentation explicite du raisonnement éthique permet à un agent d’informer son processus de prise de décision ou de juger du comportement des autres. Pour y parvenir, il ‘teste’ les actions possibles dans des simulations spécifiques afin d’évaluer leurs conséquences ou leur mérite éthique inhérent. Le résultat de la simulation donne alors un ensemble d’actions acceptables ou inacceptables, qui dicte le comportement à venir. Le cadre présenté ici est concerné par ce processus d’évaluation, plutôt que par ce que l’application de cette évaluation.

Le processus éthique est appréhendé comme une procédure en quatre étapes définie par quatre types de modèles interdépendants : un *modèle d’action*, un *modèle de causalité*, un *modèle du Bien*, et un *modèle du Juste*. Les deux premiers modèles produisent une compréhension entièrement non-éthique du monde, les deux suivants y superposent une compréhension éthique du monde. Le modèle d’action présenté ici, et qui constitue la base du cadre, est basé sur une version modifiée de l’Event Calculus à la manière de [9]. La situation est représentée en termes de *fluents*, des propriétés du monde variant dans le temps, et d’*événements* qui modifient ces fluents. Nous définissons ces modèles ici, comme illustré dans la figure 1.

Un *modèle d’action* \mathbb{A} permet à l’agent de représenter son environnement et les changements qui s’y déroulent. Il prend comme entrée un *ensemble d’actions effectuées*. Il est composé d’une *situation initiale* contenant les fluents vrais à $T=0$, une *spécification d’événements* contenant un ensemble d’événements et de relations de dépendance, et un *moteur d’événement* qui permet à la simulation d’évoluer. Il génère une *trace d’événements* de chaque simulation qui désigne pour chaque moment les événements qui s’y produisent et des fluents qui y sont vrais.

Un *modèle de causalité* \mathbb{C} piste les conséquences des actions, rendant possible un raisonnement sur la responsabilité et l’imputabilité des agents. Il prend comme entrée la *trace d’événements* produite par le modèle d’action et une *spécification d’événements* contenant un ensemble d’événements et de relations de dépendance. Il est composé d’un *moteur causal* qui permet la création d’un arbre causal représentatif de la simulation. Il génère une *trace causale* de chaque simulation qui désigne pour chaque moment les liens de cause à effets qui existent entre événements et fluents.

Un *modèle du Bien* \mathbb{G} donne une appréciation de la valeur éthique intrinsèque de finalités ou d’événements. Il est composé d’une *spécification de modalités*, d’une *spécification éthique d’événements* composée d’un ensemble d’événements et d’un ensemble de relations de dépendance éthique, et d’une ou plusieurs *théories du Bien*. Il génère une *évaluation du Bien*, évaluant les événements comme

plus ou moins en accord avec le Bien.

Un *modèle du Juste* \mathbb{R} détermine l’action la plus juste selon des circonstances données. Il prend comme entrée la *trace causale* produite par le modèle de causalité et, dans le cas où une théorie du Juste considérée contient des principes conséquentialistes, une *évaluation du Bien* produite par le modèle du Bien. Il est composé d’une ou plusieurs *théories du Juste* et, dans le cas où une théorie du Juste considérée contient des principes déontologiques, un *ensemble de spécifications déontologiques*. Il génère une *évaluation du Juste*, évaluant les actions comme plus ou moins en accord avec le Juste -et donc admissibles ou non.

Ces quatre types de modèles sont interdépendants à degrés variables. Les modèles du Bien et du Juste reposent toujours sur un modèle d’action et un modèle de causalité. Mais alors qu’un modèle de causalité est toujours nécessaire, la formulation particulière du moteur causal peut varier, par exemple pour représenter différentes définitions des causes et des conséquences. Parce que le moteur d’événement constitue la base du cadre, il est toutefois proposé qu’il soit unique et invariable. En ce qui concerne les modèles éthiques, avoir un modèle du Bien est nécessaire pour modéliser les théories du Juste qui sont conséquentialistes, ainsi que celles qui sont déontologiques lorsqu’elles comportent des contraintes conséquentialistes. Des interdépendances peuvent également exister au sein d’un type de modèle, en particulier dans le cas des théories du Juste qui peuvent faire appel l’une à l’autre. La hiérarchie bien définie entre les différents types de modèles donne au cadre la capacité non seulement de modéliser, mais aussi de comparer un nombre potentiellement illimité de théories éthiques. De plus, la compartimentation des différents types de processus permet leur analyse spécifique. Remplacer un modèle spécifique tout en maintenant les autres rend possible l’examen individualisé de ses ramifications. Sur la base de ces modèles, le *cadre d’évaluation éthique* est défini par :

$$\mathbb{F} = \langle \mathbb{A}_i, \mathbb{C}_i, \mathbb{G}_i, \mathbb{R}_i \rangle$$

Étant donné un cadre d’évaluation éthique \mathbb{F} et un ensemble \mathcal{A} d’actions exécutées α , nous définissons l’ensemble des actions admissibles comme suit :

$$\text{Admissible}(\mathbb{F}, \mathcal{A}) = \{ \alpha \in \mathcal{A} / \mathbb{A}_i, \mathbb{C}_i, \mathbb{G}_i, \mathbb{R}_i \models \text{admissible}(\alpha) \}$$

4 Moteur d’évènement

L’Event Calculus Adapté Le moteur d’évènement présenté ici correspond à l’Event Calculus complet décrit dans [27], avec les ajouts suivants. Pour répondre aux particularités de la modélisation de scénarios éthiques complexes, nous introduisons des événements automatiques en plus des actions. Ces événements automatiques se produisent lorsque toutes leurs conditions préalables, sous la forme de fluents, tiennent, sans apport de l’agent. En outre, nous

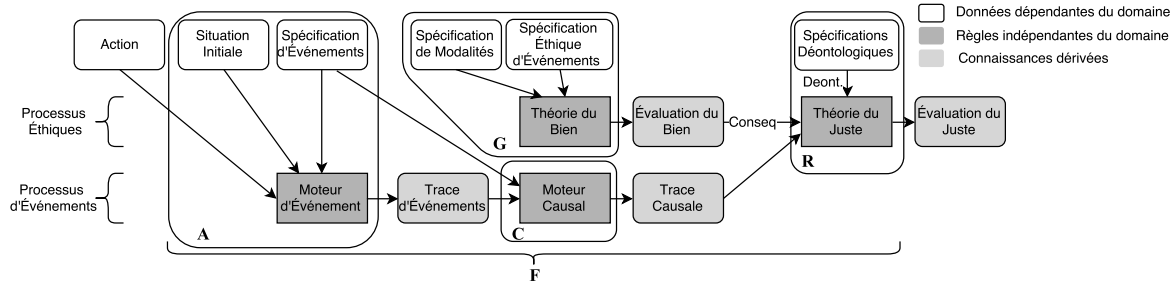


FIGURE 1 – Modèles et Modularité

faisons une distinction entre les fluents inertiels, qui restent vrais jusqu'à ce qu'ils soient terminés par une occurrence d'événement, et les fluents non inertiels qui sont vrais pour un moment seulement [24]. Enfin, nous introduisons un ensemble de simulations qui permettent à l'agent de simuler séparément et simultanément les effets de différentes actions sur le même scénario. Nous désignons l'ensemble des fonctions et des constantes comme suit : \mathcal{S} est un ensemble de simulations, \mathcal{T} un ensemble de moments ; \mathcal{F} un ensemble de fluents, \mathcal{A} un ensemble d'actions, \mathcal{U} un ensemble d'événements automatiques et \mathcal{E} un ensemble d'événements où $\mathcal{E} \equiv \mathcal{A} \cup \mathcal{U}$.

Axiomes d'Effets Les prédicats suivants caractérisent le comportement des fluents qui contribuent à la création d'événements. `initially(F)` indique que F est vrai initialement ; `effect(E,F)` indique que E peut causer F ; `initiates(S,E,F,T)` indique que E initie F à T dans S (et F n'est pas la négation d'un fluent) ; `terminates(S,E,F,T)` indique que E termine F à T dans S ; `clipped(S,F,T)` indique que F est stoppé à T dans S ; `nonInertial(F)` qualifie les fluents qui ne sont pas contraints par la loi d'inertie ; `holds(S,F,T)` indique que F est vrai à T dans S . Ces prédicats nous permettent d'axiomatiser les principes qui gouvernent les fluents : un fluent est vrai à T dans S si il a été initié par une occurrence d'événement à $T-1$ dans S ; un fluent vrai à T dans S reste vrai jusqu'à l'occurrence d'un événement qui le termine, sauf s'il est non-inertiel, auquel cas il est vrai à T seulement.

```
holds(S,F,0):-initially(F),sim(S).
initiates(S,E,F,T):-
  effect(E,F),occurs(S,E,T),not negative(S,F).
negative(S,neg(F)):-effect(E,neg(F)),sim(S).
terminates(S,E,F,T):-
  effect(E,neg(F)),occurs(S,E,T).
clipped(S,F,T):-terminates(S,E,F,T).
holds(S,F,T):-
  initiates(S,E,F,T-1),time(T).
holds(S,F,T):-
  holds(S,F,T-1),not clipped(S,F,T-1),
  not nonInertial(F),time(T).
```

Axiomes de Préconditions Les prédicats suivants caractérisent le comportement d'événements qui déterminent l'état des fluents. `prec(F,E)` indique que F est une précondition de E ; `incomplete(S,E,T)` indique que E est incomplet à T dans S ; `possible(S,E,T)` indique que E est possible à T dans S ; `occurs(S,U,T)` indique que U se produit à T dans S ; `occurs(S,A,T)` indique que A se produit à T dans S . Ces prédicats nous permettent d'axiomatiser les principes qui régissent l'occurrence des événements : un événement automatique se produit à T dans S si toutes ses préconditions sont vraies à T dans S ; une action se produit à T dans S si toutes ses préconditions sont vraies et qu'un agent effectue A à T dans S .

```
incomplete(S,E,T):-
  prec(F,E),not holds(S,F,T),sim(S),time(T).
possible(S,E,T):-
  not incomplete(S,E,T),sim(S),event(E),time(T).
occurs(S,U,T):-possible(S,U,T),auto(U).
occurs(S,A,T):-
  possible(S,A,T),performs(S,D,A,T),act(A).
```

5 Moteur Causal

Axiomes de Causalité En se basant sur l'architecture de l'Event Calculus, nous définissons la causalité en termes de conséquences. Cela nous permet de générer une trace fonctionnelle et dynamique des liens causaux. Nous la définissons de la manière suivante.

Un fluent F est une *conséquence* d'un événement E si E produit F , et les deux sont vrais. Un événement E est une *conséquence* d'un fluent F si F est une précondition à E , et les deux sont vrais.

Cette définition gère la possibilité qu'il y ait plus d'une précondition pour l'occurrence de E et que F ne soit pas considéré comme une cause de E si E ne se produit pas (par exemple car d'autres préconditions n'ont pas été remplies). Pour la modéliser, nous définissons le prédicat `cons(S,E1,T,E2)`, qui indique que l'événement $E2$ est une conséquence de l'événement $E1$ qui s'est produit dans S à T . Le moment référencé est le moment auquel s'est produit le *premier* événement de la chaîne causale. Une

chaîne causale est composée d'une série de fluents et d'événements, mais le début et la fin d'une chaîne causale sont toujours des événements.

```

cons(S, E, T, F) :-
    occurs(S, E, T), effect(E, F), holds(S, F, T+1).
cons(S, F, T, E) :-
    occurs(S, E, T), prec(F, E), holds(S, F, T).
cons(S, E1, T1, E3) :-
    cons(S, E1, T1, C2), cons(S, C2, T2, E3),
    event(E1), event(E3), T2 > T1.

```

6 Théories du Bien

Dans cette section, nous présentons deux modes de définition du Bien, relatifs aux droits aux valeurs. Ces modes sont interchangeables et peuvent également être combinés. Nous présentons ensuite un modèle pour quantifier le bien une fois qu'il a été qualifié, ce qui lui permet à la fois d'être intégré dans les théories du Juste et donne aux événements des poids significatifs. Les droits, les valeurs ou d'autres moyens de définir le bien sont appelés *modalités*.

6.1 Qualifier le Bien

6.1.1 Droits

L'*utilitarisme des droits* de Nozick postule que le Bien à maximiser consiste en la non violation des droits [25]. Un droit peut être défini comme une "*revendication justifiée que les individus et les groupes peuvent faire sur d'autres individus ou sur la société; avoir un droit c'est être en mesure de déterminer par ses choix, ce que les autres doivent ou ne doivent pas faire*" [7]. Cette définition capture le fait qu'un droit indique à la fois un état des choses pour la personne concernée (l'exercice du droit) et une contrainte imposée aux autres (l'interdiction de violer le droit). Nous définissons les règles de sorte qu'un événement impliquant des personnes et qui viole un droit est *mal* par rapport à ce droit, et un événement qui implique des personnes mais ne viole pas un droit est *bien* par rapport à ce droit. Un événement peut être mal par rapport à un droit et bien par rapport à un autre. Cependant, aucun événement impliquant des personnes n'est neutre en ce qui concerne les droits : il respecte ou non chaque droit. Ce principe du tiers exclu est rendu explicite par l'utilisation de la négation par l'échec dans la règle. Notez que les droits sont définis par `right(M)` dans la *spécification de modalités*.

```

bad(E, X, M) :- effect(E, involves(X)),
    effect(E, neg(M)), right(M).
good(E, X, M) :- effect(E, involves(X)),
    not effect(E, neg(M)), right(M).

```

6.1.2 Valeurs

Une théorie fondée sur les valeurs fournit également un moyen efficace d'évaluer le mérite initial des événements. Une valeur peut être définie comme "*une conception, explicite ou implicite, distinctive d'un individu, ou caractéristique d'un groupe, de ce qui est souhaitable et qui influence la sélection des modes, des moyens et des fins d'action disponibles.*" [19]. Une valeur est donc un type d'entité indépendante que peuvent assumer les actions et leurs conséquence. Les valeurs peuvent être générales, ou spécifiques à différents contextes, tels que le lieu de travail ou l'éducation des enfants. Nous définissons les règles de sorte qu'un événement qui démontre l'expression d'une valeur particulière est *bien* par rapport à cette valeur, et un événement qui démontre la négation d'une valeur est *mal* par rapport à cette valeur. Les autres événements sont considérés ni bons ni mauvais par rapport à celle-ci. Notez que les valeurs sont définies par `value(M)` dans la *spécification de modalités*.

```

good(E, X, M) :- effect(E, involves(X)),
    effect(E, displays(M)), value(M).
bad(E, X, M) :- effect(E, involves(X)),
    effect(E, neg(displays(M))), value(M).

```

6.2 Quantifier le Bien

Une fois déterminé le contenu du Bien, nous procédons à la quantification de ce contenu, c'est-à-dire à la 'pesée' des bonnes et mauvaises ramifications des événements. Nous définissons trois paramètres pour cela :

- Le nombre de personnes impliquées dans l'événement. Par exemple, un événement affectant cinq personnes comptera cinq fois plus qu'un événement en affectant une. Cette information est donnée dans les prédicats `good(E, X, M)` et `bad(E, X, M)` par `X`.
- La valeur relative des personnes impliquées dans l'événement. Par exemple, il est peut-être plus significatif de sauver des enfants plutôt que des adultes, ou de nuire à des personnes en bonne santé plutôt qu'à des patients en déclin. Ce paramètre est pris en compte par l'attribution à chaque groupe d'une valeur numérique, exprimée dans le prédicat `t_weight(E, G, N)` où `E` est un événement, `G` son groupe cible et `N` le poids donné.
- L'importance de la modalité affectée par l'événement. Par exemple, être bienfaisant est peut-être plus important qu'être poli, respecter le droit à la vie est peut-être plus important que respecter le droit de propriété. Ce paramètre est pris en compte par l'attribution à chaque modalité d'une valeur numérique, exprimé dans le prédicat `m_weight(M, N)` où `M` est la modalité et `N` le poids donné.

L'attribution de poids aux modalités et aux groupes est non triviale, et la méthode proposée ici en est une parmi d'autres, qui fait acte d'introduction. Il est possible, par

exemple, de l'enrichir en prenant en compte d'autres dépendances, telles que la corrélation entre certaines modalités et personnes (par exemple, l'autonomie pourrait être essentielle pour les adultes et la sécurité pour les enfants), ou l'importance de parties affectées non humaines (les animaux, l'environnement, etc.). L'étape suivante consiste à réunir tous les poids en un seul en agrégeant par un produit les différents paramètres dans les prédicats `weightedGood(E, N, M)` et `weightedBad(E, N, M)`.

Le poids global d'un événement correspond alors à la différence entre les sommes de toutes ses bonnes et mauvaises ramifications pondérées. Plus le poids d'un événement est grand, plus il participe au Bien; les événements qui ont des poids négatifs font plus de mal que de bien. Les poids sont donnés par le prédicat `weight(E, N)`. Ce prédicat permet l'intégration du Bien dans le Juste et participera à définir les théories du Juste. Il est à noter que les poids de cible et de modalité sont définis par `t_Weight(E, G, N)` et `m_Weight(M, N)` dans la *spécification de modalités*.

```
weightedGood(E, X*N1*N2, M) :- good(E, X, M),
    t_Weight(E, G, N1), m_Weight(M, N2).
weightedBad(E, X*N1*N2, M) :- bad(E, X, M),
    t_Weight(E, G, N1), m_Weight(M, N2).
weight(E, N1-N3) :-
    N1=#sum[weightedGood(E, N2, M1)=N2],
    N3=#sum[weightedBad(E, N4, M2)=N4],
    number(N1;N3), event(E).
```

7 Théories du Juste

7.1 Éthiques Consequentialistes

L'éthique conséquentialiste existe sous de nombreuses formes, allant des principes simples d'action aux théories complexes pour maximiser le bien. Nous en décrivons et modélisons cinq.

7.1.1 Proscription d'Actions Purement Préjudiciables

Le premier principe conséquentialiste affirme que les actions ayant des effets purement préjudiciables sont inadmissibles. Cette règle intuitive est pertinente pour la plupart des scénarios éthiques et peut compléter d'autres théories du Juste qui ne se concentrent que sur des actions à effets complexes. Pour mettre en œuvre cette règle, nous définissons les prédicats `badCons(S, A, T)` et `goodCons(S, A, T)`, qui indiquent respectivement qu'une action A qui se produit à T dans S provoque au moins une mauvaise ou une bonne conséquence. Nous déclarons ensuite qu'une action est inadmissible si elle n'a que des mauvaises conséquences. Toute autre action qui n'a pas été démontrée inadmissible est admissible.

```
badCons(S, A, T) :- act(A), cons(S, A, T, E), bad(E, X, M).
goodCons(S, A, T) :- act(A), cons(S, A, T, E), good(E, X, M).
```

```
imp(pureBad, A) :-
    badCons(S, A, T), not goodCons(S, A, T).
per(pureBad, A) :- act(A), not imp(pureBad, A).
```

7.1.2 Principe de la Moins Mauvaise Conséquence

Également appelé *maximum minimorum*, ce principe déclare qu'une action est inadmissible si sa pire conséquence est pire que la pire conséquence de toute autre action possible. Ce principe est particulièrement pertinent pour la prise de décision sous incertitude, où, sous 'l'hypothèse de la malchance', chaque action possible produirait sa pire conséquence. Ainsi l'agent ferait mieux de choisir l'alternative ayant le moins mauvais mauvais résultat [22]. Pour formaliser cette règle, nous déterminons d'abord une hiérarchie entre les conséquences des actions, afin d'ensuite indiquer la pire. Le prédicat `worse(E1, E2)` indique que la conséquence E1 d'une action est pire que la conséquence E2 de la même ou d'une autre action. Les prédicats `notWorstCons(S, A, T, E)` et `worstCons(S, A, T, E)` déterminent alors la limite basse d'un ordre partiel déterminé par le prédicat `worst`. Enfin, nous déclarons qu'une action A1 est inadmissible si sa pire conséquence E1 est pire que la pire conséquence E2 de toute autre action A2. Toutes les autres actions sont admissibles.

```
worse(E1, E2) :-
    cons(S1, A1, T1, E1), cons(S2, A2, T2, E2),
    weight(E1, N1), weight(E2, N2), N1 < N2.
notWorstCons(S, A, T, E1) :-
    act(A), cons(S, A, T, E1), cons(S, A, T, E2),
    worse(E2, E1), not worse(E1, E2).
worstCons(S, A, T, E) :- act(A), event(E),
    cons(S, A, T, E), not notWorstCons(S, A, T, E).
imp(leastBad, A1) :- worstCons(S1, A1, T1, E1),
    worstCons(S2, A2, T2, E2),
    worse(E1, E2), A1 != A2.
per(leastBad, A) :- act(A), not imp(leastBad, A).
```

7.1.3 Principe d'Analyse Coût-Avantage

Ce principe indique qu'une action est admissible si elle est globalement bénéfique, c'est-à-dire si ses bonnes conséquences l'emportent les mauvaises. Nous utilisons les prédicats `weightCons(S, A, T, E, N)`, qui détermine le poids N des conséquences individuelles E d'une action A qui s'est produite à T dans S, et `weightAct(A, N2)`, qui concatène ces poids², pour indiquer que A est inadmissible si ce poids total N2 est négatif, et admissible sinon.

```
weightCons(S, A, T, E, N) :-
    act(A), cons(S, A, T, E), weight(E, N).
weightAct(A, N2) :- act(A), number(N2),
    N2=#sum[weightCons(S, A, T, E, N1)=N1].
imp(benCosts, A) :- weightAct(A, N2), number(N2), N2 < 0.
per(benCosts, A) :- act(A), not imp(benCosts, A).
```

2. sans spécifier de situation car une seule action est réalisée dans chaque situation.

7.1.4 Utilitarisme de l'Acte

"C'est le plus grand bonheur du plus grand nombre qui est la mesure du bien et du mal." J. Bentham, 1776 [8].

L'utilitarisme de l'acte exige que l'on évalue une action directement selon le *principe d'utilité*, qui stipule que l'action moralement correcte est celle qui a les meilleures conséquences globales pour le bien-être ou l'utilité de la majorité des parties concernées [8]. Une action est donc admissible si, compte tenu de toutes les autres actions disponibles, elle a les meilleures conséquences dans l'ensemble. En utilisant le prédicat `weightAct` défini ci-dessus, nous déterminons un ordre de préférence entre les actions dans le domaine et déclarons qu'une action A1 est inadmissible s'il existe une autre action A2 dont le poids est supérieur. Toute autre action est admissible.

```
imp(actUti,A1):-  
  weightAct(A1,N1),weightAct(A2,N2),N1<N2.  
per(actUti,A):-act(A),not imp(actUti,A).
```

7.1.5 Utilitarisme de la Règle

"Chaque acte, dans la vie morale, tombe sous une règle; et nous devons juger la moralité ou l'immoralité de l'acte, non par ses conséquences, mais par les conséquences de son universalisation -c'est-à-dire par les conséquences de l'adoption de la règle sous laquelle cet acte tombe." J. Hospers, 1975 [28]

L'utilitarisme de la règle évalue une action en deux temps. La première étape consiste à évaluer les règles morales sur la base du principe d'utilité : il s'agit de déterminer si une règle ou un ensemble de règles morales engendrera les meilleures conséquences, supposant que tout ou la plupart des agents s'y plient. Dans la vie quotidienne, de telles règles peuvent inclure 'Ne pas voler', ou 'Gardez ses promesses'. La deuxième étape consiste à évaluer les actions individuelles relativement à ce qui a été justifié au cours de la première étape. Une action n'est admissible que si la règle à laquelle elle appartient respecte le principe d'utilité, outre son propre respect du principe. Par exemple, si 'Ne pas voler' est une règle adoptée, le vol sera toujours inadmissible, même dans l'instance où un vol produirait la plus grande utilité (par exemple, en alimentant un affamé). Le prédicat `ruleCount(R,N)` regroupe tous les poids N des actions qui appartiennent à une règle R; le prédicat `weightRule(R,N)` les concatène. Nous déclarons alors qu'une action A est inadmissible si elle est une instance d'une règle R globalement nuisible, c'est-à-dire dont les mauvaises conséquences l'emportent sur les bonnes, considérant toutes ses instanciations. Toute autre action est admissible. Il est à noter que les instances de règles et les

règles sont définies par `rule(R)` et `instance(A,R)` dans la spécification de modalités.

```
ruleCount(R,N):-  
  rule(R),instance(A,R),weightAct(A,N).  
weightRule(R,N):-  
  rule(R),number(N),N=#sum[ruleCount(R,N1)=N1].  
imp(ruleUti,A):-  
  act(A),instance(A,R),weightRule(R,N),N<0.  
per(ruleUti,A):-act(A),not imp(ruleUti,A).
```

7.2 Éthiques Déontologiques

Dans cette section, nous présentons trois doctrines déontologiques. Deux d'entre elles sont purement déontologiques, les codes de conduite et l'éthique kantienne. La doctrine du double effet comporte des contraintes conséquentialistes. Il est à noter que notre traduction logique de ces théories constitue une possibilité parmi d'autres.

7.2.1 Codes de Conduite

"J'apporterai mon aide à mes confrères ainsi qu'à leurs familles dans l'adversité. Que les hommes et mes confrères m'accordent leur estime si je suis fidèle à mes promesses; que je sois déshonoré et méprisé si j'y manque." Serment d'Hippocrate, [12]

Un code de conduite est un ensemble de règles qui décrit les obligations, les interdictions ou les responsabilités d'un individu, d'un groupe ou d'une organisation. Il spécifie les principes qui guident la prise de décision ou les procédures de ceux qui sont contraints par le code. Les codes de conduite varient dans leur portée et leur nature, allant des codes déontologiques professionnels aux commandements religieux. Le comportement et la moralité sont typiquement déterminés par un corps global, tel qu'une entreprise, un état ou un dieu. Nous illustrons ici ce type de contrainte en modélisant une règle commune qu'est l'interdiction de tuer. Une telle règle se trouve par exemple dans la Déclaration de Genève de l'Association Médicale Mondiale sous la forme de la déclaration «Je garderai le respect absolu de la vie humaine» [6], ou dans le Décalogue comme commandement 'Tu ne tueras point'(Exode 20 : 1-21). Nous modélisons une règle de ce type en déclarant qu'une action est inadmissible dans la mesure où elle provoque ou consiste en ce qui est interdit, ici, tuer.

```
imp(conduct,A):-act(A),cons(S,A,T,kill(N,G)).  
per(conduct,A):-act(A),not imp(conduct,A).
```

7.2.2 La Formule de la Fin en Elle-Même

"Agis de telle sorte que tu traites l'humanité aussi bien dans ta personne que dans la personne de tout autre toujours en même temps comme une"

fin, et jamais simplement comme un moyen." I. Kant, 1785 [18]

Cette formule est un élément d'éthique kantienne qui met l'accent sur la valeur intrinsèque de la vie humaine. C'est un impératif moral qui interdit d'utiliser les personnes comme moyen pour d'autres fins, les personnes étant des fins en elles-mêmes en vertu de leur nature d'êtres rationnels [18]. La formule contraste la valeur intrinsèque, qui est persistante et souveraine, avec la valeur instrumentale, qui dépend de ce qu'elle produit. Nous présentons le prédicat $\text{aim}(A, E)$ qui indique que le but de l'action A est de provoquer l'événement E et utilisons le fluent $\text{involves}(X)$ pour indiquer qu'au moins une personne est impliquée dans E . Nous déclarons alors qu'une action A est inadmissible si elle provoque un événement E qui implique au moins une personne, mais où E n'est pas un but de A . Toute autre action est admissible. Les buts sont définis par $\text{aim}(A, E)$ dans la *spécification déontologique*.

```
imp(kant, A) :- act(A), cons(S, A, T, E),
    effect(E, involves(X)), not aim(A, E).
per(kant, A) :- act(A), not imp(kant, A).
```

7.2.3 La Doctrine du Double Effet

'Rien n'empêche un acte d'avoir deux effets, dont un seul est voulu, tandis que l'autre est à côté de l'intention." T. Aquinas, 1485 [4]

La doctrine du double effet est un ensemble de critères éthiques utilisés pour évaluer la permissibilité éthique d'une action qui a à la fois de bonnes et de mauvaises conséquences [15]. Elle dicte qu'une personne peut licitement exécuter une action en sachant qu'elle aura des bons et mauvais effets, à condition que : 1) L'action elle-même soit bonne ou moralement neutre ; 2) Le mauvais effet ne soit pas directement voulu ; 3) Le bon effet résulte de l'acte et non du mauvais effet ; 4) Le bon effet soit plus important ou égal au mauvais effet [23]. $\text{imp}(\text{dde1}, A)$ proscrit une action si elle est intrinsèquement mauvaise, correspondant à la condition 1. $\text{imp}(\text{dde2}, A)$ proscrit une action si elle provoque un mauvais effet qui conduit à un bon effet. Cette règle correspond aux conditions 2 et 3, car nous estimons que l'utilisation d'un événement comme moyen pour arriver à un autre événement équivaut à vouloir le premier événement. $\text{imp}(\text{dde3}, A)$ proscrit une action si son effet global est mauvais. Cela correspond à la condition 4 qui équivaut au Principe d'Analyse Coût-Avantage. Toutes les autres actions sont permises par la doctrine.

```
imp(dde1, A) :- act(A), bad(A, X, M).
imp(dde2, A) :-
    act(A), cons(S, A, T, E1), cons(S, E1, T2, E2),
    bad(E1, X1, M1), good(E2, X2, M2).
imp(dde3, A) :- imp(benefitsCosts, A).
per(dde, A) :- act(A), not imp(dde1, A),
    not imp(dde2, A), not imp(dde3, A).
```

7.2.4 Discussion

La modélisation de ces théories souligne certaines de leurs caractéristiques. D'abord, nous distinguons deux types de théories du Juste : celles qui évaluent chaque action par rapport aux autres actions possibles, et celles qui évaluent chaque action de manière indépendante. Les principes *relatifs* comparent les actions et font un choix unique. Le *principe de la moins mauvaise conséquence*, l'*utilitarisme de l'acte* et l'*utilitarisme de la règle* sont de ce genre. Cependant, l'*utilitarisme de la règle* est particulier en ce sens que la permissibilité de *toutes* les actions est déterminée selon l'impact de chacune individuellement. Ainsi, toutes ou aucune des actions considérées sous une règle sont admissibles. À l'inverse, les principes *indépendants* évaluent chaque action en elle-même sans être affectés par les options disponibles. Ainsi, ils produisent des ensembles d'actions admissibles ou inadmissibles. Toutes les autres théories du Juste présentées ici sont de ce genre. Il est important de noter que différentes théories du Juste peuvent, et parfois doivent, se compléter. Par exemple, la *doctrine du double effet* ne dit rien sur les actions dont les effets sont purement mauvais et peut être utilement complétée par un principe conséquentialiste pour y remédier.

8 Preuve de Concept

Dans cette section, nous illustrons la manière dont chaque contrainte éthique décrite ci-dessus gère un dilemme éthique à l'aide d'un exemple mono-agent de prise de décision. C'est un exemple jouet qui n'a pas vocation à être réaliste, mais vise à montrer la diversité des évaluations éthiques possibles. Le code source complet est téléchargeable sur un service cloud³.

Un Dilemme Médical Un médecin (l'agent autonome) est en possession de trois différents traitements expérimentaux pour une maladie grave et handicapante. Chaque traitement a un taux de réussite différent.

- Pour 100 patients qui essaient le traitement Alpha, 15 sont guéris, 20 perdent leur vie et 65 restent inchangés.

- Pour 100 patients qui tentent le traitement Bêta, 30 sont guéris, 25 perdent leur vie et 45 restent inchangés.

- Pour 100 patients qui essaient le traitement Gamma, 50 sont guéris, 30 perdent leur vie et 20 restent inchangés. Cependant, sur les 50 patients guéris, 30 ne sont entièrement guéris que parce qu'ils ont également reçu une transplantation d'organe provenant de chacun des 30 morts. Sans la transplantation, ils auraient perdu la vie.

Le gain en termes de vies sauvées (c'est-à-dire les patients guéris moins les patients tués) pour chaque traitement est donc : **Alpha -5**; **Beta 5**; **Gamma 20**. Afin de choisir quel traitement est admissible, nous considérons

3. https://gitlab.lip6.fr/ganascia/AAMAS_2017

que le médecin simule séparément le fait de donner chaque traitement à un groupe de 100 personnes. Il administre le traitement à T=0 dans trois simulations respectives. Le cas est représenté de la manière suivante.

```

act(give(Z)):-treatment(Z).
prec(exist(Z),give(Z)):-act(give(Z)).
performs(sim1,doctor,give(alpha),0).
performs(sim2,doctor,give(beta),0).
performs(sim3,doctor,give(gamma),0).
initially(exist(Z)):-treatment(Z).
auto(cure(X,G);kill(X,G);null(X,G))
    :-people(X),group(G).
testgroup(group(alpha;beta;gamma)).
treatment(alpha;beta;gamma).
effect(give(alpha),posImpact(15,group(alpha))).
effect(give(alpha),negImpact(20,group(alpha))).
effect(give(alpha),noImpact(65,group(alpha))).
effect(give(beta),posImpact(30,group(beta))).
effect(give(beta),negImpact(25,group(beta))).
effect(give(beta),noImpact(45,group(beta))).
effect(give(gamma),posImpact(20,group(gamma))).
effect(give(gamma),negImpact(30,group(gamma))).
effect(give(gamma),noImpact(20,group(gamma))).
effect(kill(X,group(gamma)),
    posImpact(X,group(gamma))):-people(X).
prec(posImpact(X,G),cure(X,G)):-auto(cure(X,G)).
prec(negImpact(X,G),kill(X,G)):-auto(kill(X,G)).
prec(noImpact(X,G),null(X,G)):-auto(null(X,G)).
effect(cure(X,G),involves(X)):-auto(cure(X,G)),X>0.
effect(null(X,G),involves(X)):-auto(null(X,G)),X>0.
effect(kill(X,G),involves(X)):-auto(kill(X,G)),X>0.
nonInertial(posImpact(X,G);negImpact(X,G);
noImpact(X,G);involves(X)):-people(X),testgroup(G).

```

Le docteur considère en outre que venir en aide constitue le Bien, que guérir est venir en aide, tuer est l'opposé de venir en aide et laisser la maladie inchangée est n'avoir aucun impact. Il considère également que la valeur *venir en aide* (*helpfulness*) a un poids de 1 (ceci est ici trivial car il n'y a qu'une seule modalité) et que la vie de tous les patients est équivalente. Il pense aussi que donner chacun de ces traitements pourrait être généralisé comme la règle 'donner des remèdes incertains'. Enfin, son objectif en donnant des traitements est de guérir.

```

m_Weight(M,1):-modality(M).
t_Weight(E,G,1):-
    testgroup(G),effect(E,involves(X)).
modality(M):-value(M).
value(helpfulness).
effect(cure(X,G),displays(helpfulness)):-
    auto(cure(X,G)).
effect(kill(X,G),neg(displays(helpfulness))):-
    auto(kill(X,G)).
rule(uncertainCures).
instance(give(alpha;beta;gamma),uncertainCures).
aim(give(Z),cure(X,group(Z))):-
    treatment(Z),people(X).

```

TABLE 1 – Evaluations Basées sur la Valeur 'helpfulness'

	Alpha	Beta	Gamma
pureBad	Admis	Admis	Admis
leastBad	Admis	Inadmis	Inadmis
benCosts	Inadmis	Admis	Admis
actUti	Inadmis	Inadmis	Admis
ruleUti	Admis	Admis	Admis
conduct	Inadmis	Inadmis	Inadmis
kant	Inadmis	Inadmis	Inadmis
dde	Inadmis	Admis	Inadmis

TABLE 2 – Evaluations Basées sur le Droit 'life'

	Alpha	Beta	Gamma
pureBad	Admis	Admis	Admis
leastBad	Admis	Inadmis	Inadmis
benCosts	Admis	Admis	Admis
actUti	Admis	Inadmis	Inadmis
ruleUti	Admis	Admis	Admis
conduct	Inadmis	Inadmis	Inadmis
kant	Inadmis	Inadmis	Inadmis
dde	Admis	Admis	Inadmis

Les résultats de l'évaluation éthique sont résumés dans le tableau 1. Pour montrer comment le remplacement d'un module par un autre peut changer le processus d'évaluation, nous modélisons également un cas dans lequel le docteur fonde sa théorie du Bien non pas sur une *valeur* mais sur le respect du *droit* à la vie. Cette modalité est définie comme suit, et les résultats résumés dans le tableau 2.

```

modality(M):-right(M).
right(life).
effect(kill(X,G),neg(life)):-auto(kill(X,G)).

```

9 Conclusion

Le cadre présenté ici adapte et s'appuie sur l'Event Calculus pour permettre la modélisation de théories éthiques et de scénarios dans lesquels les appliquer. Défini en programmation logique, il présente une méthode et une implémentation de cette méthode. L'accent est mis sur la hiérarchie et la représentation explicite des processus de raisonnement qui déterminent la prise de décision éthique. Celles-ci permettent d'engendrer des règles avec un fort potentiel expressif qui confèrent aux agents la capacité de décider et d'expliquer leurs décisions, mais aussi de raisonner sur les actions d'autres agents. En outre, la confrontation de théories éthiques avec les contraintes logiques des langages de programmation éclaire ces théories, clarifiant leurs concepts, les relations qui les lient et les ambiguïtés potentielles qu'elles peuvent contenir. Nous envi-

sageons un certain nombre d'avenues futures pour développer le cadre actuel. Tout d'abord, nous cherchons à modéliser l'intention, qui n'est pour l'instant traitée qu'implicitement, et de modéliser les désirs des agents. Cela permettra aux agents de gérer des scénarios plus complexes et plus réalistes. En outre, nous avons l'intention de permettre la formulation de plans éthiques d'actions dans lesquels plus d'une action peut être évaluée dans une simulation, en travaillant vers un véritable domaine de planification. Enfin, nous visons à intégrer le cadre dans un système multi-agents, afin d'exploiter plus pleinement son potentiel pour faciliter la coopération ou l'intelligence collective.

Remerciements

Les auteurs remercient l'Agence Nationale de la Recherche (ANR) pour sa contribution financière sous la référence ANR-13-CORD-0006.

Références

- [1] Alexander, L et M Moore: *Deontological Ethics*. Dans Zalta, Edward N. (rédacteur) : *The Stanford Encyclopedia of Philosophy*. 2016.
- [2] Anderson, M et S Anderson: *Machine ethics*. Cambridge University Press, 2011.
- [3] Anderson, M, S Anderson et C Armen: *MedEthEx : a prototype medical ethics advisor*. 2006.
- [4] Aquinas, T: *Summa theologiae*. Xist Publishing, 2015.
- [5] Arkin, R: *Governing lethal behavior in autonomous robots*. CRC Press, 2009.
- [6] Association, World Medical et al.: *WMA declaration of Geneva*. International Journal of Person Centered Medicine, 4(3), 2015.
- [7] Beauchamp, T et J Childress: *Principles of Biomedical Ethics*. Principles of Biomedical Ethics. Oxford University Press, 2001, ISBN 9780195143317.
- [8] Bentham, J: *A fragment on government*. The Lawbook Exchange, Ltd., 2001.
- [9] Berreby, F, G Bourgne et J G Ganascia: *Modelling Moral Reasoning and Ethical Responsibility with Logic Programming*. Dans *Logic for Programming, Artificial Intelligence, and Reasoning*, pages 532–548. Springer, 2015.
- [10] Bringsjord, S et J Taylor: *The divine-command approach to robot ethics*. Robot Ethics : The Ethical and Social Implications of Robotics', MIT Press, Cambridge, MA, pages 85–108, 2012.
- [11] Cointe, N, G Bonnet et O Boissier: *Ethical Judgment of Agents' Behaviors in Multi-Agent Systems*. Dans *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pages 1106–1114, 2016.
- [12] Cos, Hippocrates of: *The Oath*. Loeb Classical Library, 1923.
- [13] Dennis, L, M Fisher, M Slavkovik et M Webster: *Formal verification of ethical choices in autonomous systems*. Robotics and Autonomous Systems, 77 :1–14, 2016.
- [14] Dennis, LA, M Fisher et A Winfield: *Towards verifiably ethical robot behaviour*. arXiv preprint arXiv :1504.03592, 2015.
- [15] Foot, P: *The problem of abortion and the doctrine of the double effect*. Applied Ethics : Critical Concepts in Philosophy, 2 :187, 2002.
- [16] Ganascia, J G: *Non-monotonic resolution of conflicts for ethical reasoning*. Dans *A Construction Manual for Robots' Ethical Systems*, pages 101–118. Springer, 2015.
- [17] Horty, J: *Nonmonotonic foundations for deontic logic*. Dans *Defeasible deontic logic*. Springer, 1997.
- [18] Kant, I: *Groundwork of the Metaphysic of Morals, trans. HJ Paton*. New York : Harper & Row, 1964.
- [19] Kluckhohn, C: *Values and value-orientations in the theory of action : An exploration in definition and classification*. 1951.
- [20] Kowalski, R: *Computational logic and human thinking : how to be artificially intelligent*. Cambridge University Press, 2011.
- [21] Lifschitz, V: *What Is Answer Set Programming ?*. Dans *AAAI*, tome 8, pages 1594–1597, 2008.
- [22] Luce, D et H Raffia: *Games and decisions*. Mineola, NY, 1985.
- [23] Mangan, J T: *Historical Analysis of the Principle of Double Effect, An*. Theological Studies, 10, 1949.
- [24] Miller, R et M Shanahan: *Some alternative formulations of the event calculus*. Dans *Computational logic : logic programming and beyond*. Springer, 2002.
- [25] Nozick, R: *Anarchy, state, and utopia*, 1974.
- [26] Pereira, LM et A Saptawijaya: *Modelling morality with prospective logic*. Dans *Progress in Artificial Intelligence*, pages 99–111. Springer, 2007.
- [27] Shanahan, M: *The event calculus explained*. Dans *Artificial intelligence today*. Springer, 1999.
- [28] Struhl, KJ et PS Rothenberg: *Ethics in perspective : a reader*. Random House, 1975.
- [29] Tufiş, M et J G Ganascia: *Grafting norms onto the BDI agent model*. Dans *A Construction Manual for Robots' Ethical Systems*. Springer, 2015.

Admissible generalizations of examples as rules

Philippe Besnard² Thomas Guyet^{3,1} Véronique Masson^{4,1}

¹ LACODAM – Team Inria/IRISA-UMR6074, France

² CNRS/IRIT-UMR5505, France

³ AGROCAMPUS-OUEST, France

⁴ Rennes University-1, France

besnard@irit.fr thomas.guyet@irisa.fr veronique.masson@irisa.fr

Abstract

Rule learning is a data mining task consisting in generating predictive rules from a tabular dataset of examples labeled by a class identifier. We focus on propositional rule induction for which some representative algorithms are CN2 or Ripper. Such algorithms extract rules of the form “IF *Conditions* THEN *class-label*”.

How to generalize examples is a central issue in a learning task. This work investigates the question of what is an admissible generalization of examples, to be called “rule elicitation”, from an expert’s viewpoint.

The overall rule learning process is viewed as a two-step process, depicted in Figure 1 :

1. Some subsets of data (some examples restricted to some attributes) are selected. In Figure 1, φ selects some possible subsets of data.
2. Each subset of data (\mathcal{A}, S) is assumed to elicit (*i.e.*, is generalized by) a unique rule. The idea of the “eliciting” function f from (\mathcal{A}, S) is to generate the rule π .

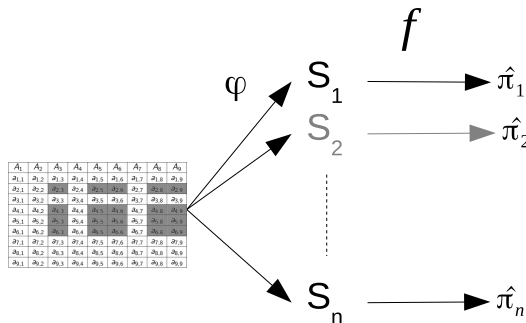


FIGURE 1 – Rule learning process: Abstract modeling

This article focuses on the f function, *i.e.*, how to elicit a rule from a subset of data. We do not tackle the question of

determining φ , *i.e.*, how to select data subsets. It is assumed that rules are generated from all possible selected subsets.¹

We introduce a formalization for generalization of examples as follows. A generalization of a set of examples S is an *admissible* rule $\pi = \widehat{S}$ where $\widehat{\cdot}$ is a closure-like operator. For any S and attribute A_i , generalizing S_i (*i.e.*, S restricted to the A_i attribute) is identified with mapping S_i to \widehat{S}_i with properties of $\widehat{\cdot}$ taken from the list of Kuratowski axioms for closure or weaker versions of them.

We prove that some significant classes of generalizations are captured by preclosure and capping operators.

The main intuition is that rule admissible subsets of the range $\text{Rng } A_i$ of an attribute A_i can be characterized as *choices* from the powerset of $\text{Rng } A_i$. Depending on what principles underly the actual choice, a different kind of closure (*i.e.*, a weakening of Kuratowski’s) embodies rule generalization through generalization.

We explicitly give families of choice functions that induce preclosure, cumulation, and capping operators. The role of these functions is to choose an admissible rule among the set of supersets of the examples S .

We illustrate such concrete functions for numerical attributes and relate them to intuitive notions of value interpolation, respectively neighborhoods and intervals.

We check the behaviour of the CN2 algorithm against the notions introduced in our formalization. Our experiments seem to show that rules are generated only from the extreme values of example sets, regardless of the actual example distribution. This behaviour corresponds to operators stronger than capping (but still weaker than Kuratowski’s). Yet, we should notice that these examples are specific cases of datasets with well-separated classes. In case of overlapping range of values, the rule choices that have been made depend on example distributions.

1. We have no practical objective, our purpose is not to design a new algorithm but to set a general framework to shed light on some characteristics of existing rule learning algorithms.

Le système SESAME : Spécifier des sémantiques argumentatives

Philippe Besnard^{2,1} Sylvie Doutre^{3,1} Dominique Longin^{2,1}

¹ IRIT (UMR 5505), France

² CNRS, Université de Toulouse, France

³ Université de Toulouse 1, France

{philippe.besnard, sylvie.doutre, dominique.longin}@irit.fr

Résumé

SESAME est un système qui, dans le cadre de l'argumentation abstraite, permet de spécifier une sémantique argumentative et qui s'attaque au problème de vérification : pour un graphe d'argumentation donné, tel ou tel ensemble d'arguments constitue-t-il une extension sous la sémantique spécifiée par l'utilisateur ? Les sémantiques qui peuvent être spécifiées au travers du système SESAME ne se limitent pas aux sémantiques déjà connues, un utilisateur peut en spécifier de nouvelles. Le système SESAME fournit en sortie une représentation logique de la sémantique saisie sous la forme d'une formule paramétrée. Appliquée à un ensemble d'arguments d'un graphe d'argumentation donné, la formule instanciée va être satisfiable si et seulement si cet ensemble est une extension pour ce graphe sous la sémantique spécifiée.

Abstract

SESAME is a tool which allows to specify argumentation semantics for abstract argumentation frameworks, and to tackle the verification problem : is a given set of arguments an extension under a semantics for a given argumentation framework ? The semantics which can be specified go well beyond the range of semantics already known, and the user can indeed specify brand new semantics of her own. The system then provides a logical encoding in the form of a parametrized formula. When applied to a given subset of arguments of a given argumentation framework, the instantiated formula will be satisfiable if and only if the set is an extension for the framework according to the specified semantics.

Mots-clefs. Argumentation, logique, spécification.

1 Introduction

Cette note est une courte introduction à SESAME (SEmantics Specification in Abstract arguMENTation), un système qui permet à son utilisateur de spécifier une sémantique argumentative, et qui en donne un codage logique sous la forme d'une formule propositionnelle paramétrée [4].

Appliqué à un système (graphe) d'argumentation donné $G = (\mathcal{A}, \mathcal{R})$, pour $S \subseteq \mathcal{A}$, le principe est que la formule instanciée est *satisfiable* si et seulement si S est une *extension* pour G selon la sémantique argumentative spécifiée par l'utilisateur.

La FIGURE 1 présente une capture d'écran de SESAME.

Rappel. Un graphe d'argumentation est un graphe orienté fini $(\mathcal{A}, \mathcal{R})$. Les éléments de \mathcal{A} sont appelés arguments et $b\mathcal{R}a$ se lit « b attaque a » pour a et b appartenant à \mathcal{A} .

Exemple de codage de la la sémantique stable.

Pour la sémantique stable de [6] par exemple, il faut construire une formule $\sigma_{(\mathcal{A}, \mathcal{R}), S}$ pour le cas $\sigma = \text{stable}$.

Définition Pour un graphe d'argumentation $(\mathcal{A}, \mathcal{R})$, une *extension stable* est un ensemble $S \subseteq \mathcal{A}$ tel que :

- il n'existe pas deux arguments a et b de S t.q. $a\mathcal{R}b$;
- pour tout argument $a \notin S$, il existe $b \in S$ t.q. $b\mathcal{R}a$.

La formule $\sigma_{(\mathcal{A}, \mathcal{R}), S}$ va traduire une reformulation de ces deux conditions :

- pour tout a dans S , pour tout b attaquant a , il est faux que b est dans S ;
- pour tout a dans \mathcal{A} , si a n'est pas dans S alors il existe b attaquant a tel que b est dans S .

Pourquoi reformuler ? Afin d'obtenir des conditions qui sont des combinaisons d'énoncés de base

« l'argument x est dans l'ensemble X »

devant se coder par des symboles propositionnels

$\varphi_{(x \in X)}$

ainsi notés pour des raisons de lisibilité (dans les faits, ce sont de simples atomes du langage propositionnel).

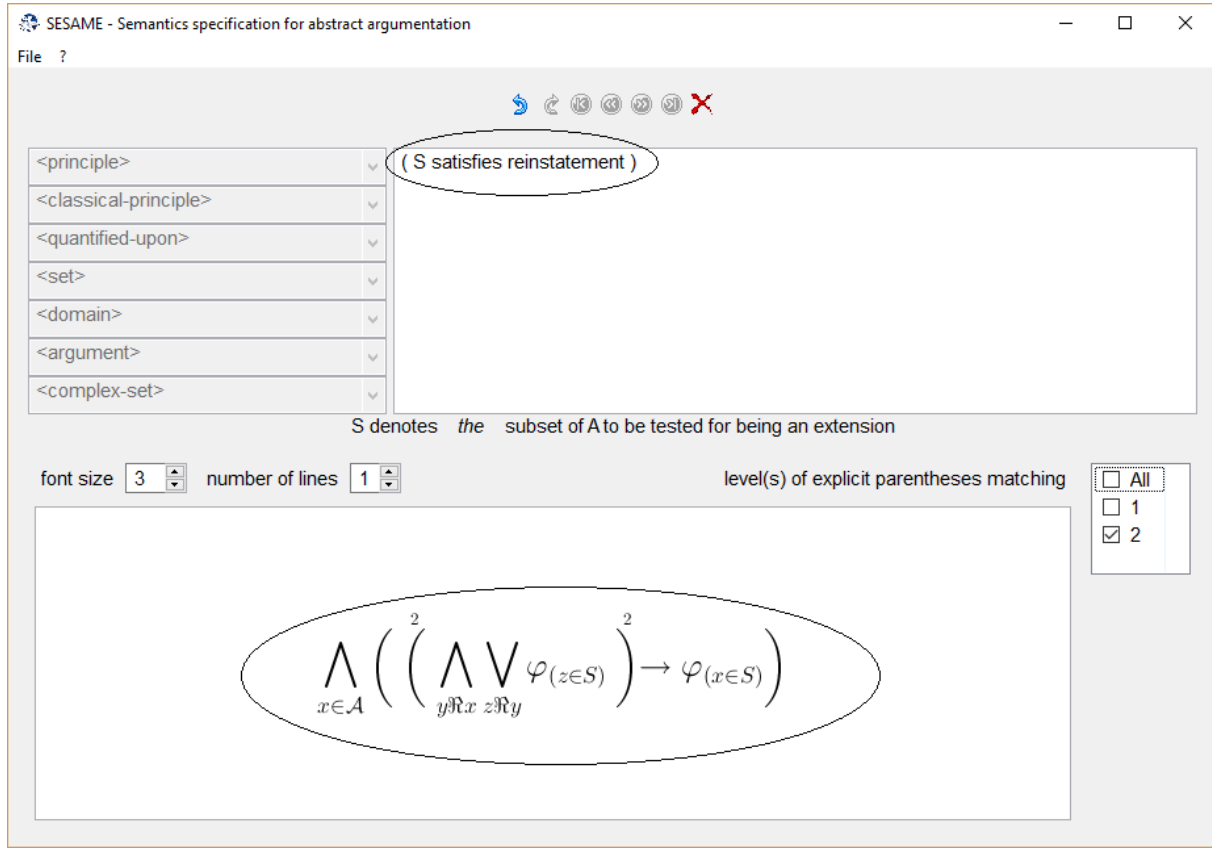


FIGURE 1 – Sémantique argumentative « S satisfait le principe de reinstatement » et la formule résultante $\sigma_{(\mathcal{A}, \mathcal{R}), S}$ (les chiffres visent à faciliter la visualisation de l'appariement des parenthèses)

Ainsi, SESAME génère une sous-formule (de $\sigma_{(\mathcal{A}, \mathcal{R}), S}$) exprimant que S remplit les deux conditions requises pour être une extension stable :

$$\left(\bigwedge_{a \in S} \bigwedge_{b \mathcal{R} a} \neg \varphi(b \in S) \right) \wedge \bigwedge_{a \in \mathcal{A}} \left(\neg \varphi(a \in S) \rightarrow \bigvee_{b \mathcal{R} a} \varphi(b \in S) \right)$$

Enfin, l'ensemble Φ_S signifiant que tous les éléments de S sont dans S et que tous les arguments hors de S ne sont pas dans S est défini par¹

$$\Phi_S \stackrel{\text{def}}{=} \{ \varphi(a \in S) \mid a \in S \} \cup \{ \neg \varphi(a \in S) \mid a \in \mathcal{A} \setminus S \}$$

d'où une formule $\sigma_{(\mathcal{A}, \mathcal{R}), S}$ pour les extensions stables :

$$\left(\bigwedge_{a \in S} \bigwedge_{b \mathcal{R} a} \neg \varphi(b \in S) \right) \wedge \bigwedge_{a \in \mathcal{A}} \left(\neg \varphi(a \in S) \rightarrow \bigvee_{b \mathcal{R} a} \varphi(b \in S) \right) \wedge \bigwedge \Phi_S$$

1. Puisqu'une condition (sémantique autre que la sémantique stable) pourrait être vraie pour un sous-ensemble de S mais pas S , il faut préciser qu'on spécifie bien tout S (et rien que S).

Au final, le système SESAME demande à l'utilisateur de spécifier une sémantique argumentative (qui n'est pas nécessairement déjà connue) sous forme d'une combinaison de principes (notion introduite par [2] dans le domaine) — à l'exception de $\bigwedge \Phi_S$ qui reste transparent pour l'utilisateur.

2 Spécifier une sémantique

L'élément central du processus de spécification d'une sémantique argumentative via SESAME est un arbre de décomposition selon une certaine grammaire. L'utilisateur développe cet arbre en cliquant sur des boutons dans la fenêtre principale, qui affiche les développements possibles des non-terminaux apparaissant dans l'arbre partiellement développé.

De fait, la gamme des sémantiques argumentatives supportées par le système SESAME est déterminée par une grammaire hors contexte telle que les énoncés primitifs sont, comme mentionnés plus haut, de la forme « l'argument x est dans l'ensemble X ». A cet effet, la grammaire dispose

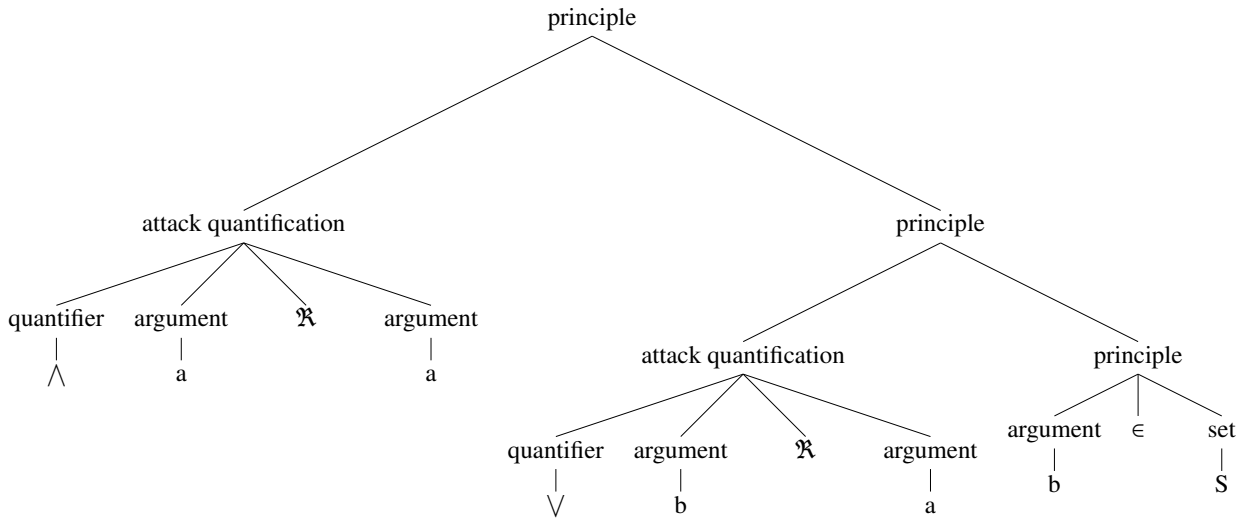


FIGURE 2 – Arbre de décomposition « tout argument s’attaquant lui-même est attaqué par au moins un argument de S »

de terminaux a, b, c, \dots et $\mathcal{A}, S, S_1, S_2, \dots$ (Attention, \mathcal{A} désigne l’ensemble des arguments du graphe alors que S désigne *le* sous-ensemble candidat au statut d’extension — par exemple, S est une extension stable si et seulement si S satisfait \dots). La grammaire permet divers types de combinaisons, booléennes et aussi par quantification.

Concrètement, quand l’utilisateur clique sur le bouton ⟨principe⟩, apparaît un menu déroulant qui indique quelles chaînes peuvent remplacer ⟨principe⟩, à savoir « il est faux que ⟨principe⟩ » ou « ⟨principe⟩ and ⟨principe⟩ » ou \dots

Après chaque choix effectué par l’utilisateur le système SESAME d’une part met à jour une description (en pseudo langage naturel) de l’arbre de décomposition et, d’autre part, affiche (en \LaTeX) la version actualisée de la formule en cours de construction.

Si par exemple l’utilisateur souhaite définir le principe que tout argument qui s’attaque lui-même doit être attaqué par au moins un argument de S , il va *de facto* développer l’arbre de décomposition de la FIGURE 2.

Cet arbre de décomposition (invisible pour l’utilisateur) produit instantanément à l’écran l’affichage de la formule

$$\bigwedge_{a \in \mathcal{A}} \bigvee_{b \in S} \varphi_{b \mathcal{R} a}$$

qui s’obtient via une séquence appropriée de clics sur les boutons de développement.

3 Aide à l’utilisateur

Naturellement, exprimer une sémantique argumentative selon les termes de la grammaire peut exiger une bonne dose d’ingéniosité de la part de l’utilisateur.

C’est une conséquence directe de l’idée que les boutons de développement correspondent aux non-terminaux de la grammaire. Chacun de ces boutons est associé à un menu déroulant comprenant plusieurs items qui correspondent aux règles développant ledit non-terminal (par exemple, le menu déroulant pour ⟨argument⟩ affiche la liste a, b, c, \dots, x, y, z). Il faut donc dépasser cela de sorte qu’en exception à cette idée, certaines options dans le menu déroulant ne se limitent pas à ces items mais soient en réalité des macros explicitées dans le but d’aider l’utilisateur dans son activité de spécification. Par exemple, la grammaire permet d’exprimer qu’un sous-ensemble des arguments est maximal pour un principe. Ou similairement, elle permet d’exprimer que les attaquants d’un certain argument forment un sous-ensemble de tel autre sous-ensemble. Il serait exagéré d’exiger qu’un utilisateur usuel fasse l’effort de déterminer comment exprimer cela à partir des seules règles de la grammaire. C’est pourquoi le bouton de développement pour le nonterminal ⟨principe⟩ comporte des options telles que : « ⟨ensemble⟩ est maximal pour ⟨principe⟩ », « ⟨ensemble-complexe⟩ est inclus dans ⟨ensemble-complexe⟩ », \dots

Si une spécification contradictoire lui est soumise, le système SESAME génère une formule insatisfaisable quelle que soit la valeur des paramètres. En revanche, le système n’effectue aucune vérification sémantique : l’utilisateur peut spécifier quelque chose de douteux, *e.g.*, que S soit une « extension » si et seulement si S contient tout argument qui s’attaque lui-même.

4 Conclusion

Le système SESAME permet à l'utilisateur d'exprimer des sémantiques argumentatives, connues ou originales, afin d'en obtenir une caractérisation sous la forme d'une formule propositionnelle paramétrée qu'il est possible d'instancier sur tout graphe argumentatif $(\mathcal{A}, \mathfrak{R})$ et tout $S \subseteq \mathcal{A}$ pour tester sa satisfaisabilité. Le système est téléchargeable à l'adresse suivante : <http://www.irit.fr/SESAME>.

Pour le moment, SESAME permet seulement de spécifier des sémantiques argumentatives de type extension-based par opposition au type ranking-based en plein essor actuellement [1, 5], et une suite logique sur laquelle nous commençons déjà à réfléchir est d'autoriser la spécification de sémantiques de ce second type. Par ailleurs, nous avons aussi le projet de coupler SESAME à un logiciel permettant de faire l'instanciation de la sémantique par un graphe argumentatif afin de vérifier *via* un solveur SAT comme TouIST [3] si un sous-ensemble S de l'ensemble des arguments est une extension ou non.

Remerciements

Ce travail a bénéficié du soutien du projet AMANDE (ANR-13-BS02-0004) de l'Agence Nationale pour la Recherche. La première version de SESAME a été développée par Van Hieu Ho.

Références

- [1] Amgoud, Leila et Jonathan Ben-Naim: *Ranking-Based Semantics for Argumentation Frameworks*. Dans Liu, Weiru, V. S. Subrahmanian et Jef Wijsen (éditeurs) : *7th International Conference on Scalable Uncertainty Management (SUM'13)*, tome 8078 de *Lecture Notes in Artificial Intelligence*, pages 134–147, Washington, DC, USA, 2013. Springer.
- [2] Baroni, Pietro et Massimiliano Giacomin: *On Principle-based Evaluation of Extension-based Argumentation Semantics*. *Artificial Intelligence*, 171(10) :675–700, 2007.
- [3] Ben Slimane, Khaled Skander, Alexis Comte, Olivier Gasquet, Abdelwahab Heba, Olivier Lezaud, Frederic Maris et Mael Valais: *Twist your logic with TouIST*. *CoRR*, 2015. [abs/1507.03663](https://arxiv.org/abs/1507.03663).
- [4] Besnard, Philippe, Sylvie Doutre, Van Hieu Ho et Dominique Longin: *SESAME - A System for Specifying Semantics in Abstract Argumentation*. Dans Thimm, Matthias, Federico Cerutti, Hannes Strass et Mauro Vallati (éditeurs) : *International Workshop on Systems and Algorithms for Formal Argumentation (SAFA)*, tome 1672, pages 40–51, <http://CEUR-WS.org>, 2016. *CEUR Workshop Proceedings*.
- [5] Bonzon, Elise, Jérôme Delobelle, Sébastien Konieczny et Nicolas Maudet: *A Comparative Study of Ranking-Based Semantics for Abstract Argumentation*. Dans Schuurmans, Dale et Michael P. Wellman (éditeurs) : *30th AAI Conference on Artificial Intelligence (AAAI-16)*, pages 914–920, Phoenix, Arizona, USA, 2016. AAAI Press.
- [6] Dung, Phan Minh: *On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-person Games*. *Artificial Intelligence*, 77(2) :321–357, 1995.

The Parameterized Complexity of Positional Games^{*}

Édouard Bonnet¹, Serge Gaspers^{2,3}, Antonin Lambilliotte⁴, Stefan Rümmele^{5,2}, Abdallah Saffidine²

¹ Middlesex University, London, UK
edouard.bonnet@dauphine.fr

² The University of New South Wales, Sydney, Australia
sergeg@cse.unsw.edu.au, abdallah.saffidine@gmail.com

³ Data61, CSIRO, Sydney, Australia

⁴ École Normale Supérieure de Lyon, Lyon, France
antonin.lambilliotte@ens-lyon.fr

⁵ The University of Sydney, Sydney, Australia
stefan.rummele@sydney.edu.au

Abstract : We study the parameterized complexity of several positional games. Our main result is that SHORT GENERALIZED HEX is $W[1]$ -complete parameterized by the number of moves. This solves an open problem from Downey and Fellows' influential list of open problems from 1999. Previously, the problem was thought of as a natural candidate for $AW[*]$ -completeness.

Our main tool is a new fragment of first-order logic where universally quantified variables only occur in inequalities. We show that model-checking on arbitrary relational structures for a formula in this fragment is $W[1]$ -complete when parameterized by formula size.

We also consider a general framework where a positional game is represented as a hypergraph and two players alternately pick vertices. In a Maker-Maker game, the first player to have picked all the vertices of some hyperedge wins the game. In a Maker-Breaker game, the first player wins if she picks all the vertices of some hyperedge, and the second player wins otherwise. In an Enforcer-Avoider game, the first player wins if the second player picks all the vertices of some hyperedge, and the second player wins otherwise.

SHORT MAKER-MAKER is $AW[*]$ -complete, whereas SHORT MAKER-BREAKER is $W[1]$ -complete and SHORT ENFORCER-AVOIDER CO- $W[1]$ -complete parameterized by the number of moves. This suggests a rough parameterized complexity categorization into positional games that are complete for the first level of the W -hierarchy when the winning configurations only depend on which vertices one player has been able to pick, but $AW[*]$ -completeness when the winning condition depends on which vertices both players have picked. However, some positional games where the board and the winning configurations are highly structured are fixed-parameter tractable. We give another example of such a game, SHORT k -CONNECT, which is fixed-parameter tractable when parameterized by the number of moves.

Mots-clés : Hex, Maker-Maker games, Maker-Breaker games, Enforcer-Avoider games, parameterized complexity theory

1 Introduction

In a *positional game* [12], two players alternately claim unoccupied elements of the board of the game. The goal of a player is to claim a set of elements that form a winning set, and/or to prevent the other player from doing so.

TIC-TAC-TOE, its competitive variant played on a 15×15 board, GOMOKU, as well as HEX are the most well-known positional games. When the size of the board is not fixed, the decision problem, whether the first player has a winning strategy from a given position in the game is PSPACE-complete for many such

^{*}An almost identical version of this paper has been accepted for publication in the proceedings of the ICALP 2017 conference.

games. The first result was established for GENERALIZED HEX, a variant played on an arbitrary graph [7]. Reisch soon followed up with results for GOMOKU [14] and HEX played on a board [15]. More recently, PSPACE-completeness was obtained for HAVANNAH [3] and several variants of CONNECT(m, n, k, p, q) [13], a framework that encompasses TIC-TAC-TOE and GOMOKU.

In a Maker-Maker game, also known as strong positional game, the winner is the first player to claim all the elements of some winning set. In a Maker-Breaker game, also known as weak positional game, the first player, Maker, wins by claiming all the elements of a winning set, and the second player, Breaker, wins by preventing Maker from doing so. In an Enforcer-Avoider game, the first player, Enforcer, wins if the second player claims all the vertices of a winning set, and the second player, Avoider, wins otherwise.

In this paper, we consider the corresponding short games, of deciding whether the first player has a winning strategy in ℓ moves from a given position in the game, and parameterize them by ℓ . The parameterized complexity of short games is known for games such as chess [18], generalized geography [1, 2], and pursuit-evasion games [19]. For HEX, played on a hexagonal grid, the short game is FPT and for GENERALIZED HEX, played on an arbitrary graph, the short game is W[1]-hard and in AW[*].

When winning sets are given as arbitrary hyperedges in a hypergraph, we refer to the three game variants as MAKER-MAKER, MAKER-BREAKER, and ENFORCER-AVOIDER, respectively. MAKER-BREAKER was first shown PSPACE-complete by [16] under the name G_{pos} (POS DNF). A simpler proof was later given by [4] who also showed PSPACE-completeness of MAKER-MAKER. To the best of our knowledge, the classical complexity of ENFORCER-AVOIDER has not been established yet.

In this paper we will show that the short game for GENERALIZED HEX is W[1]-complete, solving an open problem stated numerous times [3, 6, 5, 9, 17], we establish that the short game for a generalization of Tic-Tac-Toe is FPT, and we determine the parameterized complexity of the short games for MAKER-MAKER, MAKER-BREAKER, and ENFORCER-AVOIDER. One of our main tools is a new fragment of first-order logic where universally-quantified variables only occur in inequalities and no other relations. After giving some necessary definitions in the next section, we will state our results precisely, and discuss them. The rest of the paper is devoted to the proofs of our results, with some parts deferred to the appendix, due to space constraints.

2 Preliminaries

Finite structures. A vocabulary τ is a finite set of relation symbols, each having an associated arity. A finite structure \mathcal{A} over τ consists of a finite set A , called the universe, and for each R in τ a relation over A of corresponding arity. An (undirected) graph is a finite structure $G = (V, E)$, where E is a symmetric binary relation. A hypergraph is a finite structure $G = (V \cup E, IN)$, where $IN \subseteq V \times E$ is the incidence relation between vertices and edges. Sometimes it is more convenient to denote a hypergraph instead by a tuple $G = (V, E)$ where E is a set of subsets of V .

First-order logic. We assume a countably infinite set of variables. Atomic formulas over vocabulary τ are of the form $x_1 = x_2$ or $R(x_1, \dots, x_k)$ where $R \in \tau$ and x_1, \dots, x_k are variables. The class FO of all first-order formulas over τ consists of formulas that are constructed from atomic formulas over τ using standard Boolean connectives \neg, \wedge, \vee as well as quantifiers \exists, \forall followed by a variable. Let φ be a first-order formula. The size of (a reasonable encoding of) φ is denoted by $|\varphi|$. The variables of φ that are not in the scope of a quantifier are called free variables. We denote by $\varphi(\mathcal{A})$ the set of all assignments of elements of A to the free variables of φ such that φ is satisfied. We call \mathcal{A} a model of φ if $\varphi(\mathcal{A})$ is not empty. The class Σ_1 contains all first-order formulas of the form $\exists x_1, \dots, \exists x_k \varphi$ where φ is a quantifier free first-order formula.

Parameterized complexity. The class FPT contains all parameterized problems that can be decided by an FPT-algorithm. An FPT-algorithm is an algorithm with running time $f(k) \cdot n^{\mathcal{O}(1)}$, where $f(\cdot)$ is an arbitrary computable function that only depends on the parameter k and n is the size of the problem instance. An FPT-reduction of a parameterized problem Π to a parameterized problem Π' is an FPT-algorithm that transforms an instance (I, k) of Π to an instance (I', k') of Π' such that: (i) (I, k) is a yes-instance of Π if and only if (I', k') is a yes-instance of Π' , and (ii) $k' = g(k)$, where $g(\cdot)$ is an arbitrary computable function that only depends on k . Hardness and completeness with respect to parameterized complexity classes is defined analogously to the concepts from classical complexity theory, using FPT-reductions. The following parameterized classes will be needed in this paper: $\text{FPT} \subseteq \text{W}[1] \subseteq \text{AW}[*]$. Many parameterized complexity classes can be defined via a version of the following model checking problem.

MC(Φ)

Instance: Finite structure \mathcal{A} and formula $\varphi \in \Phi$.

Parameter: $|\varphi|$.

Problem: Decide whether $\varphi(\mathcal{A}) \neq \emptyset$.

In particular, the problem MC(Σ_1) is W[1]-complete and the problem MC(FO) is AW[*]-complete (see for example [8]).

Positional games. Positional games are played by two players on a hypergraph $G = (V, E)$. The vertex set V indicates the set of available positions, while the each hyperedge $e \in E$ denotes a winning configuration. The two players alternately claim unclaimed vertices of V until either all elements are claimed or one player wins. The notion of winning depends on the game type. In a *Maker-Maker game*, the first player to claim all vertices of any hyperedge $e \in E$ wins. In a *Maker-Breaker game*, the first player (*Maker*) wins if she claims all vertices of any hyperedge $e \in E$. If the game ends and player 1 has not won, then the second player (*Breaker*) wins. In an *Enforcer-Avoider game*, the first player (*Enforcer*) wins if the second player (*Avoider*) claims all vertices of any hyperedge $e \in E$. If the game ends and player 1 has not won, then the second player wins. A positional game is called an l -move game, if the game ends either after a player wins or both players played l moves. A winning strategy for player 1 is a move for player 1 such that for all moves of player 2 there exists a move of player 1... such that player 1 wins.

3 Results

We view a *positional game* as a hypergraph $G = (V, E)$. Two players alternately claim an unclaimed vertex from V , and the winning sets are the hyperedges E . Depending on the game, the hyperedges are either implicitly or explicitly defined. A *position* in a positional game is an allocation of vertices to the players, who have already claimed these vertices. The *empty position* is the position where no vertex is allocated to a player.

In a *Maker-Maker game*, the winner is the first player to claim all the vertices of some hyperedge. In a *Maker-Breaker game*, the first player, *Maker*, wins by claiming all the vertices of a hyperedge, and the second player, *Breaker*, wins by preventing *Maker* from doing so. In an *Enforcer-Avoider game*, the first player, *Enforcer*, wins if the second player claims all the vertices of a winning set, and the second player, *Avoider*, wins otherwise.

The first game we consider is a *Maker-Maker game* that generalizes well-known games TIC-TAC-TOE, CONNECT6, and GOMOKU (also known as FIVE IN A ROW). In $\text{CONNECT}(m, n, k, p, q)$, the vertices are cells of an $m \times n$ grid, each set of k aligned cells (horizontally, vertically, or diagonally) is a winning set, the first move by player 1 is to claim q vertices, and then the players alternate claim p unclaimed vertices at each turn. TIC-TAC-TOE corresponds to $\text{CONNECT}(3, 3, 3, 1, 1)$, CONNECT6 to $\text{CONNECT}(19, 19, 6, 2, 1)$, and GOMOKU to $\text{CONNECT}(19, 19, 5, 1, 1)$. Variations with different board sizes are also common. In the $\text{SHORT } k\text{-CONNECT}$ problem, the input is the set of $m \cdot n$ vertices, an assignment of some of these vertices to the two players, the integer p , and the parameter ℓ . The winning sets are implicitly defined. The question is whether player 1 has a winning strategy from this position in at most ℓ moves. We omit q from the problem definition of $\text{SHORT } k\text{-CONNECT}$ since we are modeling games that advanced already past the initial moves. Our first result (proved in Section 4) is that $\text{SHORT } k\text{-CONNECT}$ is fixed-parameter tractable for parameter ℓ . (In all our results, the parameter is the number of moves, ℓ .)

Theorem 1 $\text{SHORT } k\text{-CONNECT}$ is FPT.

The main reason for this tractability is the rather special structure of the winning sets. It helps reducing the problem to model checking for first-order logic on locally bounded treewidth structures, which is FPT [10].

A similar strategy was recently used to show that SHORT HEX is FPT [3]. The HEX game is played on a parallelogram board paved by hexagons, each player owns two opposite sides of the parallelogram. Players alternately claim an unclaimed cell, and the first player to connect their sides with a path of connected hexagons wins the game. Note that we may view HEX as a *Maker-Breaker game*: if the second player manages to disconnect the first players sides, he has created a path connecting his sides. [3] also considered a well-known generalization to arbitrary graphs. The GENERALIZED HEX game is played on a graph with two specified vertices s and t . The two players alternately claim an unclaimed vertex of the graph, and player 1 wins if she can connect s and t by vertices claimed by her, and player 2 wins if he can prevent player 1

from doing so. The SHORT GENERALIZED HEX problem has as input a graph G , two vertices s and t in G , an allocation of some of the vertices to the players, and an integer ℓ . The parameter is ℓ , and the question is whether player 1 has a winning strategy to connect s and t in ℓ moves.

The SHORT GENERALIZED HEX problem is known to be in AW^* and was conjectured to be AW^* -complete [3, 6, 5, 9, 17]. In fact, AW^* is thought of as the natural home for most short games [6], playing a similar role in parameterized complexity as PSPACE in classical complexity for games with polynomial length. However, [3] only managed to show that SHORT GENERALIZED HEX is $\text{W}[1]$ -hard, leaving a complexity gap between $\text{W}[1]$ and AW^* . Our next result is to show that SHORT GENERALIZED HEX is in $\text{W}[1]$. Thus, SHORT GENERALIZED HEX is in fact $\text{W}[1]$ -complete.

Theorem 2 *SHORT GENERALIZED HEX is $\text{W}[1]$ -complete.*

Our main tool is a new fragment of first-order logic for which model-checking on arbitrary relational structures is $\text{W}[1]$ -complete parameterized by the length of the formula. This fragment, which we call \forall^{\neq} -FO, is the fragment of first-order logic where universally-quantified variables appear only in inequalities.

Theorem 3 *$\text{MC}(\forall^{\neq}\text{-FO})$ is $\text{W}[1]$ -complete.*

This result is proved by reducing a formula in \forall^{\neq} -FO to a formula in Σ_1 . The \forall^{\neq} -FO logic makes it convenient to express short games where we can express that player 1 can reach a certain configuration without being blocked by player 2, no matter what configurations player 2 reaches. This is indeed the case for GENERALIZED HEX, where we are merely interested in knowing if player 1 can connect s and t without being blocked by player 2.

More generally, this is the case for SHORT MAKER-BREAKER, where the input is a hypergraph $G = (V, E)$, a position, and an integer ℓ , and the question is whether player 1 has a winning strategy to claim all the vertices of some hyperedge in ℓ moves.

Theorem 4 *SHORT MAKER-BREAKER is $\text{W}[1]$ -complete.*

The fact that SHORT MAKER-BREAKER is PSPACE-complete and $\text{W}[1]$ -complete (and *not* AW^* -complete) may challenge the intuition one has on alternation. Looking at the classical complexity (PSPACE-completeness), it seems that both players have comparable expressivity and impact over the game. As the game length is polynomially bounded, if the outcome could be determined by only guessing a sequence of moves from one player, then the problem would lie in NP. Now from the parameterized complexity standpoint, SHORT MAKER-BREAKER is equivalent under FPT reductions to guessing the k vertices of a clique (as in the seminal $\text{W}[1]$ -complete k -CLIQUE problem); no alternation there. Those considerations may explain why it was difficult to believe that GENERALIZED HEX is *not* AW^* -complete as conjectured repeatedly [17, 5, 6].

This is also in contrast to SHORT MAKER-MAKER, where the input is a hypergraph $G = (V, E)$, a position, and an integer ℓ , and the question is whether player 1 has a strategy to be the first player claiming all the vertices of some hyperedge in ℓ moves.

Theorem 5 *SHORT MAKER-MAKER is AW^* -complete.*

For the remaining type of positional games, the SHORT ENFORCER-AVOIDER problem has as input a hypergraph $G = (V, E)$, a position, and an integer ℓ , and the question is whether player 1 has a strategy to claim ℓ vertices that forces player 2 to complete a hyperedge. Again, player 1 can only block some moves of player 2, and the winning condition for player 1 can be expressed in \forall^{\neq} -FO.

Theorem 6 *SHORT ENFORCER-AVOIDER is $\text{co-W}[1]$ -complete.*

Our results suggest that a structured board may suggest that a positional game is FPT, but otherwise, the complexity depends on how the winning condition for player 1 can be expressed. If it only depends on what positions player 1 has reached, our results suggest that the problem is $\text{W}[1]$ -complete, but when the winning condition for player 1 also depends on the position player 2 has reached, the game is probably AW^* -complete.

4 SHORT k -CONNECT IS FPT

Graph G represents an $m \times n$ board in the following sense. Every board cell is represented by a vertex. Horizontal, vertical and diagonal neighbouring cells are connected via an edge. Vertex sets V_1 and V_2 represent the vertices already occupied by Player 1 and Player 2. While integer p , the number of stones to be placed during a move, is part of the input, we restrict it to values below constant k as games with $p \geq k$ are trivial.

SHORT k -CONNECT

Instance: A graph $G = (V, E)$ representing an $m \times n$ board, occupied vertices $V_1, V_2 \subseteq V$, and integer p and l .

Parameter: l .

Problem: Decide whether Player 1 has a winning strategy with at most l moves.

Theorem 1 SHORT k -CONNECT is FPT.

Proof. We reduce SHORT k -CONNECT to first-order model checking MC(FO) on a bounded local treewidth structure. Using a result by Frick and Grohe [10], it follows that SHORT k -CONNECT is FPT. Let (G, V_1, V_2, p, l) be an instance of SHORT k -CONNECT, where $G = (V, E)$. We construct instance (\mathcal{A}, φ) of MC(FO) as follows. Let $EDGE$ be a binary relation symbol and let $V1$ and $V2$ be unary relation symbols. Then \mathcal{A} is the $\{EDGE, V1, V2\}$ -structure $(V, EDGE^{\mathcal{A}}, V1^{\mathcal{A}}, V2^{\mathcal{A}})$ with $EDGE^{\mathcal{A}} := E$, $V1^{\mathcal{A}} := V_1$, and $V2^{\mathcal{A}} := V_2$. FO-formula φ is defined as $\varphi \equiv \exists x_1^1 \exists x_2^1 \dots \exists x_1^p \forall y_1^1 \dots \forall y_1^p \exists x_2^1 \dots \exists x_2^p \forall y_2^1 \dots \exists x_1^p \exists u_1 \exists u_2 \dots \exists u_k \forall v_1 \forall v_2 \dots \forall v$

$$\begin{aligned} \psi \equiv & \bigvee_{i=0}^l \left[\text{legal}P1_i(x_1^1, \dots, x_1^p, y_1^1, \dots, x_i^p) \wedge \left(\neg \text{legal}P2_i(x_1^1, \dots, x_1^p, y_1^1, \dots, x_i^p) \vee \right. \right. \\ & \left. \left(\text{config}P1_i(x_1^1, \dots, x_i^p, u_1, \dots, u_k) \wedge \bigwedge_{j=1}^{k-2} \text{aligned}(u_j, u_{j+1}, u_{j+2}) \wedge \right. \right. \\ & \left. \left. \left(\neg \text{config}P2_i(y_1^1, \dots, y_i^p, v_1, \dots, v_k) \vee \neg \bigwedge_{j=1}^{k-2} \text{aligned}(v_j, v_{j+1}, v_{j+2}) \right) \right) \right] \\ \text{path}(u, v, w) \equiv & EDGE(u, v) \wedge EDGE(v, w), \\ \text{hor_vert}(u, v, w) \equiv & \exists x \exists y \text{path}(u, v, w) \wedge \text{path}(u, x, w) \wedge \text{path}(u, y, w) \wedge \text{path}(x, v, y) \wedge \\ & \forall z \left[(z \neq v \wedge z \neq x \wedge z \neq y) \rightarrow \neg \text{path}(u, z, w) \right], \\ \text{diag}(u, v, w) \equiv & \text{path}(u, v, w) \wedge \forall x \left[x \neq v \implies \neg \text{path}(u, x, w) \right], \\ \text{aligned}(u, v, w) \equiv & \text{hor_vert}(u, v, w) \vee \text{diag}(u, v, w). \end{aligned}$$

Variables x_i^j represent the j th stone in Player 1's i th move and variables y_i^j represent the j th stone in Player 2's i th move. The sequences $u_1 \dots u_k$ and $v_1 \dots v_k$ represent possible winning configurations for Player 1 and Player 2. The overall structure of ψ is the following. The first disjunction ranging from $i = 0$ to $i = l$ represents the number of moves Player 1 needs to win the game. We then ensure that the x variables represent legal moves by Player 1. Further, either variables y do not represent legal moves by Player 2, or Player 1 achieved a winning configuration. For the latter, we assure that variables u represent aligned vertices occupied by Player 1. Finally, we check that Player 2 did not achieve a winning configuration before, that is vertices v do not represent aligned vertices occupied by Player 2.

Formula $\text{path}(u, v, w)$ expresses that there is a path of length 2 between vertices u and w via v . Formula $\text{hor_vert}(u, v, w)$ expresses that vertices u, v , and w are aligned horizontally or vertically in this order. A case analysis shows that u, v and w are horizontally or vertically aligned if and only if there are exactly three nodes at distance 1 of u and w , and that v is in the middle of the other two. Formula $\text{diag}(u, v, w)$ expresses that vertices u, v , and w are diagonally aligned in this order. This is the case if there exists no other length 2 path between u and w . Formula $\text{aligned}(u, v, w)$ expresses that vertices u, v , and w are aligned (in that order). Formula $\text{legal}P1_i$ (see Appendix A) ensures that variables x_i^j represent legal moves of Player 1, that is vertices not contained in V_1 or V_2 or previously played vertices. Analogously, $\text{legal}P2_i$ ensures that variables y_i^j represent legal moves of Player 2. Formula, $\text{config}P1_i$ (see Appendix A) expresses that

variables u_1, \dots, u_k form a valid configuration of exactly k vertices out of the set of V_1 or vertices played by Player 1. Analogously, $configP^2_i$ states that variables v_1, \dots, v_k form a valid configuration of exactly k vertices out of the set of V_2 or vertices played by Player 2. The size of φ is polynomial in l, k , and p . Since k is a constant and p is bounded by k , we have an FO formula polynomial in our parameter l . Graph G represents a grid with diagonals. Hence, G has maximum degree 8 and therefore bounded local treewidth. It follows from Frick and Grohe [10] that Short Connect is FPT. \square

5 MC(\forall^{\neq} -FO) is $w[1]$ -complete

The class \forall^{\neq} -FO contains all first-order formulas of the form $Q_1x_1Q_2x_2Q_3x_3\dots Q_kx_k\varphi$, with $Q_i \in \{\forall, \exists\}$ and φ being a quantifier free first-order formula such that every \forall -quantified variable x_i only occurs in inequalities, that is in relations of the form $x_i \neq x_j$ for some variable x_j .

Theorem 3 MC(\forall^{\neq} -FO) is $W[1]$ -complete.

Proof. Hardness: Every Σ_1 formula is contained in the class \forall^{\neq} -FO. Hence, $W[1]$ -hardness follows from $W[1]$ -completeness of $MC(\Sigma_1)$.

Membership: By reduction to $MC(\Sigma_1)$. Let (\mathcal{A}, φ) be an instance of $MC(\forall^{\neq}\text{-FO})$. If φ contains only existential quantifiers then (\mathcal{A}, φ) is already an instance of $MC(\Sigma_1)$. Hence, let $\varphi = Q_1x_1Q_2x_2\dots Q_{i-1}x_{i-1}\forall x_i\exists x_{i+1}\exists x_{i+2}\dots\exists x_k\psi$ with $Q_j \in \{\forall, \exists\}$ for $1 \leq j < i$, ψ is in negation normal form and $|\varphi| = l$. That is, x_i is the rightmost of the universal quantified variables. In order to reduce (\mathcal{A}, φ) to an instance of $MC(\Sigma_1)$, we need a way to remove all universal quantifications. We will show how to eliminate the universal quantification of x_i . This technique can then be used to iteratively eliminate all the universal quantifiers. Let $\varphi_1(x_1, \dots, x_{i-1})$ be the subformula $\varphi_1(x_1, \dots, x_{i-1}) \equiv \forall x_i\exists x_{i+1}\dots\exists x_k\psi$. We will show that we can replace $\varphi_1(x_1, \dots, x_{i-1})$ by

$$\varphi_2(x_1, \dots, x_{i-1}) \equiv \exists y_i\exists y_{i+1}\dots\exists y_k \left(\psi[y_i/x_i, y_{i+1}/x_{i+1}, \dots, y_k/x_k] \wedge \right. \quad (1)$$

$$\bigwedge_{j=1}^{i-1} \exists y_{i+1}^j \exists y_{i+2}^j \dots \exists y_k^j \psi[x_j/x_i, y_{i+1}^j/x_{i+1}, y_{i+2}^j/x_{i+2}, \dots, y_k^j/x_k] \wedge \quad (2)$$

$$\left. \bigwedge_{j=i+1}^k \exists y_{i+1}^j \exists y_{i+2}^j \dots \exists y_k^j \psi[y_j/x_i, y_{i+1}^j/x_{i+1}, y_{i+2}^j/x_{i+2}, \dots, y_k^j/x_k] \right). \quad (3)$$

This reduction is an FPT-reduction, since the size of formula φ_2 is a function of the size of formula φ_1 . Let c_1, \dots, c_{i-1} be arbitrary but fixed elements of the universe A of \mathcal{A} . We will show that $\varphi_1(x_1, \dots, x_{i-1}) \equiv \varphi_2(x_1, \dots, x_{i-1})$ by proving (a) $\varphi_1(c_1, \dots, c_{i-1}) \rightarrow \varphi_2(c_1, \dots, c_{i-1})$ and (b) $\varphi_2(c_1, \dots, c_{i-1}) \rightarrow \varphi_1(c_1, \dots, c_{i-1})$. For (a) assume that $\varphi_1(c_1, \dots, c_{i-1})$ is true. This means, $\varphi_1[c_i/x_i]$ is true for all $c_i \in A$, that is for all $c_i \in A$ there exists an assignment to x_{i+1}, \dots, x_k such that ψ is true. Part (1) of $\varphi_2(c_1, \dots, c_{i-1})$ asks for some $c_i \in A$ such that there exists an assignment to x_{i+1}, \dots, x_k such that ψ is true. Part (2) asks for the existence of an assignment to x_{i+1}, \dots, x_k such that ψ is true for each of the cases where x_i is one of the elements c_1, \dots, c_{i-1} . Part (3) asks for the existence of an assignment to x_{i+1}, \dots, x_k such that ψ is true for each of the cases where x_i is one of the elements that are assigned to x_{i+1}, \dots, x_k in the model of Part (1). All these are special cases of the universal quantification over x_i , hence $\varphi_2(c_1, \dots, c_{i-1})$ is true.

For direction (b) assume towards a contradiction that $\varphi_1(c_1, \dots, c_{i-1})$ is false and that $\varphi_2(c_1, \dots, c_{i-1})$ is true. Since φ_1 is false, there exists $c_i \in A$ such that $\varphi_1[c_i/x_i]$ is false. We perform a case distinction on the value c_i . First let $c_i = c_j$ for $1 \leq j < i$. Then let c_{i+1}, \dots, c_k be the assignments to variables y_{i+1}^j, \dots, y_k^j in the model of φ_2 . The j th conjunct of Part (3) of φ_2 states that ψ holds for $x_i = x_j$ using the assignment c_{i+1}, \dots, c_k . Hence, assigning c_{i+1}, \dots, c_k to variables x_{i+1}, \dots, x_k in φ_1 is a model for $\varphi_1[c_i/x_i]$, which contradicts our assumption. As the next case, let c_{i+1}, \dots, c_k be the assignment to variables y_{i+1}, \dots, y_k in the model of φ_2 and let $c_i = c_j$ for $i < j \leq k$. Let c'_{i+1}, \dots, c'_k be the assignments to variables y_{i+1}^j, \dots, y_k^j in the model of φ_2 . The j th conjunct of Part (2) of φ_2 states that ψ holds for $x_i = x_j = c_j$ using the assignment c'_{i+1}, \dots, c'_k . Hence, assigning c'_{i+1}, \dots, c'_k to variables x_{i+1}, \dots, x_k in φ_1 is a model for $\varphi_1[c_i/x_i]$, which contradicts our assumption. For the last case, let c_i be one of the remaining values. Let l_1, \dots, l_m be all the literals in ψ that contain x_i . All of them are inequalities of the

form $x_i \neq x_j$ for $j \neq i$. Let c'_i be the assignment to y_i in the model of φ_2 . Let l'_1, \dots, l'_m be the literals in $\psi[y_i/x_i, y_{i+1}/x_{i+1}, \dots, y_k/x_k]$ in Part (1) of φ_2 that correspond to l_1, \dots, l_m . We have no knowledge about the truth value of these literals l'_j with $1 \leq j \leq m$, but all of the literals l_j in ψ evaluate to true when assigning c_{i+1}, \dots, c_k to the variables x_{i+1}, \dots, x_k . Since ψ is in negation normal form and the literals l_1, \dots, l_m never occur in unnegated form, that is as equalities, changing the truth value of these literal from false to true will never result in changing the truth value of the whole formula from true to false. But since c'_i together with c_{i+1}, \dots, c_k is a model of Part (1) of φ_2 , it holds that for all values of c_i that we consider in this case, that $\varphi_1[c_i/x_i]$ is true, which contradicts our assumption. This completes the case distinction and we have $\varphi_1(x_1, \dots, x_{i-1}) \equiv \varphi_2(x_1, \dots, x_2)$. \square

6 SHORT GENERALIZED HEX is $w[1]$ -complete

SHORT GENERALIZED HEX

Instance: Graph $G = (V, E)$, vertices $s, t \in V$, vertex sets $V_1, V_2 \subseteq V$ with $V_1 \cap V_2 = \emptyset$, and integer l .

Parameter: l .

Problem: Decide whether Player 1 has a winning strategy with at most l moves in the generalized Hex game (G, s, t, V_1, V_2) .

A generalized Hex game (G, s, t, V_1, V_2) is a positional game (V', E') , where the positions V' and the winning configurations E' are defined as follows. Set V' contains all vertices of G , that is $V' = V$. Set E' contains a set of vertices $\{v_1, \dots, v_k\}$ if and only if $\{v_1, \dots, v_k\} \cup \{s, t\}$ form an $s - t$ path in G . Additionally, vertices in V_1 and V_2 are already claimed by player 1 and player 2, respectively. Since the set of winning configurations of SHORT GENERALIZED HEX is only defined implicitly, the input size of SHORT GENERALIZED HEX can be exponential smaller than the number of winning configurations.

Theorem 2 SHORT GENERALIZED HEX is $W[1]$ -complete.

Proof. Hardness is already known [3]. For membership, we reduce SHORT GENERALIZED HEX to $MC(\forall^{\neq}\text{-FO})$. Let (G, s, t, V_1, V_2, l) be an instance of SHORT GENERALIZED HEX, where $G = (V, E)$. Claimed vertices V_1 and V_2 can be preprocessed: (i) every $v \in V_1$ and its incident edges are removed from G and the neighbourhood of v is turned into a clique; (ii) every $v \in V_2$ and its incident edges are removed from G . Hence, w.l.o.g. we assume that $V_1 = V_2 = \emptyset$. We construct an instance (\mathcal{A}, φ) of $MC(\forall^{\neq}\text{-FO})$ as follows. Let $EDGE$ be a binary relation symbol and let S and T be unary relation symbols. Then \mathcal{A} is the $\{EDGE, S, T\}$ -structure $(V, EDGE^{\mathcal{A}}, S^{\mathcal{A}}, T^{\mathcal{A}})$ with $EDGE^{\mathcal{A}} := E$, $S^{\mathcal{A}} := \{s\}$, and $T^{\mathcal{A}} := \{t\}$. The $\forall^{\neq}\text{-FO}$ -formula φ is defined as $\varphi = \exists s \exists t \exists x_1 \forall y_1 \exists x_2 \forall y_2 \dots \forall y_{l-1} \exists x_l \exists z_1 \exists z_2 \dots \exists z_l \psi$, with

$$\psi \equiv S(s) \wedge T(t) \wedge \left(EDGE(s, t) \vee \bigvee_{i=1}^l \bigvee_{j=1}^l \left(EDGE(s, z_1) \wedge EDGE(z_j, t) \wedge \right. \right. \\ \left. \left. path_{i,j}(x_1, \dots, x_i, z_1, \dots, z_j) \wedge diff_i(x_1, y_1, \dots, y_{i-1}, x_i) \right) \right), \\ path_{i,j}(x_1, \dots, x_i, z_1, \dots, z_j) \equiv \bigwedge_{h=1}^{j-1} EDGE(z_h, z_{h+1}) \wedge \bigwedge_{h=1}^j \bigvee_{k=1}^i z_h = x_k, \\ diff_i(x_1, y_1, \dots, x_{i-1}, y_{i-1}, x_i) \equiv \bigwedge_{1 \leq j < k \leq i} x_j \neq x_k \wedge \bigwedge_{1 \leq j < k \leq i} y_j \neq x_k.$$

The intuition of φ is the following. The variables x_i, y_i , and z_i represent the moves of Short, the moves of Cut, and the ordered (s, t) -path induced by Short's moves, respectively. The variables s and t represent the vertices of the same name. Formula φ expresses that there is either a direct edge between s and t or a $s-t$ path of length j was played. The main disjunctions (\bigvee) ensure that we consider wins that take up to l moves, and build $s-t$ path of length up to l . Subformula $path_{i,j}$ will be true if and only if the z variables form a path using only values of the selected values for the x variables. Subformula $diff_i$ ensures that all x variables are pairwise distinct and they are distinct from all y variables with smaller index.

We have $|\varphi| = \mathcal{O}(l^4)$, so this is indeed an FPT-reduction and $W[1]$ -membership follows. \square

7 SHORT MAKER-BREAKER is $W[1]$ -complete

SHORT MAKER-BREAKER

Instance: Hypergraph $G = (V, E)$, vertex sets $V_1, V_2 \subseteq V$ with $V_1 \cap V_2 = \emptyset$, and integer l .

Parameter: l .

Problem: Decide whether Player 1 has a winning strategy with at most l if vertices V_1 and V_2 are already claimed by Player 1 and Player 2, respectively.

Theorem 4 SHORT MAKER-BREAKER is $W[1]$ -complete.

Proof. For membership, we reduce SHORT MAKER-BREAKER to $MC(\forall^\neq\text{-FO})$. Let (G, V_1, V_2, l) be an instance of SHORT MAKER-BREAKER, where $G = (V, E)$ is a hypergraph. Claimed vertices V_1 and V_2 can be preprocessed: (i) every $v \in V_1$ is removed from V and every hyperedge $e \in E$; (ii) every $v \in V_2$ is removed from V and every hyperedge $e \in E$ with $v \in e$ is removed from E . Hence, w.l.o.g. we assume that $V_1 = V_2 = \emptyset$. We construct an instance (\mathcal{A}, φ) of $MC(\forall^\neq\text{-FO})$ as follows. Let IN and $SIZE$ be binary relation symbols. Then \mathcal{A} is the $\{IN, SIZE\}$ -structure $(V \cup E \cup \{1, \dots, |V|\}, IN^{\mathcal{A}}, SIZE^{\mathcal{A}})$ with $IN^{\mathcal{A}} := \{(x, e) \mid x \in V, e \in E, x \in e\}$ and $SIZE^{\mathcal{A}} := \{(e, i) \mid e \in E, |e| = i\}$. Hence, the universe of \mathcal{A} consists of the vertices of G , an element for each hyperedge, and an element for some bounded number of integers. The $\forall^\neq\text{-FO}$ -formula φ is defined as $\varphi \equiv \exists x_1 \forall y_1 \dots \forall y_{l-1} \exists x_l \exists e \exists z_1 \exists z_2 \dots \exists z_l \psi$, with

$$\psi \equiv \bigvee_{1 \leq j \leq i \leq l} \left(\text{diff}_i(x_1, y_1, \dots, x_i) \wedge SIZE(e, j) \wedge \bigwedge_{k=1}^j \bigvee_{h=1}^i z_k = x_h \wedge \bigwedge_{1 \leq k < h \leq j} z_k \neq z_h \wedge \bigwedge_{k=1}^j IN(z_k, e) \right).$$

The subformula $\text{diff}_i(x_1, y_1, \dots, x_i)$ refers to the subformula with same name used in the proof of Theorem 2. That is, it ensures that all x variables are pairwise distinct and that they are distinct from all y variables with smaller index. The intuition of φ is the following. The variables x_i and y_i represent the moves of Maker and the moves of Breaker, respectively. The variables z_i represent the vertices forming the winning configuration of Maker and e represents the hyperedge of this winning configuration. The first disjunction ensures that we consider wins that take up to l moves. The second disjunction ensures that we consider winning configurations that consist of up to i vertices. After checking that e has the correct size ($SIZE(e, j)$), we encode that the values of the z variables are contained in the hyperedge represented by e and that these variables are pairwise disjoint and selected among the moves of Maker (the x variables).

We have $|\varphi| = \mathcal{O}(l^4)$, so this is indeed an FPT-reduction and $W[1]$ -membership follows.

For hardness, we reduce k -MULTICOLORED CLIQUE to SHORT MAKER-BREAKER. The reduction is essentially the same as the reduction used for showing $W[1]$ -hardness of GENERALIZED HEX [3]. The crucial observation is that the construction of [3] contains only a polynomial number of possible $s - t$ paths. Hence, we can encode every such $s - t$ -path as a unique hyperedge denoting a winning configuration in SHORT MAKER-BREAKER. \square

8 SHORT MAKER-MAKER is $AW[1]$ -complete

SHORT MAKER-MAKER

Instance: Hypergraph $G = (V, E)$, vertex sets $V_1, V_2 \subseteq V$ with $V_1 \cap V_2 = \emptyset$, and integer l .

Parameter: l .

Problem: Decide whether Player 1 has a winning strategy with at most l if vertices V_1 and V_2 are already claimed by Player 1 and Player 2.

Theorem 5 SHORT MAKER-MAKER is $AW[1]$ -complete.

Proof. For membership, we reduce SHORT MAKER-MAKER to $MC(\text{FO})$. Let (G, V_1, V_2, l) be an instance of SHORT MAKER-MAKER, where $G = (V, E)$ is a hypergraph. We construct an instance (\mathcal{A}, φ) of $MC(\text{FO})$ as follows. Let $V1$, $V2$, and $EDGE$ be unary relation symbols. Let IN be a binary relation symbol. Then \mathcal{A} is the $\{V1, V2, EDGE, IN\}$ -structure $(V \cup E, V1^{\mathcal{A}}, V2^{\mathcal{A}}, EDGE^{\mathcal{A}}, IN^{\mathcal{A}})$ with $V1^{\mathcal{A}} := \{x \mid x \in V_1\}$, $V2^{\mathcal{A}} := \{x \mid x \in V_2\}$, $EDGE^{\mathcal{A}} := \{e \mid e \in E\}$, and $IN^{\mathcal{A}} := \{(x, e) \mid x \in V, e \in E, x \in e\}$. Hence, the universe of \mathcal{A} consists of the vertices and the hyperedges of G . The FO-formula φ is defined as $\varphi \equiv \exists x_1 \forall y_1 \dots \forall y_{l-1} \exists x_l \psi$, with

$$\psi \equiv \bigvee_{i=0}^l \text{legal}P1_i(x_1, y_1, \dots, x_l) \wedge \left(\neg \text{legal}P2_{i-1}(x_1, y_1, \dots, x_l) \vee \right. \\ \left. (\text{win}P1_i(x_1, y_1, \dots, x_l) \wedge \neg \text{win}P2_{i-1}(x_1, y_1, \dots, x_l)) \right).$$

$$\text{win}P1_i(x_1, y_1, \dots, x_l) \equiv \exists e \forall z \text{EDGE}(e) \wedge (\neg \text{IN}(z, e) \vee V1(z) \vee \bigvee_{j=1}^i z = x_j),$$

$$\text{win}P2_i(x_1, y_1, \dots, x_l) \equiv \exists e \forall z \text{EDGE}(e) \wedge (\neg \text{IN}(z, e) \vee V2(z) \vee \bigvee_{j=1}^i z = y_j).$$

Variable x_j represent Player 1's j th move and variable y_j represent Player 2's j th move. The first disjunction represents the number of moves i that Player 1 needs to win the game. Formula $\text{legal}P1_i$ (see Appendix B) ensures that variables $(x_j)_{1 \leq j \leq i}$ represent legal moves of Player 1, that is vertices not contained in V_1 or V_2 or previously played vertices. Analogously, $\text{legal}P2_i$ ensures that variables $(y_j)_{1 \leq j \leq i}$ represent legal moves of Player 2. Formula $\text{win}P1_i$ ensures that Player 1 has won within the i first moves, that is, it has completed a hyperedge with V_1 and variables up to x_i . Analogously, $\text{win}P2_i$ ensures that Player 2 has won within the i first moves. We have $|\varphi| = \mathcal{O}(l^3)$ and $|\mathcal{A}| = \mathcal{O}(|G|^2)$, so this is indeed an FPT-reduction and AW[*]-membership follows.

For hardness, we reduce from the AW[*]-complete problem SHORT GENERALIZED GEOGRAPHY on bipartite graphs. The reduction is deferred to the appendix. \square

9 SHORT ENFORCER-AVOIDER is co-W[1]-complete

SHORT ENFORCER-AVOIDER

Instance: Hypergraph $G = (V, E)$, vertex sets $V_1, V_2 \subseteq V$ with $V_1 \cap V_2 = \emptyset$, and integer l .

Parameter: l .

Problem: Decide whether Player 1 has a winning strategy with at most l moves if vertices V_1 and V_2 are already claimed by Player 1 and Player 2, respectively.

Theorem 6 SHORT ENFORCER-AVOIDER is co-W[1]-complete.

Proof. We show that the co-problem of SHORT ENFORCER-AVOIDER is W[1]-complete. The co-problem of SHORT ENFORCER-AVOIDER is to decide whether for all strategies of Enforcer, there exists a strategy of Avoider such that during the first l moves, Avoider does not claim a hyperedge. Again, vertices V_1 and V_2 are already claimed by Enforcer and Avoider, respectively. We prove W[1]-hardness by a parameterized reduction from INDEPENDENT SET and W[1]-membership by reduction to MC(\forall^\neq -FO).

In the W[1]-complete INDEPENDENT SET problem [5], the input is a graph $G = (V, E)$ and an integer parameter k , and the question is whether G has an independent set of size k , i.e., a set of k pairwise non-adjacent vertices. We construct a positional game $G' = (V', E')$ by replacing each vertex of G by a clique of size $k+1$. The vertex set V' has vertices $v(1), \dots, v(k+1)$ for each vertex $v \in V$, and hyperedges $E' = \{\{v(i), v(j)\} : v \in V \text{ and } 1 \leq i < j \leq k+1\} \cup \{\{u(i), v(j)\} : uv \in E \text{ and } 1 \leq i, j \leq k+1\}$. We claim that G has an independent set of size k if and only if Avoider does not claim a hyperedge in the first k moves in the positional game G' starting from the empty position, that is $V_1 = V_2 = \emptyset$. For the forward direction, suppose $I = \{v_1, \dots, v_k\}$ is an independent set of G of size k . Then, a winning strategy for Avoider is to claim an unclaimed vertex from $\{v_i(1), \dots, v_i(k+1)\}$ at round $i \in \{1, \dots, k\}$. We note that Enforcer cannot claim all the vertices from $\{v_i(1), \dots, v_i(k+1)\}$, since there are not enough moves to do so, and Avoider does not complete a hyperedge with this strategy. On the other hand, suppose Avoider has a winning strategy in k moves. For an arbitrary play by Enforcer, let $\{v_1(i_1), \dots, v_k(i_k)\}$ denote the vertices claimed by Player 1. Then, $v_i \neq v_j$ and $v_i v_j \notin E$ for any $1 \leq i < j \leq k$, since Player 1 would otherwise claim all the vertices of a hyperedge. Therefore, $\{v_1, \dots, v_k\}$ is an independent set of G of size k .

For membership, we reduce to MC(\forall^\neq -FO). Let (G, V_1, V_2, l) be an instance of the co-problem of SHORT ENFORCER-AVOIDER where $G = (V, E)$ is a hypergraph. First we do some preprocessing. We remove all vertices from G that are contained in V_2 , that is the vertices already claimed by Avoider. If this results in an empty hyperedge, the instance is a no instance. Otherwise, we remove all hyperedges that contain a vertex in V_1 , that is the vertices already claimed by Enforcer, since Avoider will never lose via these

edges anymore. Finally, we remove all vertices from G that are contained in V_1 . Let $G = (V, E)$ now refer to the outcome of this preprocessing. By construction all vertices of G are unoccupied and some vertices might not be contained in any hyperedge. If G contains less than $2l$ vertices we can solve the problem via brute force in FPT time. Hence, in what follows we assume that there are at least $2l$ unoccupied vertices in G . We construct an instance (\mathcal{A}, φ) of $\text{MC}(\forall^\neq\text{-FO})$ as follows. Let EDGE_i be a i -ary relation symbol for $1 \leq i \leq l$. Then \mathcal{A} is the $\{\text{EDGE}_1, \dots, \text{EDGE}_l\}$ -structure $(V, \text{EDGE}_1^{\mathcal{A}}, \dots, \text{EDGE}_l^{\mathcal{A}})$ with $\text{EDGE}_i^{\mathcal{A}} := \{(v_1, \dots, v_l) \mid e \in E, |e| = i, e = \{v_1, \dots, v_l\}\}$, that is $\text{EDGE}_i^{\mathcal{A}}$ contains all hyperedges of cardinality i . The $\forall^\neq\text{-FO}$ -formula φ is defined as

$$\varphi \equiv \forall y_1 \exists x_1 \forall y_2 \exists x_2 \dots \exists x_l \text{diff}_l(y_1, x_1, \dots, x_l) \wedge \bigwedge_{1 \leq i \leq l} \bigwedge_{\{z_1, \dots, z_i\} \subseteq \{x_1, \dots, x_i\}} \neg \text{EDGE}_i(z_1, \dots, z_i),$$

where $\text{diff}_i(y_1, x_1, \dots, x_i) \equiv \bigwedge_{1 \leq j < k \leq i} x_j \neq x_k \wedge \bigwedge_{1 \leq j \leq k \leq i} y_j \neq x_k$.

Subformula $\text{diff}_i(y_1, x_1, \dots, x_i)$ ensures that all x variables are pairwise distinct and they are distinct from all y variables with index less or equal theirs. The intuition of φ is the following. The variables x_i and y_i represent the moves of Avoider and the moves of Enforcer, respectively. Avoider wins if the x variables do not cover a whole hyperedge after l moves. We only have to check hyperedges of size up to l . Hence, for each cardinality $i \leq l$, we check for all subsets z_1, \dots, z_i of the x variables that they do not form a hyperedge. Formula φ does not pose any restrictions on the y variables, that is we do not force Enforcer to pick unoccupied vertices. We call a move by Enforcer that picks an already occupied vertex cheating. To prove correctness, we need to show that whenever Enforcer has a winning strategy σ_E that involves cheating, Enforcer also has a winning strategy σ'_E without cheating. We construct σ'_E as follows. We follow strategy σ_E while σ_E does not perform a cheating move. If the next move would be a cheating move, we play a random unoccupied vertex instead and keep track of this vertex in a new set V_r . The next time we need to select a move, we construct a board state s by removing all vertices in V_r from the picks of Enforcer and query strategy σ_E on this state s . If the answer is an unoccupied vertex, we perform this move normally. If instead the answer is a previously played vertex (which might be in V_r), we play a random unoccupied vertex instead and add it to V_r . Since σ_E was a winning strategy, so is σ'_E . Hence, formula φ does not need to check if the y variables correspond to unoccupied vertices. The construction can be done by an FPT algorithm since for each hyperedge $e \in E$ of cardinality i , we create $i! \leq l!$ entries in the EDGE_i relation. We have $|\varphi| = \mathcal{O}(l^l)$, so this is indeed an FPT reduction and $\text{W}[1]$ -membership follows. \square

10 Conclusion

We have seen that the parameterized complexity of short positional games depends crucially on whether both players compete for achieving winning sets, or whether the game can be seen as one player aiming to achieve a winning set and the other player merely blocking the moves of the first player. Naturally, blocking moves correspond to inequalities in first-order logic, and our $\forall^\neq\text{-FO}$ fragment of first-order logic therefore captures that the universal player can only block moves of the existential player. Our $\text{W}[1]$ -completeness of $\text{MC}(\forall^\neq\text{-FO})$ has been used several times in this paper, but our transformation of $\forall^\neq\text{-FO}$ formulas into Σ_1 formulas may have other uses. As a concrete example related to positional games, [3] established that **SHORT HEX** is FPT by expressing the problem as a FO formula, and making use of Frick and Grohe's meta-theorem [10], similarly as we did in Section 4. This establishes that the problem is FPT but the running time is non-elementary in l . However, we remark that their FO formula is actually a $\forall^\neq\text{-FO}$ formula of size polynomial in l . Our transformation gives an equivalent Σ_1 formula whose length is single-exponential in l , and the meta-theorem of [11] then gives a running time for solving **SHORT HEX** that is triply-exponential in l .

References

- [1] ABRAHAMSON K. R., DOWNEY R. G. & FELLOWS M. R. (1993). Fixed-parameter intractability II (extended abstract). In *Proceedings of the 10th Annual Symposium on Theoretical Aspects of Computer Science (STACS 93)*, volume 665 of *Lecture Notes in Computer Science*, p. 374–385: Springer.
- [2] ABRAHAMSON K. R., DOWNEY R. G. & FELLOWS M. R. (1995). Fixed-parameter tractability and completeness IV: on completeness for $\text{W}[P]$ and PSPACE analogues. *Annals of Pure and Applied Logic*, **73**(3), 235–276.

- [3] BONNET É., JAMAIN F. & SAFFIDINE A. (2016). On the complexity of connection games. *Theoretical Computer Science*, **644**, 2–28.
- [4] BYSKOV J. M. (2004). *Maker-Maker and Maker-Breaker Games are PSPACE-Complete*. Rapport interne RS-04-14, BRICS, Department of Computer Science, Aarhus University.
- [5] DOWNEY R. G. & FELLOWS M. R. (1999). *Parameterized complexity*. New York: Springer-Verlag.
- [6] DOWNEY R. G. & FELLOWS M. R. (2013). *Fundamentals of Parameterized Complexity*. Springer.
- [7] EVEN S. & TARJAN R. E. (1976). A combinatorial problem which is complete in polynomial space. *Journal of the ACM*, **23**(4), 710–719.
- [8] FLUM J. & GROHE M. (1998). *Parameterized Complexity Theory*. Springer.
- [9] FOMIN F. V. & MARX D. (2012). FPT suspects and tough customers: Open problems of downey and fellows. In H. L. BODLAENDER, R. DOWNEY, F. V. FOMIN & D. MARX, Eds., *The Multivariate Algorithmic Revolution and Beyond - Essays Dedicated to Michael R. Fellows on the Occasion of His 60th Birthday*, volume 7370 of *Lecture Notes in Computer Science*, p. 457–468: Springer.
- [10] FRICK M. & GROHE M. (2001). Deciding first-order properties of locally tree-decomposable structures. *Journal of the ACM*, **48**(6), 1184–1206.
- [11] GROHE M. & WÖHRLE S. (2004). An existential locality theorem. *Annals of Pure and Applied Logic*, **129**(1), 131–148.
- [12] HALES A. W. & JEWETT R. I. (1963). Regularity and positional games. *Transactions of the American Mathematical Society*, **106**, 222–229.
- [13] HSIEH M. Y. & TSAI S.-C. (2007). On the fairness and complexity of generalized k -in-a-row games. *Theoretical Computer Science*, **385**(1), 88–100.
- [14] REISCH S. (1980). Gobang ist PSPACE-vollständig. *Acta Informatica*, **13**(1), 59–66.
- [15] REISCH S. (1981). Hex ist PSPACE-vollständig. *Acta Informatica*, **15**, 167–191.
- [16] SCHAEFER T. J. (1978). On the complexity of some two-person perfect-information games. *Journal of Computer and System Sciences*, **16**(2), 185–225.
- [17] SCOTT A. (2009). *On the Parameterized Complexity of Finding Short Winning Strategies in Combinatorial Games*. PhD thesis, University of Victoria.
- [18] SCOTT A. & STEGE U. (2008). Parameterized chess. In *Proceedings of the 3rd International Workshop on Parameterized and Exact Computation (IWPEC 2008)*, volume 5018 of *Lecture Notes in Computer Science*, p. 172–189: Springer.
- [19] SCOTT A. & STEGE U. (2010). Parameterized pursuit-evasion games. *Theoretical Computer Science*, **411**(43), 3845–3858.

A Subformulas for Theorem 1

$$\text{legalP1}_i(x_1^1, \dots, x_1^p, y_1^1, \dots, x_i^p) \equiv \bigwedge_{j=1}^i \bigwedge_{t=1}^p \left[\neg V1(x_j^t) \wedge \neg V2(x_j^t) \wedge \bigwedge_{r=1}^{j-1} \bigwedge_{q=1}^t (x_j^t \neq x_r^q) \wedge \bigwedge_{q=1}^{t-1} (x_j^t \neq x_j^q) \wedge \bigwedge_{r=1}^{j-1} \bigwedge_{q=1}^t (x_j^t \neq y_r^q) \right],$$

$$\text{legalP2}_i(x_1^1, \dots, x_1^p, y_1^1, \dots, x_i^p) \equiv \bigwedge_{j=1}^{i-1} \bigwedge_{t=1}^p \left[\neg V1(y_j^t) \wedge \neg V2(y_j^t) \wedge \bigwedge_{r=1}^{j-1} \bigwedge_{q=1}^t (y_j^t \neq y_r^q) \wedge \bigwedge_{q=1}^{t-1} (y_j^t \neq y_j^q) \wedge \bigwedge_{r=1}^{j-1} \bigwedge_{q=1}^t (y_j^t \neq x_r^q) \right].$$

$$\text{configP1}_i(x_1^1, \dots, x_i^p, u_1, \dots, u_k) \equiv \bigwedge_{j=1}^k \left[\left(V1(u_j) \vee \bigvee_{r=1}^i \bigvee_{q=1}^p u_j = x_r^q \right) \wedge \bigwedge_{r=1}^{j-1} u_j \neq u_r \right],$$

$$\text{configP2}_i(y_1^1, \dots, y_i^p, v_1, \dots, v_k) \equiv \bigwedge_{j=1}^k \left[\left(V2(v_j) \vee \bigvee_{r=1}^{i-1} \bigvee_{q=1}^p v_j = y_r^q \right) \wedge \bigwedge_{r=1}^{j-1} v_j \neq v_r \right].$$

B Subformulas for Theorem 5

$$\begin{aligned} \text{legalP1}_i(x_1, y_1, \dots, x_i) &\equiv \bigwedge_{1 \leq j \leq i} [\neg V1(x_j) \wedge \neg V2(x_j)] \wedge \bigwedge_{1 \leq j < k \leq i} [x_j \neq x_k \wedge y_j \neq x_k], \\ \text{legalP2}_i(x_1, y_1, \dots, x_i) &\equiv \bigwedge_{1 \leq j \leq i} [\neg V1(y_j) \wedge \neg V2(y_j)] \wedge \bigwedge_{1 \leq j < k \leq i} (y_j \neq y_k) \wedge \bigwedge_{1 \leq j \leq k \leq i} x_j \neq y_k. \end{aligned}$$

C AW[*]-hardness of SHORT MAKER-MAKER

Reduction from the AW[*]-complete problem SHORT GENERALIZED GEOGRAPHY on bipartite graphs. From an instance $B = (X \uplus Y, F, v_0)$, k of SHORT GENERALIZED GEOGRAPHY, with $v_0 \in X$, we build a hypergraph $G = (V, E)$, l of size polynomial in $|B|$ which will be an equivalent SHORT MAKER-MAKER instance.

In our reduction, the hypergraph G mainly involves two distinguished vertices $\exists, \forall \in V$ and gadgets corresponding to vertices and edges of B . In the initial setup, the vertex \exists is assumed to have already been claimed by Player 1 and the vertex \forall to have already been claimed by Player 2. Our construction ensures that all the hyperedges of E contain exactly one vertex in $\{\exists, \forall\}$. We thus partition the hyperedges between the ones that can make Player 1 win and the ones that can make Player 2 win.

Formally, G is defined as indicated in Equations (4) and (5). It uses gadgets detailed in the rest of this section. The parameter is linearly preserved from the input parameter: $l = 9(k + 1) + 6$.

$$V = \{\exists, \forall\} \cup \bigcup_{u \in X} V^\exists(u) \cup \bigcup_{u \in Y} V^\forall(u) \cup V_4^D(\{\exists\}) \cup V_4^D(\{\forall\}) \quad (4)$$

$$E = \{\{\forall, a^{v_0}\}\} \cup \bigcup_{u \in X} E^\exists(u) \cup \bigcup_{u \in Y} E^\forall(u) \cup D_4^\exists(\{\exists\}) \cup D_4^\forall(\{\forall\}) \quad (5)$$

$$V_1 = \{\exists\}, V_2 = \{\forall\}$$

C.1 Terminology

A *useless* 3-threat for Player 1 is a 3-threat that can be defended, and for which after the 3-threat and its defense, Player 1 has not achieved anything. Formally, the threat and its defense are two vertices which, once played, do not appear in any other hyperedges that could make one player or their opponent win. Note that those threats can be disregarded for Player 1 but not for Player 2. Indeed, Player 2 could use a series of useless 3-threats to win by delaying the game.

A *losing* 3-threat for a player is a 3-threat that can be met with a counter-attack winning in a constant number of moves; more precisely in at most 6 moves.

A *living* 3-threat is a non losing 3-threat; if it is for Player 1, it should in addition be non useless.

C.2 Delay gadget

As a building block of the forthcoming existential and universal gadgets, we introduce the following delay gadgets where $? \in \{\exists, \forall\}$. If $? = \exists$ (resp. $? = \forall$), we say that the delay gadget belongs to Player 1 (resp. to Player 2).

$$D_1^?(S) := \{S \cup \{?, x_1^S\}, S \cup \{?, x_2^S\}, S \cup \{?, x_3^S\}\} \quad (6)$$

$$D_2^?(S) := \bigcup_{i,j \in [3]} \{S \cup \{?, x_i^S, y_j^S\}\} = \begin{cases} S \cup \{?, x_1^S, y_1^S\}, S \cup \{?, x_2^S, y_1^S\}, S \cup \{?, x_3^S, y_1^S\}, \\ S \cup \{?, x_1^S, y_2^S\}, S \cup \{?, x_2^S, y_2^S\}, S \cup \{?, x_3^S, y_2^S\}, \\ S \cup \{?, x_1^S, y_3^S\}, S \cup \{?, x_2^S, y_3^S\}, S \cup \{?, x_3^S, y_3^S\} \end{cases} \quad (7)$$

$$D_4^?(S) := \bigcup_{g,h,i,j \in [3]} \{S \cup \{?, x_g^S, y_h^S, z_i^S, t_j^S\}\} \quad (8)$$

$$V_1^D(S) := \{x_1^S, x_2^S, x_3^S\} \quad (9)$$

$$V_2^D(S) := \bigcup_{i \in [3]} \{x_i^S, y_i^S\} = \{x_1^S, x_2^S, x_3^S, y_1^S, y_2^S, y_3^S\} \quad (10)$$

$$V_4^D(S) := \bigcup_{i \in [3]} \{x_i^S, y_i^S, z_i^S, t_i^S\} \quad (11)$$

The elements x_i^S , y_i^S , z_i^S , and t_i^S (with $i \in [3]$) will only appear in the corresponding delay gadgets. For any set S , we will introduce at most one set among $D_1^\forall(S)$, $D_1^\exists(S)$, $D_2^\forall(S)$, $D_2^\exists(S)$, $D_4^\forall(S)$, and $D_4^\exists(S)$. This implies that existing x_i^S and y_i^S (with $i \in [3]$) are well-defined.

Lemma 1

Let $\delta \in \{1, 2, 4\}$ and $S \subseteq V$ be a set of vertices such that $D_\delta^\exists(S) \subseteq E$ (resp. $D_\delta^\forall(S) \subseteq E$). If all vertices of S have been claimed by Player 1 (resp. Player 2), and if no more than one vertex of $V_\delta^D(S)$ has been claimed by the opponent, then she (resp. he) has an unstopable δ -threat.

Proof. The two statements have identical proofs by switching Player 1 and Player 2. We therefore only give a proof for a delay gadget $D_\delta^\exists(S)$. Assume that Player 1 has played all the vertices of S . Without loss of generality, assume that the vertex claimed by the opponent, if any, is x_1^S . Recall that we assume that \exists has already been claimed by Player 1 and \forall has been claimed by Player 2.

For $\delta = 1$, Player 1 has at least two 1-threats, playing in x_2^S or x_3^S , and Player 2 cannot block them both. Thus, if Player 2 claims x_i^S (with $i \in [3]$), she claims x_j^S with $j \neq i \in [3]$ and wins.

For $\delta = 2$, Player 1 has several 2-threats. If Player 2 claims x_i^S (resp. y_i^S) for some $i \in [3]$, Player 1 claims x_j^S (resp. y_j^S) for some $j \neq i \in \{2, 3\}$ and obtains an unstopable 1-threat.

For $\delta = 4$, the reasoning is similar and omitted. \square

Corollary 1 Let $\delta \in \{1, 2, 4\}$ and $S \subseteq V$ be a set of vertices such that $D_\delta^\exists(S) \subseteq E$ (resp. $D_\delta^\forall(S) \subseteq E$). If Player 1 (resp. Player 2) claims all vertices in S and no more than one vertex of $V_\delta^D(S)$ has been claimed by the opponent, then if it is that player's turn, they can force a win in δ moves unless the opponent has a $\delta - 1$ -threat. If it is the opponent's turn, then Player 1 (resp. Player 2) can force a win in δ moves unless the opponent has a δ -threat.

C.3 Existential vertex gadget

For each vertex $u \in X$ in the existential partition of the SHORT GENERALIZED GEOGRAPHY instance, we introduce in G the following hyperedges:

$$\begin{aligned} E^\exists(u) = & D_2^\exists(\{a^u, b^u\}) \cup D_2^\forall(\{b^u, e^u\}) \cup D_2^\forall(\{b^u, g^u\}) \\ & \cup \bigcup_{v \in N(u)} D_1^\exists(\{a^u, c_v^u, d_v^u\}) \\ & \cup D_1^\forall(\{b^u, d_v^u, e^u\}) \cup D_1^\exists(\{c_v^u, e^u, f^u\}) \\ & \cup D_1^\forall(\{d_v^u, f^u, g^u\}) \cup D_1^\exists(\{c_v^u, g^u, h^u\}) \\ & \cup D_2^\forall(\{d_v^u, i_v^u\}) \cup D_2^\exists(\{i_v^u, a^u\}) \end{aligned}$$

In terms of vertices of G introduced by the gadget, each vertex $u \in X$ gives rise to a set $V^\exists(u)$ that contains all the vertices needed by the delay sub-gadgets along with $\{a^u, b^u, e^u, f^u, g^u, h^u\} \cup \bigcup_{v \in N(u)} \{c_v^u, d_v^u, i_v^u\}$.

Lemma 2

Consider the gadget for an existential vertex $u \in X$ such that no element of $V^\exists(u) \setminus \{a^u\}$ has been claimed yet. Assume that a^u has been played by Player 1 and that it is Player 2's turn. If Player 2 has no non-losing 3-threats in the whole board, then for each $v \in N(u)$ such that a^v has not been claimed yet, Player 1 has a strategy $\sigma^\exists(u, v)$ that ensures either that Player 2 plays a^v after no more than 8 moves all of which belonging to $V^\exists(u)$ and that there are no non-losing 3-threats left for Player 2 in the gadget or that Player 1 wins in no more than 14 moves.

Proof. We exhibit the strategy for Player 1 and show that Player 2's answers are forced to prevent Player 1 from winning. By assumption Player 2 has no non-losing 3-threats anywhere else on the board and no vertices claimed in $V^\exists(u)$, so unless Player 2 play b^u , Player 1 wins in 6 moves by claiming b^u herself via Corollary 1 applied to $D_2^\exists(\{a^u, b^u\})$. Although Player 2 has now sets of 3-threats which involve e^u and g^u , he does not have any 2-threats. Player 1 plays c_v^u which forces Player 2 to claim d_v^u by Corollary 1 applied to $D_1^\exists(\{a^u, c_v^u, d_v^u\})$. Player 1 plays e^u which forces Player 2 to claim f^u . Player 1 plays g^u which forces Player 2 to claim h^u . Player 1 plays i_v^u . At this point, 8 moves have been played, Player 2 has no 3-threats left in the gadget, so Player 2 is forced to play a^v lest Player 1 plays a^v and wins in a total of 14 moves by Corollary 1 applied to $D_2^\exists(\{i_v^u, a^v\})$.

Since Player 1 has claimed e^u , g^u , and i_v^u , the only local hyperedges remaining for Player 2 are $D_2^\forall(\{d_w^u, i_w^u\})$ for $w \neq v$, and none of them feature a 3-threat. \square

Lemma 3

Consider the gadget for an existential vertex $u \in X$ such that no element of $V^\exists(u) \setminus \{a^u\}$ has been claimed yet. Assume that for any vertex $v \in Y$, a^v has not been claimed by Player 1. Assume that a^u has been played by Player 1 and that it is Player 2's turn. If Player 1 has no living 3-threats elsewhere on the board, then Player 2 has a strategy $\sigma^\forall(u)$ that ensures either 1) that after no less than 8 moves, all of which either belong to $V^\exists(u)$ or are not in any live existential hyperedge, Player 2 plays a^v for some v and there are no living 3-threats left for Player 1, or it is Player 2's turn and there is no living 3-threat for Player 1; or 2) that Player 2 wins.

Proof. We exhibit a local strategy for Player 2, any move by Player 1 in a non-living 3-threat elsewhere on the board is responded to accordingly. Player 2 plays b^u creating sets of 3-threats in $D_2^\forall(\{b^u, e^u\})$ and $D_2^\forall(\{b^u, g^u\})$. Playing either of e^u and g^u is losing for Player 1 because Player 2 can play in the other vertex. Therefore, Player 1 needs to play in a 3-threat to avoid losing. Notwithstanding the non-living 3-threats, the only 3-threats for Player 1 can be found in the gadgets $D_1^\exists(\{a^u, c_v^u, d_v^u\})$ for $v \in N(u)$.

As long as Player 1 plays in d_w^u for some w , Player 2 replies in the corresponding c_w^u voiding the threat. As soon as Player 1 plays a move other than d_v^u in $D_1^\exists(\{a^u, c_v^u, d_v^u\})$ for some v , Player 2 can answer d_v^u , voiding the threat, and play proceeds as follows. Player 1 has no 2-threats and so replying e^u is forced to avoid losing via Corollary 1 applied to $D_1^\forall(\{b^u, d_v^u, e^u\})$. Player 2 plays f^u which forces Player 1 to claim g^u . Player 2 plays h^u threatening to play i_v^u . Therefore, Player 1 needs to either play in 3-threats via the gadgets $D_1^\exists(\{a^u, c_w^u, d_w^u\})$ for some $w \in N(u)$ such that d_w^u has not been claimed yet, or Player 1 has to play in i_v^u herself. As long as Player 1 plays in d_w^u for some w , Player 2 replies in the corresponding c_w^u voiding the threat.

Eventually, Player 1 has to play in i_v^u . If a^v has already been claimed by Player 2, then Player 2 is left with no 3-threat to defend. Otherwise, Player 2 plays a^v . \square

C.4 Universal vertex gadget

For each $u \in Y$, we introduce in G the following hyperedges:

$$\begin{aligned} E^\forall(u) = & D_2^\forall(\{a^u, b^u\}) \cup D_2^\exists(\{b^u, g^u\}) \cup D_2^\exists(\{b^u, i^u\}) \\ & \cup \bigcup_{v \in N(u)} D_1^\forall(\{a^u, c_v^u, d_v^u\}) \\ & \cup D_1^\exists(\{b^u, d_v^u, e_v^u\}) \cup D_1^\forall(\{c_v^u, e_v^u, f^u\}) \\ & \cup D_1^\exists(\{b^u, c_v^u, j_v^u\}) \cup D_1^\forall(\{d_v^u, j_v^u, f^u\}) \\ & \cup D_1^\exists(\{c_v^u, f^u, g^u\}) \cup D_1^\exists(\{d_v^u, f^u, g^u\}) \cup D_1^\forall(\{e_v^u, g^u, h^u\}) \\ & \cup D_1^\exists(\{f^u, h^u, i^u\}) \cup D_2^\forall(\{e_v^u, a^v\}) \end{aligned}$$

In terms of vertices of G introduced by the gadget, each vertex $u \in Y$ gives rise to a set $V^\forall(u)$ that contains all the vertices needed by the delay sub-gadgets along with $\{a^u, b^u, f^u, g^u, h^u, i^u\} \cup \bigcup_{v \in N(u)} \{c_v^u, d_v^u, e_v^u, j_v^u\}$.

We observe that the only shared vertices between the different existential and universal gadgets are a^u for $u \in X \cup Y$. For instance, in the universal gadget, each a^v with $v \in N(u)$ is the "starting vertex" of the existential gadget encoding the vertex $v \in X$.

Lemma 4

Consider the gadget for a universal vertex $u \in Y$ such that no element of $V^\forall(u) \setminus \{a^u\}$ has been claimed yet. Assume that for any vertex $v \in X$, a^v has not been claimed by Player 2. Assume that a^u has been played by Player 2 and that it is Player 1's turn. If Player 2 has no non-losing 3-threats elsewhere on the board, then Player 1 has a strategy $\sigma^\exists(u)$ that ensures either 1) that after no more than 8 moves, all of which belong to $V^\forall(u)$, Player 1 plays a^v for some v and there are no non-losing 3-threats left for Player 2 or it is Player 1's turn and there are no non-losing 3-threats for Player 2; or 2) that Player 1 wins in no more than 14 moves.

Proof. We exhibit a local strategy for Player 1. Player 1 plays b^u creating sets of 3-threats in $D_2^\exists(\{b^u, g^u\})$ and $D_2^\exists(\{b^u, i^u\})$. Claiming either of g^u and i^u is losing for Player 2 because Player 1 can play in the other vertex. Therefore, Player 2 needs to play in a 3-threat to avoid losing. The only non-losing 3-threats for Player 2 can be found in the gadget $D_1^\forall(\{a^u, c_v^u, d_v^u\})$ for $v \in N(u)$.

If Player 2 claims d_v^u , Player 1 plays c_v^u , forcing Player 2 to claim j_v^u . Player 1 plays f^u , forcing Player 2 to claim g^u . At this point, Player 1 can play i^u and win by Corollary 1 applied to $D_2^\exists(\{b^u, i^u\})$.

If instead of d_v^u Player 2 starts by claiming c_v^u , then Player 1 plays d_v^u , forcing Player 2 to claim e_v^u . Player 1 plays f^u , forcing Player 2 to claim g^u . Player 1 plays h^u , forcing Player 2 to claim i^u . If a^v has already been claimed by Player 1, then Player 1 is left with no 3-threat to defend. Otherwise, Player 1 plays a^v . \square

Lemma 5

Consider the gadget for a universal vertex $u \in Y$ such that no element of $V^\forall(u) \setminus \{a^u\}$ has been claimed yet. Assume that a^u has been played by Player 2 and that it is Player 1's turn.

If Player 1 has no living 3-threats on the whole board, then for each $v \in N(u)$ such that a^v has not been claimed yet, Player 2 has a strategy $\sigma^\forall(u, v)$ that ensures either that Player 1 plays a^v after no less than 8 moves all of which either belong to $V^\forall(u)$ or are not in any live existential hyperedge and that there are no living 3-threats left for Player 1 in the gadget; or that Player 2 wins.

Proof. We exhibit the strategy for Player 2 and show that Player 1's answers are forced to prevent Player 2 from winning. By assumption Player 1 has no living 3-threats anywhere else on the board and no vertices claimed in $V^\forall(u)$, so unless Player 1 plays b^u , Player 2 wins in 6 moves by claiming b^u himself via Corollary 1 applied to $D_2^\forall(\{a^u, b^u\})$. Although Player 1 has now sets of 3-threats which involve e^u and g^u , she does not have any 2-threats. Player 2 plays c_v^u which forces Player 1 to claim d_v^u by Corollary 1 applied to $D_1^\forall(\{a^u, c_v^u, d_v^u\})$. Player 2 plays e^u which forces Player 1 to claim f^u . Player 2 plays g^u which forces Player 1 to claim h^u . Player 2 plays i^u . At this point, 8 moves have been played, Player 1 has no 3-threats left in the gadget, so Player 1 is forced to play a^v lest Player 2 plays a^v and wins in a total of 14 moves by Corollary 1 applied to $D_2^\forall(\{e_v^u, a^v\})$.

Since Player 2 has claimed g^u and i^u , the only local hyperedges remaining for Player 1 are in $D_2^\exists(\{b^u, d_w^u, e_w^u\})$ and $D_2^\exists(\{b^u, c_w^u, j_w^u\})$ for $w \neq v$, and none of them is feature a living 3-threat. \square

C.5 Correctness of the reduction

To show that YES SHORT GENERALIZED GEOGRAPHY instances are mapped to YES SHORT MAKER-MAKER instances and that NO instances are mapped onto NO instances, we prove that any Player 1 winning strategy in SHORT GENERALIZED GEOGRAPHY gives rise to a winning strategy for Player 1 in the corresponding SHORT MAKER-MAKER instance, and conversely for Player 2 winning/delaying strategies.

Assume that Player 1 can ensure a win within k moves in SHORT GENERALIZED GEOGRAPHY with strategy τ , and let us construct a strategy σ ensuring a Player 1 win within l moves in SHORT MAKER-MAKER. After Player 1 starts with move a^{v_0} , we use τ , Lemma 2, and Lemma 4 to create σ such that whenever τ prescribes that the token moves from a vertex $u \in X$ to $v \in Y$, we use $\sigma^\exists(u, v)$ to leave the u -gadget and enter the v -gadget. When the SHORT MAKER-MAKER game enters a u -gadget with $u \in Y$, we use $\sigma^\exists(u)$ to select moves in the gadget until the u -gadget is left and enters a v -gadget with $v \in X$. If a^v is already claimed by Player 1, then Player 2 has no non-losing threats and Player 1 can enter the $D_4^\exists(\{\exists\})$ gadget and win by Corollary 1. Otherwise, we then update the SHORT GENERALIZED GEOGRAPHY game with Player 2 moving the token to v . Eventually, the SHORT GENERALIZED GEOGRAPHY game reaches a vertex $u \in Y$ such that all neighbors have been visited before and the game ends. In the SHORT MAKER-MAKER instance, Player 1 follows $\sigma^\exists(u)$ and then wins by entering the $D_4^\exists(\{\exists\})$ gadget. If τ guarantees that at most $k' \leq k$ moves are played before Player 1 wins, then σ guarantees that at most $9(k' + 1) + 6 \leq l$ moves are played before Player 1 wins.

In the case of a NO SHORT GENERALIZED GEOGRAPHY instance, Player 2 has a strategy τ such that either Player 2 wins, or the game goes for longer than k moves. A corresponding SHORT MAKER-MAKER strategy σ can be derived such that either Player 2 wins in the SHORT MAKER-MAKER game, or the game goes for longer than $9(k + 1) + 6 = l$ moves. The construction is dual to the one above and relies on Lemma 3 and Lemma 5.

Une Sémantique Graduée Paramétrique pour la Persuasion

Elise Bonzon¹ Jérôme Delobelle² Sébastien Konieczny² Nicolas Maudet³

¹ LIPADE, Université Paris Descartes, Paris, France

² CRIL, CNRS - Université d'Artois, Lens, France

³ Sorbonne Universités, UPMC Univ Paris 06, CNRS - LIP6, UMR 7606, 75005 Paris

elise.bonzon@mi.parisdescartes.fr {delobelle,konieczny}@cril.fr nicolas.maudet@lip6.fr

Résumé

Dans cet article, nous nous intéressons à la question de savoir si les sémantiques graduées existantes permettent de capturer des principes utilisés en persuasion comme le pro-cataleptis (prévenir d'une attaque en proposant directement un contre-argument afin d'anticiper celle-ci) et l'atténuation (les longues lignes d'argumentation deviennent inefficaces). Cependant, certaines propriétés, largement acceptées par les sémantiques graduées existantes, sont incompatibles avec ces principes. Nous proposons ainsi une sémantique paramétrique basée sur le principe de propagation de valeurs, permettant de contrôler la portée des arguments pouvant avoir une influence sur l'évaluation des autres arguments. Nous étudions également les propriétés satisfaites par notre méthode (en identifiant en particulier les valeurs garantissant l'acceptabilité ou non de certaines propriétés), et apportons des résultats expérimentaux visant à étudier la diversité des ordres obtenues pour différentes valeurs du paramètre.

Abstract

In this paper we question the ability of the existant ranking semantics for argumentation to capture persuasion settings, emphasizing in particular the phenomena of pro-cataleptis (the fact that it is often efficient to anticipate the counter-arguments of the audience), and of fading (the fact that long lines of argumentation become ineffective). It turns out that some widely accepted principles of ranking-based semantics are incompatible with a faithful treatment of these phenomena. We thus propose a parametrized semantics based on propagation of values, which allows to control the scope of arguments to be considered for evaluation. We investigate its properties (identifying in particular threshold values guaranteeing that some properties hold), and report of experimental results showing that the family of rankings that may be returned have limited diversity.

1 Introduction

Récemment, la quête d'une méthode visant à donner un sens aux réseaux d'arguments contradictoires a stimulé un certain nombre de travaux. Pris dans leur forme abstraite, de tels réseaux, appelés *systèmes d'argumentation*, ont été introduits par Dung [9].

Partageant l'idée selon laquelle identifier les ensembles d'arguments mutuellement acceptables (extensions) n'est parfois pas suffisant, de nombreuses sémantiques ont été proposées permettant d'attribuer une valeur à chacun des arguments ou de retourner un ordre/classement entre les arguments [3, 7, 15, 8, 14, 1, 17, 11, 18, 2, 6].

Chacune de ces propositions possède un certain mérite, avec des exemples bien conçus affirmant, dans certaines situations au moins, qu'elle devrait être la méthode de choix. Quand il s'agit de comparer ces approches (au-delà de leurs propriétés formelles comme la convergence ou l'unicité de la solution), les choses deviennent plus difficiles. Cela est dû au fait que la comparaison n'est pas aussi évidente en premier lieu puisque différentes sémantiques mettent l'accent sur différentes propriétés. Les travaux de Bonzon et al. [5] initient cette étude en comparant la plupart des ces sémantiques en se basant sur un ensemble d'axiomes existant. Cependant, il s'avère que la pertinence de certains axiomes soit très dépendante du contexte de l'application rendant ainsi la comparaison des sémantiques difficiles.

Dans cet article, nous définissons une nouvelle sémantique graduée applicable dans le contexte de la persuasion. Dans ce contexte, certains aspects, comme ce qui constitue une argumentation efficace, ont été largement étudiés, et peuvent constituer une base intéressante pour la comparaison. Nous nous concentrerons sur deux phénomènes bien documentés :

- *Procataleptis* : anticiper les contre-arguments d'une audience [20] afin de potentiellement renforcer ses propres

arguments. Nous illustrons ce principe par un exemple introduit par Besnard et Hunter [4][p.85] : l’argumentaire d’un vendeur destiné à persuader un acheteur de voiture potentiel :

- (a1) La voiture x avec un moteur diesel est une voiture familiale de haute performance [...]
- (a2) En général, les moteurs diesel ont une performance inférieure à celle des moteurs à essence
- (a3) Avec ces nouveaux moteurs, la différence de performance [...] reste négligeable
- (a4) Les moteurs diesel ont en effet de bonnes performances, compte tenu du fait qu’ils consomment moins de carburant et que le diesel est moins cher que l’essence.

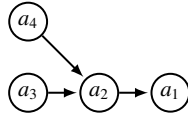


FIGURE 1 – Exemple de procatalepsis illustré par un système d’argumentation

Outre le fait que l’exemple soit antérieur aux récents scandales du diesel, ce qui est frappant, c’est qu’il contredit fortement un axiome “Void Precedence” satisfait (à notre connaissance) par toutes les sémantiques graduées existantes et qui considère les arguments non-attaqués comme les plus acceptables. En effet, cet argumentaire de vente semble plus convainquant que de simplement proposer l’argument (a1) seul.

- **Atténuation** : Les longues lignes d’argumentation deviennent inefficaces dans la pratique, pour la simple raison que le public perd facilement la trace des relations entre les arguments. Ceci est soutenu par le travail récent de Tan et al. [19] montrant (dans le cadre de leur étude sur une analyse approfondie des débats se déroulant sur le subreddit “ChangeMyView”) que les arguments situés à une distance supérieure à 5 d’un autre argument n’ont plus d’impact sur le score de celui-ci. Alors que certaines sémantiques graduées [16, 18] intègrent des fonctionnalités pouvant être utilisées pour réduire la force des arguments par rapport à leur distance, d’autres ne prennent pas en considération ce principe.

Nous concluons en montrant que les sémantiques graduées actuelles sont mal équipées pour être utilisée en persuasion. Notre travail vise donc à concevoir une sémantique graduée adaptée pour la persuasion, permettant de capturer à la fois le principe d’atténuation et le principe de procatalepsis.

Nous proposons donc une sémantique graduée qui permet de prendre en compte ces principes. Cette méthode comporte un paramètre permettant de réguler la façon dont

ces principes sont respectés. Notre vision est que, équipé de notre sémantique graduée, un vendeur faisant face à différents argumentaires de vente peut décider lequel sera le plus à son avantage. De manière générale, cette contribution pourrait être utilisée comme élément visant à développer des stratégies pour les techniques de persuasion calculatoire [12].

Le papier est organisé comme suit. Nous commençons par fournir, en Section 2, l’état de l’art requis sur les systèmes d’argumentation et les sémantiques graduées. La section 3.1 présente le principe de propagation que nous utilisons pour construire notre sémantique dans la section 3.2. Cette sémantique possède comme paramètre un facteur d’atténuation δ permettant de contrôler la vitesse de convergence et le classement obtenu. Plus important, nous étudions, dans la section 4.1, la relation qu’il existe entre ce facteur et la propriété “Void Precedence”. Enfin, dans l’objectif de comparer notre méthode avec celles existantes, nous proposons une étude axiomatique de notre sémantique (Section 4.2) ainsi qu’un exemple commun (Section 5).

2 Préliminaires

Dans cette section, nous rappelons brièvement quelques éléments clés sur les systèmes d’argumentation abstraits proposés par Dung [9].

Définition 1 (Dung 1995) *Un système d’argumentation (AF) est un couple $F = \langle \mathcal{A}, \mathcal{R} \rangle$ où \mathcal{A} est un ensemble fini d’entités abstraites appelées arguments et \mathcal{R} une relation binaire sur \mathcal{A} , i.e. $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$, appelée relation d’attaque. Notons $\text{Arg}(F) = \mathcal{A}$.*

Un des objectifs des systèmes d’argumentation est d’identifier les ensembles d’arguments qui peuvent être conjointement acceptés selon différents critères d’acceptabilité. Dans le cadre de Dung [9], l’acceptabilité d’un argument dépend de son appartenance à des ensembles d’arguments appelés extensions, tandis que les sémantiques graduées visent à classer les arguments du plus acceptable au moins acceptable.

Définition 2 *Une sémantique graduée σ associe à chaque système d’argumentation $F = \langle \mathcal{A}, \mathcal{R} \rangle$ un ordre \geq_F^σ sur \mathcal{A} , où \geq_F^σ est un pré-ordre (une relation réflexive et transitive) sur \mathcal{A} . $a \geq_F^\sigma b$ signifie que a est au moins aussi acceptable que b ($a \approx_F^\sigma b$ équivaut à $a \geq_F^\sigma b$ et $b \geq_F^\sigma a$, et $a >_F^\sigma b$ équivaut à $a \geq_F^\sigma b$ et $b \not\geq_F^\sigma a$).*

Quand il n’existe aucune ambiguïté concernant la sémantique graduée et le système d’argumentation en question, nous utiliserons \geq au lieu de \geq_{AF}^σ . Introduisons maintenant quelques notations permettant de définir nos sémantiques graduées par la suite.

Notation 1 Soient $F = \langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation et deux arguments $a, b \in \mathcal{A}$. Un **chemin** de a vers b , noté $p(a, b)$, est une séquence de noeuds $s = \langle a_0, \dots, a_n \rangle$ telle que pour chacun des noeuds de la séquence, il existe une attaque vers le noeud suivant : $a_0 = a$, $a_n = b$ et $\forall i < n, (a_i, a_{i+1}) \in \mathcal{R}$. Sa **longueur** est notée $|p(a, b)|$ et correspond au nombre d'attaques dont il est composé.

Notation 2 Soit $\Delta_n(a) = \{b \mid \exists p(b, a), \text{ avec } |p(b, a)| = n\}$ l'ensemble des arguments dont il existe un chemin de longueur n partant de cet argument vers a . Un argument $b \in \Delta_n(a)$ est un **défenseur** (resp. **attaquant**) de a si n est pair (resp. impair). Un chemin de b vers a est une **branche défensive** (resp. **branche attaquante**) si b est un défenseur (resp. attaquant) de a . $\Delta^B(a)$ (resp. $\Delta^A(a)$) correspond à l'ensemble de toutes les branches défensives (resp. attaquantes) de a .

Bien que notre méthode soit générale, dans le contexte de cet article, nous étudions également les cadres d'argumentation ayant une forme arborescente où un argument a , appelé *argument racine* possède uniquement des branches défensives (i.e. $\Delta^B(a) = \emptyset$ et $\Delta^A(a) \neq \emptyset$). De tels cadres sont appelés **argumentaires persuasifs**. Le système d'argumentation en introduction (Figure 1) est un exemple d'argumentaire persuasif avec a_1 comme argument racine.

3 Propagation à profondeur variable

3.1 Le principe de propagation

La sémantique que nous proposons dans ce papier suit le principe de *propagation* déjà utilisé dans certaines sémantiques graduées existantes [18, 6]. L'idée est d'abord d'assigner une valeur initiale positive à chaque argument du système d'argumentation. Les arguments peuvent commencer avec la même valeur initiale [18] ou avec des valeurs différentes, comme dans [6] où les arguments non-attaqués se voient attribuer une valeur qui est au moins aussi importante que celles des arguments attaqués. Ensuite, chaque argument propage sa valeur dans le système d'argumentation, en alternant sa polarité selon la longueur du chemin considéré (négativement si c'est un attaquant et positivement si c'est un défenseur).

En s'inspirant de ces définitions, nous définissons formellement le principe de propagation, en incluant un facteur d'atténuation δ diminuant l'impact des arguments situés de plus en plus loin dans le graphe (plus la longueur du chemin i est longue, plus la valeur de δ^i sera petite puisque $\delta < 1$). Entre autres, un tel facteur d'atténuation garantira la convergence de calcul du score de tous les arguments.

Définition 3 Soit $\langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation. L'évaluation P de $a \in \mathcal{A}$, à l'étape i , est donnée par :

$$P_i^{\epsilon, \delta}(a) = \begin{cases} v_\epsilon(a) & \text{si } i = 0 \\ P_{i-1}^{\epsilon, \delta}(a) + (-1)^i \delta^i \sum_{b \in \Delta_i(a)} v_\epsilon(b) & \text{sinon} \end{cases}$$

avec $\delta \in]0, 1[$ un facteur d'atténuation et $v_\epsilon : \mathcal{A} \rightarrow \mathbb{R}^+$ une fonction d'évaluation qui assigne un poids initial à chaque argument, avec $\epsilon \in [0, 1]$ tel que :

$$v_\epsilon(b) = \begin{cases} 1 & \text{si } \Delta_1(b) = \emptyset \\ \epsilon & \text{sinon} \end{cases}$$

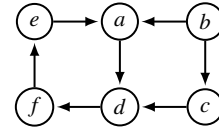


FIGURE 2 – Système d'argumentation F_1

Exemple 1 Commençons par calculer l'évaluation P de chaque argument dans F_1 (voir Figure 2) avec les valeurs $\epsilon = 0.5$ et $\delta = 0.4$. Les résultats sont représentés dans la table 1.

Focalisons nous sur l'argument f . Etant attaqué, celui-ci commence donc avec un score de 0.5 ($P_0^{0.5, 0.4}(f) = 0.5$). Lors de la phase de propagation, il reçoit tout d'abord la valeur négative de son attaquant direct d qui est lui aussi attaqué : $P_1^{0.5, 0.4}(f) = P_0^{0.5, 0.4}(f) - 0.4 \times v_{0.5}(d) = 0.3$. Ensuite, durant la seconde étape ($i = 2$), f reçoit positivement les valeurs de a et c atténuées par δ^2 : $P_2^{0.5, 0.4}(f) = P_1^{0.5, 0.4}(f) + 0.4^2 \times (v_{0.5}(a) + v_{0.5}(c)) = 0.46$. Quand $i = 3$, f reçoit négativement la valeur de b (qui est non-attaqué) et la valeur de 0.5 de e tous deux atténués par δ^3 : $P_3^{0.5, 0.4}(f) = P_2^{0.5, 0.4}(f) - 0.4^3 \times (v_{0.5}(b) + v_{0.5}(e)) = 0.364$. Et ainsi de suite.

$P_i^{0.5, 0.4}$	a	b	c	d	e	f
0	0.5	1	0.5	0.5	0.5	0.5
1	-0.1	1	0.1	0.1	0.3	0.3
2	-0.02	1	0.1	0.34	0.38	0.46
3	-0.052	1	0.1	0.308	0.316	0.364
⋮	⋮	⋮	⋮	⋮	⋮	⋮
14	-0.0402	1	0.1	0.3161	0.3506	0.3736

TABLE 1 – Calcul de l'évaluation P de chaque argument dans F_1 avec $\epsilon = 0.5$ et $\delta = 0.4$

La proposition garantit la convergence de l'évaluation P pour chacun des arguments présents dans un système d'argumentation.

Proposition 1 Soient $\langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation et un facteur d'atténuation $\delta \in]0, 1[$. Pour tout $a \in \mathcal{A}$, la séquence $\{P_i^{\epsilon, \delta}(a)\}_{i=0}^{+\infty}$ converge.

Ainsi, la valeur de propagation d'un argument a correspond à la valeur $P^{\epsilon, \delta}(a) = \lim_{i \rightarrow +\infty} P_i^{\epsilon, \delta}(a)$.

Exemple 1 (cont.) La valeur de propagation de chaque argument présent dans le système d'argumentation F_1 (voir Figure 2) correspond à la dernière ligne grisée ($i = 14$) de la table 1.

3.2 La méthode de propagation en deux phases

Les arguments non-attaqués jouent un rôle important dans l'évaluation de l'acceptabilité des arguments dans les sémantiques de Dung. Bien que, comme expliqué dans l'introduction, ces arguments ne doivent pas toujours être considérés comme les plus acceptables, ils gardent tout de même un rôle spécifique permettant, par exemple, de distinguer les branches attaquantes et défensives (comme cela a été suggéré dans l'approche d'évaluation globale de Cayrol et Lagasque-Schiex [7]). En s'inspirant de cela, notre méthode est composée de deux phases :

1) Durant la première phase, seuls les arguments non-attaqués propagent leurs valeurs dans le graphe d'argumentation. Pour cela, tous les arguments attaqués commencent avec une valeur initiale de 0 ($\epsilon = 0$). A l'issue de cette propagation, une première comparaison est effectuée sur la base de leur valeur de propagation.

2) Lors de la seconde phase (permettant de départager les éventuels arguments possédant encore un niveau d'acceptabilité identique), nous recommençons la phase de propagation mais, cette fois, avec une valeur initiale non nulle pour les arguments attaqués ($\epsilon \neq 0$).

Définition 4 Soient $\epsilon \in]0, 1[$ et $\delta \in]0, 1[$. La sémantique graduée **Propagation à Profondeur Variable**¹ $vdp^{\epsilon, \delta}$ associe à tout système d'argumentation $\langle \mathcal{A}, \mathcal{R} \rangle$ un classement \succeq sur \mathcal{A} tel que $\forall a, b \in \mathcal{A}$:

$$\begin{aligned} a \succeq b \text{ ssi} \\ P^{0, \delta}(a) > P^{0, \delta}(b) \text{ ou,} \\ (P^{0, \delta}(a) = P^{0, \delta}(b) \text{ et } P^{\epsilon, \delta}(a) \geq P^{\epsilon, \delta}(b)) \end{aligned}$$

Exemple 1 (cont.) En appliquant la définition précédente, il est d'abord nécessaire de calculer la valeur de propagation pour chaque argument avec $\epsilon = 0$. Nous obtenons les valeurs suivantes : $P^{0, 0.4}(a) = -0.4105$, $P^{0, 0.4}(b) = 1$, $P^{0, 0.4}(c) = -0.4$, $P^{0, 0.4}(d) = 0.1642$, $P^{0, 0.4}(e) = 0.0263$ et $P^{0, 0.4}(f) = -0.0656$.

Et en comparant ces valeurs de propagation, nous obtenons le classement suivant :

$$b > d > e > f > c > a$$

1. Variable-Depth Propagation en anglais

Notons qu'il n'existe aucune paire d'arguments possédant les mêmes scores, il n'est donc pas nécessaire d'effectuer la seconde étape dans ce cas précis.

Une préoccupation pourrait être que la valeur de ϵ , utilisée dans la deuxième phase, pourrait changer le classement obtenu. Nous montrons que ce n'est pas le cas :

Proposition 2 Soit $F = \langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation et $\delta \in]0, 1[$. $\forall \epsilon, \epsilon' \in]0, 1[$,

$$vdp^{\epsilon, \delta} = vdp^{\epsilon', \delta}$$

Notons que même si différentes valeurs de ϵ ne changent en rien le classement obtenu entre les arguments, il reste essentiel pour distinguer les arguments non-attaqués et les arguments attaqués dans la fonction d'évaluation v_ϵ (définition 3). Cependant, ce paramètre est un élément purement interne sans aucun effet sur le résultat de la procédure. C'est pourquoi, pour une meilleure lisibilité, vdp^δ sera utilisé à la place de $vdp^{\epsilon, \delta}$ pour décrire notre sémantique graduée paramétrique en général.

A l'inverse, deux valeurs de δ peuvent produire des classements différents. En effet, ce paramètre permet de choisir la portée d'influence des arguments dans le système d'argumentation tout en garantissant la convergence de l'évaluation P . Par exemple, avec une valeur de δ proche de 0, seuls les arguments proches (et donc une partie restreinte de l'AF) seront pris en compte lors du calcul des valeurs de propagation. A l'inverse, une valeur de δ proche de 1, permettra à la (quasi) totalité des arguments (s'il existe un chemin) d'être pris en considération.

Suivant le principe d'atténuation utilisé en persuasion, il semble naturel de supposer que les arguments situés à une longue distance d'un autre argument deviennent inefficaces.

En termes de conception, il semble intéressant de pouvoir contrôler ce paramètre pour spécifier une profondeur maximale au delà de laquelle les arguments n'ont plus d'influence sur le score des autres arguments. Afin de mieux comprendre comment prendre en considération ce principe, considérons l'algorithme suivant permettant de calculer la valeur de propagation. D'abord, une valeur positive est assignée à chaque argument ($\forall a \in \mathcal{A}$, $P_0^{\epsilon, \delta}(a) = 1$ si a n'est pas attaqué ou $P_0^{\epsilon, \delta}(a) = \epsilon$ sinon). Puis, à l'étape $i \in \mathbb{N}^*$, nous ajoutons (ou retirons) au score accumulé jusqu'à l'étape précédente ($P_{i-1}^{\epsilon, \delta}(a)$), les valeurs atténuées (v_ϵ et δ^i) reçues des arguments situés au début d'un chemin de longueur i (Δ_i) : $P_i^{\epsilon, \delta}(a) = P_{i-1}^{\epsilon, \delta}(a) + (-1)^i \delta^i \sum_{b \in \Delta_i(a)} v_\epsilon(b)$. Le

processus s'arrête quand, entre deux étapes, la différence avec l'étape précédente pour toutes les évaluations P est strictement plus petite qu'un degré de précision fixé μ , i.e. $\forall a \in \mathcal{A}$, $|P_i^{\epsilon, \delta}(a) - P_{i-1}^{\epsilon, \delta}(a)| < \mu$. Ainsi, étant donné un degré de précision, il est possible de choisir un δ en fonction de la profondeur maximale espérée.

Proposition 3 Soient F un système d'argumentation, $i \in \mathbb{N}^*$ une profondeur maximale et $\mu > 0$ un degré de précision. Si $\delta < \sqrt[i]{\frac{\mu}{\max_{a \in \text{Arg}(F)} (|\Delta_i(a)|)}}$ alors la séquence $\{P_i^{\epsilon, \delta}(a)\}_{i=0}^{+\infty}$ converge avant l'étape $i + 1$.

Exemple 1 (cont.) Supposons que la profondeur maximale espérée est de 5. En utilisant la formule précédente, avec un degré de précision de $\mu = 10^{-4}$, δ devrait être inférieur à $\sqrt[5]{\frac{10^{-4}}{3}} \simeq 0.127$. Ainsi, une valeur proche de cette limite, par exemple $\delta = 0.12$, assure que seuls les arguments situés jusqu'à une distance de 5 (inclus) seront considérés.

Utiliser ce principe d'atténuation apporte également certains avantages calculatoires. En effet, le nombre d'étapes nécessaires pour déterminer la valeur de propagation de chaque argument étant plus petit que si tout le système d'argumentation devait être parcouru, le classement entre les arguments est donc retourné plus rapidement.

4 Propriétés pour les sémantiques graduées

Dans cette section, nous vérifions quelles propriétés sont satisfaites par $\text{vd}\rho$. Nous commençons par discuter de la propriété *Void Precedence* (Section 4.1), avant de lister les différentes propriétés existantes dans la littérature (Section 4.2). Différentes valeurs de δ pouvant retourner différents classements, nous proposons, dans la section 4.3, une étude expérimentale montrant que la diversité des classements obtenus reste assez faible.

4.1 Void Precedence

L'un des points où $\text{vd}\rho$ se distingue des autres sémantiques graduées existantes est qu'un argument attaqué peut éventuellement avoir un meilleur score (et donc être plus acceptable) qu'un argument non-attaqué. En effet, quand un argument possède plusieurs branches défensives et peu de branches attaquantes, celui-ci reçoit beaucoup de valeurs positives et peu de valeurs négatives. Ainsi, en fonction de la valeur de δ , cet argument peut obtenir un score supérieur à celui d'un argument non-attaqué.

Void Precedence (VP) Un argument non-attaqué doit être strictement plus acceptable qu'un argument attaqué.

$$\Delta_1(a) = \emptyset \text{ et } \Delta_1(b) \neq \emptyset \Rightarrow a > b$$

Illustrons cela sur l'argumentaire persuasif de la figure 1 :

Exemple 2 L'argument a_1 possède deux branches défensives. Ainsi la valeur de propagation de chacun des arguments, avec $\delta = 0.95$ et $\epsilon = 0$, est $P^{0.0.95}(a_1) = 1.805$, $P^{0.0.95}(a_2) = -1.9$ and $P^{0.0.95}(a_3) = P^{0.0.95}(a_4) = 1$. Avec une valeur de ϵ non nulle ($\epsilon = 0.5$), nous obtenons les valeurs de propagation suivantes : $P^{0.5.0.95}(a_1) = 1.83$,

$P^{0.5.0.95}(a_2) = -1.4$ and $P^{0.5.0.95}(a_3) = P^{0.5.0.95}(a_4) = 1$. Par conséquent, il est possible d'inférer le classement suivant où a_1 est plus acceptable que les arguments non-attaqués a_3 et a_4 :

$$a_1 > a_3 \simeq a_4 > a_2$$

Nous montrons que le paramètre δ possède une limite en dessous de laquelle VP est satisfaite.

Proposition 4 Soient $\langle \mathcal{A}, \mathcal{R} \rangle$ un système d'argumentation et $\delta^M = \sqrt{\frac{1}{\max_{a \in \mathcal{A}} (|\Delta_2(a)|)}}$.

Si $\delta < \delta^M$ alors $\text{vd}\rho^\delta$ satisfait VP

Exemple 2 (cont.) L'argument a_1 possède le plus grand nombre de défenseurs directs avec $|\Delta_2(a_1)| = 2$. Afin de satisfaire la propriété VP, la valeur de δ doit être : $\delta < \sqrt{1/2} \simeq 0.7071$. Donc si $\delta = 0.7$, les valeurs de propagation sont $P^{0.0.7}(a_1) = 0.98$, $P^{0.0.7}(a_2) = -1.4$ et $P^{0.0.7}(a_3) = P^{0.0.7}(a_4) = 1$ quand $\epsilon = 0$ et $P^{0.5.0.7}(a_1) = 1.13$, $P^{0.5.0.7}(a_2) = -0.9$, $P^{0.5.0.7}(a_3) = P^{0.5.0.7}(a_4) = 1$ quand $\epsilon = 0.5$. Ce qui permet d'obtenir le classement suivant :

$$a_3 \simeq a_4 > a_1 > a_2$$

Il est important de noter que notre méthode se distingue des autres approches en ce qui concerne la propriété VP, mais dans une certaine mesure seulement. Par exemple, dans un argumentaire persuasif, une ligne de défense unique n'est pas suffisante pour devenir plus convaincante qu'un argument non-attaqué (i). D'un autre côté, si cette condition n'est pas remplie, une condition simple (ii) peut être énoncée afin de violer VP dans les argumentaires persuasifs :

Proposition 5 Soit $PP = \langle \mathcal{A}, \mathcal{R} \rangle$ un argumentaire persuasif avec $a \in \mathcal{A}$ comme argument racine. Alors,

(i) si $|\Delta^B(a)| < 2$ alors $\text{vd}\rho^\delta$ satisfait VP ;

(ii) si $|\Delta^B(a)| \geq 2$ et $\delta > \sqrt{\frac{m}{|\Delta^B(a)|}}$ avec m la longueur de la plus longue branche défensive de a alors $\text{vd}\rho^\delta$ viole VP.

Le procatalepsis et le principe d'atténuation peuvent attirer la valeur de δ dans des directions opposées. La compréhension de cette interaction peut fournir des informations précieuses, en particulier dans le contexte de la persuasion. Supposons par exemple que le persuadeur sait qu'une valeur de δ donnée est attendue, correspondant au profil d'un public spécifique, cette valeur étant fixée, il est possible de conclure qu'un certain nombre de branches de défenses sera requis. Par conséquent, au lieu de développer, disons, deux longues lignes de persuasion, le persuadeur favorisera plutôt le déploiement d'un certain nombre de lignes alternatives dans son argumentaire persuasif.

Il s'avère que dans le contexte de notre méthode, la propriété VP coïncide avec une autre propriété étudiée dans la

littérature, à savoir *Defense Precedence* (DP), qui indique que, lorsque deux arguments possèdent le même nombre d’attaquants directs, celui étant défendu au moins une fois doit être plus acceptable que celui qui n’est jamais défendu. Par conséquent, nous avons :

Proposition 6 vdp^δ satisfait VP ssi DP est satisfait

Il est important de noter que ce n’est pas le cas en général puisqu’il existe des sémantiques qui satisfont VP mais pas DP. En d’autres termes, jouer avec le paramètre δ permet également de contrôler la propriété DP.

4.2 Autres propriétés

Plusieurs autres propriétés ont été introduites et étudiées dans la littérature (voir le travail de Bonzon et al. [5] pour un aperçu). Afin de mieux comprendre le comportement de notre méthode et pouvoir la comparer aux autres sémantiques, nous vérifions, dans cette section, quelles sont les propriétés satisfaites par vdp^δ (en omettant certaines d’entre elles lorsqu’il existe des incompatibilités connues). Concernant les propriétés étudiées dans ce papier, nous fournissons uniquement leur définition informelle mais invitons le lecteur à se référer à [5] pour les définitions formelles.

Soient $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ un système d’argumentation et deux arguments $a, b \in \mathcal{A}$.

Abstraction (Abs) Le classement sur \mathcal{A} doit être défini uniquement sur la base de la relation d’attaque entre les arguments.

Indépendance (In) L’ordre entre deux arguments a et b doit être indépendant des arguments qui ne sont connectés ni à a ni à b .

Total (Tot) Tous les arguments sont comparables entre eux.

Non-attaqués Equivalence (NaE) Tous les arguments non-attaqués ont le même niveau d’acceptabilité.

Les *propriétés locales* se concentrent sur les attaquants/défenseurs directs :

Counter-Transitivity (CT) Si les attaquants directs de b sont au moins aussi nombreux et acceptables que ceux de a , alors a doit être au moins aussi acceptable que b .

Strict Counter-Transitivity (SCT) Si CT est satisfaite et que les attaquants directs de b sont strictement plus nombreux ou acceptables que ceux de a , alors a doit être strictement plus acceptable que b .

Defense Precedence (DP) Soient deux arguments avec le même nombre d’attaquants directs. Un argument défendu doit être strictement plus acceptable qu’un argument non défendu.

Les *propriétés globales* précisent comment le classement devrait être affecté sur la comparaison des branches attaquantes et défensives :

Addition of Defense Branch (+DB) Ajouter une branche défensive à n’importe quel argument attaqué améliore son acceptabilité.

Increase of Attack Branch (\uparrow AB) Augmenter la longueur d’une branche attaquante d’un argument améliore son acceptabilité.

Addition of Attack Branch (+AB) Ajouter une branche attaquante à un argument dégrade son acceptabilité.

Increase of Defense Branch (\uparrow DB) Augmenter la longueur d’une branche défensive d’un argument dégrade son acceptabilité.

Attack vs Full Defense (AvsFD) Un argument sans aucune branche attaquante est strictement plus acceptable qu’un argument uniquement attaqué par un argument non-attaqué.

Pour les argumentaires persuasifs, cette propriété (AvsFD) peut être simplement reformulée “un argument sera plus acceptable s’il possède un argumentaire persuasif plutôt que d’un attaquant unique”.

Vérifions maintenant lesquelles de ces propriétés sont satisfaites par vdp :

Proposition 7 Soit $\delta \in]0, 1[$. vdp^δ satisfait les propriétés Abs, In, Tot, NaE, +AB et AvsFD.

Nous montrons également que certaines propriétés globales sont satisfaites par vdp quand le facteur d’atténuation δ est assez grand.

Proposition 8 Avec $\delta \in]\delta^m, 1[$ tel que $\delta^m = \sqrt[i]{\frac{\mu}{\max_{a \in \text{Arg}(F)} (|\Delta_i(a)|)}}$ où i représente la longueur de la branche ajoutée ou étendue alors vdp^δ satisfait également +DB, \uparrow DB et \uparrow AB.

La table 2 résume ces résultats. À titre de comparaison, nous incluons également les résultats de certaines sémantiques de la littérature où le même ensemble de propriétés [5] a déjà été vérifié. C’est le cas de la sémantique utilisée pour les systèmes d’argumentation sociaux (SAF) [14] restreint au cadre des systèmes d’argumentation de Dung, la sémantique Categorizer (Cat) [3, 17], la Discussion-based semantics (Dbs) et la Burden-based semantics (Bbs) [1], la sémantique globale basée sur les tuples (Tuples*) [7] et la sémantique introduite par Matt et Toni (M&T) [15].

Une première remarque est que vdp satisfait les propriétés jugées “basiques” selon [5] (Abs, In, +AB, NaE et Tot), à l’exception de VP (voir discussion dans la section 4.1).

Nous pouvons également noter que vdp satisfait AvsFD dans tous les cas, ainsi que la propriété +DB pour un δ particulier. En effet, la possibilité de mieux classer les arguments possédant plusieurs défenseurs plutôt qu’un argument défendu une unique fois semble intéressant. Par exemple, en persuasion, une affirmation défendue par différents arguments peut-être plus crédible qu’une affirmation défendue seulement une fois.

Propriétés	SAF	Cat	DBs	Bbs	Tuples*	M&T	vdp $^\delta$	vdp $^{\delta'}$
Abs	✓	✓	✓	✓	✓	✓	✓	✓
In	✓	✓	✓	✓	✓	✓	✓	✓
Tot	✓	✓	✓	✓	×	✓	✓	✓
NaE	✓	✓	✓	✓	✓	✓	✓	✓
+AB	✓	✓	✓	✓	✓	✓	✓	✓
AvsFD	×	×	×	×	✓	✓	✓	✓
+DB	×	×	×	×	✓	×	×	✓
↑AB	✓	✓	✓	✓	✓	×	×	✓
↑DB	✓	✓	✓	✓	✓	×	×	✓
VP	✓	✓	✓	✓	✓	✓	×	✓
DP	✓	✓	✓	✓	×	×	×	✓
CT / SCT	✓	✓	✓	✓	×	×	×	×

TABLE 2 – Résumé des propriétés satisfaites par vdp ($\forall \delta$, et avec $\delta^m < \delta' < \delta^M$) ainsi que certaines sémantiques graduées existantes où le même ensemble de propriétés ont déjà été vérifié. Les cellules grisées représentent les résultats fournis dans ce papier.

Enfin, les seules propriétés n'étant jamais satisfaites par vdp sont CT et SCT. Cependant, il est facile de démontrer que ces propriétés sont incompatibles avec +DB. Intuitivement, lorsque +DB considère qu'ajouter une défense doit renforcer un argument, SCT dit que le fait d'ajouter n'importe quelle branche (ce qui inclut les branches défensives) doit diminuer son acceptabilité.

4.3 Diversité des classements

Une caractéristique intéressante de notre sémantique est donc que l'utilisateur peut choisir si VP est satisfait ou non, donnant ainsi lieu à des classements différents. Cependant, on pourrait craindre que la diversité des classements soit si élevée que la sémantique devienne trop sensible aux légères modifications de δ . Les résultats résumés dans la table 3 montre cependant une certaine proximité entre les classements obtenues pour différentes valeurs de δ .

δ	0.001	0.2	0.4	0.6	0.8	0.9
0.001	100	99.94	99.45	95.91	89.38	86.26
0.2	99.94	100	99.48	95.87	89.37	86.36
0.4	99.45	99.48	100	96.29	89.87	86.70
0.6	95.91	95.87	96.29	100	93.18	90.14
0.8	89.38	89.37	89.87	93.18	100	96.84
0.9	86.26	86.36	86.70	90.14	96.84	100

TABLE 3 – Degrés de similarité entre différentes valeurs de δ

Pour obtenir ces résultats, nous avons appliqué notre sémantique pour plusieurs valeurs de $\delta \in \{0.001, 0.2, 0.4, 0.6, 0.8, 0.9\}$ sur 1000 systèmes d'argu-

mentation générés aléatoirement². Nous avons ensuite calculé le degré de similarité qu'il existe entre deux classements issus de deux δ différents en utilisant la distance de Kendall tau [13].

Définition 5 Soient $\langle \mathcal{A}, \mathcal{R} \rangle$ et $\tau_{\sigma_1}, \tau_{\sigma_2}$ les classements retournés par les sémantiques graduées σ_1 et σ_2 respectivement. La **distance de Kendall Tau** entre τ_{σ_1} et τ_{σ_2} est calculée comme suit :

$$K(\tau_{\sigma_1}, \tau_{\sigma_2}) = \frac{\sum_{(i,j) \in \mathcal{A}} \overline{K}_{i,j}(\tau_{\sigma_1}, \tau_{\sigma_2})}{0.5 \times |\mathcal{A}| \times (|\mathcal{A}| - 1)}$$

avec :

- $\overline{K}_{i,j}(\tau_{\sigma_1}, \tau_{\sigma_2}) = 1$ si $i >^{\sigma_1} j$ et $i >^{\sigma_2} j$, ou $i <^{\sigma_1} j$ et $i <^{\sigma_2} j$, ou $i \simeq^{\sigma_1} j$ et $i \simeq^{\sigma_2} j$,
- $\overline{K}_{i,j}(\tau_{\sigma_1}, \tau_{\sigma_2}) = 0$ si $i >^{\sigma_1} j$ et $i <^{\sigma_2} j$ ou vice versa,
- $\overline{K}_{i,j}(\tau_{\sigma_1}, \tau_{\sigma_2}) = 0.5$ si $i >^{\sigma_1} j$ ou $i <^{\sigma_1} j$ et $i \simeq^{\sigma_2} j$ ou vice versa.

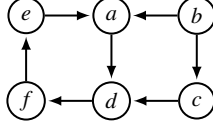
Le plus petit degré de similarité (86.26%) est logiquement observé entre la plus petite et la plus grande valeur de δ . Mais ce degré reste globalement très élevé, montrant que la sémantique reste assez stable lorsque ce paramètre varie.

5 Comparaison avec les sémantiques existantes

Nous allons maintenant montrer qu'en général, les différentes sémantiques proposées dans la littérature peuvent

2. Les algorithmes de génération sont basés sur ceux utilisés pour produire les benchmarks de la compétition ICCMA'15, voir <http://argumentationcompetition.org/2015/results.html>

renvoyer un large éventail de classements. Pour cela, nous appliquons ces sémantiques sur l'exemple de la figure 2 que nous rappelons ici.



La gamme de sémantiques considérées ici est plus importante que dans la section précédente, car nous incluons des sémantiques récentes pour lesquelles les propriétés axiomatiques n'ont pas encore été étudiées dans la littérature (du moins pas entièrement). Cependant, deux catégories de sémantiques graduées sont exclues de cette étude puisqu'elles possèdent des caractéristiques rendant difficile la comparaison avec les autres sémantiques. La première concerne les sémantiques retournant un pré-ordre partiel entre les arguments (*i.e.* certains arguments peuvent être incomparables) comme la sémantique globale *Tuples** introduite par Cayrol et Lagasque-Schiex [7] ou la sémantique introduite par Grossi et Modgil [11]. La seconde catégorie est composée de sémantiques retournant un ensemble de classements pour un même système d'argumentation [10]. Ainsi, nous considérons les sémantiques *Cat*, *M&T*, *SAF*, *Dbs* and *Bbs* déjà étudiées dans la section 4.2, ainsi que la sémantique utilisant les *fuzzy label (FL)* [8], α -Burden-based semantics (α -*Bbs*) [2], la sémantique de comptage (*CS*) [18] et les sémantiques de propagation *Propa ϵ* , *Propa $_{1+\epsilon}$* , *Propa $_{1-\epsilon}$* introduites par Bonzon et al. [6]. Tous les classements sont représentés dans la table 4.

Sémantiques	Classements
M&T	$b > d \simeq e > a \simeq c \simeq f$
FL	$b > d > e > f > a \simeq c$
SAF	$b > d > e > f > c > a$
α -BBS ($\alpha = 5$)	
<i>Propa$_{1-\epsilon}$</i>	
vdp^δ ($\delta = 0.3$)	
Dbs/Bbs	$b > f > e > c > d > a$
CS	
α -BBS ($\alpha = 0.5$)	
<i>Propaϵ</i> ($\epsilon = 0.8$)	
Cat	
<i>Propaϵ</i> ($\epsilon = 0.3$)	$b > f > e > d > c > a$
<i>Propa$_{1+\epsilon}$</i>	
vdp^δ ($\delta = 0.8$)	
	$d > b > e > c > f > a$

TABLE 4 – Classements obtenus avec les différentes sémantiques graduées sur F_1 (figure 2).

Notre sémantique se distingue clairement des autres sémantiques par le fait qu'avec $\delta \geq \frac{\sqrt{2}}{2}$, d , qui est attaqué, est strictement plus acceptable que b qui n'est pas attaqué.

En effet, grâce au cycle pair, d reçoit uniquement des poids positifs provenant du seul argument non-attaqué b . Cependant, d ne fait pas toujours partie des “meilleurs” arguments comme c'est le cas avec les sémantiques considérant qu'une nouvelle branche d'attaque peut renforcer un argument (*i.e.* les sémantiques qui satisfont +DB et AvsFD). En effet, les sémantiques considérant, à l'inverse, qu'une défense est une attaque affaiblie (*i.e.* les sémantiques qui satisfont SCT) jugent que même si d est défendu, il reste néanmoins directement attaqué une fois de plus que c , e et f . Le raisonnement inverse tient également avec l'argument f considéré comme un des “meilleurs” arguments pour les sémantiques satisfaisant SCT alors qu'il reste plutôt acceptable pour les sémantiques satisfaisant +DB et AvsFD. L'argument a reste quant à lui toujours l'argument le moins acceptable puisqu'il est attaqué directement par b mais également par e . C'est pourquoi il est toujours considéré comme moins acceptable que c qui est uniquement attaqué par b .

6 Conclusion

Baucoup de sémantiques graduées ont été proposées récemment dans la littérature. Malgré des études détaillées de leurs propriétés, il reste difficile de déterminer quelle sémantique est plus appropriée pour un contexte d'application donné. Dans ce papier, nous avons pris le problème à l'envers en proposant une sémantique appliquée au contexte de la persuasion, qui prend en considération deux principes bien connus dans la pratique : le procatalepsis et l'atténuation. Il s'avère qu'aucune des sémantiques existantes n'est vraiment appropriée pour ce contexte là – toutes s'engagent par exemple en faveur de la propriété “Void Precedence” qui est en contradiction avec l'idée du procatalepsis. C'est pourquoi nous avons introduit une nouvelle sémantique paramétrique basée sur la notion de propagation. Un facteur d'atténuation est utilisé pour permettre à la fois la convergence et la gestion de l'impact, plus ou moins important en fonction de la longueur du chemin, d'un argument sur un autre. Nous montrons que, grâce à ce facteur d'atténuation, le principe d'atténuation peut être capturé en fixant une profondeur maximale d'influence. Pour certaines valeurs de ce paramètre, VP n'est pas satisfait, ce qui permet de capturer le principe de procatalepsis dans les argumentaires persuasifs. Nous vérifions également quelles sont les autres propriétés satisfaites par notre méthode, tout en proposant une étude expérimentale visant à montrer à quel point le classement entre deux valeurs de ce paramètre reste proche. Nous pensons que cette méthode est un outil utile pour la persuasion, permettant par exemple d'évaluer l'impact relatif que peut avoir différents argumentaires persuasifs. Les travaux futurs incluent de tester cette sémantique sur des outils de persuasion calculatoire existants. Notre méthodologie peut s'avérer inté-

ressante dans des contextes différents : en se questionnant, par exemple, sur la pertinence d'une sémantique proposée dans la littérature dans d'autres contextes d'application (comme la négociation). Cela permettrait de détecter les phénomènes n'ayant éventuellement pas été capturés correctement et qui demanderaient d'éventuels ajustements.

7 Remerciements

Ce travail a bénéficié d'une aide de l'Agence Nationale de la Recherche portant la référence ANR-13-BS02-0004 dans le cadre du projet AMANDE.

Références

- [1] Amgoud, Leila et Jonathan Ben-Naim: *Ranking-Based Semantics for Argumentation Frameworks*. Dans *Proc. of the 7th International Conference on Scalable Uncertainty Management, (SUM'13)*, pages 134–147, 2013.
- [2] Amgoud, Leila, Jonathan Ben-Naim, Dragan Doder et Srdjan Vesic: *Ranking Arguments With Compensation-Based Semantics*. Dans *Proc. of the 15th International Conference on Principles of Knowledge Representation and Reasoning, (KR'16)*, pages 12–21, 2016.
- [3] Besnard, Philippe et Anthony Hunter: *A logic-based theory of deductive arguments*. *Artificial Intelligence*, 128(1-2) :203–235, 2001.
- [4] Besnard, Philippe et Anthony Hunter: *Elements of Argumentation*. MIT Press, 2008.
- [5] Bonzon, Elise, Jérôme Delobelle, Sébastien Konieczny et Nicolas Maudet: *A Comparative Study of Ranking-based Semantics for Abstract Argumentation*. Dans *Proc. of the 30th AAAI Conference on Artificial Intelligence (AAAI'16)*, pages 914–920, 2016.
- [6] Bonzon, Elise, Jérôme Delobelle, Sébastien Konieczny et Nicolas Maudet: *Argumentation Ranking Semantics based on Propagation*. Dans *Proc. of the 6th International Conference on Computational Models of Argument (COMMA'16)*, pages 139–150, 2016.
- [7] Cayrol, Claudette et Marie-Christine Lagasquie-Schiex: *Graduality in Argumentation*. *Journal of Artificial Intelligence Research*, 23 :245–297, 2005.
- [8] Costa Pereira, Célia da, Andrea Tettamanzi et Serena Villata: *Changing One's Mind : Erase or Rewind ?* Dans *Proc. of the 22nd International Joint Conference on Artificial Intelligence, (IJCAI'11)*, pages 164–171, 2011.
- [9] Dung, Phan Minh: *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.
- [10] Gabbay, Dov M.: *Equational approach to argumentation networks*. *Argument & Computation*, 3(2-3) :87–142, 2012.
- [11] Grossi, Davide et Sanjay Modgil: *On the Graded Acceptability of Arguments*. Dans *Proc. of the 24th International Joint Conference on Artificial Intelligence, (IJCAI'15)*, pages 868–874, 2015.
- [12] Hunter, Anthony: *Opportunities for Argument-Centric Persuasion in Behaviour Change*. Dans *Proc. of the 14th European Conference on Logics in Artificial Intelligence, (JELIA'14)*, pages 48–61, 2014.
- [13] Kendall, M. G.: *A New Measure of Rank Correlation*. *Biometrika*, 30(1/2) :81–93, juin 1938, ISSN 00063444.
- [14] Leite, João et João Martins: *Social Abstract Argumentation*. Dans *Proc. of the 22nd International Joint Conference on Artificial Intelligence, (IJCAI'11)*, pages 2287–2292, 2011.
- [15] Matt, Paul-Amaury et Francesca Toni: *A Game-Theoretic Measure of Argument Strength for Abstract Argumentation*. Dans *Proc. of the 11th European Conference on Logics in Artificial Intelligence, (JELIA'08)*, pages 285–297, 2008.
- [16] Pu, Fuan, Jian Luo et Guiming Luo: *Some Supplementaries to the Counting Semantics for Abstract Argumentation*. Dans *Proc. of the 27th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'15)*, pages 242–249, 2015.
- [17] Pu, Fuan, Jian Luo, Yulai Zhang et Guiming Luo: *Argument Ranking with Categoriser Function*. Dans *Proc. of the 7th International Conference on Knowledge Science, Engineering and Management, (KSEM'14)*, pages 290–301, 2014.
- [18] Pu, Fuan, Jian Luo, Yulai Zhang et Guiming Luo: *Attacker and Defender Counting Approach for Abstract Argumentation*. Dans *Proc. of the 37th Annual Meeting of the Cognitive Science Society, (CogSci'15)*, 2015.
- [19] Tan, Chenhao, Vlad Niculae, Cristian Danescu-Niculescu-Mizil et Lillian Lee: *Winning Arguments : Interaction Dynamics and Persuasion Strategies in Good-faith Online Discussions*. Dans *Proc. of the 25th International Conference on World Wide Web, (WWW'16)*, pages 613–624, 2016.
- [20] Walton, Douglas: *Dialog Theory for Critical Argumentation*. John Benjamins Publishing, 2007.

Jeu d'apprentissage basé sur le débat pour la construction de preuves mathématiques

Nadira Boudjani, Abdelkader Gouaich, Souhila Kaci

LIRMM - University of Montpellier
Montpellier, France

{boudjani, gouaich, kaci}@lirmm.fr

Résumé

Le débat est une méthode reconnue efficace pour l'apprentissage. C'est un processus interactif dans lequel les apprenants coopèrent en échangeant des arguments et des contre-arguments afin de résoudre un problème.

Nous proposons un jeu de débat pour l'apprentissage des mathématiques. En particulier apprendre comment structurer et construire des preuves mathématiques. Le cadre d'argumentation de Dung et ses extensions sont utilisés pour extraire les arguments acceptables qui forment la preuve. De plus, cela permet aux instructeurs de fournir des retours aux apprenants sans surcharge d'information.

Abstract

Debate is a valuable and effective method of learning. It is an interactive process in which learners cooperate by exchanging arguments and counter-arguments to solve a common question. We propose a debate-based learning game for mathematics classroom to teach how to structure and build mathematical proofs. Dung's argumentation framework and its extensions are used as a means to extract acceptable arguments that form the proof. Moreover this allows instructors to provide feedbacks to learners without information overload.

1 Introduction

Debate has been used as a learning method since antiquity notably by sophists such as Protagoras, Plato and Socrates. From that time, the principles of this method remained unchanged : a group of learners explore a common question by exchanging opinions, ideas in the form of arguments and counter-arguments. Such debates are based on a collaborative and progressive learning process. What has changed however are technologies that nowadays enable autonomous and ubiquitous learning.

The literature of educational science witnesses numerous examples of debate-based learning [20, 15]. Furthermore, the use of debate in several areas such as medicine, natural sciences and humanities has been experimentally demonstrated as an effective learning tool [5, 10]. In short, we can synthesize main advantages of debate-based learning as follows :

- (i) it allows a group of heterogeneous learners with different backgrounds to collectively solve a common problem by using their own skills [2],
- (ii) it improves critical thinking skills, reasoning and communication within a group since each learner has to justify and defend her point of view and constructively criticize others [5, 8],
- (iii) it increases motivation and involvement of learners by improving self-esteem and promoting social interaction [14],
- (iv) it makes explicit reasoning processes that led to conclusions. This provides instructors with information to identify misunderstandings and take actions to correct them.

The context We are interested in this paper in learning how to structure and build mathematical proofs. Mathematical skills are increasingly becoming a central criterion in skills evaluation as they are an important selection criterion for academic and professional applications. Therefore shortcomings in mathematics may be highly detrimental to students in both academic and professional opportunities. In the context of mathematical didactics several works have shown advantages of using debate in classes [7]. The object of the debate can be either to build a mathematical proof or to falsify a claim using sound mathematical deductions.

The problem & Contribution Despite the benefits witnessed in debates-based classes, we can mention some

limitations of current approaches. The first limitation concerns learner’s motivation and involvement. Although it has been demonstrated that learners show a high motivation in debate-based courses [10], it has also been acknowledged that this is influenced by instructor’s animation abilities and the fact that these courses are formal and mandatory. In order to promote an autonomous and ubiquitous learning beyond institutional environment, we propose a *gamification* approach that considers the debate as a genuine game with its intrinsic motivation levers. The second limitation concerns assessment of debate outcomes and its effect on instructors’ workload. In fact, assessment is fundamental in order to provide learners with continuous feedbacks. This task is often performed manually by the instructor who has to understand, evaluate and provide feedbacks based on the state of the debate. However, debate data are often overloaded by details of intermediate and erroneous stages of reasoning. While these stages are important to understand how learners have come to final conclusions, they do overload instructors with unnecessary details during assessment phases. As the basic ingredients in debates are arguments and counter-arguments, we use formal *argumentation* framework as a *means* to extract final conclusions from a debate before its assessment by instructors.

The remainder of this paper is structured as follows. In Section 2 we present necessary background on formal argumentation frameworks. In Section 3 we present our debate-based learning game. In Section 4 we illustrate our system on an example. Lastly we conclude.

2 Background on formal argumentation frameworks

Artificial intelligence witnesses a large amount of contributions in argumentation theory. In particular Dung’s argumentation framework is a pioneer work in the topic [6].

Definition 1 (Dung’s Framework) *An argumentation framework (AF) is a tuple $\langle \mathcal{A}, \text{Def} \rangle$, where \mathcal{A} is a finite set of arguments and $\text{Def} \subseteq \mathcal{A} \times \mathcal{A}$ is a binary defeat¹ relation. Given $A, B \in \mathcal{A}$, $A \text{ Def } B$ stands for “A defeats B”.*

The outcome of Dung’s AF is sets of arguments, called *extensions*, that are robust against defeats [6]. We say that $\mathcal{T} \subseteq \mathcal{A}$ *defends* A if $\forall B \in \mathcal{A}$ s.t. $B \text{ Def } A$, $\exists C \in \mathcal{T}$ such that $C \text{ Def } B$. We say that $\mathcal{T} \subseteq \mathcal{A}$ is *conflict-free* if $\nexists A, B \in \mathcal{T}$ such that $A \text{ Def } B$. A subset $\mathcal{T} \subseteq \mathcal{A}$ of arguments is *admissible* iff it is conflict-free and defends all elements in \mathcal{T} . Different acceptability semantics have been proposed. In particular $\mathcal{T} \subseteq \mathcal{A}$ is a *complete extension* of $\langle \mathcal{A}, \text{Def} \rangle$ iff it is admissible and contains all arguments it defends. \mathcal{T}

1. called *attack* in [6].

is a grounded extension $\langle \mathcal{A}, \text{Def} \rangle$ iff it is the minimal (for set inclusion) complete extension. For other semantics, see [6].

Several authors have considered a new kind of interaction called the *support* relation [13, 12, 4]. An abstract bipolar argumentation framework is an extension of Dung’s framework such that both defeat and support relations are considered.

Definition 2 (Bipolar argumentation framework) *An abstract bipolar argumentation framework (BAF) is a tuple $\langle \mathcal{A}, \text{Def}, \text{Supp} \rangle$, where $\langle \mathcal{A}, \text{Def} \rangle$ is Dung’s AF and $\text{Supp} \subseteq \mathcal{A} \times \mathcal{A}$ is a binary support relation. For $A, B \in \mathcal{A}$, $A \text{ Supp } B$ means “A supports B”.*

Defeat and support relations are combined to compute new defeat relations and recover Dung’s framework from which acceptable extensions are computed [13, 12, 4].

In this paper we are interested in deductive reasoning, i.e. a conclusion is derived from a set of premises.

Definition 3 (Argument) *Let Γ be a set of formulas constructed from a given language \mathcal{L} . An argument over Γ is a pair $A = \langle \Delta, \alpha \rangle$ such that*

- (i) $\Delta \subseteq \Gamma$,
- (ii) $\Delta \not\vdash_* \perp$,
- (iii) $\Delta \vdash_* \alpha$ and,
- (iv) for all $\Delta' \subset \Delta$, $\Delta' \not\vdash_* \alpha$,
where \vdash_* is the inference symbol.

Given two arguments $\langle \Delta, \alpha \rangle$ and $\langle \Delta', \alpha' \rangle$, we say that $\langle \Delta, \alpha \rangle$ *undercuts* $\langle \Delta', \alpha' \rangle$ iff for some $\phi \in \Delta'$, α and ϕ are contradictory w.r.t. the language at hand. $\langle \Delta, \alpha \rangle$ *rebuts* $\langle \Delta', \alpha' \rangle$ iff α and α' are contradictory. Then, $\langle \Delta, \alpha \rangle$ *defeats* $\langle \Delta', \alpha' \rangle$ iff $\langle \Delta, \alpha \rangle$ rebuts or undercuts $\langle \Delta', \alpha' \rangle$.

3 A debate-based learning game

Roughly our system builds on learners who exchange arguments for some purpose. The Oracle (who is the instructor) opens the discussion by providing a first argument of the form $\langle P, C \rangle$, where P is a set of premises and C is a conclusion. Then learners are engaged in a debate in which they exchange arguments to construct a proof for C given P . A set of relations is provided by the Oracle to connect arguments. A learner may add an argument/relation or pass her turn. The discussion is closed when decided by the learners or stopped by the Oracle after a time out. The output of the debate is a debate graph which is composed of arguments and relations constructed during the debate. The graph is submitted to the Oracle for evaluation. We first discuss and give our design choices. Then we propose a game-based modeling of our learning system.

3.1 Design choices

Different factors need to be considered for the game. Before we present our game-based modeling of the learning system let us expose these factors :

- *Argumentation mechanisms* : Three mechanisms have been distinguished in the literature to model the exchange of arguments between agents [17] : In a *Direct mechanism* every agent may propose a set of arguments at once. Then the process terminates. This mechanism is not appropriate in our setting because a learning process needs to be progressive. In a *synchronous mechanism* every agent may propose any set of arguments at the same time. The process is repeated until no agent wants to make more arguments. In a *dialectical mechanism* an order is assumed over agents to provide their arguments. Four variants (rigid, non rigid) × (single, multiple) can be obtained. A mechanism is rigid when an agent who passed her turn will no longer be allowed to propose arguments. The mechanism is not rigid if the agent is not discarded in such a situation. Also an agent may propose a single argument or multiple arguments when she takes her turn. *We use a dialectical, non rigid and single argument mechanism.*

- *Construction of the arguments & their validity* : Arguments may be provided by the Oracle or constructed by learners. In the second case, the Oracle provides a set of propositions upon which arguments will be constructed. *We choose the second option.* Now whatever arguments are constructed or provided we need to check their validity which refers to the satisfaction of conditions (i)-(iv) in Definition 3.

Condition (i) will always be satisfied as arguments will be constructed from a set of propositions provided by the Oracle.

An argument that does not satisfy condition (ii) should be removed.

An argument that does not satisfy condition (iv) should be modified to make its set of premises minimal.

Let us now consider condition (iii). For example the argument $\langle \{p\}, q \rangle$ is not valid w.r.t. (iii) because q does not logically follow from p . This argument should be removed. On the other hand, the argument $\langle \{(2 - \epsilon)(n + 2) < 2n + 1\}, 2 - \epsilon < \frac{2n+1}{n+2} \rangle$ will be considered as valid although $2 - \epsilon < \frac{2n+1}{n+2}$ does not logically follow from $(2 - \epsilon)(n + 2) < 2n + 1$ because the set of premises of the argument is not complete. In fact we need an additional constraint, namely $n + 2 > 0$. Such an argument can however be accepted in a debate. Then learners have to cooperate in order to defeat the argument or defend it by completing its set of

premises. This is coherent with a debate-based learning process in which arguments can be both collaboratively and progressively constructed.

We distinguish between two ways to deal with an argument that should be removed (e.g. $\langle \{p\}, q \rangle$ or condition (ii) not satisfied) or modified (condition (iv)). First notice that learners in the same group cooperate in order to solve a problem. Therefore they may be allowed to have a chat box by which they can discuss in order to convince a learner that her argument is not valid and should be removed/modified. We expect a cooperation from all learners. If not then the non valid argument will be refused by the Oracle (when the latter evaluates the debate graph at the end of the debate) which leads to the failure of the group to construct a correct proof. Another way to control the validity of an argument is to delegate this task to the Oracle in which case an argument is added to the debate graph only when it is valid. A penalty on the score of the group is applied each time a learner proposes a non valid argument. *We choose the first option.*

If constructed arguments do not comply with Definition 3 and not repaired/removed by learners then the group will fail to construct a correct proof.

- *Relations between arguments and their validity* :

- *Defeat relation* : This relation is syntactically defined and should be in the background of learners. In contrast to existing works [16, 17] in which the defeat relation is automatically stated by the system as soon as an agent proposes an argument we do believe that this relation should be stated by the learners themselves. This is a part of the learning process. Two ways to control the validity of a defeat relation are possible. Either we authorize defeats on defeats which can be captured by hierarchical argumentation frameworks [3], or the Oracle accepts a defeat relation in the debate graph only when it is valid. *We choose the second option in our game.*

- *Support relation* : we do not control the support relation because the objective of the game is to construct a proof which is built using the support relation. So it is up to learners to discuss/agree on a given support relation without the intervention of the Oracle.

- *Termination of the game* : The game terminates when the group stops or the Oracle decides so. *We choose the first option in our game.* The debate graph is submitted to the Oracle for evaluation.

- *Score function* : A score which is a penalty degree is given to a group of learners when its debate graph is evaluated by the Oracle. We define a penalty degree

as the number of irrelevant arguments present in the debate graph plus the number of non valid defeat relations refused by the Oracle during the debate.

3.2 Game-based modeling of the learning system

The main objective of the game is to create a debate environment for learners so that they can exchange arguments to prove the Oracle's claim. Learners need to be maintained engaged in a game. For this purpose we use social levers of motivation. In particular we divide learners into groups that will be made in competition. The Oracle initiates the debate by setting the question under the form of an argument.

The group that wins the game is the one that manages to build a debate graph accepted by the Oracle with a minimal penalty. To construct such a graph, learners have to use their domain knowledge to build arguments and correctly set relationships between arguments. This section formalizes the game by describing states of the game and actions that are transitions among states.

3.2.1 State of the game

To construct arguments we use a universe of discourse based on a given \mathcal{L} -language. We assume that this language is at least equipped with a conjunction operator. Given a set of arguments \mathcal{A} and a set of relation labels L (e.g. $L = \{\text{defeat}, \text{support}\}$), a state of a debate is a sequence of relations indexed by natural numbers and labels : $S : L \times \mathbb{N} \rightarrow 2^{\mathcal{A} \times \mathcal{A}}$. $S(l, k)_{l \in L, k \in \mathbb{N}}$ represents the content of the relation l at the k th step. \mathcal{S} denotes the set of all states.

3.2.2 Actions of the game

Adding a relation This action adds a new relationship between two (new or existing) arguments. This action takes as input the label of the relation and two arguments.

Definition 4 Given a relation label $l \in L$ and a couple of arguments (A, B) , the add action is a transition between states such that $\forall S, S' \in \mathcal{S}$, $S \xrightarrow{\text{add}(l,A,B)} S'$ iff

$$\exists k \in \mathbb{N}, \forall r \in L, \begin{cases} \forall m > k, S'(r, m) = S(r, m) = \emptyset \\ \forall n < k, S(r, n) = S'(r, n) \\ S(r, k) = \emptyset \\ S'(r, k) = S(r, k-1) \text{ for } r \neq l \\ S'(l, k) = S(l, k-1) \cup \{(A, B)\} \end{cases}$$

The *add* action makes a transition from S to S' iff the following conditions hold : (i) the game states contain only a finite number of known relation graphs. In other words, there exists an integer k for which all relation graphs of subsequent steps are empty, (ii) S and S' are equal until k th step. This means that S' copies S until the k th step, (iii) game state

S does not contain any information about the relation l at step k , (iv) finally, game state S' at k th step is equal to the graph of l at the $(k-1)$ th step to which the couple (A, B) is added.

Removing a relation During a debate, a player can remove a relation between arguments from the debate graph. This action takes as input the label of the relation and the couple to be removed. Specification of the remove action is similar to that of the add action, except that S' at k th step contains previous graph of l from which the pair (A, B) has been removed. That is $S'(l, k) = S(l, k-1) \setminus \{(A, B)\}$, the remaining conditions being identical.

Append action The goal of this action is to make an argument's premises and conclusion more specific. It is an auxiliary action that is built as a composition of a remove and add actions. Before we define this action, let us introduce a connector (\wedge) between arguments. Given $A = \langle P_1, C_1 \rangle$ and $B = \langle P_2, C_2 \rangle$, the notation $A \wedge B$ is an abbreviation for $\langle P_1 \cup P_2, C_1 \wedge C_2 \rangle$.

Definition 5 The append relation between states S and S' is defined as :

$$\forall S, S' \in \mathcal{S}, S \xrightarrow{\text{append}(l,A,B,C)} S' \text{ iff } \exists S'', S \xrightarrow{\text{remove}(l,A,B)} S'' \xrightarrow{\text{add}(l,A \wedge C,B)} S'$$

Submit action This action ends the game and submits the final debate state to the Oracle for evaluation.

4 Example : Mathematical proof

In this section we exemplify our debate-based learning system. Let

Q : "Prove that $\forall \epsilon > 0 \exists N \in \mathbb{N}$ such that $(n \geq N \Rightarrow 2 - \epsilon < \frac{2n+1}{n+2} < 2 + \epsilon)$ ".

We have the Oracle and 6 learners (players) : $l_1, l_2, l_3, l_4, l_5, l_6$.

The Oracle provides a first argument A_0 corresponding to Q. We have

$$A_0 = \langle \{\epsilon > 0, \exists N \in \mathbb{N}, n \geq N\}, 2 - \epsilon < \frac{2n+1}{n+2} < 2 + \epsilon \rangle.$$

We also have $L = \{\text{defeat}, \text{support}\}$ and a set of propo-

2. Notice that $P_1 \cup P_2$ may be inconsistent but remind that the validity of arguments is decided by learners.

sitions \mathcal{P} :

$$\begin{array}{lll} \{n \in \mathbb{N}, & \frac{2n+1}{n+2} < 2, & 2 - \epsilon < \frac{2n+1}{n+2}, \\ \epsilon > 0, & N > \frac{3}{\epsilon} - 2, & n > \frac{3}{\epsilon} - 2, \\ n = 1, & \neg(2 - \epsilon < \frac{2n+1}{n+2}), & \neg(\frac{2n+1}{n+2} < 2), \\ \exists N \in \mathbb{N}, & (2 - \epsilon)(n + 2) < 2n + 1, & \neg(N = \lceil \frac{3}{\epsilon} - 2 \rceil + 1), \\ \epsilon = -1, & \frac{2n+1}{n+2} < 2 + \epsilon, & \neg(\epsilon = -1), \\ n = -3, & \neg(n = -3) & n \geq N. \end{array}$$

The aim of the game is to demonstrate \mathcal{Q} . Let $Group_1$ and $Group_2$ be two groups of learners : $G_1 = \{l_1, l_2, l_3\}$ and $G_2 = \{l_4, l_5, l_6\}$. Each group has to demonstrate the proof individually. Not only the groups have to argue to construct the proof but they also have to do that as fast as possible and with minimal penalty degree.

We first illustrate the game on $Group_1$. Initially, the debate graph contains only A_0 . The discussion is dialectical : a random order is defined over the set of learners. Let l_1, l_2, l_3 be this order. Table 1 illustrates first states of the game.

The complete debate graph is given in Figure 1, where

$$\begin{array}{l} A_1 : \langle \{\epsilon > 0, \exists N \in \mathbb{N}, n \geq N\}, 2 - \epsilon < \frac{2n+1}{n+2} \rangle \\ A_2 : \langle \{\epsilon = -1, n = 1\}, \neg(2 - \epsilon < \frac{2n+1}{n+2}) \rangle \\ A_3 : \langle \{\epsilon > 0, \exists N \in \mathbb{N}, n \geq N\}, \frac{2n+1}{n+2} < 2 + \epsilon \rangle \\ A_4 : \langle \{\epsilon > 0, \frac{2n+1}{n+2} < 2\}, \frac{2n+1}{n+2} < 2 + \epsilon \rangle \\ A_5 : \langle \{\epsilon > 0\}, \epsilon > 0 \rangle \\ A_6 : \langle \{n \in \mathbb{N}\}, \frac{2n+1}{n+2} < 2 \rangle \\ A_7 : \langle \{n \in \mathbb{N}\}, n \in \mathbb{N} \rangle \\ A_8 : \langle \{n \in \mathbb{N}, (2 - \epsilon)(n + 2) < 2n + 1\}, 2 - \epsilon < \frac{2n+1}{n+2} \rangle \\ A_9 : \langle \{\epsilon > 0, n > \frac{3}{\epsilon} - 2\}, (2 - \epsilon)(n + 2) < 2n + 1 \rangle \\ A_{10} : \langle \{n \geq N, N > \frac{3}{\epsilon} - 2\}, n > \frac{3}{\epsilon} - 2 \rangle \\ A_{11} : \langle \{N > \frac{3}{\epsilon} - 2\}, N > \frac{3}{\epsilon} - 2 \rangle \\ A_{12} : \langle \{n \geq N\}, n \geq N \rangle \\ A_{13} : \langle \{n = -3\}, \neg(\frac{2n+1}{n+2} < 2) \rangle \\ A_{14} : \langle \{n \in \mathbb{N}\}, \neg(n = -3) \rangle \\ A_{15} : \langle \{n \in \mathbb{N}, \epsilon > 0\}, \frac{2n+1}{n+2} < 2 + \epsilon \rangle \\ A_{16} : \langle \{\epsilon > 0\}, \neg(\epsilon = -1) \rangle \end{array}$$

In this example learners may add a symmetric defeat between $\langle \{\epsilon > 0\}, \epsilon > 0 \rangle$ and $\langle \{\epsilon = -1\}, \epsilon = -1 \rangle$. In this case learners need to discuss in a chat box and convince the learner who added the defeat relation from the latter to the former to remove her defeat because $\epsilon > 0$ appears in the premises of A_0 so it is a fact that overrides $\epsilon = -1$. We do not make any explicit distinction between propositions because distinguishing facts and giving them priority should be done by the learners. This is a part of the learning process.

Assuming that the game ends when the learners agree to submit the debate graph to the Oracle. Thus $Group_1$ submits the graph of the *state final* using the action *submit*. Then the Oracle evaluates the debate graph

by computing acceptable arguments. To this aim, she computes the argumentation framework corresponding to the debate graph in order to compute acceptable extensions. The debate graph is composed of nodes representing arguments and two types of relations, namely defeat and support relations. Therefore bipolar AF is suitable. We have $BAF = \langle \mathcal{A}, \text{Def}, \text{Supp} \rangle$, where $\mathcal{A} = \{A_0, A_1, A_1 \wedge A_3, A_2, A_3, A_4, A_5, A_6, A_5 \wedge A_6, A_7, A_8, A_9, A_7 \wedge A_9, A_{10}, A_5 \wedge A_{10}, A_{11} \wedge A_{12}, A_{13}, A_{14}, A_{15}, A_{16}\}$, $\text{Def} = \{(A_2, A_1), (A_{16}, A_2), (A_{13}, A_6), (A_{14}, A_{13})\}$ and $\text{Supp} = \{(A_1 \wedge A_3, A_0), (A_8, A_1), (A_9 \wedge A_7, A_8), (A_{10} \wedge A_5, A_9), (A_{11} \wedge A_{12}, A_{10}), (A_{15}, A_3), (A_4, A_{15}), (A_5 \wedge A_6, A_4), (A_7, A_6), (A_5, A_{16}), (A_7, A_{14})\}$ ³. In this example “ A Supp B ” is interpreted as “the acceptance of A is necessary for the acceptance of B ” [4, 12].

As we previously indicated a bipolar argumentation framework can be translated into Dung’s argumentation framework by combining defeat and support relations. As we are looking for the proof of a given question, we need to compute the set of arguments that all learners agree on. To this aim the grounded extension is suitable as it is the set of arguments that are not defeated and those that are defeated but defended by acceptable arguments. Note that if the debate graph contains several valid proofs then all corresponding arguments will appear in the grounded extension. Given the grounded extension, the path of a proof can be retrieved from the debate graph thanks to the support relations.

Without entering in the details of computation, the grounded extension associated to the above BAF is $\{A_0, A_1, A_3, A_1 \wedge A_3, A_4, A_5, A_6, A_5 \wedge A_6, A_7, A_8, A_9, A_7 \wedge A_9, A_{10}, A_5 \wedge A_{10}, A_{11} \wedge A_{12}, A_{14}, A_{15}, A_{16}\}$.

The Oracle evaluates the grounded extension. Then she informs the group if it won/failed the game and returns a penalty degree (if any). The winning group is the one that correctly constructed the proof with lowest penalty degree. The Oracle also returns irrelevant arguments to winners and wrong arguments/relations to losers.

To this aim, the Oracle has two sets of arguments :

- (i) acceptable arguments : we distinguish between two kinds of acceptable arguments, namely relevant arguments which are arguments of the proof ($A_0, A_1, A_3, A_1 \wedge A_3, A_4, A_5, A_6, A_5 \wedge A_6, A_7, A_8, A_9, A_7 \wedge A_9, A_{10}, A_5 \wedge A_{10}, A_{11} \wedge A_{12}, A_{15}$) and irrelevant arguments (A_{14}, A_{16}).
- (ii) rejected arguments (A_2, A_{13}).

To win the game the grounded extension must not contain rejected arguments. The ideal situation is to build a debate graph such that its grounded extension contains only the relevant arguments.

3. A_{11} and A_{12} need not to be considered separately. In fact $A_{11} \wedge A_{12}$ is needed and sufficient.

# state	player	argument	action	state
0	\emptyset	A_0	\emptyset	$S_0 = \emptyset$
1	l_1	A_1	$add('support', A_1, A_0)$	$S_1('support', 1) = \{(A_1, A_0)\}$
2	l_2	A_2	$add('defeat', A_2, A_1)$	$S_2('support', 2) = \{(A_1, A_0)\}$ $S_2('defeat', 2) = \{(A_2, A_1)\}$
3	l_3	A_3	$append('support', A_1, A_0, A_3)$	$S_3('support', 3) = \{(A_1 \wedge A_3, A_0)\}$ $S_3('defeat', 3) = \{(A_2, A_1)\}$
4	l_1	\emptyset	$add('defeat', A_3, A_1)$ (not valid)	$S_4('support', 4) = \{(A_1 \wedge A_3, A_0)\}$ $S_4('defeat', 4) = \{(A_2, A_1)\}$
...

TABLE 1 – First game states during the game.

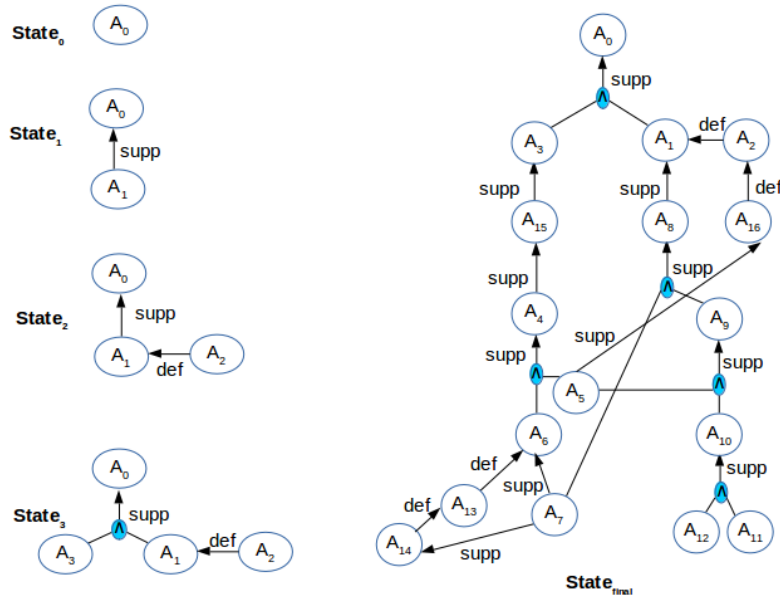


FIGURE 1 – Debate graph during the game.

Back to our example, the grounded extension contains all the arguments of the proof. Now the oracle verifies its sequence according to the support relation. From the debate graph, the sequence of the proof given by the group is :

$$\{(A_7, A_6 \wedge A_5, A_4, A_{15}, A_3) \wedge (A_{12} \wedge A_{11}, A_{10}, A_9 \wedge A_7, A_8, A_1), A_0\}.$$

The argument A_{14} is ignored because it is irrelevant. The sequence of the proof is correct. The group won the game with a penalty degree 3 due to a non valid defeat (A_3 Def A_1) and two irrelevant arguments (A_{14}, A_{16}).

Now let the debate graph in Figure 2 be given by $Group_2$. The grounded extension associated to the debate graph is $\{A_2, A_3, A_6, A_5 \wedge A_6, A_7, A_8, A_9, A_7 \wedge A_9, A_{10}, A_5 \wedge A_{10}, A_{11} \wedge A_{12}, A_{14}, A_{15}\}$.

The Oracle evaluates the grounded extension. We observe that not all arguments of the proof belong to the grounded extension. Therefore the Oracle informs $Group_2$ that it failed the game. Moreover the Oracle gives a feedback to $Group_2$ by pointing out mistakes/comments. In particular A_2 should not appear in the grounded extension because the defeat from A_2 to A_1 is not true and A_2 is not defeated by an acceptable argument. In addition, we have one irrelevant argument (A_{14}) and three relevant arguments are missed to prove the question (A_0, A_1, A_4).

5 Conclusion & Ongoing work

Argumentation debates are a popular method for learning. We use this framework in the context of mathema-

- [15] Park, C., Ch. Kier et K. Jugdev: *Debate as a Teaching Strategy in Online Education : A Case Study*. Canadian J. of Learning and Technology, 37(3), 2011, ISSN 1499-6685.
- [16] Rahwan, I. et K. Larson: *Game Theory*. Dans *Argumentation in Artificial Intelligence*, pages 321–339. 2009.
- [17] Thimm, M. et A.J. García: *Classification and strategic issues of argumentation games on structured argumentation frameworks*. Dans *AAMAS'10*, pages 1247–1254, 2010.
- [18] Weigand, E.: *Argumentation : The Mixed Game*. *Argumentation*, 20(1) :59–87, 2006, ISSN 0920-427X, 1572-8374.
- [19] Yuan, T., V. Svansson, D. Moore et A. Grierson: *A Computer Game for Abstract Argumentation*. 2007.
- [20] Zemplén, G.: *History of Science and Argumentation in Science Education*, pages 129–140. SensePublishers, Rotterdam, 2010, ISBN 978-94-6091-349-5.

Séquences temporelles d'informations qualitatives : raisonner sur la topologie de régions mobiles de taille constante *

Quentin Cohen-Solal Maroua Bouzid Alexandre Niveau

Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France
 {quentin.cohen-solal, maroua.bouzid-mouaddib, alexandre.niveau}@unicaen.fr

Résumé

Nous présentons une approche générale pour le raisonnement sur des séquences temporelles d'informations qualitatives. Cette approche, qui s'appuie sur le cadre récemment introduit des *multialgèbres*, est de façon générale plus efficace computationnellement que les techniques de la littérature. Nous appliquons notre approche au cas spécifique des séquences d'informations topologiques sur des régions de taille constante; nous montrons que le formalisme obtenu dispose d'une procédure complète pour la décision de la cohérence, et nous identifions ses trois sous-classes maximales contenant toutes les relations basiques.

Abstract

Relying on the recently introduced *multi-algebras*, we present a general approach for reasoning about temporal sequences of qualitative information that is generally computationally more efficient than existing techniques. Applying our approach to the specific case of sequences of topological information about constant-size regions, we show that the resulting formalism has a complete procedure for deciding consistency, and we identify its three maximal tractable subclasses containing all basic relations.

1 Introduction

La prise en compte des informations temporelles *et* spatiales est essentielle à la résolution de tâches complexes dans un environnement physique et dynamique. Un certain nombre de *formalismes qualitatifs* de différentes expressivités ont été proposés pour décrire le monde et raisonner sur celui-ci [voir par ex. Chen *et al.*, 2015]. Certains, en particulier, permettent de raisonner à la fois avec des informations de nature temporelle et de nature spatiale. Par exemple, les *séquences temporelles d'informations qualitatives*, sur lesquelles porte cet article, sont des séquences

de descriptions, chacune représentant l'état du monde à un moment spécifique. Parmi les tâches de raisonnement associées à de telles séquences, on peut citer leur ordonnancement [Sioutis *et al.*, 2015] ainsi que leur construction lors d'une interpolation visant à déterminer les états intermédiaires entre deux états du monde – utile par exemple dans le cadre de la planification spatiale [Ragni et Wölfl, 2005; Westphal *et al.*, 2011; Wolter et Kreutzmann, 2015]. La tâche fondamentale de raisonnement est cependant la *décision de la cohérence* de ces séquences temporelles. Ce problème de décision est complexe, car en général la cohérence temporelle et la cohérence spatiale ne peuvent être décidées indépendamment. De plus, la cohérence d'une description dépend des propriétés spatio-temporelles des *entités* décrites. Par exemple, garantir la continuité du mouvement est essentiel dans le cadre du raisonnement à propos d'environnements physiques [Galton, 2000; Hazarika et Cohn, 2001; Muller, 2002]. Il peut également être nécessaire d'assurer que certains attributs des entités – tels que la taille, la position, l'orientation ou la forme – soient préservés au cours du temps, et d'imposer des restrictions sur le mouvement *entre* les instants consécutifs de la séquence. Ces contraintes créent des *interdépendances* qui affectent la cohérence des descriptions.

Exemple 1. Considérons la séquence temporelle suivante : une boîte bleue est dans une boîte rouge durant une première période; durant une seconde période, la boîte rouge est dans une boîte verte, et soit la boîte bleue est dans la boîte rouge, soit la rouge est dans la bleue; durant une troisième période, la boîte verte est dans la boîte bleue. Cette description est cohérente pour des boîtes déformables mais est incohérente pour des boîtes rigides. En effet, avec des boîtes rigides, la description de la première période permet de déduire que la boîte bleue est dans la boîte verte durant la seconde période, ce qui est contradictoire avec la description de la troisième période.

*La version originale (en anglais) de cet article a été publiée dans les actes de la conférence IJCAI 2017 [Cohen-Solal *et al.*, 2017c].

De nombreuses approches représentent les interdépendances en utilisant des *graphes de voisinage (conceptuel)* [Freksa, 1992; Egenhofer, 2010]. Notons que l'incohérence de la description de l'exemple 1 ne dépend que du fait que la *taille* des objets soit constante : le fait que les boîtes soient indéformables n'est pas utilisé directement. La préservation de la taille est ainsi une interdépendance temporelle importante, requise en particulier pour raisonner à propos d'objets rigides, mais également nécessaire lorsque l'on considère des régions déformables ayant une taille constante, comme des corps solides articulés, des corps déformables (pour certaines déformations) ou des liquides. Malheureusement, peu de formalismes portent sur des régions de taille constante et, comme nous allons le voir, ils ont tous des désavantages significatifs.

Dans cet article, nous proposons un cadre général pour raisonner à propos des séquences temporelles d'informations qualitatives, basé sur les *multialgèbres* récemment introduites [Cohen-Solal *et al.*, 2017a,b]. Notre approche permet de représenter de nombreux types d'interdépendances, comme celles des graphes de voisinage, et elle a de meilleures propriétés computationnelles que les approches connues. En outre, nous instancions notre cadre dans le cas spécifique des *séquences d'informations topologiques sur des régions de taille constante évoluant continûment* (telles que les corps rigides ou articulés). D'une part, nous montrons que le formalisme résultant est *complet* pour la décision de la cohérence; d'autre part, en appliquant les théorèmes du cadre des multialgèbres, nous identifions les trois *sous-classes* de ce formalisme qui sont maximales pour la traitabilité et contiennent toutes les relations *basiques*.

L'article est organisé comme suit : la section suivante présente les formalismes qualitatifs, le raisonnement spatio-temporel et les travaux connexes. Dans la section 3, nous présentons le cadre formel récemment introduit des multialgèbres, montrons comment il peut être utilisé pour raisonner sur les séquences temporelles, et décrivons ses avantages computationnels. Pour finir, la section 4 est dédiée aux propriétés fondamentales du cas spécifique des informations topologiques sur des régions de taille constante.

2 Contexte et travaux connexes

2.1 Formalismes qualitatifs

Nous rappelons les concepts principaux relatifs au raisonnement sur le temps et sur l'espace [Ligozat, 2013]. Dans le contexte du raisonnement qualitatif, les informations à propos des entités temporelles et spatiales du monde sont représentées par des *relations binaires* provenant d'un ensemble fini \mathcal{B} , constituant le vocabulaire de base des descriptions. Par exemple, la taille relative entre deux entités x et y est entièrement caractérisée par l'un des éléments de l'ensemble $\mathcal{B}_{PA} = \{<, =, >\}$. Les relations de \mathcal{B} sont qua-

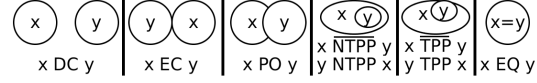


FIGURE 1 – Les 8 relations de RCC8 dans le plan.

lifiées de *basiques*. Les relations basiques sont disjointes (deux entités satisfont au plus une relation basique) et généralement exhaustives (deux entités satisfont au moins une relation basique). Une *relation (composite)* est un ensemble de relations basiques, c'est-à-dire un sous-ensemble de \mathcal{B} , ce qui permet de représenter une incertitude sur la relation basique satisfaite. Par exemple, « $x \{<, =\} y$ » signifie que soit $x < y$ soit $x = y$, alors que « $x \mathcal{B}_{PA} y$ » ne donne aucune information. L'ensemble $\mathcal{A} = 2^{\mathcal{B}}$ de toutes les relations construites à partir de \mathcal{B} forme une *algèbre non associative* [Ligozat, 2013, ch. 11]. L'*algèbre des points* PA [Vilain *et al.*, 1989], dont l'ensemble des relations basiques est \mathcal{B}_{PA} , est un des exemples les plus simples. Une autre algèbre bien connue est RCC8, qui décrit les *relations topologiques* [Cohn *et al.*, 1997] entre *régions*, c'est-à-dire, dans cet article, des sous-ensembles non vides, clos et réguliers de \mathbb{R}^d . Les relations basiques de RCC8 sont illustrées dans la figure 1.

Les descriptions du monde sont souvent données sous forme de *réseaux de contraintes qualitatives*. Un réseau N sur une algèbre \mathcal{A} est un ensemble de relations de \mathcal{A} (des *contraintes*), une pour chaque paire de *variables d'entité*. L'ensemble des variables est noté E . De plus, nous notons N^{xy} la relation entre les variables x et y (implicite, $N^{yx} = \overline{N^{xy}}$). Par exemple, le réseau tel que $N^{xy} = \{\text{PO}\}$, $N^{yz} = \{\text{TPP}\}$ et $N^{xz} = \{\text{DC}\}$ décrit la topologie de trois entités inconnues x, y, z . Notons que toutes les relations sont basiques dans ce réseau : il est qualifié de *scénario*. Une question fondamentale à propos d'une telle description est sa cohérence : peut-on trouver des entités qui satisfont chacune de ses contraintes? Les algèbres sont en fait dotées d'opérateurs sur leur relations : l'*intersection* \cap , l'*inversion* $\bar{\cdot}$, et la *composition (faible)* \diamond , qui permettent une forme de raisonnement déductif. Leur application peut retirer des relations basiques d'une relation composite et ainsi diminuer l'incertitude sur la relation basique satisfaite. Nous avons les implications suivantes :

$$\begin{aligned} x r y &\iff y \bar{r} x \\ x r_1 y \wedge x r_2 y &\iff x (r_1 \cap r_2) y \\ x r_1 y \wedge y r_2 z &\implies x (r_1 \diamond r_2) z \end{aligned}$$

Par exemple, des deux premières contraintes du scénario N décrit précédemment, nous pouvons inférer que la relation basique entre x et z doit être dans $\{\text{PO}, \text{NTPP}, \text{TPP}\}$; puisque cet ensemble est disjoint de la relation N^{xz} , nous en concluons que ce scénario est incohérent.

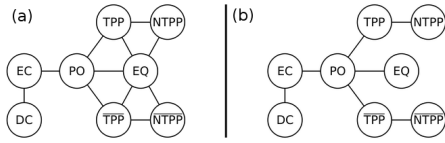


FIGURE 2 – Graphes de voisinage de RCC8 : (a) si la taille des régions peut changer; (b) pour des régions de taille constante.

2.2 Graphes de voisinage

Les graphes de voisinage ont été proposés pour raisonner à propos du changement [Freksa, 1992]. Ils encodent la façon dont les relations entre entités peuvent changer au cours du temps : deux relations basiques sont connectées dans un graphe de voisinage si en déformant continûment deux entités satisfaisant la première relation, on peut faire en sorte qu’elles satisfassent la seconde relation sans qu’aucune autre relation ne soit satisfaite durant la déformation. Les deux graphes de la figure 2 [Gerevini et Nebel, 2002] montrent le voisinage des relations topologiques, pour des régions de taille variable et de taille constante. Supposons que TPP soit satisfaite entre deux régions ; la figure 2.a montre que si on déforme ces régions continûment jusqu’à ce que TPP ne soit plus satisfaite, la nouvelle relation ne peut être que NTPP, PO ou EQ en général. Cependant, si la taille des régions est constante, alors la figure 2.b indique que la relation ne peut être que NTPP ou PO.

Une des applications des graphes de voisinage est de garantir qu’une *séquence temporelle* de scénarios N_1, \dots, N_m – chaque N_i décrivant la situation à un instant fixé t_i – est compatible avec la propriété de *continuité sans relation intermédiaire* (ou *changement qualitatif continu*) [Westphal *et al.*, 2013], qui est notamment utile dans le cadre de la planification qualitative [Ragni et Wöflf, 2005; Westphal *et al.*, 2011]. Une séquence temporelle de scénarios est considérée cohérente s’il existe des entités évoluant continûment qui satisfont chaque réseau N_i à l’instant t_i ; la continuité sans relation intermédiaire ajoute la condition qu’à n’importe quel moment *entre* deux instants fixés et consécutifs t_i et t_{i+1} , chaque paire d’entités $\{x, y\}$ doit satisfaire soit N_i^{xy} soit N_{i+1}^{xy} . Par exemple, la séquence « x TPP y , puis x DC y » n’a aucune solution sans relation intermédiaire; en fait, une séquence temporelle (r_1, \dots, r_m) de relations basiques de RCC8 satisfait la continuité sans relation intermédiaire si et seulement si elle *respecte* le graphe de voisinage de RCC8, c’est-à-dire que pour $1 \leq i < m$, soit $r_i = r_{i+1}$, soit (r_i, r_{i+1}) est un arc dans le graphe.

2.3 Travaux connexes

Il existe de nombreuses études sur le changement qualitatif, en particulier basé sur les graphes de voisinage [e.g., Wu *et al.*, 2014; Santos et Moreira, 2009; Bogaert *et al.*, 2006; Van de Weghe et De Maeyer, 2005; Cohn et Haza-

rika, 2001; Zimmermann et Freksa, 1993], et sur ses applications [e.g., Cui *et al.*, 1992; Dylla et Wallgrün, 2007]. En particulier, les graphes de voisinage ont été utilisés dans le cadre de la simulation qualitative [Cohn *et al.*, 1998] et des formalismes de planification [Dylla et Moratz, 2004].

D’un autre côté, le raisonnement à propos des séquences temporelles n’est pas largement étudié : sur le sujet nous n’avons connaissance que de l’approche générale de Westphal *et al.* [2013], qui duplique toutes les variables pour chaque instant afin d’ajouter des *contraintes de transition* (non binaires) imposant les interdépendances.

Cependant, des formalismes spatio-temporels de plus haute expressivité ont été étudiés, dans lesquels l’ordre des entités temporelles est incertain. Les descriptions du formalisme de Gerevini et Nebel [2002] consistent en des contraintes entre des intervalles temporels ainsi que des contraintes de RCC8 entre des régions, satisfaites durant ces intervalles : par ex., un intervalle I est avant ou après un autre intervalle, et durant I la relation $\{DC, PO\}$ est satisfaite entre deux régions. Ragni et Wöflf [2006] ont introduit un formalisme similaire pour les relations directionnelles entre des points de l’espace. L’approche de Sioutis *et al.* [2014] a une expressivité différente – en particulier les entités temporelles sont des points et non des intervalles.

Nous ne connaissons pas d’autre formalisme spatio-temporel qualitatif et algébrique, mais le raisonnement spatio-temporel est également étudié dans le cadre des logiques spatio-temporelles [Bennett *et al.*, 2002], qui sont plus expressives, mais moins performantes : alors que le problème de décision de la cohérence est typiquement NP-complet pour les approches algébriques, il est généralement PSPACE-difficile pour les logiques.

Seules deux approches, à notre connaissance, traitent de la préservation de la taille dans le contexte de relations topologiques. La première est la logique de Galton [1993], basée sur la *logique temporelle d’Allen*. Elle permet de raisonner sur les relations topologiques entre corps rigides, mais nécessite de connaître la relation de taille précise pour chaque paire de régions. La seconde est le formalisme de Gerevini et Nebel [2002] mentionné précédemment. Il impose la préservation de la taille grâce à la *combinaison taille-topologie*, en dupliquant toutes les variables pour chaque intervalle temporel. Il garantit également une évolution sans relation intermédiaire grâce au graphe de voisinage de la figure 2.b. Cependant, il n’est pas clair que ce formalisme permet de garantir la *continuité* du mouvement; de plus, aucun algorithme n’a été proposé pour raisonner à propos des descriptions non basiques.

3 Multialgèbres des séquences temporelles

Les multialgèbres ont récemment été introduites [Cohen-Solal *et al.*, 2017a,b] dans le but d’englober au sein d’un cadre algébrique commun plusieurs formes de combi-

naisons, notamment les intégrations lâches [Wölfl et Westphal, 2009] et les représentations multiéchelles [Cohen-Solal *et al.*, 2015], mais également les séquences temporelles. Cette dernière application a été illustrée avec l'exemple simple des séquences temporelles d'informations directionnelles (c'est-à-dire des relations de PA) qui impose la continuité sans relation intermédiaire. Nous montrons ici comment cette approche se généralise à n'importe quel formalisme qualitatif, et comment elle peut être adaptée au contexte de la préservation de la taille, en l'illustrant dans le cadre des relations de RCC8.

3.1 Séquences temporelles et projections

La relation entre deux entités dans une séquence temporelle d'informations qualitatives peut être vue comme un m -uplet de relations, la i -ème relation étant celle satisfaite au i -ème instant de la séquence. Ainsi, la séquence dans laquelle une région x est à l'intérieur d'une autre région y , qui la chevauche ensuite, pour finalement en être disjointe, correspond au triplet $(\{TPP, NTPP\}, \{PO\}, \{EC, DC\})$. Les multialgèbres permettent de représenter les relations de cette forme : une multialgèbre étant en effet un produit cartésien $\mathcal{A}_1 \times \dots \times \mathcal{A}_m$, une *relation d'une multialgèbre* est un m -uplet $R = (R_1, \dots, R_m)$ de relations classiques $R_i \in \mathcal{A}_i$. La multialgèbre des séquences d'informations topologiques avec m instants est ainsi simplement $RCC8^m$. Les algèbres dans le produit peuvent cependant aussi être distinctes, ce qui permet de combiner des informations de nature hétérogène au sein d'une même relation : les multialgèbres généralisent en fait les *intégrations lâches*. Par exemple, la combinaison bien connue des tailles qualitatives et de la topologie [Gerevini et Renz, 2002] correspond à la multialgèbre $RCC8 \times PA$. Sa relation $(\{DC\}, \{<, =\})$ décrit le fait que deux régions sont déconnectées et que la première est de taille inférieure ou égale.

Sur de nombreux aspects, une multialgèbre se comporte comme un ensemble d'algèbres indépendantes : on peut généraliser plusieurs notions classiques aux multialgèbres de manière transparente en les considérant *composante par composante*. Ainsi, pour deux relations R et R' , nous écrivons $R \subseteq R'$ si $R_i \subseteq R'_i$ est satisfaite pour chaque i , et R est dit *basique* si chaque R_i est basique. De plus, les multialgèbres sont dotées d'opérateurs de composition, d'inversion et d'intersection qui s'appliquent composante par composante (par ex., $R \diamond R'$ est défini par $\forall i: (R \diamond R')_i = R_i \diamond R'_i$). Cela fait sens pour les séquences temporelles puisque cela revient à appliquer les opérateurs classiques indépendamment sur les descriptions de chaque instant.

Néanmoins, les éléments du m -uplet ne sont pas toujours indépendants. Par exemple, dans le but de garantir la continuité sans relation intermédiaire sur les séquences temporelles, chaque paire de relations successives doit satisfaire le graphe de voisinage. Avec les multialgèbres,

b	TPP	NTPP	PO	EC	DC	EQ
$\overset{PA}{RCC8} b$	<	<	\mathcal{B}_{PA}	\mathcal{B}_{PA}	\mathcal{B}_{PA}	=
b	<			=		
$\overset{RCC8}{PA} b$	TPP	NTPP	PO	EC	DC	EQ

TABLE 1 – Projections des relations basiques entre RCC8 et PA.

les interdépendances de ce type entre composantes individuelles sont imposées par des opérateurs additionnels, appelés *projections*. Il y a une projection pour chaque couple d'algèbres dans le produit cartésien, qui représente les interdépendances entre les relations de ces deux algèbres. Formellement, une projection $\overset{j}{r}_i: \mathcal{A}_i \mapsto \mathcal{A}_j$ est une fonction associant à chaque relation r de l'algèbre \mathcal{A}_i la relation r' de l'algèbre \mathcal{A}_j correspondant à la sémantique suivante : si r est satisfaite, alors r' l'est également. Par exemple, la table 1 indique les projections des relations basiques de la combinaison taille-topologie. La projection de $\{TPP\}$ vers PA est $\overset{PA}{RCC8}\{TPP\} = \{<\}$, puisqu'une région strictement incluse dans une autre ne peut avoir qu'une taille plus petite. Les projections des relations non basiques peuvent être déduites de la table en utilisant les formules $\overset{j}{r}_i = \bigcup_{b \in r} \overset{j}{r}_i\{b\}$ et $\overset{j}{\bar{r}}_i = \bigcup_{b \in \bar{r}} \overset{j}{\bar{r}}_i\{b\}$.

Les projections peuvent être utilisées pour garantir qu'une séquence temporelle respecte un graphe de voisinage G quelconque. Puisque le respect de G impose des restrictions sur les relations des instants successifs, mais n'impose rien dans les autres cas, la projection de $b \in \mathcal{B}$ d'un instant i vers un autre instant j est définie par :

$$\overset{j}{r}_i\{b\} = \begin{cases} \{b' : \{b, b'\} \in G\} \cup \{b\} & \text{if } |i - j| = 1, \\ \mathcal{B} & \text{if } |i - j| > 1. \end{cases}$$

Ainsi, étant donné une algèbre \mathcal{A} et un graphe de voisinage G , on peut construire une multialgèbre représentant les séquences temporelles d'informations provenant de \mathcal{A} dans le contexte de la continuité sans relation intermédiaire. Dans le cas des informations topologiques, on obtient la définition suivante.

Définition 2. Soit $m \in \mathbb{N}^*$. La multialgèbre TT_{sri} de la *topologie temporalisée sans relation intermédiaire* est le produit cartésien $RCC8^m$ doté des projections imposant le graphe de voisinage de la figure 2.a.

Cependant, si on veut raisonner à propos de régions de taille constante, les graphes de voisinage ne suffisent pas. Considérons la relation $(TPP, PO, \overline{TPP})$: elle respecte le graphe de la figure 2.b, bien qu'aucune paire de régions de taille constante ne la satisfasse. Ce problème peut être résolu en raisonnant *explicitement* sur les tailles des régions, en plus du raisonnement spatial. Dans le cadre des multialgèbres, cela peut être effectué en ajoutant l'algèbre des points au produit cartésien. Pour le raisonnement topologique, on obtient $RCC8^m \times PA$: chaque relation est un

$(m+1)$ -uplet contenant m relations topologiques (une pour chaque instant) ainsi qu'une relation en plus indiquant la taille relative des deux régions. Les projections vers et depuis l'algèbre ajoutée sont exactement celles de la combinaison taille-topologie – qui est donc un cas particulier ($m = 1$) de cette multialgèbre.

Definition 3. Soit $m \in \mathbb{N}^*$. La multialgèbre TT_{tc} de la *topologie temporalisée des régions de taille constante* est le produit cartésien $\text{RCC8}^m \times \text{PA}$ doté des projections suivantes : pour tout b , $\overset{j}{r}_i b = \mathcal{B}_{\text{RCC8}}$ (aucune interdépendance) pour $1 \leq i, j \leq m$, et $\overset{m+1}{r}_i b$ et $\overset{i}{r}_{m+1} b$ sont définies par la table 1 pour $1 \leq i \leq m$.

Definition 4. Soit $m \in \mathbb{N}^*$. La multialgèbre $\text{TT}_{\text{sri,tc}}$ de la *topologie temporalisée des régions de taille constante sans relation intermédiaire* est le produit cartésien $\text{RCC8}^m \times \text{PA}$ doté des projections $\overset{j}{r}_i$ imposant le graphe de voisinage de la figure 2.b pour $1 \leq i, j \leq m$, et des projections $\overset{m+1}{r}_i$ et $\overset{i}{r}_{m+1}$ de TT_{tc} pour $1 \leq i \leq m$.

Ces deux multialgèbres imposent uniquement la préservation de la taille *relative*, mais pour certains formalismes cela suffit à garantir l'existence d'une solution avec des régions de taille constante. Cette technique d'ajout d'une algèbre dans le produit cartésien est *a priori* applicable à la préservation d'autres attributs, tels que l'orientation, la position ou la vitesse.

3.2 Sémantique des séquences temporelles

Nous avons vu que les multialgèbres sont dotées des opérateurs de composition, d'inversion, d'intersection et de projection qui permettent de raisonner à propos de leurs relations. Pour prouver que le raisonnement d'une multialgèbre est correct et/ou complet, nous avons besoin de définir la *sémantique* de ses relations, c'est-à-dire la façon dont elles sont interprétées. Cela nécessite d'explicitier un *domaine* – l'ensemble des entités considérées – et une *fonction d'interprétation* associant à chaque relation tous les couples d'entités qui sont considérés comme des *réalisations* correctes de cette relation.

Dans le contexte des multialgèbres définies précédemment pour traiter les séquences temporelles, nous raisonnons à propos de régions qui se déplacent continûment ; une entité est alors ce que nous appelons une *évolution continue d'une région*, c'est-à-dire une fonction continue indiquant à chaque instant l'état courant de la région. Nous considérons deux domaines : l'ensemble \mathfrak{R}^I de toutes les évolutions continues durant un intervalle $I \subseteq \mathbb{R}$, et son sous-ensemble strict $\mathfrak{R}_{\text{tc}}^I$ contenant toute les évolutions continues d'une région de taille constante. Plus formellement, \mathfrak{R}^I (resp. $\mathfrak{R}_{\text{tc}}^I$) est l'ensemble des fonctions continues f de I vers l'ensemble des régions de \mathbb{R}^d (resp. vers l'ensemble des régions mesurables de \mathbb{R}^d satisfaisant $\exists s \in \mathbb{R}_+^* : \forall t \in I : \mu(f(t)) = s$, avec μ une mesure de taille

de région). Nous pouvons maintenant définir la sémantique de TT_{tc} (nous ne définissons pas celles de TT_{sri} et $\text{TT}_{\text{sri,tc}}$ pour des raisons d'espace restreint) :

Definition 5. Soit $m \in \mathbb{N}^*$, et soit $(\tau_1, \dots, \tau_m) \in \mathbb{R}^m$ une séquence ordonnée de m instants distincts. La fonction d'interprétation de TT_{tc} est la fonction $\varphi_{\text{TT}_{\text{tc}}} : \text{RCC8}^m \times \text{PA} \rightarrow (\mathfrak{R}_{\text{tc}}^{[\tau_1, \tau_m]})^2$ associant à chaque relation R l'ensemble de tous les couples d'évolutions de régions (f, f') satisfaisant R , c'est-à-dire vérifiant les contraintes de RCC8 à l'instant correspondant ($\forall i \in \{1, \dots, m\} : f(\tau_i) R_i f'(\tau_i)$) ainsi que les contraintes de taille $(\mu(f(\tau_1)) R_{m+1} \mu(f'(\tau_1)))$.

Cette sémantique satisfait de très bonnes propriétés : le triplet $(\text{TT}_{\text{tc}}, \mathfrak{R}_{\text{tc}}^{[\tau_1, \tau_m]}, \varphi_{\text{TT}_{\text{tc}}})$ constitue un *formalisme lâchement combiné* [Cohen-Solal et al., 2017a,b]. Cela garantit qu'une relation est cohérente si et seulement si elle contient une relation basique cohérente, et que le raisonnement est *sain* : utiliser les opérateurs pour éliminer des relations basiques d'une relation ne change pas son interprétation.

Notons que, tandis que la partie syntaxique (la multialgèbre) impose seulement la préservation des tailles relatives, la sémantique (la fonction d'interprétation) requiert bien la préservation de la taille des entités.

3.3 Décision de la cohérence des séquences

Nous avons utilisé les relations des multialgèbres pour représenter des séquences temporelles entre deux entités. À l'instar du cadre classique, des descriptions plus complexes peuvent être établies à l'aide de *réseaux de contraintes qualitatives*, dont les contraintes sont des relations de multialgèbres. L'exemple suivant est un réseau N (et même un scénario) sur $\text{RCC8}^3 \times \text{PA} : N^{xy} = (\text{TPP}, \text{PO}, \text{PO}, <)$, $N^{yz} = (\text{PO}, \text{TPP}, \text{PO}, <)$, et $N^{xz} = (\text{PO}, \text{PO}, \overline{\text{TPP}}, >)$.

Grâce à la section précédente, nous pouvons formellement définir une *solution* d'un réseau comme un assignement à chaque variable d'entité de \mathbf{E} d'une entité du domaine U considéré tel que chaque contrainte soit satisfaite, c'est-à-dire un ensemble $\{u_x\}_{x \in \mathbf{E}} \subseteq U$ satisfaisant $\forall x, y \in \mathbf{E} : (u_x, u_y) \in \varphi(N^{xy})$. Notons qu'avec les formalismes lâchement combinés, un réseau est cohérent si et seulement s'il contient un scénario cohérent. Pour décider la cohérence d'un réseau, une méthode intuitive consiste à utiliser les opérateurs de la multialgèbre pour retirer des relations basiques irréalisables des contraintes, et ce itérativement. Lorsque les opérateurs ne retirent plus de relations, le réseau résultant est dit *algébriquement clos*.

Definition 6. Un réseau N est dit *algébriquement clos* lorsqu'il est à la fois clos par composition ($\forall x, y, z \in \mathbf{E} : N^{xz} \subseteq N^{xy} \diamond N^{yz}$) et clos par projection ($\forall x, y \in \mathbf{E} : N_j^{xy} \subseteq \overset{j}{r}_i N_i^{xy}$ pour tous i, j distincts).

Calculer la clôture algébrique d'un réseau peut donner l'ensemble vide comme relation : c'est le cas pour le réseau N dans l'exemple précédent. L'ensemble vide étant

une contrainte insatisfaisable, et le raisonnement étant sain, cela signifie que le réseau initial est incohérent. Remarquons au passage que dans le cas du réseau N , sans expliciter les contraintes de taille, on n'aurait pas pu détecter son incohérence.

Un réseau algébriquement clos ne contenant aucune relation vide est qualifié d'*algébriquement cohérent*, au sens où le raisonnement algébrique ne peut prouver son incohérence. En remplaçant N^{xz} par (PO, PO, TPP, <) dans l'exemple précédent, on obtient un réseau algébriquement cohérent, qui est en fait cohérent. Cependant, en général, on ne peut rien conclure à propos de la cohérence d'un réseau algébriquement cohérent – même si c'est un scénario. C'est une propriété fondamentale d'un formalisme que tous ses scénarios algébriquement clos soient cohérents. En effet, cette propriété (qui dépend de sa fonction d'interprétation) fournit une procédure *complète* pour décider la cohérence – une simple recherche d'un scénario clos, typiquement en utilisant un algorithme de *branch and bound* (séparation et évaluation) qui applique la clôture algébrique. Un formalisme ou un fragment satisfait la propriété plus forte de *traitabilité algébrique* lorsque clôturé algébriquement un réseau, ce qui peut être effectué en temps polynomial, suffit à décider sa cohérence.

3.4 Avantages computationnels des multialgèbres

Les multialgèbres sont bien adaptées pour raisonner sur les séquences temporelles. En effet, contrairement aux autres approches comme celle de Westphal *et al.* [2013] ou de Gerevini et Nebel [2002], il n'y a aucun besoin de dupliquer les variables d'entités à chaque instant de la séquence, afin de pouvoir ajouter des contraintes d'interdépendances entre une entité et elle-même à un instant différent (par exemple pour préserver la taille). En particulier, la clôture algébrique peut être imposée en temps $\Theta(m \cdot n^3 + m^2 \cdot n^2)$ avec les multialgèbres, alors qu'elle nécessite un temps $\Theta(m^3 \cdot n^3)$ si les variables sont dupliquées. Ne pas dupliquer les variables a un second avantage : pour chercher un scénario clos d'un réseau (afin de décider sa cohérence dans les cas intraitables), l'algorithme de séparation et évaluation a un facteur de branchement significativement plus petit. Cependant, contrairement à celle de Westphal *et al.* [2013], l'approche des multialgèbres ne peut empêcher des changements qualitatifs sur des couples distincts d'entités d'avoir lieu *en parallèle* entre les instants.

4 Topologie temporalisée avec préservation de la taille

Nous nous penchons à présent sur TT_{tc} , c'est-à-dire sur le cas de la *préservation de la taille* pour les séquences temporelles d'informations *topologiques* ; les solutions doivent être continues, mais il *peut* y avoir des relations inter-

médiaires. Nous montrons que le raisonnement avec TT_{tc} est complet, et nous identifions ses sous-classes traitables maximales contenant toutes les relations basiques.

4.1 Complétude du raisonnement pour TT_{tc}

Cette section montre un résultat fondamental : la clôture algébrique suffit à décider la cohérence de tous les scénarios sur TT_{tc} . Remarquons que la preuve est très spécifique à RCC8 – il n'y a aucune garantie que pour une algèbre \mathcal{A} quelconque, on ait un résultat similaire pour les scénarios de la multialgèbre temporalisée $\mathcal{A}^m \times PA$. En particulier, une condition nécessaire est la cohérence de tous les scénarios algébriquement clos pour la combinaison $\mathcal{A} \times PA$ (c'est-à-dire le cas $m = 1$) ; ce n'est notamment pas le cas si \mathcal{A} est l'*algèbre des intervalles* [Ligozat, 2013, ch. 8].

Proposition 7. *Les scénarios algébriquement clos sur TT_{tc} sont cohérents (pour des régions de \mathbb{R}^d , $d \geq 2$).*

Démonstration. Soit N un scénario algébriquement clos sur TT_{tc} , avec n variables d'entité x_1, \dots, x_n et m instants τ_1, \dots, τ_m . Nous montrons qu'il est possible d'assigner à chaque x_i un élément f_i de $\mathfrak{R}_{tc}^{[\tau_1, \tau_m]}$ (une évolution continue d'une région d'une taille constante) tel que chaque contrainte soit satisfaite. Puisque nous avons seulement besoin de prouver l'existence d'une solution, dans le but de rendre cette preuve aussi simple et claire que possible, nous choisissons une forme spécifique pour les régions et leur mouvement. En particulier, les relations intermédiaires étant autorisées, nous pouvons traiter chaque instant indépendamment. Plus précisément, il nous suffit de montrer que pour tout scénario sur RCC8 algébriquement clos, en partant d'une configuration fixée avec des régions disjointes deux à deux, il est possible de déplacer continûment ces régions une par une, sans modifier leur taille, pour obtenir une configuration satisfaisant le scénario. Cette évolution continue pouvant être inversée, la configuration initiale peut être utilisée comme une configuration *intermédiaire* entre toute paire d'instant successifs, ce qui assure que l'évolution est continue tout au long de la séquence.

Taille des régions. Puisque N est clos par composition et que les scénarios clos sur PA sont cohérents, nous pouvons associer à chaque entité un *rang* entier, noté $\text{rang}(x_i) \in \{1, \dots, n\}$, tel que les rangs satisfont les contraintes de taille. Sans perte de généralité, nous supposons que les x_i sont ordonnés de sorte que leur rang soit croissant, et que les différents rangs sont des entiers successifs ($\forall i \in \{2, \dots, n\} : \text{rang}(x_{i-1}) \leq \text{rang}(x_i) \leq \text{rang}(x_{i-1}) + 1$). Les entités de rang 1 sont les plus petites régions, les entités de rang 2 sont les secondes plus petites, etc. Évidemment, le rang d'une entité ne dépend pas de l'instant considéré, la taille d'une entité ne devant pas changer entre les instants.

Définition des régions. Considérons un scénario clos sur RCC8, et notons R^{ij} la relation basique de RCC8 qui

doit être satisfaite entre x_i et x_j à cet instant. Nous construisons chaque région comme une union disjointe de *sous-régions*, à savoir des *boules* fermées de \mathbb{R}^d : cela nous permet de faire interagir une région avec chacune des autres régions le plus indépendamment possible. Pour chaque i , les sous-régions de x_i ont toutes le même rayon, égal à $3^{\text{rang}(x_i)}$ pour qu'il soit plus de deux fois plus grand que celui des boules de rang inférieur (la raison sera donnée plus tard). Les boules restent disjointes à tout moment, même lorsqu'elles bougent (ce n'est jamais un problème dans \mathbb{R}^d , $d \geq 2$), afin que la taille de x_i reste constante. Chaque région x_i comporte n sous-régions *primaires*, notées P_i^j ($1 \leq j \leq n$), qui seront utilisées pour imposer la satisfaction des relations EC et PO avec chaque autre région x_j . Puisque x_i peut avoir à inclure *toutes* les régions de plus petit rang (à cause des relations TPP et NTPP), qui, dans le pire des cas, peuvent être toutes déconnectées (DC), nous ajoutons à x_i autant de sous-régions *secondaires* que de sous-régions de rang strictement inférieur. Ainsi, pour chaque k tel que $\text{rang}(x_k) < \text{rang}(x_i)$, pour chaque sous-région P de x_k , nous ajoutons à x_i une sous-région $P_i(P)$ destinée à couvrir P si nécessaire. Remarquons que les entités de même rang ont exactement le même nombre de sous-régions, et donc la même taille (puisque les sous-régions d'une entité restent toujours disjointes).

Configuration initiale. Initialement, toutes les boules sont disjointes et leurs centres sont alignés sur une droite, choisie arbitrairement, appelée *axe* (nous choisissons également arbitrairement des directions *droite* et *gauche* sur cet axe). Hormis durant les déplacements, le centre de chaque boule reste toujours sur l'axe. Nous appelons *zone d'interaction* $Z(P)$ de la boule P , la zone centrée autour de la position *initiale* de P (les zones d'interaction ne bougent jamais). Nous supposons que les zones d'interaction sont disjointes, et que chacune est suffisamment grande pour contenir une boule de rayon 3^{n+1} (plus grande que n'importe quelle sous-région). Initialement, chaque $Z(P)$ contient uniquement la sous-région P . Nous nous assurons qu'il y a en permanence *au plus une* sous-région de chaque entité dans chaque zone d'interaction : c'est une façon simple de garantir que les sous-régions d'une entité restent disjointes (et donc que la taille est préservée). Nous notons $\mathcal{Z}_i = \{Z(P_i) \mid P_i \text{ subregion of } x_i\}$ l'ensemble des zones d'interaction de l'entité x_i .

Si $R^{ij} = \text{EQ}$ pour certains i, j , nous pouvons bouger chaque sous-région de x_j afin qu'elle devienne égale à une sous-région de x_i . Puisque le réseau est clos, cela est toujours possible (x_i et x_j ont le même rang et $\forall k: R^{ik} = R^{jk}$). Nous pouvons ensuite *identifier* x_i et x_j , c'est-à-dire verrouiller leurs sous-régions deux à deux afin qu'elles bougent toujours simultanément et que x_i et x_j restent ainsi égales à tout moment. Ainsi, dans la suite, nous pouvons supposer sans perte de généralité que $i \neq j \iff R^{ij} \neq \text{EQ}$.

Étape 1 : contact et chevauchement. Pour chaque

couple i, j tel que $i < j$, on procède comme suit : (i) si $R^{ij} = \text{PO}$, on bouge P_j^i pour faire coïncider son centre avec celui de P_i^j (voir la figure 3.a,b); (ii) si $R^{ij} = \text{EC}$, on bouge P_j^i pour faire coïncider son point le plus à gauche avec le point le plus à droite de P_i^j (voir la figure 3.c,d); (iii) si $R^{ij} = \text{TPP}$, on bouge P_j^i pour faire coïncider son point le plus à droite avec le point le plus à droite de P_i^j (voir la figure 3.e). Il est clair, à cette étape, qu'il y a au plus deux sous-régions dans chaque zone (et qu'elles appartiennent à des régions distinctes, ainsi la taille de chaque région n'a pas changé), qu'aucune sous-région P_i^j telle que $i < j$ n'a bougé, et que toutes les relations PO et EC sont satisfaites.

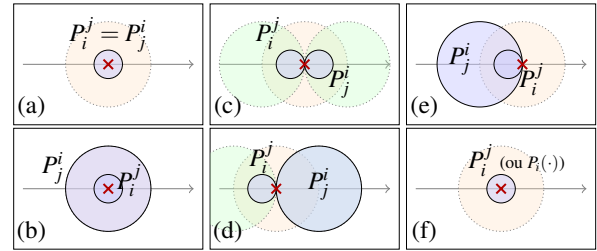


FIGURE 3 – Les boules bleues montrent toutes les configurations possibles à la fin de l'étape 1 dans une zone d'interaction $Z(P_i^j)$ non vide : (a) $R^{ij} = \text{PO}$ et $\text{rang}(x_i) = \text{rang}(x_j)$; (b) $R^{ij} = \text{PO}$ et $\text{rang}(x_i) < \text{rang}(x_j)$; (c) $R^{ij} = \text{EC}$ et $\text{rang}(x_i) = \text{rang}(x_j)$; (d) $R^{ij} = \text{EC}$ et $\text{rang}(x_i) < \text{rang}(x_j)$; (e) $R^{ij} = \text{TPP}$; (f) $R^{ij} \in \{\text{DC}, \text{NTPP}\}$ (c'est également la configuration dans toutes les zones d'interaction de boules secondaires). Les croix rouges indiquent le *point d'interaction*. Les boules pâles en pointillés montrent les positions possibles durant l'étape 2 des boules secondaires les plus petites (en vert, le cas particulier).

Étape 2 : inclusions. Pour que le scénario soit satisfait, il ne reste plus qu'à satisfaire les relations TPP et NTPP entre les régions. Remarquons que nous avons garanti à l'étape 1 une connexion intérieure pour chaque relation TPP, il ne nous reste ainsi qu'à imposer les inclusions. C'est la partie la plus difficile : nous devons couvrir toutes les sous-régions des régions incluses sans que cela n'affecte les autres entités, et nous devons également prendre garde à ne pas annuler les déplacements précédents. Pour ce faire, (i) nous traitons les entités dans l'ordre croissant de leur tailles, ce qui assure que lors du traitement de x_i , plus aucune sous-région de x_k avec $k \leq i$ ne peut encore bouger; et (ii) pour chaque entité, nous n'itérons pas sur ses relations mais sur ses *zones d'interaction*, en gérant toutes les boules devant se trouver là (puisque'on ne reviendra jamais sur une zone déjà traitée). Notons que les sous-régions de x_i qui ont bougé vers une autre zone ont forcément déjà été traitées, les sous-régions pouvant uniquement bouger vers une zone d'une entité de rang strictement inférieur.

Concrètement, la procédure est comme suit. Pour chaque i de 1 à n , on considère chaque zone d'interaction $Z(P_i) \in \mathcal{Z}_i$. Si elle est vide, P_i a bougé vers une autre zone : on ne fait rien. Sinon, il ne peut y avoir que six configurations, montrées dans la figure 3. Pour chaque k de $i+1$ à n , on

regarde s'il faut ajouter une sous-région de x_k dans cette zone. Ce n'est jamais nécessaire s'il y en a déjà une, mais dans le cas contraire il y a deux cas où il faut en ajouter une : (i) $R^{jk} \in \{TPP, NTPP\}$, donc P_i doit être couverte par une sous-région de x_k ; (ii) la zone contient une boule d'une autre entité x_j , qui a bougé ici durant l'étape 1 ou un peu plus tôt dans cette boucle, et $R^{jk} \in \{TPP, NTPP\}$ (ce cas peut concerner plusieurs boules, chacune d'une entité distincte). S'il y a au moins une boule à couvrir *et* aucune sous-région de x_k n'est déjà présente, alors on déplace la sous-région $P_k(P_i)$ de sa position initiale (par construction, elle n'a jamais bougé) vers $Z(P_i)$, afin qu'elle couvre *toutes* les boules qui doivent être couvertes et ne chevauche *aucune* boule qui ne doit pas être chevauchée.

Pour ce faire, hormis dans un cas particulier, il suffit de simplement faire coïncider le centre de $P_k(P_i)$ avec le *point d'interaction* de $Z(P_i)$ (voir la figure 3), qui dépend uniquement de sa configuration à la fin de l'étape 1. Le cas particulier a lieu lorsque la zone a servi à imposer EC (figure 3.c,d) *et* qu'il faut couvrir la boule de gauche (resp. de droite) *sans* chevaucher la boule de droite (resp. de gauche). Dans ce cas, on rend $P_k(P_i)$ tangent au point d'interaction, en positionnant son centre du côté gauche (resp. droit) de l'axe (voir la boule verte en pointillés dans la figure 3.c,d). On peut prouver, avec la table de composition de RCC8, qu'aucune inclusion ni aucun chevauchement résultant de ces manipulations ne contredit les contraintes du scénario (en particulier, si $P_k(P_i)$ est plus grande que toutes les boules présentes, alors elle les couvre toutes). \square

Par conséquent, chercher un scénario algébriquement clos suffit à décider la cohérence de tout réseau sur TT_{ic} : le raisonnement dans le cadre de TT_{ic} est donc complet pour la décision de la cohérence.

4.2 Sous-classes algébriquement traitables de TT_{ic}

La proposition 7 implique que la décision de la cohérence pour TT_{ic} est dans NP (exhiber un scénario clos prouve la cohérence d'un réseau). Cependant, puisque tout réseau sur RCC8 est équivalent à un réseau sur TT_{ic} de taille polynomiale, décider la cohérence dans le cadre de TT_{ic} est en fait NP-complet. En conséquence, nous nous penchons dans cette section sur l'identification de fragments traitables, en particulier de sous-ensembles de relations de TT_{ic} qui sont clos par composition, intersection et inversion – ce que l'on appelle des *sous-classes* [voir Ligozat, 2013]. Nos résultats découlent de l'application des théorèmes de Cohen-Solal *et al.* [2017a,b], qui fournissent des conditions suffisantes à ce qu'une sous-classe d'une multialgèbre soit *algébriquement traitable*. Les contraintes d'espace nous empêchent de détailler ces conditions, mais les preuves qui suivent contiennent tous les arguments requis pour les vérifier en utilisant l'article d'origine comme référence.

Avant d'identifier les sous-classes traitables maximales de TT_{ic} , nous prouvons la traitabilité d'une sous-classe intermédiaire, à savoir $RCC8_{max}^m \times PA_{max}$, avec $RCC8_{max}$ la *sous-algèbre (non convexe) maximale pour la distributivité* de RCC8 [Long et Li, 2015] et $PA_{max} = \{<, =, >, \neq, \mathcal{B}_{PA}\}$. En fait, par la preuve du résultat suivant, cette sous-classe vérifie une propriété intéressante : pour décider la cohérence, il est suffisant d'appliquer une clôture par projection, suivie d'une clôture par composition (cette sous-classe est *simple* [Cohen-Solal *et al.*, 2017a,b]). Ce n'est pas le cas en général ; à cause des interdépendances des relations d'un même instant et d'instant différents, pour clore algébriquement un réseau, il peut être nécessaire d'alterner clôture par projection puis clôture par composition jusqu'à atteindre un point fixe.

Proposition 8. *La sous-classe $RCC8_{max}^m \times PA_{max}$ de TT_{ic} est algébriquement traitable.*

Démonstration. Nous affaiblissons toutes les projections de PA (\mathcal{A}_{m+1}) vers RCC8 (\mathcal{A}_i avec $i \leq m$) : nous considérons que $\uparrow_{PA}^{RCC8} b = \mathcal{B}_{RCC8}$ pour tout $b \in \mathcal{B}_{PA}$. Nous allons prouver que la sous-classe avec des projections affaiblies est algébriquement traitable, ce qui implique directement que la sous-classe avec les projections d'origine l'est également.

Nous appliquons le théorème de découpage de Cohen-Solal *et al.* [2017a, Th. 24] sur cette sous-classe avec les projections affaiblies, en utilisant la fonction identité pour chaque affinement. **(I)** Les scénarios algébriquement clos pour les projections affaiblies sont cohérents, par la proposition 7, puisqu'ils sont clos pour les projections d'origine. **(II)** C1 : les réseaux sur $RCC8_{max}$ et PA_{max} qui sont clos par composition sont *minimaux* [Long et Li, 2015]. **(III)** C2 : la sous-classe est *simple* car (i) $RCC8_{max} \times PA_{max}$ ($m = 1$) avec les projections affaiblies est simple [Cohen-Solal *et al.*, 2017a,b], (ii) entre deux relations R_i et R_j sur RCC8, il n'y a aucune interdépendance directe ($\uparrow_{RCC8}^{RCC8} b = \mathcal{B}_{RCC8}$ pour tout b), et (iii) être clos par projection est une propriété locale. **(IV)** C3 : les relations \uparrow -cohérentes sont cohérentes. En effet, c'est vrai pour les relations basiques (proposition 7) ; pour les relations non basiques, on peut choisir une relation basique pour PA puis prendre une relation compatible de RCC8 pour chaque instant indépendamment, puisque, une fois encore, R_i et R_j sont indépendants lorsque $i, j \leq m$. Remarquons que l'on peut prendre une relation compatible de RCC8 car nous avons $\forall b \in R_{m+1} : \exists b' \in R_i : b \in \uparrow_{RCC8}^{PA} b'$ pour $1 \leq i \leq m$, puisque $R_{m+1} \subseteq \uparrow_{RCC8}^{PA} R_i$, et parce que (b', b) est cohérent. **(V)** Pour finir, par des arguments similaires à ceux de la condition C2, la sous-classe est \uparrow -close. \square

Nous identifions à présent les trois sous-classes de TT_{ic} qui sont maximales pour la traitabilité et contiennent toutes les relations basiques. Elles sont construites sur les sous-classes classiques de RCC8 : $\widehat{\mathcal{H}}_8$, \mathcal{C}_8 et \mathcal{Q}_8 [Renz, 1999].

Theorem 9. Les sous-classes traitables maximales de TT_{ic} qui contiennent $\widehat{\mathcal{B}}_{\text{RCC8}}$ sont $\widehat{\mathcal{H}}_8^m \times \text{PA}$, $\mathcal{C}_8^m \times \text{PA}$ et $\mathcal{Q}_8^m \times \text{PA}$.

Démonstration. Soit $\mathcal{S} \in \{\widehat{\mathcal{H}}_8, \mathcal{C}_8, \mathcal{Q}_8\}$. Nous utilisons le théorème d’affinement de Cohen-Solal *et al.* [2017a, Th. 26] pour montrer que $\mathcal{S} \times \text{PA}$ est algébriquement traitable, en l’affinant vers $\text{RCC8}_{\text{max}}^m \times \text{PA}_{\text{max}}$ par $H = (h_{\mathcal{S}}, \dots, h_{\mathcal{S}}, h_{\text{max}})$, avec $h_{\mathcal{S}}$ l’affinement usuel de \mathcal{S} vers les relations basiques [Gerevini et Renz, 2002, Fig. 2] et h_{max} l’affinement de PA vers les relations basiques de *dimension maximale* [voir Cohen-Solal *et al.*, 2017a]. **(I)** C1 : $\mathcal{S}^m \times \text{PA}$ est algébriquement stable par H , car (i) c’est le cas pour $m = 1$ [Cohen-Solal *et al.*, 2017a], (ii) il n’y a aucune interdépendance directe entre les instants de la séquence, et (iii) être clos par projection et être clos par composition sont des propriétés locales. **(II)** C2 : $\text{RCC8}_{\text{max}}^m \times \text{PA}_{\text{max}}$ est algébriquement traitable (proposition 8). **(III)** $\mathcal{S}^m \times \text{PA}$ est $\hat{\tau}$ -close, en utilisant les mêmes arguments que pour C1. Par le théorème d’affinement, $\mathcal{S}^m \times \text{PA}$ est donc algébriquement traitable.

Maintenant, puisque $\widehat{\mathcal{H}}_8$, \mathcal{C}_8 et \mathcal{Q}_8 sont les trois seules sous-classes de RCC8 qui contiennent toutes les relations basiques et qui sont maximales pour la traitabilité (en supposant $\text{P} \neq \text{NP}$) [Renz, 1999], $\widehat{\mathcal{H}}_8^m \times \text{PA}$, $\mathcal{C}_8^m \times \text{PA}$ et $\mathcal{Q}_8^m \times \text{PA}$ sont ainsi maximales également. \square

5 Conclusion et perspectives

Nous proposons une approche générale, basée sur les multialgèbres, pour raisonner à propos des séquences temporelles d’informations qualitatives, qui est plus performante que les approches de l’état de l’art. Nous introduisons en particulier le nouveau formalisme de la *topologie temporalisée des régions de taille constante*, dont le raisonnement est complet pour la décision de la cohérence. De plus, nous avons identifié ses sous-classes traitables maximales contenant toutes les relations basiques.

Cet article est limité aux séquences temporelles, mais les idées présentées peuvent notamment être appliquées à l’approche temporellement plus expressive de Sioutis *et al.* [2014], lui fournissant une procédure de raisonnement complète pour les relations topologiques entre régions de taille constante.

Nous planifions de continuer l’étude de la topologie temporalisée en nous intéressant à la continuité sans relation intermédiaire, avec et sans préservation de la taille; nous conjecturons que le raisonnement pour décider la cohérence est également complet dans ces cas. Nous avons aussi pour objectif de montrer que les multialgèbres permettent d’imposer des *diagrammes de dominance* [Galton, 2000], qui sont des structures plus générales que les graphes de voisinage, et de traiter la contrainte de continuité sans *état* intermédiaire [Westphal *et al.*, 2013].

Références

- BENNETT, B., COHN, A. G., WOLTER, F. et ZAKHARYASHEV, M. (2002). Multi-dimensional modal logic as a framework for spatio-temporal reasoning. *Applied Intelligence*, 17(3):239–251.
- BOGAERT, P., Van de WEGHE, N., COHN, A. G., WITLOX, F. et DE MAEYER, P. (2006). The qualitative trajectory calculus on networks. *In Proc. of the International Conference on Spatial Cognition*, pages 20–38. Springer.
- CHEN, J., COHN, A. G., LIU, D., WANG, S., OUYANG, J. et YU, Q. (2015). A survey of qualitative spatial representations. *The Knowledge Engineering Review*, 30(01):106–136.
- COHEN-SOLAL, Q., BOUZID, M. et NIVEAU, A. (2015). An algebra of granular temporal relations for qualitative reasoning. *In Proc. of IJCAI*, pages 2869–2875.
- COHEN-SOLAL, Q., BOUZID, M. et NIVEAU, A. (2017a). Checking the consistency of combined qualitative constraint networks. *In Proc. of AAAI*, pages 1084–1090.
- COHEN-SOLAL, Q., BOUZID, M. et NIVEAU, A. (2017b). Décision de la cohérence des réseaux de contraintes qualitatives combinés. *Revue d’Intelligence Artificielle*, 31(1-2):41–70.
- COHEN-SOLAL, Q., BOUZID, M. et NIVEAU, A. (2017c). Temporal sequences of qualitative information : Reasoning about the topology of constant-size moving regions. *In Proc. of IJCAI*.
- COHN, A. G., BENNETT, B., GOODAY, J. et GOTTS, N. M. (1997). Qualitative spatial representation and reasoning with the region connection calculus. *GeoInformatica*, 1(3):275–316.
- COHN, A. G., GOTTS, N. M., CUI, Z., RANDELL, D. A., BENNETT, B. et GOODAY, J. (1998). Exploiting temporal continuity in qualitative spatial calculi. *Spatial and Temporal Reasoning in Geographic Information Systems*, pages 5–24.
- COHN, A. G. et HAZARIKA, S. M. (2001). Qualitative spatial representation and reasoning : An overview. *Fundamenta informaticae*, 46(1-2):1–29.
- CUI, Z., COHN, A. G. et RANDELL, D. A. (1992). Qualitative simulation based on a logic of space and time. *In Proc. of AAAI*, pages 679–684.
- DYLLA, F. et MORATZ, R. (2004). Exploiting qualitative spatial neighborhoods in the situation calculus. *In Proc. of the International Conference on Spatial Cognition*, pages 304–322. Springer.

- DYLLA, F. et WALLGRÜN, J. O. (2007). Qualitative spatial reasoning with conceptual neighborhoods for agent control. *Journal of Intelligent and Robotic Systems*, 48(1):55–78.
- EGENHOFER, M. J. (2010). The family of conceptual neighborhood graphs for region-region relations. *In Proc. of the International Conference on Geographic Information Science*, pages 42–55. Springer.
- FREKSA, C. (1992). Temporal reasoning based on semi-intervals. *Artif. Intell.*, 54(1-2):199–227.
- GALTON, A. (1993). Towards an integrated logic of space, time, and motion. *In Proc. of IJCAI*, pages 1550–1555.
- GALTON, A. (2000). *Qualitative spatial change*. Oxford University Press.
- GEREVINI, A. et NEBEL, B. (2002). Qualitative spatio-temporal reasoning with RCC-8 and Allen’s interval calculus : Computational complexity. *In Proc. of ECAI*, pages 312–316.
- GEREVINI, A. et RENZ, J. (2002). Combining topological and size information for spatial reasoning. *Artif. Intell.*, 137(1):1–42.
- HAZARIKA, S. M. et COHN, A. G. (2001). Qualitative spatio-temporal continuity. *In Proc. of the International Conference on Spatial Information Theory*, pages 92–107. Springer.
- LIGOZAT, G. (2013). *Qualitative spatial and temporal reasoning*. John Wiley & Sons.
- LONG, Z. et LI, S. (2015). On distributive subalgebras of qualitative spatial and temporal calculi. *In Spatial Information Theory*, pages 354–374. Springer.
- MULLER, P. (2002). Topological spatio-temporal reasoning and representation. *Computational Intelligence*, 18(3):420–450.
- RAGNI, M. et WÖLFL, S. (2005). Temporalizing spatial calculi : On generalized neighborhood graphs. *In Proc. of KI*, pages 64–78. Springer.
- RAGNI, M. et WÖLFL, S. (2006). Temporalizing cardinal directions : From constraint satisfaction to planning. *In Proc. of KR*, pages 472–480.
- RENZ, J. (1999). Maximal tractable fragments of the region connection calculus : A complete analysis. *In Proc. of IJCAI*, pages 448–455.
- SANTOS, M. Y. et MOREIRA, A. (2009). Conceptual neighborhood graphs for topological spatial relations. *In Proc. of the World Congress on Engineering*, volume 1, pages 12–18.
- SIOUTIS, M., CONDOTTA, J.-F., SALHI, Y. et MAZURE, B. (2014). A qualitative spatio-temporal framework based on point algebra. *In Proc. of AIMSA*, pages 117–128. Springer.
- SIOUTIS, M., CONDOTTA, J.-F., SALHI, Y., MAZURE, B. et RANDELL, D. A. (2015). Ordering spatio-temporal sequences to meet transition constraints : Complexity and framework. *In Proc. of AI Applications and Innovations*, pages 130–150. Springer.
- Van de WEGHE, N. et DE MAEYER, P. (2005). Conceptual neighbourhood diagrams for representing moving objects. *In Proc. of the International Conference on Conceptual Modeling*, pages 228–238. Springer.
- VILAIN, M., KAUTZ, H. et van BEEK, P. (1989). Constraint propagation algorithms for temporal reasoning : A revised report. *In WELD, D. S. et de KLEER, J., éditeurs : Readings in Qualitative Reasoning About Physical Systems*, pages 373–381. Morgan Kaufmann Publishers Inc.
- WESTPHAL, M., DORNHEGE, C., WÖLFL, S., GISSLER, M. et NEBEL, B. (2011). Guiding the generation of manipulation plans by qualitative spatial reasoning. *Spatial Cognition & Computation*, 11(1):75–102.
- WESTPHAL, M., HUÉ, J., WÖLFL, S. et NEBEL, B. (2013). Transition constraints : A study on the computational complexity of qualitative change. *In Proc. of IJCAI*, pages 1169–1175.
- WÖLFL, S. et WESTPHAL, M. (2009). On combinations of binary qualitative constraint calculi. *In Proc. of IJCAI*, pages 967–973.
- WOLTER, D. et KREUTZMANN, A. (2015). Analogical representation of rcc-8 for neighborhood-based qualitative spatial reasoning. *In Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz)*, pages 194–207. Springer.
- WU, J., CLARAMUNT, C. et DENG, M. (2014). Towards a qualitative representation of movement. *In Proc. of the International Conference on Conceptual Modeling*, pages 191–200. Springer.
- ZIMMERMANN, K. et FREKSA, C. (1993). Enhancing spatial reasoning by the concept of motion. *Prospects for Artificial Intelligence*, pages 140–147.

Fonctions préservées par analogie : une méthode pour étendre des données booléennes

Analogy-preserving functions: A way to extend Boolean samples

Miguel Couceiro¹, Nicolas Hug, Henri Prade, Gilles Richard

1. LORIA, Campus Scientifique, 54506 Vandœuvre-lès-Nancy

2. IRIT, Université Paul Sabatier, 31062 Toulouse Cedex 9

miguel.couceiro@inria.fr, {nicolas.hug, henri.prade, gilles.richard}@irit.fr

Résumé

L'extension des ensembles d'exemples est une question importante en apprentissage automatique. En effet quand les exemples disponibles sont en quantité limitée, les performances des méthodes classiques de classification peuvent diminuer, et il peut être désirable de construire des exemples additionnels. Dans cet article, nous considérons l'usage du raisonnement analogique, et plus particulièrement des proportions analogiques pour étendre les ensembles d'exemples. On fait l'hypothèse que les classes sont déterminées par une fonction (partiellement connue). On examine les conditions requises sur les fonctions pour garantir une extension sans erreur des ensembles d'exemples, dans un cadre booléen. Dans ce but, on introduit la notion de fonctions préservant l'analogie (fonctions PA), et nous établissons qu'elles coïncident avec la classe des fonctions booléennes affines. Ce résultat théorique est complété par une étude expérimentale de PA-fonctions approchées, ce qui suggère qu'elles peuvent rester adéquates pour étendre des ensembles d'exemples.

Mots clés : proportion analogique, extension d'ensembles de données d'entraînement.

Abstract

Training set extension is an important issue in machine learning. Indeed when the examples at hand are in a limited quantity, the performances of standard classifiers may significantly decrease and it can be helpful to build additional examples. In this paper, we consider the use of analogical reasoning, and more particularly of analogical proportions for extending training sets. Here the ground truth labels are considered to be given by a (partially known) function. We examine the conditions that are required for such functions to ensure an error-free extension in a Boolean setting. To this end, we introduce the notion of Analogy Preserving (AP) functions, and we prove that their class is the class of affine Boolean functions. This theoretical result is comple-

ted with an empirical investigation of *approximate* functions AP, which suggests that they remain suitable for training set extension.

Keywords : proportion analogique, training set extension.

1 Introduction

La capacité à apprendre à partir de peu d'exemples est une faculté importante du cerveau humain, et joue un grand rôle dans l'élaboration des catégories cognitives chez l'enfant [14]. Cela contraste avec les algorithmes en apprentissage qui habituellement requièrent des ensembles de données suffisamment grands. La question d'apprendre à partir de quelques exemples n'est pas nouvelle, voir par exemple [12] dans le contexte de la reconnaissance des formes.

Le raisonnement analogique, qui est reconnu comme un moyen puissant d'établir des parallèles entre des entités apparemment indépendantes, peut également être employé pour construire de nouveaux exemples à partir d'un petit ensemble de données [2].

L'analogie a une longue histoire et a été principalement étudiée d'un point de vue psychologique [9, 13], mais des modélisations logiques ont aussi été introduites : ainsi [10] a donné une modélisation de l'analogie en logique du premier ordre qui apparaît trop restrictive, et une approche plus sophistiquée basée sur une logique d'ordre supérieure a été proposée [15], qui correspond à la l'idée de la théorie de la mise en parallèle de structures ("structure mapping theory") de [13].

En suivant des idées venant de l'anthropologie structuraliste [17] et de la linguistique computationnelle [18, 30], une modélisation en logique propositionnelle des proportions analogiques, i.e., des énoncés de la forme "**a** est à **b**

ce que \mathbf{c} est à \mathbf{d} ", a été introduite par [23, 25]. L'approche du raisonnement analogique basée sur les proportions, dont l'intérêt cognitif a été souligné depuis longtemps [27], a aussi montré qu'elle pouvait présenter un intérêt en classification sur problèmes tests [1, 4].

Les auteurs de [16] ont montré que ce type de processus de classification analogique pouvait être formalisé en deux grandes étapes : Tout d'abord une *extension analogique* d'un ensemble de données est construite (comme expliqué en Section 3), et ensuite un algorithme de type k -NN est appliqué à cet ensemble de données étendu. Comme on peut s'y attendre, les performances du classifieur analogique dépendent grandement de la qualité de l'extension. Dans cet article, on introduit la classe PA des fonctions préservées par l'analogie qui garantissent une extension sans erreur.

L'article est organisé comme suit. En Section 2 nous proposons une vue d'ensemble des différentes méthodes couramment utilisées pour étendre un ensemble de données. En Section 3 on rappelle des éléments de base sur l'analogie et son expression en logique booléenne (sous forme de proportion analogique), en pointant l'existence de deux modélisations potentielles. On rappelle aussi le processus d'extension d'un ensemble de données par l'analogie et on introduit la notion de fonctions PA. La Section 4 porte sur la caractérisation théorique des fonctions PA. En section 5 on définit et on étudie empiriquement les fonctions PA *ap-prochées*. On montre ensuite leur intérêt pour l'extension des ensembles de données en pratique.

2 Extension d'un ensemble de données

L'extension d'un ensemble de données ("training set") d'un référentiel donné \mathcal{X} est une idée simple pour améliorer les capacités de généralisation d'un classifieur. Il s'agit d'ajouter à un ensemble de données S de nouveaux exemples, mais en veillant à préserver la *qualité* de S .

Formellement, on part d'un ensemble $S = \{x^{(i)} \in \mathcal{X} | i \in [1, n]\}$ d'exemples (n est supposé petit), où $x^{(i)}$ est un élément d'un produit cartésien $\mathcal{X} = X_1 \times \dots \times X_m$. A chaque élément $x^{(i)} \in S$, on associe une valeur de la fonction cible $f(x^{(i)}) = y^{(i)} \in Y$. En cas de régression, $y^{(i)} \in \mathbb{R}$, et en cas de classification $y^{(i)}$ appartient à un ensemble fini et est appelé "classe" ou **étiquette**.

Plusieurs méthodes ont été proposées pour étendre un ensemble de données avec de nouveaux exemples. On peut ainsi obtenir un nouvel exemple en partant de 1, 2 ou 3 exemples connus.

1. Avec un seul exemple, une façon naturelle de procéder est d'utiliser l'approche classique par voisinage : étant donné l'exemple $(a, f(a))$, on peut générer un nouvel exemple $(b, f(b))$ où b est proche de a et $f(b)$ est proche de $f(a)$. En classification, $f(b)$ peut être choisi comme $f(a)$.

2. Avec deux exemples, l'option précédente reste applicable et conduit à interpoler le nouvel exemple à partir des deux qui sont donnés. Une option un peu différente est la procédure dite de "Feature Knockout" [32], qui consiste à construire un troisième exemple en modifiant un attribut choisi au hasard du premier exemple par sa valeur dans le second exemple. Ce procédé possède des propriétés intéressantes et est en fait équivalent à une technique populaire de régularisation (Tikhonov) en régression linéaire. Une idée apparentée est utilisée dans un proposition récente [5] qui introduit une mesure d'étrangeté ("oddness") par rapport à une classe qui est calculée sur la base de paires constituées de deux plus proches voisins dans la même classe ; la méthode revient à remplacer les deux voisins par un représentant fictif de la classe.
3. Avec trois exemples $(a, f(a)), (b, f(b)), (c, f(c))$, les options précédentes restent possibles et conduisent à construire un quatrième exemple qui est dans un certain sens entre les trois autres : nous avons encore une forme d'interpolation. Une idée assez différente est d'extrapoler le quatrième item sur la base d'une proportion analogique [2]. Dans cette perspective, le quatrième élément n'est pas nécessairement dans le voisinage des trois autres.

Dans cet article, nous étudions cette dernière option en profondeur. Cet article est la version française de [6].

3 Proportions analogiques pour l'extension d'ensembles de données

Comme indiqué plus haut, les classifieurs analogiques [16] étendent d'abord l'ensemble des données, puis appliquent une procédure type k -NN à l'ensemble de données ainsi augmenté. Dans la suite, on se focalise sur la première étape, l'extension étant basée sur la notion de proportion analogique.

L'idée de **proportion** fut introduite dans la Grèce Ancienne dans un cadre numérique avec en particulier les exemples de :

1. la **proportion arithmétique**, où a, b, c, d sont en proportion si $a - b = c - d$;
2. la **proportion géométrique**, où a, b, c, d sont en proportion si $\frac{a}{b} = \frac{c}{d}$.

Ces exemples capturent implicitement l'idée de proportion analogique qui correspond à un énoncé de la forme *a est à b comme c est à d*.

Dans cette section, on rappelle tout d'abord les propriétés de base des proportions analogiques, et on décrit ensuite le processus de construction de l'extension analogique d'un ensemble de données. Dans la suite, X dénote un ensemble non vide, et les éléments de X seront dénotés par des lettres minuscules a, b, c, d, \dots . De plus, les m -uplets (ou vecteurs)

sur X seront notés par lettres en caractères gras $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}$, .. la i -ème composante d'un m -uplet \mathbf{a} sera noté a_i .

3.1 Proportions analogiques

Comme leur contrepartie numérique, une **proportion analogique**¹ sur un ensemble non vide X est une relation quaternaire A sur X qui satisfait les 3 axiomes suivants [11, 19] :

1. **Identité** : pour tout $a, b \in X$, $A(a, b, a, b)$.
2. **Symétrie** : pour tout $a, b, c, d \in X$, si $A(a, b, c, d)$, alors $A(c, d, a, b)$.
3. **Permutabilité centrale** : pour tout $a, b, c, d \in X$, si $A(a, b, c, d)$, alors $A(a, c, b, d)$.

Il y a différentes manières de définir une analogie sur un ensemble X , selon la structure sous-jacente et les opérateurs disponibles. Dans [22, 30, 21], on trouvera des exemples pour des structures matricielles, des mots sur un alphabet, des treillis, etc. Quand $X = \mathbb{B} = \{0, 1\}$, des expressions logiquement équivalentes de la proportion analogique ont été données [23, 25] :

$$A(a, b, c, d) \text{ si } (a \wedge \neg b \leftrightarrow c \wedge \neg d) \wedge (\neg a \wedge b \leftrightarrow \neg c \wedge d) \\ \text{ou de manière équivalente, si} \\ (a \wedge d \leftrightarrow b \wedge c) \wedge (a \vee d \leftrightarrow b \vee c),$$

où \leftrightarrow est le connecteur d'équivalence : $x \leftrightarrow x' = 1$ si $x = x'$ et 0 sinon. La première expression exprime que a diffère de b comme c diffère de d et inversement b diffère de a comme d diffère de c . La seconde équivalence traduit le fait que l'analogie se comporte comme une proportion numérique pour ce qui est des extrêmes et des moyens.

Nous ferons référence dans la suite à ce modèle booléen de l'analogie comme le modèle **standard**. En fait, il y a d'autres modèles de l'analogie qui satisfont les trois axiomes qui peuvent être construits dans \mathbb{B} [24]. Parmi eux, un présente un intérêt particulier : il s'agit du modèle proposé dans [17], qui est obtenu en relaxant le modèle standard sous la forme $(a \leftrightarrow b) \leftrightarrow (c \leftrightarrow d)$. On référera aux modèles standard et de Klein par A_S et A_K respectivement. Contrairement au modèle standard, celui de Klein satisfait une propriété apparemment attractive : $A_K(a, \neg a, b, \neg b)$.

La Table 1 montre les 8 lignes où les proportions du modèle de Klein tiennent. Pour les 8 configurations de valeurs de a, b, c, d restantes qui n'apparaissent pas dans la table, aucune ne correspond à une proportion valide pour aucun des deux modèles. On peut voir que le modèle de Klein, quoique attractif à première vue, obéit à la propriété suivante qui ne semble pas naturelle pour une analogie : $A_K(a, b, c, d) \iff A_K(b, a, c, d)$. Il se trouve cependant que tous les résultats théoriques dans cet article seront valides à la fois pour le modèle standard et le modèle de

1. dans le reste de l'article, le terme *analogie* signifie toujours *proportion analogique*.

a	b	c	d	A_S	A_K
0	0	0	0	1	1
1	1	1	1	1	1
0	0	1	1	1	1
1	1	0	0	1	1
0	1	0	1	1	1
1	0	1	0	1	1
0	1	1	0	0	1
1	0	0	1	0	1

TABLE 1 – Valuations de $a : b :: c : d$ pour le modèle standard et le modèle de Klein de l'analogie. Les valuations manquantes donnent 0 pour A_S et A_K .

Klein ; on utilisera la notation $a : b :: c : d$ qui vaudra pour $A_S(a, b, c, d)$ ou $A_K(a, b, c, d)$ indifféremment. En Section 5, on verra cependant que, en dépit de propriétés théoriques similaires, les résultats empiriques montrent que le modèle standard est préférable.

On peut vérifier sur la Table 1 que A_S et A_K satisfont aussi une sorte de *propriété d'indépendance au codage* :

$$a : b :: c : d \iff \neg a : \neg b :: \neg c : \neg d,$$

ce qui garantit que 0 et 1 jouent des rôles symétriques. Notons par ailleurs que les analogies sur l'ensemble X induisent une analogie sur X^m où pour $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d} \in X^m$,

$$\mathbf{a} : \mathbf{b} :: \mathbf{c} : \mathbf{d} \text{ si } a_i : b_i :: c_i : d_i \text{ pour chaque } i \in [1, m].$$

A partir de cette définition et en utilisant la Table 1 on peut déduire la propriété suivante qui sera utilisée dans de futures preuves :

Property 1. *Pour tout $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d} \in \mathbb{B}^m$ tel que $\mathbf{a} : \mathbf{b} :: \mathbf{c} : \mathbf{d}$, on a $h(\mathbf{a}, \mathbf{b}) = h(\mathbf{c}, \mathbf{d})$, $h(\mathbf{a}, \mathbf{c}) = h(\mathbf{b}, \mathbf{d})$ et $h(\mathbf{a}, \mathbf{d}) = h(\mathbf{b}, \mathbf{c})$, où $h(\mathbf{x}, \mathbf{x}')$ est la distance de Hamming, définie comme le nombre de composantes qu'on doit changer pour transformer \mathbf{x} en \mathbf{x}' (ou l'inverse).*

3.2 Equation analogique et principe inférentiel

Une fois la notion de proportion analogique définie sur un ensemble X , étant donnés 3 éléments a, b, c de X , l'exigence que la relation $a : b :: c : x$ soit vraie définit une équation où x est l'inconnue. Selon l'ensemble X , l'analogie A et les éléments $a, b, c \in X$, on peut rencontrer une des trois situations : l'équation n'a pas de solution, l'équation a une solution unique, ou l'équation a plusieurs solutions.

Dans le cas booléen, $A_S(a, b, c, x)$ a une solution si et seulement si $(a \leftrightarrow b) \vee (a \leftrightarrow c)$ est vrai. Ainsi $A_S(1, 0, 0, x)$ et $A_S(0, 1, 1, x)$ n'ont pas de solution. Quand la solution existe, elle est donnée par $x = (a \leftrightarrow b) \leftrightarrow c$ ou plus simplement par $x = c$ si $a \leftrightarrow b$, et par $x = b$ si $a \leftrightarrow c$. Notons que pour A_K , l'équation a toujours une solution.

Le processus de résolution d'équation analogique joue un rôle essentiel dans le **principe d'inférence analogique** qui peut s'écrire schématiquement comme suit :

$$\frac{\mathbf{a} : \mathbf{b} :: \mathbf{c} : \mathbf{d}}{f(\mathbf{a}) : f(\mathbf{b}) :: f(\mathbf{c}) : f(\mathbf{d})},$$

pour des m -uplets $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d} \in X^m$ et une fonction $f: X^m \rightarrow X$. En essence, il pose que si des m -uplets $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d} \in X^m$ sont en analogie, alors leurs images par f (et dans notre cas, leurs étiquettes) sont aussi en analogie. Ce principe peut être vu comme un cas particulier d'un autre principe appelé quelquefois *saut analogique* [10]; voir [26] sur ce point.

Le principe d'inférence analogique requière que la fonction f satisfasse deux conditions :

- (i) l'équation $f(\mathbf{a}) : f(\mathbf{b}) :: f(\mathbf{c}) : y$ a une solution quand $\mathbf{a} : \mathbf{b} :: \mathbf{c} : \mathbf{d}$ tient,
- (ii) la solution y est égal à $f(\mathbf{d})$.

Dans la sous-section suivante, on rappelle comment ce principe peut être utilisé pour prédire des valeurs inconnues et étendre un jeu d'essai. Dans le reste de l'article on se focalise sur le cas booléen ($X = \mathbb{B}$), et on recherche une caractérisation complète des fonctions booléennes qui sont complètement compatibles avec ce principe.

3.3 Extension d'un jeu d'essai par analogie

Etant donné un jeu d'essai $S \subseteq \mathbb{B}^m$ et une fonction $f: \mathbb{B}^m \rightarrow \mathbb{B}$, on définit ce qu'on appelle l'**extension analogique** de S utilisant f , comme suit :

$$\mathbf{E}_S(f) = \{\mathbf{x} \in \mathbb{B}^m \mid \exists (\mathbf{a}, \mathbf{b}, \mathbf{c}) \in S^3, \mathbf{a} : \mathbf{b} :: \mathbf{c} : \mathbf{x} \text{ et } f(\mathbf{a}) : f(\mathbf{b}) :: f(\mathbf{c}) : y \text{ a une solution}\}$$

Intuitivement, $\mathbf{E}_S(f)$ peut être vu comme l'ensemble de tous les $\mathbf{x} \in \mathbb{B}^m$ qui sont en analogie avec au moins un 3-uplet dans S , pourvu que l'équation sur les étiquettes associées soit aussi soluble. Il est clair que $S \subseteq \mathbf{E}_S(f)$, et on dénote $\mathbf{E}_S(f) \setminus S$ par $\mathbf{E}_S^*(f)$. Chaque élément de $\mathbf{E}_S(f)$ reçoit une **étiquette analogique** $\bar{\mathbf{x}}_f$. Pour les éléments de $\mathbf{E}_S^*(f)$, $\bar{\mathbf{x}}_f$ est défini comme la prédiction la plus commune parmi les solutions candidates y . Pour les éléments de S , on rend simplement pour $\bar{\mathbf{x}}_f$ la valeur $f(\mathbf{x})$ qui est connue. Voici une description algorithmique du processus :

1. Tout d'abord, ajouter chaque $\mathbf{x} \in S$ à $\mathbf{E}_S(f)$. Alors, pour tout $\mathbf{a}, \mathbf{b}, \mathbf{c} \in S$ tel que $f(\mathbf{a}) : f(\mathbf{b}) :: f(\mathbf{c}) : y$ est soluble et tel qu'il existe $\mathbf{x} \in \mathbb{B}^m \setminus S$ with $\mathbf{a} : \mathbf{b} :: \mathbf{c} : \mathbf{x}$, ajouter \mathbf{x} à $\mathbf{E}_S(f)$ et conserver y comme un candidat pour $\bar{\mathbf{x}}_f$. Techniquement $\mathbf{x} \in \mathbf{E}_S^*(f)$.
2. Puis pour chaque $\mathbf{x} \in \mathbf{E}_S^*(f)$, effectuer une **procédure de vote majoritaire** : prendre pour $\bar{\mathbf{x}}_f$ le candidat le plus commun parmi toutes les solutions y en cas de d'ex-æquo, prendre une des valeurs au hasard). Pour les éléments de S , $\bar{\mathbf{x}}_f$ est simplement $f(\mathbf{x})$.

L'extension analogique peut alors être utilisée, comme un ensemble de données augmenté, par n'importe quel classifieur, en utilisant les étiquettes analogiques calculées comme si c'était la vérité terrain [16]. Il est donc naturel de s'attendre à ce que pour chaque $\mathbf{x} \in \mathbf{E}_S^*(f)$ on ait $\bar{\mathbf{x}}_f = f(\mathbf{x})$. La notion de fonction PA va nous aider à formaliser cette espérance.

3.4 Fonctions préservant l'analogie

Définition 1. On dit que l'extension $\mathbf{E}_S(f)$ est **saine** si $\bar{\mathbf{x}}_f = f(\mathbf{x})$, pour chaque $\mathbf{x} \in \mathbf{E}_S^*(f)$. Aussi, si $\mathbf{E}_S(f)$ est sain pour tous les $S \subseteq \mathbb{B}^m$, on dit alors que f **préservé l'analogie** (on parle alors de "PA fonction").

La Proposition 1 fournit une définition équivalente des PA fonctions.

Proposition 1. Une fonction $f: \mathbb{B}^m \rightarrow \mathbb{B}$ est PA si et seulement si pour tout $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d} \in \mathbb{B}^m$, f satisfait la condition suivante :

$$\left\{ \begin{array}{l} \mathbf{a} : \mathbf{b} :: \mathbf{c} : \mathbf{d} \text{ et} \\ f(\mathbf{a}) : f(\mathbf{b}) :: f(\mathbf{c}) : y \text{ est soluble} \end{array} \right. \implies y = f(\mathbf{d})$$

Démonstration. si f satisfait cette condition, alors il est clair d'après la sous-section 3.3 que pour tout $S \subseteq \mathbb{B}^m$ et $\mathbf{x} \in \mathbf{E}_S^*(f)$, tous les candidats y pour $\bar{\mathbf{x}}_f$ sont égal à $f(\mathbf{x})$, aussi $\bar{\mathbf{x}}_f$ sera toujours $f(\mathbf{x})$, ce qui rend la fonction f PA.

Si f ne satisfait pas cette condition, alors il existe $\mathbf{a}, \mathbf{b}, \mathbf{c}$, et $\mathbf{d} \in \mathbb{B}^m$ tels que $\mathbf{a} : \mathbf{b} :: \mathbf{c} : \mathbf{d}$ mais la solution y pour $f(\mathbf{a}) : f(\mathbf{b}) :: f(\mathbf{c}) : y$ n'est pas égal à $f(\mathbf{d})$. En prenant $S_0 = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$ on obtient $\mathbf{E}_{S_0}^*(f) = \{\mathbf{d}\}$, et puisque $\bar{\mathbf{d}}_f = y \neq f(\mathbf{d})$, $\mathbf{E}_{S_0}(f)$ n'est pas saine, alors f n'est pas PA. \square

Si f est PA, alors l'extension $\mathbf{E}_S(f)$ est saine pour tout S , c'est donc une extension pertinente de S qui peut être utilisée en toute confiance pour la classification. Dans la suite de l'article, on apporte une réponse au problème suivant :

Problème 1. Donner une description complète des fonctions $f: \mathbb{B}^m \rightarrow \mathbb{B}$ qui assurent une extension $\mathbf{E}_S(f)$ saine pour tout jeu d'essai $S \subseteq \mathbb{B}^m$. En d'autres termes, il s'agit d'identifier toutes les PA fonctions.

Notons tout d'abord que beaucoup de fonctions "naturelles" ne sont pas PA. Considérons par exemple la fonction binaire $f(x_1, x_2) = x_1 \wedge x_2$, avec $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d} \in \mathbb{B}^2$ donnés dans la Table 2. On a $\mathbf{a} : \mathbf{b} :: \mathbf{c} : \mathbf{d}$ et $f(\mathbf{a}) : f(\mathbf{b}) :: f(\mathbf{c}) : y$ est soluble, mais la solution est $y = 0$, qui est différente de $f(\mathbf{d}) = 1$, donc f n'est pas PA. Cela est à relier au fait que les proportions analogiques ne sont pas stables par combinaison conjonctive. Ce serait aussi le cas pour la disjonction [25].

Dans la section suivante, on présente une description complète des fonctions PA et on répond ainsi au Problème 1.

	x_1	x_2	$f(\cdot)$
a	0	0	0
b	0	1	0
c	1	0	0
d	1	1	1

TABLE 2 – $f(x_1, x_2) = x_1 \wedge x_2$ n'est pas PA.

4 La classe des fonctions PA

Il nous faut tout d'abord rappeler quelques notions de base de la théorie des variables essentielles pour les fonctions.

4.1 Variables essentielles et sections de fonctions

Pour $i \in [1, m]$, $\alpha \in \mathbb{B}^m$ et $c \in \mathbb{B}$, soit α_i^c m -uplet de \mathbb{B}^m obtenu en remplaçant α_i par c dans α . Une variable x_i est dite **inessentielle** dans $f: \mathbb{B}^m \rightarrow \mathbb{B}$ si pour tout $\alpha \in \mathbb{B}^m$ et $c \in \mathbb{B}$, $f(\alpha_i^c) = f(\alpha_i)$. Sinon, x_i est dite **essentielle** dans f , ou encore que f dépend de x_i . En termes simples, une variable essentielle est une variable qui a la *capacité* de changer la valeur de f . Par exemple dans $f(x_1, x_2, x_3) = x_1 \wedge x_3$, x_1 et x_3 sont des variables essentielles tandis que x_2 est inessentielle. On notera $\text{ess}(f)$ le nombre de variables essentielles de f (ou **arité essentielle**).

Deux fonctions $f: \mathbb{B}^m \rightarrow \mathbb{B}$ et $g: \mathbb{B}^n \rightarrow \mathbb{B}$ sont dites **équivalentes** s'il existe deux applications $\sigma: [1, m] \rightarrow [1, n]$ et $\sigma': [1, m] \rightarrow [1, n]$ telles que

$$\begin{aligned} f(x_1, \dots, x_m) &= g(x_{\sigma(1)}, \dots, x_{\sigma(m)}) \text{ et} \\ g(x_1, \dots, x_n) &= f(x_{\sigma'(1)}, \dots, x_{\sigma'(m)}). \end{aligned}$$

En d'autres termes, f et g sont équivalentes si l'une peut être obtenue à partir de l'autre par permutation des variables, addition de variables inessentiels, ou identification de variables inessentiels. Par exemple, $f(x_1, x_2, x_3) = x_1 \wedge x_3$ et $g(x_1, x_2) = x_1 \wedge x_2$ sont des fonctions équivalentes. Notons que deux fonctions équivalentes ont nécessairement le même nombre de variables essentielles. Pour plus de détails sur la théorie des variables essentielles de fonctions, voir [8, 7, 28, 31].

Dans nos démonstrations, on utilisera la propriété suivante :

Property 2. Soit $f: \mathbb{B}^m \rightarrow \mathbb{B}$ et $g: \mathbb{B}^n \rightarrow \mathbb{B}$ des fonctions équivalentes. Alors f est PA si et seulement si g est PA.

Ceci peut être vérifié en remarquant que l'analogie dans \mathbb{B}^m est définie composante par composante, que la permutation de variables n'a pas d'effet sur les équations et leur solution. De plus, la manipulation de variables inessentiels ne change pas la valeur de la fonction f , et donc la propriété PA tient encore.

Nous définissons maintenant le concept de **section** d'une fonction, aussi connu sous le nom de *restriction*, ou encore de manière équivalente comme le résultat d'une *application partielle* en informatique. Soit f une fonction $\mathbb{B}^m \rightarrow \mathbb{B}$, et (I, J) une partition de $[1, m]$. Pour $\mathbf{x} \in \mathbb{B}^m$ et $\alpha \in \mathbb{B}^{|J|}$, la I -section (ou simplement la section) $f_I^\alpha: \mathbb{B}^{|J|} \rightarrow \mathbb{B}$ est la fonction qui est obtenue en fixant toutes les variables de I aux valeurs des composantes de α . Notons que l'arité de f_I^α est $|J|$, et que $\text{ess}(f_I^\alpha) \leq \text{ess}(f)$. Par exemple, considérons la fonction f à trois variables : $f(x_1, x_2, x_3) = (x_1 \wedge x_2) \vee x_3$. La section $f_{\{1,3\}}^{(1,0)}$ est définie par $f_{\{1,3\}}^{(1,0)}(x_2) = (1 \wedge x_2) \vee 0 = x_2$.

Un résultat important à propos des sections qui sera utilisé dans les autres preuves est énoncé par la Propriété 3, qu'on peut vérifier en notant que $x : x :: x : x$ pour tout $x \in \mathbb{B}$:

Property 3. Si $f: \mathbb{B}^m \rightarrow \mathbb{B}$ est PA, alors toute section de f est aussi PA.

Dans ce qui suit, l'opérateur ET, ' \wedge ', sera noté ' \cdot ' pour se conformer à une notation algébrique. De plus, ' $+$ ' dénotera maintenant l'addition modulo-2, équivalente à l'opérateur XOR. Notons que $x + 1 = \neg x$.

4.2 Les fonctions affines

Nous sommes maintenant en position d'examiner des exemples de fonctions PA. Nous allons montrer que toute fonction affine est PA.

Proposition 2. Soit L la classe de toutes les fonctions affines, i.e. les fonctions de la forme :

$$f(x_1, \dots, x_m) = \alpha_1 \cdot x_1 + \dots + \alpha_m \cdot x_m + \alpha,$$

avec $\alpha_1, \dots, \alpha_m, \alpha \in \mathbb{B}$. Toute fonction affine (aussi appelée **linéaire** quand $\alpha = 0$) est PA.

Démonstration. Soit $f: \mathbb{B}^m \rightarrow \mathbb{B} \in L$. En remarquant que f est PA si et seulement si $f + 1 = \neg f$ est PA, on peut supposer sans perte de généralité que $\alpha = 0$. De plus, en considérant que f dépend essentiellement de $n \leq m$ variables (n est le nombre de α_i égal à 1), f est équivalente à la fonction $g: \mathbb{B}^n \rightarrow \mathbb{B}$ définie par $g(x_1, \dots, x_n) = x_1 + \dots + x_n$. En prenant en compte la Propriété 2, on a juste à prouver que g est PA pour montrer que f est aussi PA.

La fonction g a la propriété remarquable² que si on change la valeur de n'importe quel x_i cela change la valeur de g : $\forall_i, g(x_1, \dots, x_i, \dots, x_n) = \neg g(x_1, \dots, \neg x_i, \dots, x_n)$. Compte tenu de cette propriété, il est facile de voir que :

$$\forall \mathbf{x}, \mathbf{x}' \in \mathbb{B}^n, g(\mathbf{x}) = g(\mathbf{x}') \iff h(\mathbf{x}, \mathbf{x}') \text{ est pair,}$$

2. Ceci est la raison pour laquelle les fonctions affines conduisent à des problèmes de classification qui sont, en fait, hautement **non** linéairement séparables.

où h est la distance de Hamming.

Soit $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d} \in \mathbb{B}^n$ tels que les deux hypothèses de la définition de PA soient satisfaites, i.e.

$$\mathbf{a} : \mathbf{b} :: \mathbf{c} : \mathbf{d} \quad \text{et} \quad g(\mathbf{a}) : g(\mathbf{b}) :: g(\mathbf{c}) : y \quad \text{est soluble.}$$

Comme l'équation est soluble, la Table 1 nous indique qu'il y a trois cas possibles (on utilise la Propriété 1) :

1. $g(\mathbf{a}) = g(\mathbf{b})$, et dans ce cas la solution est $y = g(\mathbf{c})$. Comme $g(\mathbf{a}) = g(\mathbf{b})$, alors $h(\mathbf{a}, \mathbf{b})$ est pair, ainsi que $h(\mathbf{c}, \mathbf{d})$. Alors, $g(\mathbf{c}) = g(\mathbf{d})$ et $y = g(\mathbf{d})$.
2. $g(\mathbf{a}) = g(\mathbf{c})$, et dans ce cas la solution est $y = g(\mathbf{b})$. Comme $g(\mathbf{a}) = g(\mathbf{c})$, alors $h(\mathbf{a}, \mathbf{c})$ est pair, ainsi que $h(\mathbf{b}, \mathbf{d})$. Donc $g(\mathbf{b}) = g(\mathbf{d})$ et $y = g(\mathbf{d})$.
3. $\neg g(\mathbf{a}) = g(\mathbf{b}) = g(\mathbf{c})$, et dans ce cas la solution est $y = g(\mathbf{a})$. Comme $g(\mathbf{b}) = g(\mathbf{c})$, alors $h(\mathbf{b}, \mathbf{c})$ est pair, ainsi que $h(\mathbf{a}, \mathbf{d})$. Donc $g(\mathbf{a}) = g(\mathbf{d})$ et $y = g(\mathbf{d})$.

Le troisième cas est seulement pertinent pour le modèle de Klein. Dans tous les cas on a $y = g(\mathbf{d})$, ce qui montre que g est PA, et par suite tout $f \in L$. \square

4.3 Une description complète des fonctions PA

Nous venons de voir que toute fonction affine est PA. Nous allons maintenant établir un résultat plus fort : les fonctions affines sont les **seules** fonctions PA.

Pour cela on va utiliser la représentation polynomiale des fonctions booléennes. Un **monôme** est un terme de la forme :

$$\mathbf{x}_I = \prod_{i \in I} x_i,$$

pour un ensemble quelconque possiblement vide d'entiers positifs I , où $|I|$ est appelé le **degré** de \mathbf{x}_I . On prend la convention que 1 est le monôme vide \mathbf{x}_\emptyset . Un **polynôme** est une somme de monômes et son degré est le plus grand degré de ses monômes. Il est bien connu [29, 33] que toute fonction $f : \mathbb{B}^m \rightarrow \mathbb{B}$ est représentée de façon unique par un tel polynôme, aussi appelé sa Forme Normale Algébrique :

$$f(x_1, \dots, x_m) = \sum_{I \subseteq \{1, \dots, m\}} a_I \cdot \mathbf{x}_I$$

où chaque a_I appartient à \mathbb{B} . Notons que la fonction constante 0 est représentée par $a_\emptyset \cdot \mathbf{x}_\emptyset$ avec $a_\emptyset = 0$. Le degré d'une fonction $f : \mathbb{B}^m \rightarrow \mathbb{B}$, noté $d(f)$, est défini comme le degré du polynôme unique représentant f .

Notons que la classe des fonctions de degré au plus 1 est exactement la classe L des fonctions affines, qui sont PA. On va montrer que la classe des PA fonctions est la classe des fonctions affines en prouvant que si une fonction f est PA, alors $d(f) \leq 1$. On considère d'abord le cas où $d(f) = 2$.

Property 4. Soit $f : \mathbb{B}^m \rightarrow \mathbb{B}$ avec $d(f) = 2$. f n'est pas PA.

Démonstration. Considérons f avec $d(f) = 2$ et $\text{ess}(f) \geq 2$. On note \mathbf{x}_I un des monômes de f de degré 2. On considère la section f_J^0 , où $J = [1, m] \setminus I$, où $\mathbf{0}$ dénote le vecteur constant 0 in $\mathbb{B}^{|J|}$. Toutes les variables qui ne font pas partie du monôme \mathbf{x}_I ont été mises à 0. La section f_J^0 a un monôme unique de degré 2 (\mathbf{x}_I) et $\text{ess}(f) = 2$. f_J^0 est nécessairement équivalente à une des fonctions suivantes :

$$\begin{cases} f_1(x_1, x_2) = x_1 \cdot x_2 + \alpha \\ f_2(x_1, x_2) = x_1 \cdot x_2 + x_1 + \alpha \\ f_3(x_1, x_2) = x_1 \cdot x_2 + x_1 + x_2 + \alpha \end{cases}$$

Il est assez facile de trouver des exemples de $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d} \in \mathbb{B}^2$ pour montrer qu'aucune de ces fonctions n'est PA (voir la Table 2), so f_J^0 ne peut pas être PA non plus. Comme f_J^0 est une section f , f n'est pas PA. \square

Ce dont nous avons besoin maintenant c'est d'une propriété qui nous permettrait de décroître le degré d'une fonction sans changer sa propriété PA. C'est le rôle de la Propriété 5.

Property 5. Soit $f : \mathbb{B}^m \rightarrow \mathbb{B}$ une fonction telle que $d(f) = k \geq 2$. Alors il y a une section g de f telle que $d(g) = k - 1$.

Démonstration. Supposons que $d(f) = k \geq 2$, et soit \mathbf{x}_I un monôme de f de degré maximum, i.e. $|I| = k$. A nouveau, considérons la section $g = f_J^0$ où $J = [1, m] \setminus I$ et $\mathbf{0}$ dénote le vecteur constant 0 de $\mathbb{B}^{|J|}$. Il est clair que g est représenté par un polynôme qui a un unique monôme de degré maximal k , c'est-à-dire \mathbf{x}_I , et peut-être quelques autres monômes de degré strictement inférieur à k . Prenons un $i \in I$: alors $g' = g_{\{i\}}^1$ est une section de g de degré (et d'arité) $k - 1$. Comme g' est une section de g , c'est aussi une section de f , ce qui complète la preuve. \square

Nous sommes maintenant en position d'établir le résultat principal.

Proposition 3. La classe des fonctions PA est la classe L des fonctions affines.

Démonstration. Nous avons vu que toute fonction affine est PA, i.e. si $d(f) \leq 1$, alors $f \in PA$. D'autre part, la Propriété 4 nous dit que si $d(f) = 2$, alors $f \notin PA$. Aussi supposons que $d(f) \geq 3$. Par des applications successives de la Propriété 5, il s'ensuit que qu'il y a une section g de f avec $d(g) = 2$. Comme g n'est pas PA, f n'est pas PA non plus à cause de la Propriété 3. En résumé, si $d(f) \geq 2$ alors f n'est pas PA, at donc la classe des fonctions PA est exactement L . \square

On peut donc apporter une réponse finale au problème initial : **la classe des fonctions qui assurent une extension**

saine pour tout jeu d'essai $S \subseteq \mathbb{B}^m$ est la classe L des fonctions affines. Si la fonction n'est pas affine, alors il existe un jeu d'essai $S_0 \subseteq \mathbb{B}^m$ pour lequel $\mathbf{E}_{S_0}(f)$ n'est pas sain.

Cependant, tandis que ce résultat théorique est intéressant en soi, il est évident que les fonctions purement affines ne sont pas représentatives de ce que l'on peut rencontrer dans la réalité. Ceci conduit au **Problème 2** : Que reste-t-il de la qualité de l'extension analogique $\mathbf{E}_S(f)$ quand f s'écarte d'être PA de différentes manières ? Le but de la section suivante est de commencer à étudier empiriquement cette question.

5 Fonctions approximativement PA et expériences

Ici, nous étudions d'abord la qualité de l'extension analogique quand la fonction f n'est pas exactement affine. Etant donné un échantillon S , nous définissons $\omega(S, f)$ (ou simplement ω) comme la **qualité** de l'extension $\mathbf{E}_S(f)$:

$$\omega(S, f) = P_{\mathbf{x} \in \mathbf{E}_S^*(f)} [\bar{\mathbf{x}}_f = f(\mathbf{x})],$$

où $P_{\mathbf{x}}$ est la distribution uniforme sur \mathbb{B}^m .

Ici, $\mathbf{x} \in \mathbb{B}^m$ est considérée comme une variable aléatoire. Par définition, $\omega(S, f) = 1$ pour tout S si et seulement si f est PA. Il est clair que $\omega(S, f) = \omega(S, \neg f)$ pour tout S et pour toute f , simplement à cause de la propriété de *code independence* (voir Section 3.1).

Etant données 2 fonctions booléennes f et g , on définit leur distance comme $\text{dist}(f, g) = P_{\mathbf{x}} [f(\mathbf{x}) \neq g(\mathbf{x})]$. On dit que f est ϵ -proche de g si $\text{dist}(f, g) \leq \epsilon$, et que f est ϵ -proche d'un ensemble Σ s'il $\exists g \in \Sigma$ telle que f est ϵ -proche de g .

Nous étudions les variations de ω quand f est ϵ -proche de L . En fait, comme l'ensemble des fonctions affines est constitué de l'ensemble des fonctions linéaires ainsi que de leurs compléments, des propriétés de ω , on déduit qu'il suffit de considérer que f est ϵ -proche de l'ensemble des fonctions linéaires. Ce qui est très utile car, en pratique, on peut utiliser le test BLR [3] qui permet d'interroger une fonction partiellement connue pour découvrir si elle est ϵ -proche de l'ensemble des fonctions linéaires.

Partant de $g: \mathbb{B}^8 \rightarrow \mathbb{B}$ linéaire définie comme $g(\mathbf{x}) = x_1 + \dots + x_8$, on introduit du bruit en niant certaines entrées $g(\mathbf{x})$ avec une fréquence ϵ . On obtient ainsi une fonction f_ϵ qui est nécessairement ϵ -proche de l'ensemble des fonctions linéaires.

Dans la Figure 1, nous présentons les valeurs de $\omega(S, f_\epsilon)$ avec le modèle Standard pour différentes tailles de S (exprimées comme un pourcentage de $|\mathbb{B}^m|$). Les résultats sont des moyennes sur 50 expériences. D'autres expériences ont été faites avec d'autres fonctions linéaires (i.e. avec moins de variables essentielles ou avec une arité différente). Les résultats obtenus sont similaires.

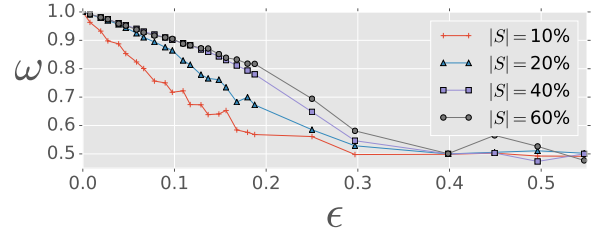


FIGURE 1 – Valeur de ω pour des fonctions ϵ -proches de L .

	ω_S	ω_K	β_S	β_K
Monk 1	.96	.96	.73	.62
Monk 2	.96	.84	.69	.60
Monk 3	.98	.95	.87	.77

TABLE 3 – ω et β pour les modèles Standard et Klein sur les problèmes Monks.

Notons qu'il suffit de considérer $\epsilon \in [0, \frac{1}{2}]$, car si g est ϵ -proche de L , $\neg g$ est $(1 - \epsilon)$ -proche de L et les 2 fonctions ont donc le même ω , ainsi les courbes sont symétriques par rapport à l'axe $\epsilon = \frac{1}{2}$.

Quand $\epsilon = 0$, on obtient $\omega = 1$, comme attendu par la Proposition 3. On observe une décroissance presque linéaire en ω quand ϵ varie de 0.3 – 0.4 puis conduisant à un plateau où $\omega = \frac{1}{2}$, indiquant que les labels analogiques $\bar{\mathbf{x}}_f$ sont plus ou moins aléatoires. De plus, ω décroît plus vite pour de petits échantillons S .

Cela est dû au fait que les labels $\bar{\mathbf{x}}_f$ résultent d'un vote majoritaire et que le nombre de candidats devient plus petit quand $|S|$ décroît, altérant ainsi la qualité de la prédiction. Reste à déterminer une dépendance fonctionnelle entre ω , ϵ et $|S|$.

Maintenant, notons que, même si la fonction f est loin d'être PA, la qualité ω de l'extension $\mathbf{E}_S(f)$ peut rester haute.

Pour illustrer ce fait, définissons la valeur β qui sera un indicateur de combien f est loin d'être parfaitement PA. Pour chaque $\mathbf{x} \in \mathbf{E}_S^*(f)$, on définit $\beta_{\mathbf{x}}$ comme la proportion de candidats y menant au label correct, i.e. la proportion de y tels que $y = f(\mathbf{x})$. β est définie comme la moyenne de tous les $\beta_{\mathbf{x}}$. Clairement, la fonction f est PA si et seulement si $\beta = 1$ pour tout S , i.e. si $\beta_{\mathbf{x}} = 1$ pour tout $\mathbf{x} \in \mathbf{E}_S^*(f)$ et pour tout S .

La Table 3 donne les valeurs de ω et β pour les modèles Standard et Klein (respectivement $\omega_S, \omega_K, \beta_S$ et β_K) sur 3 jeux de données du dépôt UCI, précisément les 3 problèmes 'Monk'³ [20]. Les résultats sont une moyenne sur 100 expériences, où S est à chaque fois choisi aléatoirement comme 30% de l'univers X .

On observe que sur chaque jeu de données, β_S est significativement plus petit que 1. Cela suggère que les fonctions

3. Comme ils sont nominaux, nous les binarisons.

sous jacentes sont hautement non PA, car en moyenne, il y a une grande proportion (aux alentours de 20%) de candidats y qui prédisent le mauvais label. Cependant, ω_S n'est jamais plus petit que 96%, ce qui génère des extensions de bonne qualité. C'est là que le vote majoritaire entre en jeu : dans certains cas, ce vote compense les fausses prédictions. C'est ce qui se produit dans 96%, 96% et 98% des cas respectivement.

Ici encore, nous cherchons à obtenir des garanties théoriques sur cette procédure de vote majoritaire.

Notons aussi que le modèle de Klein produit des résultats de la même qualité ou de qualité inférieure au modèle standard. Cela suggère que le modèle standard, qui définit la même classe de fonctions PA, est plus utile en pratique. Notons que la différence entre ω_S et ω_K a été observée de manière consistante au travers de diverses expériences que nous ne mentionnons pas ici faute de place. Cela pourrait s'expliquer par le fait que le modèle de Klein satisfait une propriété peu attendue d'une analogie : $A_K(a, b, c, d) \iff A_K(b, a, c, d)$.

6 Conclusion et pistes futures

Dans ce papier, nous avons démontré l'intérêt de l'usage des proportions analogiques pour étendre un échantillon de valeurs booléennes. Nous avons introduit la notion de fonctions PA, qui sont les fonctions pour lesquelles nous sommes sûrs d'obtenir une extension sans erreur. Après avoir identifié la classe des fonctions PA comme étant celle des fonctions linéaires, nous avons discuté comment un tel résultat théorique pourrait être utile dans les cas réels.

Nous avons étudié 2 manières de dévier du cas PA. Premièrement, en étudiant les fonctions approximativement linéaires et en observant les variations dans la qualité de l'extension analogique.

Deuxièmement, en étudiant des ensembles de données dont la fonction sous jacente est clairement non linéaire mais pour lesquels les classifieurs analogiques sont performants. Ces 2 investigations empiriques suggèrent que nous devons rechercher des garanties théoriques sur la qualité de l'extension analogique dans ces 2 contextes.

Il est aussi clair que ce type de résultat doit aussi être recherché pour des attributs nominaux (et pas seulement booléens.)

Acknowledgements. Ce travail bénéficie du soutien du Labex ANR-11-LABX-0040-CIMI (Centre International de Mathématiques et d'Informatique) dans le cadre du programme ANR-11-IDEX-0002-02, projet ISIPA.

Références

[1] Bayouhd, S., L. Miclet et A. Delhay: *Learning by*

analogy : A classification rule for binary and nominal data. pages 678–683, 2007.

- [2] Bayouhd, S., H. Mouchère, L. Miclet et E. Anquetil: *Learning a Classifier with Very Few Examples : Analogy Based and Knowledge Based Generation of New Examples for Character Recognition.* Dans *Proc. 18th Europ. Conf. on Machine Learning (ECML'07)*, pages 527–534. Springer-Verlag, 2007.
- [3] Blum, M., M. Luby et R. Rubinfeld: *Self-testing/Correcting with Applications to Numerical Problems.* *J. Comput. Syst. Sci.*, 47(3) :549–595, 1993, ISSN 0022-0000.
- [4] Bounhas, M., H. Prade et G. Richard: *Analogical classification : A new way to deal with examples.* Dans *Proc. 21st Europ. Conf. on Artificial Intelligence (ECAI'14)*, pages 135–140. IOS Press, 2014.
- [5] Bounhas, M., H. Prade et G. Richard: *Not being at odds with a class : A new way of exploiting neighbors for classification.* Dans *Proc. 22nd Europ. Conf. on Artificial Intelligence (ECAI'16)*, tome 285, pages 1662–1663. IOS Press, 2016.
- [6] Couceiro, M., N. Hug, H. Prade et G. Richard: *Analogy-preserving functions : A way to extend Boolean samples.* Dans *Proc. 26th Int. J. Conf. Artif. Intel. IJCAI'17, Melbourne*, 2017.
- [7] Couceiro, M. et E. Lehtonen: *Generalizations of Swierczkowski's lemma and the arity gap of finite functions.* *Discrete Math.*, 309(20) :5905–5912, 2009.
- [8] Couceiro, M. et M. Pouzet: *On a quasi-ordering on Boolean functions.* *Theor. Comput. Sci.*, 396(1-3) :71–87, 2008.
- [9] Dastani, M., B. Indurkha et R. Scha: *Analogical projection in pattern perception.* *J. of Experimental and Theoretical Artificial Intelligence*, 15(4) :489–511, 2003.
- [10] Davies, T. R. et S. J. Russell: *A logical approach to reasoning by analogy.* Dans McDermott, J. P. (éditeur) : *Proc. 10th Int. Joint Conf. on Artificial Intelligence (IJCAI'87)*, pages 264–270. Morgan Kaufmann, 1987.
- [11] Dorolle, M.: *Le Raisonnement par Analogie.* PUF, Paris, 1949.
- [12] Fei-Fei Li, R. Fergus et P. Perona: *One-shot learning of object categories.* *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(4) :594–611, 2006.
- [13] Gentner, D.: *Structure-Mapping : A Theoretical Framework for Analogy.* *Cognitive Science*, 7(2) :155–170, 1983.
- [14] Gentner, D., K. J. Holyoak et B. N. Kokinov (éditeurs): *The Analogical Mind : Perspectives from Cog-*

- nitive Science*. Cognitive Science, and Philosophy. MIT Press, Cambridge, MA, 2001.
- [15] Gust, H., K. U. Kühnberger et U. Schmid: *Metaphors and heuristic-driven theory projection (HDTP)*. Theoretical Computer Science, 354(1) :98 – 117, 2006.
- [16] Hug, N., H. Prade, G. Richard et M. Serrurier: *Analogical classifiers : A theoretical perspective*. Dans *Proc. 22nd Europ. Conf. on Artificial Intelligence (ECAI'16)*, pages 689–697. IOS Press, 2016.
- [17] Klein, S.: *Analogy and mysticism and the structure of culture (and Comments & Reply)*. Current Anthropology, 24 (2) :151–180, 1983.
- [18] Lepage, Y.: *Analogy and formal languages*. Electr. Notes Theor. Comput. Sci., 53, 2001.
- [19] Lepage, Y.: *De l'analogie rendant compte de la commutation en linguistique*. Habilit. à Diriger des Recher., Univ. J. Fourier, Grenoble, 2003.
- [20] Lichman, M.: *UCI Machine Learning Repository*, 2013. <http://archive.ics.uci.edu/ml>.
- [21] Miclet, L., S. Bayouhd et A. Delhay: *Analogical Dissimilarity : Definition, Algorithms and Two Experiments in Machine Learning*. J. Artif. Intell. Res. (JAIR), 32 :793–824, 2008.
- [22] Miclet, L. et A. Delhay: *Relation d'analogie et distance sur un alphabet défini par des traits*. Rapport technique 1632, IRISA, July 2004.
- [23] Miclet, L. et H. Prade: *Handling analogical proportions in classical logic and fuzzy logics settings*. Dans *Proc. 10th Eur. Conf. on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'09)*, pages 638–650. Springer, LNCS 5590, 2009.
- [24] Prade, H. et G. Richard: *Boolean analogical proportions - Axiomatics and algorithmic complexity issues*. Dans Antonucci, A., L. Cholvy et O. Papini (rédacteurs) : *Proc. 14th Eur. Conf. on Symb. and Quantit. Appr. to Reasoning with Uncertainty (ECSQARU'17)*, Lugano, Jul. 7-10, LNAI 10369. Springer, 2012.
- [25] Prade, H. et G. Richard: *From Analogical Proportion to Logical Proportions*. Logica Universalis, 7(4) :441–505, 2013.
- [26] Prade, H. et G. Richard: *A discussion of analogical-proportion based inference*. Dans *Proc. 2nd Computational Analogy Workshop at ICCBR-17, Trondheim*, 2017.
- [27] Rumelhart, D. E. et A. A. Abrahamson: *A model for analogical reasoning*. Cognitive Psychol., 5 :1–28, 2005.
- [28] Salomaa, A.: *On essential variables of functions, especially in the algebra of logic*. Ann. Acad. Sci. Fenn. Ser. A I. Math., 339 :3–11, 1963.
- [29] Stone, M. H.: *The Theory of Representation for Boolean Algebras*. Trans. of the American Mathematical Society, 40(1) :37–111, 1936.
- [30] Stroppa, N. et F. Yvon: *Analogical Learning and Formal Proportions : Definitions and Methodological Issues*. Rapport technique D004, ENST-Paris, 2005.
- [31] Willard, R.: *Essential arities of term operations in finite algebras*. Discrete Math., 149(1-3) :239–259, 1996.
- [32] Wolf, L. et I. Martin: *Regularization Through Feature Knock Out*. MIT Computer Science and Artificial Intelligence Laboratory, (CBCL Memo 242), 2004.
- [33] Zhegalkin, I. I.: *On the Technique of Calculating Propositions in Symbolic Logic*. Mat. Sb, 43 :9–28, 1927.

Réseau bayésien d'un critère de décision dans un problème de décision répété et multicritères

Véronique Delcroix¹

¹ LAMIH, UMR 8201, UVHC, Valenciennes, France

veronique.delcroix@univ-valenciennes.fr

Résumé

Cet article concerne les problèmes répétés de décision multi-critère. Il s'agit plus précisément des problèmes de choix, tel que le choix d'une voiture, d'un fauteuil roulant, ou le choix d'études post-bac, qui se répètent fréquemment, avec des acteurs et dans des contextes différents, mais toujours autour du même type d'alternatives. Dans ces problèmes, les alternatives sont décrites par un ensemble d'attributs qui reste fixe. Chaque nouveau cas de décision correspond à un ensemble de facteurs concernant les acteurs de la décision, le contexte, l'environnement, les contraintes et les niveaux d'importance des critères. Cet article identifie deux types de critères de décision et propose pour chacun un modèle local défini par un réseau bayésien. Le réseau bayésien obtenu pour un problème répété de décision multi-critères peut être utilisé pour n'importe quel cas de décision. Il permet d'une part d'évaluer une alternative en fonction du cas de décision et d'autre part, de fournir des conseils sur les attributs des alternatives qui soient bien adaptés aux caractéristiques du cas de décision.

Abstract

A *repeated*¹ multi-criteria decision problem is a choice problem that occurs frequently, regarding always the same kind of alternatives, but with different actors and context. Examples of such problems are the choice of a car, or a wheelchair, or studies continuation. In those problems, the alternatives are described by a set of attributes that remain fix; each new decision case includes information about the actors of the decision, the context, the environment, and the values of constraints and importance of criteria. This paper identifies two kinds of decision criteria and proposes a specific local structure of a Bayesian network for each of

them. The obtained Bayesian network obtained for a repeated multi-criteria decision problem can be used for any decision case. First, it allows the evaluation of an alternative according to the criteria and the decision case. Second, it provides guidance about the attributes of alternatives that best suit to the decision case characteristics.

1 Introduction

Les problèmes de décision multi-critères sont fréquents dans la vie quotidienne : acheter des chaussures, tout comme se poser la question "qu'est-ce qu'on mange ce soir ?" sont des décisions qui dépendent de plusieurs critères éventuellement conflictuels, et de différents paramètres concernant le contexte, les acteurs, etc. Répondre à ces questions nécessite de combiner ces éléments pour trouver un compromis satisfaisant. Ce type de problème se répète de nombreuses fois et chaque nouveau cas de décision implique des acteurs différents, dans des contextes différents. Les réponses sont donc différentes à chaque fois, à cause de ces différents *cas de décision*. C'est ce que nous appelons un problème de décision multi-critères *répété*. L'aide à la décision dans ce type de problème doit prendre en compte les caractéristiques des alternatives, les évaluations des alternatives suivant chacun des critères, ainsi que les paramètres de la situation de décision.

Dans la littérature, les termes *problèmes de décision répétés* font souvent référence à des situations où un décideur unique se trouve face plusieurs fois à des décisions de même type. Dans certains cas, ces décisions sont toutes faites sur une même période de temps, en vue d'atteindre un objectif, comme par exemple la recherche de prix [11]. Dans d'autres cas, une même personne se trouve plusieurs fois face au même type de décision, mais à des périodes différentes. Dans ce cas, elle peut utiliser l'expérience acquise lors des décisions passées. C'est le cas pour les inves-

1. Since this problem seems never to be addressed in the literature, there is no usual term to define it. In previous papers[8], we proposed the term *recurrent* multi-criteria decision problem; further to several remarks, we prefer now the term *repeated*, despite it is mostly used to refer to decision problems where a decision maker has to deal several times with the same kind of decision

tisseurs [13]. Les problèmes de décision répétés sont aussi considérés dans la théorie des jeux, où le joueur doit choisir une action à chaque itération du jeu. Nous avons trouvé un autre exemple de décision répété en économie, mettant en jeu différents acteurs dans chaque situation [14], mais il ne s'agit pas d'aide à la décision multi-critères.

Dans le présent article, il est question d'aide à la décision pour les problèmes de décision répétés, multicritères et multi-attributs. Notre objectif est d'apporter une aide appropriée aux acteurs du problème, telle que l'évaluation d'une alternative sur chacun des critères en fonction du cas de décision, ou certains conseils sur les valeurs des caractéristiques de l'alternative. Pour cela, l'idée est d'embarquer dans un modèle unique l'ensemble des éléments du raisonnement qui mène à un bon choix en fonction de la situation. Un tel modèle contient les éléments de connaissance suivant : (1) les caractéristiques directes et dérivées des alternatives, et leurs interactions ; (2) les paramètres qui peuvent influencer sur le choix, y compris les facteurs externes et la situation de décision ; (3) la façon dont une alternative est évaluée suivant chaque critère, en prenant en compte les caractéristiques du cas de décision qui influence cette évaluation.

Ces éléments de connaissance comportent une part d'incertitude due à l'ignorance, et une part de variabilité naturelle des éléments considérés. Nous proposons d'utiliser des modèles graphiques probabilistes, et en particulier des réseaux bayésiens, car ils sont des modèles de connaissances qui intègrent l'incertitude. Cet article concerne la construction du réseau bayésien pour des problèmes répétés de décision multicritères et plus précisément la modélisation des critères de décision. Nous identifions deux types de critères, objectifs et subjectifs, et nous proposons pour chacun une structure locale dans un réseau bayésien. Un *critère objectif* est associé à une caractéristique de l'alternative dont l'évaluation est objective et connue des acteurs de la décision. Un tel critère est évalué par un *degré d'adéquation* entre la valeur de la caractéristique de l'alternative et la variable correspondant du côté du décideur. Un *critère subjectif* est associé à un concept d'un degré plus élevé ; il est évalué par un *niveau de satisfaction* qui dépend de l'évaluation de l'alternative suivant cet axe de signification et de son importance du point de vue du décideur.

Cet article commence par introduire les problèmes de décision répétés et multicritères, et les compare avec ceux habituellement considérés dans l'aide à la décision multicritères (ADMC). La partie 3 propose une classification des critères en deux types dans le cadre des problèmes de décisions répétés et multi-critères, et décrit le graphe local du réseau bayésien associé. La partie 4 présente deux utilisations de ce modèle : l'évaluation et le conseil. Les deux dernières parties sont consacrées aux travaux connexes, la conclusion et les perspectives.

2 Problèmes de décision répétés et multi-critères

Un problème de décision répété, multi-critères et multi-attributs est défini par un ensemble d'attributs qui caractérisent les alternatives et un ensemble de critères. Une alternative est définie par une affectation complète de l'ensemble des attributs. Le problème consiste à choisir une alternative (ou d'un sous ensemble d'alternatives) qui soit satisfaisante pour l'ensemble des critères. Un problème de décision est dit *répété* s'il se produit souvent, à chaque fois dans une situation différente et avec des acteurs différents. Un *cas de décision* (ou une situation de décision) est défini par l'ensemble des caractéristiques des acteurs, du contexte et de l'environnement qui peuvent avoir une influence sur le choix. Dans un problème répété de décision multi-critères, l'ensemble des attributs et l'ensemble des critères sont les mêmes pour tous les cas de décision, alors que l'importance des critères et les valeurs des caractéristiques du cas de décision varient.

Les problèmes de décision répétés et multi-critères présentent plusieurs différences par rapport à ceux habituellement traités par les approches d'ADMC. Les méthodes d'ADMC traitent généralement un problème unique (non répété) et reposent sur l'hypothèse que le décideur est fortement impliqué dans le processus de décision, et qu'il est guidé par un analyste. A l'inverse, puisque les problèmes de décision répétés concernent un large panels de "décideurs", nous considérons la possibilité que le décideur soit relativement "ignorant" ou peu impliqué dans la décision, ou simplement pressé, et nous préférons parler des *acteurs* de la décision. Par suite, il n'est pas pertinent de questionner ces acteurs sur leurs préférences, alors que la plupart des méthodes d'ADMC sont basées sur des comparaisons d'alternatives par paires, suivant chacun des critères.

Critères et attributs des alternatives. On peut définir un critère comme un "outil" permettant de d'évaluer et de comparer des alternatives suivant un axe de signification particulier, ou un point de vue [16]. Plus précisément, un critère est une fonction réelle g sur l'ensemble des alternatives, telle qu'il soit possible de comparer deux alternatives a et b suivant un certain point de vue, sur la seule base des nombres $g(a)$ et $g(b)$ [4]. Dans notre contexte, l'évaluation d'une alternative suivant un critère dépend du cas de décision. Par exemple, lors du choix d'une voiture ou d'un fauteuil roulant, ce qui convient pour une personne n'est pas adapté pour une autre. Dans cet article, un critère est un moyen d'évaluer une alternative² suivant un axe de signification en fonction d'un cas de décision. Nous considérons le problème de décision à partir du tout début du processus de décision, à un stade où aucun sous-ensemble d'alternatives n'a encore été pré-défini pour un cas de décision. A ce

2. ou un sous-ensemble d'alternatives décrites par les valeurs d'un sous-ensemble des attributs

stade initial, les alternatives sont potentiellement définies grâce au domaine de l'ensemble des attributs. A l'inverse, un problème typique d'ADMC est défini par une matrice de décision qui rassemble les performances de n alternatives pré-sélectionnées suivant les m critères, accompagnée des poids des critères. Ainsi, la matrice de décision est un point de départ dans les approches classiques d'ADMC, alors que dans notre approche, les recommandations obtenues peuvent aider les acteurs à obtenir cette matrice.

L'objectif de notre proposition est de fournir une forme d'aide aux acteurs de la décision à un stade très préliminaire, en se basant sur la seule description du cas de décision (Quoi, Qui, Où, Quand, Comment, Combien, Pourquoi, ...). Pour cela, nous proposons la construction d'un réseau bayésien unique pour un problème de décision répété et multi-critères, composé des modèles de chacun des critères. Ce réseau bayésien est ensuite réutilisé pour chaque nouveau cas de décision.

3 La modélisation d'un critère de décision dans un réseau bayésien

Dans cette partie, nous commençons par rappeler les définitions de base concernant les réseaux bayésiens ; ensuite, nous présentons les *critères objectifs* qui sont évalués par un degré d'adéquation entre un attribut d'une alternative et la variable correspondant du cas de décision ; enfin, nous présentons les *critères subjectifs*, qui sont évalués par le niveau de satisfaction donné par une alternative sur un axe de signification et en tenant compte du cas de décision. Pour chacun de ces deux types de critères, nous proposons une structure locale du graphe d'un réseau bayésien, et nous donnons différents exemples illustratifs.

3.1 Réseau Bayésien

Un réseau bayésien est défini par un ensemble de variables aléatoires $\mathbf{X} = X_1, \dots, X_n$, un graphe orienté sans circuits $G = (\mathbf{X}, \mathbf{E})$, et un ensemble de paramètres Θ contenant les distributions de probabilités locales associées à chaque variables de \mathbf{X} . Ensemble, G et Θ définissent une distribution de probabilités P sur \mathbf{X} qui se factorise comme suit : $P(\mathbf{X} = \mathbf{x}) = \prod_{X_i \in \mathbf{X}} P(X_i = x_i \mid pa(X_i))$, où $pa(X_i) \subset \mathbf{X}$ représente les parents de X_i dans le graphe G .

Nous considérons un réseau bayésien *causal*, ce qui signifie que chaque arc orienté représente une influence causale directe entre une cause et un effet. Pour satisfaire cette définition, un réseau bayésien causal doit nécessairement inclure toutes les causes communes de deux noeuds quelconques du réseau³. Cette propriété est utile

3. En effet, lorsque deux noeuds X et Y ont une cause commune, ils sont dépendants. Si leur cause commune n'appartient pas à l'ensemble des variables du graphe, la dépendance entre les variables X et Y ne peut pas

pour construire le réseau bayésien regroupant tous les critères du problème de décision répétés.

Dans cet article, l'ensemble des variables du réseau bayésien d'un problème répété de décision multi-critères contient les attributs des alternatives, ainsi que toutes les caractéristiques du cas de décision qui doivent être considérées. Il contient aussi les attributs dérivés ou composés à partir d'autres caractéristiques. Par exemple, la capacité d'accélération d'une voiture dépend de son poids et de la puissance du moteur. Ceci est représenté dans le réseau bayésien par trois variables et deux liens causaux. Le réseau bayésien contient aussi d'autres variables associées aux critères qui seront présentées dans la suite.

Dans la suite de cet article, nous utilisons une structure particulière du graphe appelée *V-structure* : trois noeuds X , Y et Z forment une *V-structure* dans un graphe orienté si X et Y ont un fils commun Z et il n'y a pas d'arc entre X et Y .

Voici les notations utilisées dans cet article : les lettres en majuscule représentent des variables aléatoires, les minuscule représentent leurs valeurs. Les majuscules en gras représentent des ensembles de variables et les minuscules en gras désignent des ensembles de valeurs. La notation $P(x)$ représente $P(X = x)$, et $P(X \mid y)$ représente $P(X \mid Y = y)$.

Les exemples et les résultats présentés dans cet article ont été réalisés avec la bibliothèque Agrum / pyAgrum⁴, qui est dédiée aux modèles graphiques probabilistes, dont les réseaux bayésiens. Nous avons choisi pyAgrum et l'environnement des notebook Ipython car ils facilitent l'utilisation de la bibliothèque en C++ Agrum notamment en termes d'affichage. Nous avons beaucoup apprécié l'aide rapide des auteurs de la bibliothèque⁵ à propos des fonctions et algorithmes utilisés.

3.2 Définition des critères objectifs et subjectifs

Lorsqu'on parle de critère de décision, il est fréquent de confondre un attribut d'une alternative et le critère associé. L'exemple le plus courant est le prix qui fait référence à la fois à la caractéristique d'une alternative et au critère correspondant. Cependant, il existe un autre type de critères de décision qui font référence à des concepts plus abstraits tels que le confort, la sécurité et la stabilité, pour lesquels il n'existe aucune définition de leur évaluation qui soit communément partagée.

Le premier type de critère, appelé critère objectif, est caractérisé comme suit :

- le critère est associé à un attribut X_{alt} de l'alternative⁶ dont l'évaluation est objective

être représentée par un arc entre elles puisque aucune variable n'est une cause de l'autre. L'absence d'arc n'est pas non plus correcte puisqu'elle est interprétée comme une indépendance entre X et Y .

4. <https://pourge.lip6.fr/projects/pyagrum>

5. un grand merci à Pierre-Henri Wuillemin.

6. caractéristique directe ou combinée.

- il existe une variable X_{dc} du coté du cas de décision qui doit correspondre en un certain sens avec l'attribut X_{alt} de l'alternative. La variable X_{dc} peut représenter la valeur préférée de l'attribut, ou une sorte de contrainte.
- les deux variables X_{dc} et X_{alt} sont exprimées en utilisant la même unité ou un cadre de référence commun et connu par les acteurs de la décision.

Un critère objectif est évalué par un *degré d'adéquation* DoM_X ⁷ entre un attribut X_{alt} de l'alternative et la variable X_{dc} correspondant du cas de décision. Dans cette définition, correspondre ne signifie pas "avoir la même valeur ou la même distribution de probabilités", cela signifie que les deux variables doivent correspondre suivant une certaine règle, définie par un expert, et qui peut dépendre d'autres paramètres tel que des facteurs externes, d'autres caractéristiques du cas de décision, ou d'autres attributs de l'alternative.

Le second type de critères est appelé *critère subjectif*. Il concerne des éléments de plus haut niveau pour lesquels il n'y a pas de variable correspondant du coté du cas de décision. Un critère subjectif peut être associé avec un concept abstrait qui n'est pas facilement mesurable. Il peut aussi être associé avec un élément complexe, ou technique, dont la mesure n'est pas connue par les acteurs de la décision. Un tel critère est subjectif du point de vue des acteurs de la décision puisqu'il leur est difficile d'exprimer leur valeur préférée ou une contrainte dans la même unité que l'attribut de l'alternative. Par exemple la capacité d'accélération d'une voiture est associée avec un critère subjectif car elle est évaluée par le ratio *carWeight / horsePower*, (voir la Figure 4) et cette grandeur ne permet généralement pas à un utilisateur de formuler aisément son exigence concernant la capacité de accélération de la voiture. De ce fait, la variable correspondant à son attente sur ce critère est une variable subjective. Cependant, l'acheteur d'une voiture peut donner l'importance que ce critère a pour lui.

Un critère subjectif C est évalué par le *niveau de satisfaction* SAT_C fourni par une alternative en fonction de l'évaluation de l'alternative IND_C pour le critère subjectif et de l'importance du critère IMP_C dans le cas de décision. L'évaluation du niveau de satisfaction d'un critère subjectif est tel que : (1) quand une alternative présente une très bonne qualité (ou évaluation) pour le critère, la satisfaction est très bonne quelle que soit le niveau d'importance du critère ; (2) plus la qualité de alternative est élevée, plus le niveau de satisfaction est élevé, mais plus le niveau d'importance du critère est bas, et plus facilement les acteurs de la décision seront satisfaits ; (3) quand le critère est indifférent pour un cas de décision, le niveau de satisfaction est élevé quelle que soit la qualité de l'alternative pour ce critère. D'après ces exigences, le niveau de satisfaction peut

être défini par la fonction $SAT_C = IND_C^{IMP_C}$.

3.3 Construction du réseau bayésien d'un problème répété de décision multi-critères

Pour un problème répété de décision multi-critères, nous voulons construire un réseau bayésien regroupant les connaissances en lien avec le choix d'une alternative, quelle que soit la situation de décision. Le réseau bayésien est le même pour tous les cas de décision. Pour cela, le graphe du réseau bayésien contient toutes les variables liées aux éléments impliqués dans le choix : la connaissance sur les alternatives, les paramètres d'une situation de décision et les critères d'évaluation. Ce modèle permet la simulation et le raisonnement sur le problème de décision avec différentes entrées possibles : des informations sur un cas de décision donné, des informations sur une alternative, et / ou des exigences liées aux critères.

Nous présentons maintenant les sous-graphes associés à un critère objectif et à un critère subjectif. Les exemples de critères sélectionnés pour illustrer ces modèles sont extraits de deux problèmes de décision différents répétés et multi-critères : le choix d'une voiture et le choix d'un fauteuil roulant manuel (FRM) [17, 8].

Un sous-graphe du réseau bayésien associé au problème du choix d'une voiture est présenté Figure 6. Il rassemble les sous-graphes de deux critères et plusieurs caractéristiques d'une voiture. Le sous-graphe de chaque critère est expliqué en détail avec les figures 1 et 4.

Dans ce type de problème, les acteurs de la décision peuvent être relativement ignorants des éléments du problème et des critères. Ils sont supposés être capable de décrire leur situation, mais c'est la seule exigence sur eux. Ils ne sont pas supposés avoir déjà fait une pré-sélection d'alternatives, ni avoir à répondre à des questions sur leur préférences entre une alternative et une autre en relativement à un critère.

3.3.1 Réseau bayésien local d'un critère objectif

Le degré d'adéquation d'un critère objectif est représenté dans le réseau bayésien par une variable dont le domaine est l'intervalle $[0, 1]$. Une valeur proche de zéro représente un "très mauvais" degré d'adéquation, alors qu'une valeur proche de un représente un "très bonne" adéquation. L'évaluation du degré d'adéquation nécessite un ensemble de variables qui est exactement l'ensemble des parents du noeud *degré d'adéquation* dans le graphe du réseau bayésien.

Un critère objectif est représenté dans réseau bayésien causal par une V-structure composée de au moins trois noeuds : une caractéristique X_{alt} d'une alternative, la caractéristique X_{dc} du cas de décision qui doit "correspondre" avec X_{alt} , et le degré d'adéquation (DoM_X) (leur fils commun). le noeud DoM_X peut avoir d'autres parents qui

7. degree of match

représentent les paramètres de la règle utilisée pour évaluer le degré d'adéquation. Ces paramètres peuvent être des attributs du cas de décision, d'autres attributs des alternatives, ou des facteurs externes. La table de probabilités conditionnelles de la variable DoM_X traduit la façon dont le degré d'adéquation est évalué.

Trois exemples de critères objectifs sont décrits ci-dessous dans les figures 1, 2 et 3. Le premier critère est tiré du problème répété du choix d'une voiture, alors que les deux exemples suivants sont extraits du problème du choix d'un fauteuil roulant manuel. Dans chaque exemple, nous décrivons le critère objectif et les variables impliquées dans l'évaluation du degré d'adéquation. Pour chaque critère, nous présentons seulement le sous-graphe du réseau bayésien local. Les valeurs numériques des tables de probabilités conditionnelles ne sont pas détaillées car cela n'apporte rien. En revanche, les résultats des différents cas d'utilisations du réseau bayésien sont présentés dans la dernière partie de cet article.

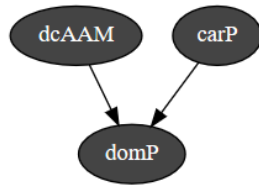


FIGURE 1 – Sous-graphe d'un critère objectif : le prix, dans le problème répété du choix d'une voiture. Ce critère est évalué par le degré d'adéquation ($domP$) entre le prix de la voiture ($carP$) et le montant disponible du cas de décision ($dcAAM$).

La figure 1 montre le sous-graphe associé au critère du prix lors du choix d'une voiture. Un bon choix de voiture implique que le prix de la voiture soit en adéquation avec le budget de l'acheteur. Nous ne partageons que partiellement l'idée que "le moins cher sera le mieux (en terme de prix)", car un bon prix dans une situation peut être inacceptable dans une autre situation, puisque les acheteurs ont des budgets différents et des attentes différentes. La fonction probabiliste pour évaluer le degré d'adéquation ($domP$) en termes de prix doit être définie par des experts de la vente de voiture.

La figure 2 montre le sous-graphe du critère associé avec la largeur du siège dans le problème du choix d'un Fauteuil Roulant Manuel (FRM). La meilleure largeur du siège dépend de la largeur de hanche de la personne, mais aussi de l'utilisation du FRM. En général, la meilleure situation est d'avoir un siège 2 cm plus large que les hanches de la personne. Cependant, lorsque le FRM est utilisé pour la pratique d'un sport, la personne doit être serrée dans le FRM, et un écart de -1 cm est recommandé. Enfin, pour les

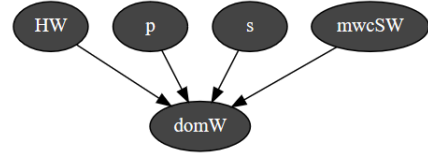


FIGURE 2 – Sous-graphe d'un critère objectif : la largeur du siège dans le problème du choix d'un Fauteuil Roulant Manuel (FRM). Ce critère est évalué par le degré d'adéquation ($domW$) entre la largeur du siège ($mwcSW$) et la largeur de hanche de la personne (HW); il dépend de deux autres facteurs : si la personne est (ou risque d'être) enceinte (p) ou si le FRM doit être utilisé pour la pratique d'un sport (s).

femmes enceintes, un écart supérieur est nécessaire. Ces règles aboutissent à définir quatre variables comme parents de la variable degré d'adéquation $domW$. La définition de la table de probabilités conditionnelles du nœud $domW$ est guidée par ces règles.

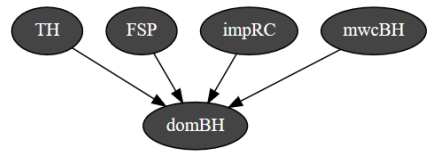


FIGURE 3 – Sous-graphe d'un critère objectif : la hauteur du dossier dans le problème du choix d'un FRM. Ce critère est évalué par le degré d'adéquation ($domBH$) entre la hauteur du dossier ($mwcBH$) et la hauteur de tronc de la personne (TH). Il dépend de deux autres facteurs : l'importance du confort de repos ($impRC$) et la fréquence d'auto-propulsion (FSP).

La figure 3 montre le sous-graphe du critère associé avec la hauteur de dossier lors du choix d'un FRM. La meilleure hauteur de dossier d'un FRM dépend de son usage. Lorsque le FRM est principalement dédié au repos (le confort de repos est très important), le dossier doit être assez haut pour que la personne puisse y reposer sa tête. À l'inverse, lorsque la fréquence d'auto-propulsion est élevée, le dossier doit être assez bas pour que les omoplates de la personne passent au-dessus du dossier lorsque la personne met ses coudes en arrière. Ainsi, la règle d'évaluation du degré d'adéquation ($domBH$) est une fonction probabiliste qui dépend de quatre variables : la hauteur de dossier du FRM ($mwcBH$), la hauteur de tronc de la personne (TH), l'importance du confort de repos ($impRC$) et la fréquence d'auto-propulsion (FSP).

3.3.2 Le réseau bayésien local d'un critère subjectif

Le niveau de satisfaction d'un critère subjectif est représenté dans le réseau bayésien par un indice dans l'in-

tervalle $[0, 1]$, où une valeur proche de zéro représente un “très mauvais” niveau de satisfaction alors qu’une valeur proche de un représente un “très bon” niveau de satisfaction. Le niveau de satisfaction SAT_C d’un critère C dépend de deux variables exactement : l’importance du critère (IMP_C) dans le cas de décision et l’indice d’évaluation du critère (IND_C), qui est l’évaluation d’une alternative suivant le critère subjectif. Du fait que l’importance d’un critère est une valeur subjective, la variable IMP_C est aussi définie par un indice dans l’intervalle $[0,1]$ où une valeur proche de zéro signifie que le critère n’est “pas important” alors qu’une valeur proche de un signifie que le critère est “très important”. La variable IND_C est aussi définie par un indice dans l’intervalle $[0,1]$ où une valeur proche de zéro représente une “très mauvaise évaluation” suivant le critère, alors qu’une valeur proche de un représente une “très bonne évaluation”.

Un critère subjectif C est représenté dans un réseau bayésien causal par une V-structure composée de trois nœuds IMP_C , IND_C et SAT_C qui dépend des deux premières variables. Contrairement aux critères objectifs, la V-structure d’un critère subjectif est composée de trois nœuds exactement. En effet, tous les paramètres impliqués dans l’évaluation d’une alternative sont parents du nœud IND_C , y compris les facteurs externes et les caractéristiques du cas de décision.

Les figures 4 et 5 donnent des exemples de critères subjectifs. Du fait que la V-structure associée à n’importe quel critère subjectif est toujours identique, nous montrons pour ces exemples un sous-graphe du réseau bayésien qui inclut le détail de l’évaluation d’une alternative sur le critère.

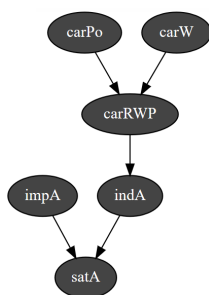


FIGURE 4 – Un critère subjectif : la capacité d’accélération lors du choix d’une voiture. Elle dépend du rapport ($carRWP$) entre le poids de la voiture ($carW$) et sa puissance ($carPo$).

La figure 4 montre le sous-graphe du critère “accélération” lors du choix d’une voiture. Le niveau de satisfaction ($sata$) de l’acheteur d’une voiture pour le critère d’accélération dépend de la capacité d’accélération de la voiture ($indA$) et de l’importance qu’il accorde à ce critère ($impA$). L’évaluation d’une voiture suivant ce critère est donnée par le rapport ($carRWP$) entre le poids de la voiture ($carW$) et sa

puissance (nombre de chevaux) ($carPo$).

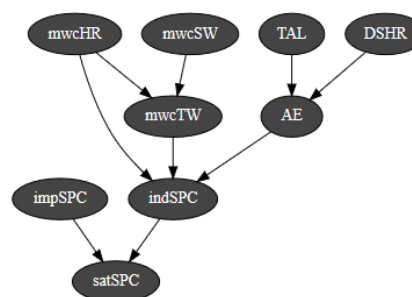


FIGURE 5 – Un critère subjectif : le confort d’auto-propulsion lors du choix d’un fauteuil roulant manuel (FRM). L’évaluation du confort d’auto-propulsion $indSPC$ dépend de l’équipement du FRM avec des mains courantes ($mwcHR$), de la largeur totale du FRM ($mwcTW$) et de l’angle du coude quand la personne tient le sommet de la main courante (AE).

La figure 5 montre le sous-graphe du critère subjectif de confort d’auto-propulsion lors du choix d’un FRM. Le niveau de satisfaction ($satsPC$) donné par un FRM pour le critère de confort d’auto-propulsion dépend de la facilité avec laquelle le FRM peut être auto-propulsé par l’utilisateur ($indSPC$) et de l’importance de ce critère ($impSPC$) pour l’utilisateur. L’auto-propulsion nécessite que le FRM soit équipé avec des mains courantes ($mwcHR$). Le niveau de confort d’auto-propulsion ($indSPC$) augmente quand la largeur totale du FRM ($mwcTW$) diminue et quand l’angle du coude (AE) est entre 100 et 120° au moment où la personne tient le sommet de la main courante.

Cet exemple illustre le fait que l’évaluation d’une alternative suivant un critère subjectif ne dépend pas uniquement des caractéristiques de l’alternative. L’évaluation d’un fauteuil roulant manuel suivant le critère de confort d’auto-propulsion ($indSPC$) dépend de différentes caractéristiques du FRM ($mwcTW$ et $mwcHR$) mais aussi de la personne (AE).

Dans cette partie, nous avons expliqué comment construire les sous-graphes de critères objectifs et subjectifs. En les combinant, on obtient le réseau bayésien d’un problème de décision répété et multi-critères. La partie suivante présente comment utiliser un tel réseau bayésien dans le but de fournir une forme d’aide à la décision.

4 Évaluation et conseils dans un cas de décision

Les deux principaux objectifs de la modélisation d’un problème de décision répété et multi-critères avec un réseau bayésien sont l’évaluation d’une alternative, et la production de conseils concernant les attributs des alternatives,

adaptés au cas de décision.

4.1 Évaluation d’une alternative dans un cas de décision

L’objectif est d’évaluer une alternative, (ou un sous-ensemble d’alternatives) dans un cas de décision donné et suivant chaque critère. L’évaluation est obtenue pour tous les critères par inférence dans le réseau bayésien complet.

Input : Les observations sur un ensemble (ou un sous-ensemble) des attributs des alternatives à évaluer (\mathbf{obs}_{alt}) et les observations sur l’ensemble des caractéristiques du cas de décision (\mathbf{obs}_{dc}). Toutes ces observations peuvent être des observations classiques ou des observations virtuelles (observations de vraisemblance) [15].

Output (variables cibles) : les distributions de probabilités *a posteriori* $P(Dom_X | \mathbf{obs}_{alt}, \mathbf{obs}_{dc})$ du degré d’adéquation d’un critère objectif et *a posteriori* $P(SAT_C | \mathbf{obs}_{alt}, \mathbf{obs}_{dc})$ du niveau de satisfaction d’un critère subjectif.

4.2 Conseil sur les attributs d’une alternative

Un autre objectif est de fournir des conseils⁸ aux acteurs de la décision concernant les attributs d’une alternative qui serait adaptée à leur situation.

L’idée est de simuler une situation idéale où les acteurs d’un cas de décision donné seraient très satisfaits par une alternative sur tous les critères. Dans cette situation idéale, nous cherchons à savoir quelles sont les valeurs des attributs de l’alternative qui donne ces bons niveaux de satisfaction et bons degrés d’adéquation pour tous les critères. Pour cela, nous propageons dans le réseau bayésien les observations du cas de décision et des niveaux de satisfaction et des degrés d’adéquation élevés pour tous les critères.

Notre objectif est d’obtenir les valeurs des attributs de l’alternative qui explique cette situation idéale. Cependant, l’explication doit concerner *uniquement* les attributs de l’alternative (et aucune caractéristique du cas de décision ou des facteurs externes); pour cela, nous entrons des observations sur *tous* les noeuds du réseau bayésien, à l’exception des attributs des alternatives.

Les conseils sur les attributs des alternatives peuvent avoir différentes formes : une alternative virtuelle décrite par une affectation de tous les attributs, ou une distribution de probabilités *a posteriori* sur chaque attribut de l’alternative.

8. Nous préférons parler de *conseils* plutôt que de recommandation car un système de recommandation est généralement vu comme une sous classe des systèmes de filtrage d’information qui cherchent à prédire la note qu’un utilisateur donnerait à un système (Wikipedia), alors que notre proposition n’est basée ni sur les choix précédents du même utilisateur, ni sur aucune données.

Input : les observations de toutes les caractéristiques du cas de décision (\mathbf{obs}_{dc}), ainsi que les niveaux d’importance des critères subjectifs, et l’affectation d’un très bon niveau de satisfaction (\mathbf{sat}_{max}) ou degré d’adéquation (\mathbf{dom}_{max}) pour tous les critères; les seuls noeuds non observés sont tous des attributs des alternatives.

Output : les noeuds cibles sont les attributs des alternatives. On peut calculer la distribution de probabilités *a posteriori* de chaque attribut X de l’alternative dans la situation idéale $P(X | \mathbf{obs}_{dc}, \mathbf{sat}_{max}, \mathbf{dom}_{max})$. Une autre résultat est l’affectation la plus probable de l’ensemble des attributs d’une alternative $argmax_{att_{alt}} P(ATT_{alt} = att_{alt} | obs_{dc}, sat_{Max}, dom_{Max})$ où ATT_{alt} est l’ensemble des attributs d’une alternative, nommé en anglais the “most probable explanation” (MPE). Ce résultat définit une alternative idéale donnée par l’ensemble att_{alt} qui explique le mieux la situation idéale où les critères sont tous bien satisfaits. Il est probable qu’aucune alternative réelle ne corresponde à l’ensemble de valeurs att_{alt} . Cependant, la partie du réseau bayésien composée des caractéristiques d’une alternative intègre les liens de (in)dépendances entre ces attributs, de telle sorte qu’une affectation telle que att_{alt} présente une certaine cohérence.

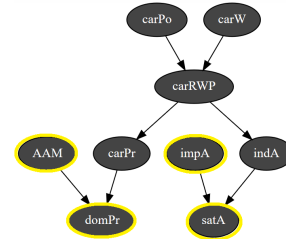


FIGURE 6 – Sous-graphe du réseau bayésien du problème de décision répété et multi-critères du choix d’une voiture, incluant deux critères. Les noeuds avec des observations sont surlignés.

La figure 6 présente le réseau bayésien du problème répété du choix d’une voiture, avec les deux critères présentés sur les figures 1 et 4 (le prix et la capacité d’accélération d’une voiture). Pour obtenir des conseils sur les caractéristiques de la voiture, nous entrons des observations sur les quatre noeuds qui ne sont pas des caractéristiques de la voiture. Ces noeuds sont surlignés sur la figure 6. Nous entrons six jeux d’observations pour ces quatre noeuds, chacun décrivant la situation idéale d’un acheteur imaginaire. Chaque cas de décision est décrit par les observations sur les variables (AAM et $impA$) : la variable *montant disponible* (AAM) prend l’une des 3 valeurs : entre 5 et 10 K€, entre 11 et 15 K€, entre 16 et 20 K€; la variable *importance du critère accélération* ($impA$) prend l’une des deux valeurs : *peu important*, ou *très important*. Les variables *degré d’adéquation pour le prix* ($domPr$) et *niveau de sa-*

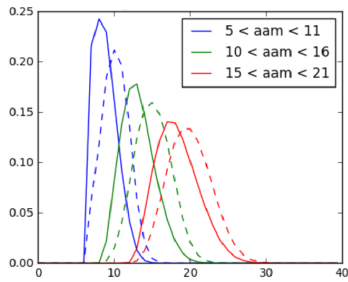


FIGURE 7 – Chaque distribution $P(carPr \mid obs_{dc}, sat_{Max}, dom_{Max})$ exprime une forme de conseil sur le prix d’une voiture dans un des six cas de décision, définis par le montant disponible (aam) et l’importance du critère accélération : peu important (ligne pleine), ou très important (pointillés).

tisfaction pour le critère accélération reçoivent une observation disjonctive qui représente la valeur “très haut”.

La figure 7 montre le prix “recommandé” de la voiture dans chacun des six cas de décision décrits par les observations. Le conseil est donné sous la forme d’une distribution de probabilités *a posteriori* de la variable prix de la voiture ($carPr$) sachant les observations de la situation idéale. Le prix “recommandé” est plus élevé lorsque le montant disponible est élevé et que le critère accélération est très important.

Ces résultats illustrent comment le réseau bayésien peut être utilisé pour produire automatiquement des conseils pour les acteurs d’un problème de décision répété et multi-critères à partir de la description de leur cas de décision. Ces conseils ont pour but de donner des orientations aux acteurs d’une décision en leur indiquant des valeurs (ou des ensembles de valeurs) à préférer pour les attributs de l’alternative, au regard de leur situation. Il ne s’agit pas de proposer directement une alternative.

5 Travaux connexes

La définition d’un système d’aide à la décision unique pour un problème donné de décision répété et multi-critères semble être un problème qui n’a jamais été abordé dans la littérature. Cependant les modèles graphiques probabilistes (MGP) ont déjà été proposés dans l’ADMC. Nous décrivons ci-dessous comment les critères de décision sont représentés dans ces modèles. Nous présentons ensuite d’autres travaux de la littérature qui proposent d’utiliser une V-structure dans un réseau bayésien avec des objectifs et dans des contextes différents. Nous les comparons au réseau bayésien que nous proposons pour modéliser les critères objectifs et subjectifs.

Des MGP sont proposés dans [10, 19, 18] dans le cadre de l’ADMC. Dans [10], un réseau bayésien causal est uti-

lisé pour évaluer une alternative dans le cas des critères incertains, en vue d’obtenir la matrice de décision avant d’utiliser une des méthodes classiques d’ADMC. Chaque critère incertain est une caractéristique de l’alternative qui dépend de facteurs externes et d’autres caractéristiques de l’alternative. La valeur préférée du côté du décideur n’est pas représentée dans le modèle, et l’objectif est pas d’utiliser le modèle pour différents cas de décision. Les noeuds associés avec ces critères peuvent être soit des noeuds cibles en vue d’évaluer une alternative sur ces critères, ou bien ils peuvent être fixés avec la valeur souhaitée, dans le but de fournir des conseils sur les autres caractéristiques. Bien que les exemples traités dans cet article [10] puissent être considérés comme des problèmes de décision répété et multi-critères (par exemple le choix d’un mode de transport et de l’heure de départ pour atteindre l’aéroport), la modèle proposé ne contient pas les caractéristiques du cas de décision (ou du décideur) et de ce fait nécessite de modifier les paramètres du modèle pour chaque nouveau cas de décision.

Dans [19], un autre type de MGP appelé diagramme d’influence⁹ [12] est proposé pour classer les alternatives. Du fait que ce modèle ne contient pas d’information sur le cas de décision, la fonction d’utilité doit être redéfinie pour chaque nouveau cas de décision. Dans [18], un diagramme d’influence permet d’obtenir une liste ordonnée des alternatives qui correspondent bien à un cas de décision et le même modèle peut être utilisé dans différents cas de décision. Cependant, l’utilisation d’un diagramme d’influence présente deux contraintes. Ce modèle est conçu pour faire des choix séquentiels, ce qui signifie que les actions doivent être ordonnées, et que le meilleur choix pour une action n’est obtenu qu’après avoir fait un choix pour l’action précédente. Dans [19], les noeuds d’action du diagramme d’influence représentent les caractéristiques de l’alternative, et ils sont ordonnés. Dans [18], le diagramme d’influence contient un seul noeud d’action dont les actions possibles sont un ensemble prédéfini d’alternatives, ce qui ne correspond pas avec l’approche proposée dans le présent article. Une autre limite des diagrammes d’influence est que la fonction d’utilité est une fonction d’agrégation complète, avec par conséquent l’hypothèse sous-jacente de substituable entre les critères. Cette limite est écartée dans l’extension des diagrammes d’influence proposée dans [9] pour prendre en compte plusieurs critères dans la fonction d’utilité sans les agréger. Cependant, le modèle est limité à de très petits nombres critères et d’alternatives, ce qui n’est pas le cas dans notre approche.

La V-structure proposée dans cet article pour modéliser un critère peut être comparée avec le noeud d’utilité d’un diagramme d’influence. En effet, un noeud d’utilité

9. un diagramme d’influence contient trois types de noeuds : les noeuds classiques, comme ceux d’un réseau bayésien, les noeuds d’action, et les noeuds d’utilité

contient une fonction d'utilité dont les paramètres sont les noeuds parent. L'inférence dans un diagramme d'influence permet d'obtenir l'utilité de chaque valeur d'un noeud d'action étant donné les observations sur d'autres noeuds. Un noeud d'utilité permet donc aussi bien l'évaluation d'une action que le conseil sur le choix de la meilleure action, celle ayant la valeur d'utilité la plus élevée. Dans un diagramme d'influence, les différentes fonctions d'utilité doivent être combinées, ce qui implique d'agréger les critères. Au contraire, dans le réseau bayésien proposé dans cet article, il n'y a pas d'agrégation explicites des critères. Un autre avantage du modèle proposé est que le niveau de satisfaction requis pour les critères peut être ajusté : si aucun compromis n'est possible pour obtenir un niveau élevé de satisfaction ou un degré d'adéquation élevé pour tous les critères, à cause des conflits entre critères et des contraintes liées au cas de décision, alors il est possible de diminuer les niveaux de satisfaction et les degrés d'adéquation exigés. Une comparaison plus complète des réseaux bayésiens et des diagrammes d'influence pour les problèmes de décision répétés et multi-critères est présentée dans [6, 7].

Les travaux présentés ci-dessous ne sont pas dans le domaine de l'ADMC, mais ils proposent des MGP contenant une structure locale spécifique du graphe qui est très proche de celles que nous proposons pour modéliser un critère. Dans [5], le MGP est un modèle probabiliste relationnel¹⁰ (PRM). Les auteurs proposent un modèle de recommandations personnalisés basé sur les PRM en se basant sur les préférences de l'utilisateur sur les critères de décision. La prédiction du lien entre un item et une requête est basée sur un ensemble de variables binaires intermédiaires appelées *facteurs de décision*. Ces facteurs évaluent si un attribut d'un item est proche ou non du critère exprimé par l'utilisateur dans sa requête. Chaque facteur de décision est représenté dans le graphe par une V-structure dont les deux parents sont un attribut d'un item et l'attribut correspondant dans la recherche de l'utilisateur. Ces facteurs de décision sont très similaires aux degrés d'adéquation du présent article : les deux sont basés sur la comparaison des valeurs de deux variables qui représentent des informations de même nature. Les facteurs de décision sont ensuite combinés pour définir une variable binaire *exists* dont la valeur est 1 quand un item a été visité par l'utilisateur au cours de sa session. La recommandation d'un item est basée sur la prédiction de la valeur de cette variable. La façon dont les facteurs de décision sont combinés peut être comparée à l'agrégation de critères. Cependant, cette proposition et la notre exploitent des informations très différentes et adressent des problèmes différents : d'un côté, il s'agit de proposer des items à un internaute sur la base des items déjà visités et de sa requête, alors que la présente proposition est basée sur un modèle de connaissance pour un problème de déci-

sion répété et multicritères et sur la description d'un cas de décision.

Dans [3, 8], les auteurs proposent une V-structure implicite pour exprimer le fait que deux variables discrètes doivent partager la même distribution de probabilités. Ils introduisent le concept de *variable de cohérence* défini comme suit pour deux variables A et B : lorsque la variable de cohérence est égale à 1 (ou "vrai") cela impose que les deux variables partagent la même distribution de probabilités sachant les mêmes prémisses. Cette proposition peut être comparée au *degré d'adéquation* proposé dans le présent article. En effet, elle correspond à un cas particulier où la fonction qui définit le degré d'adéquation impose que les deux variables aient la même distribution de probabilités. Pour mémoire, dans notre proposition, le degré d'adéquation est un indice dans $[0, 1]$ qui peut être défini par n'importe quelle fonction décrivant la meilleure adéquation entre les deux variables considérées, avec la possibilité de prendre en compte différents paramètres.

6 Conclusion

Dans cet article, nous avons présenté la construction d'un modèle de connaissance pour un problème de décision répété et multi-critères, à l'aide d'un réseau bayésien. Du fait que ce type de problème se répète souvent, avec à chaque fois des cas de décision différents, il est intéressant de proposer un modèle unique qui embarque la connaissance sur les alternatives et leur évaluation en suivant un ensemble de critères en prenant en compte la description du cas de décision. Dans ce contexte, notre contribution est l'identification de deux types de critères et la proposition d'un réseau bayésien local pour les modéliser. Les critères objectifs, évalués par un degré d'adéquation entre un attribut d'une alternative et la variable correspondant du côté du cas de décision ; et les critères subjectifs évalués par un niveau de satisfaction fourni par une alternative dans un cas de décision, suivant un certain axe de signification. Il nous semble que n'importe quel critère d'un problème de décision répété et multi-critères relève de l'un de ces deux types. De plus, nous espérons que ce travail peut aider à une meilleure définition des critères de décision au cours de l'analyse d'un problème de décision multicritère. En effet, certains soulignent la nécessité d'une meilleure définition des critères de décision dans les problèmes d'ADMC, en vue de réduire le flou et améliorer la clarté [1].

Une des limites du présent travail concerne des restrictions sur le réseau bayésien, liés à la façon dont les conseils sont générés. En effet, comme l'explication de la situation idéale ne doit concerner que les caractéristiques de l'alternative, nous avons imposé d'observer toutes les autres variables du réseau bayésien. Cette contrainte nous a conduit à limiter le modèle du cas de décision à une seule variable par critère. Une perspective en cours est de permettre la

10. Probabilistic Relational Model

gestion d'un modèle plus complet du cas de décision, en utilisant un sous-ensemble d'observations. Cette perspective doit aussi permettre de gérer plusieurs réseaux bayésiens qui modélisent séparément les caractéristiques des alternatives, celles du cas de décision et les différents critères, plutôt qu'un unique réseau bayésien qui regroupe tout. Notre principal objectif est de proposer un cadre plus général permettant les échanges entre un ensemble de réseaux bayésiens causaux modélisant les différentes entités en jeu dans une situation que l'on veut modéliser. La modélisation des problèmes de décision répétés et multi-critères est une des applications ciblées. Dans cet objectif, nous souhaitons proposer et évaluer l'intérêt d'un modèle distribué dans lequel les échanges sont basés sur des interventions probabilistes [2].

Remerciements

Ce travail est soutenu par le projet ELSAT2020 (Ecomobilité Logistique Sécurité et Adaptabilité dans les Transports à l'Horizon 2020) et cofinancé par l'Union Européenne avec le Fonds européen de développement régional, l'Etat et la Région Hauts de France. Les auteurs remercient le support de ces institutions.

Références

- [1] Ananda, Jayanath et Gamini Herath: *A critical review of multi-criteria decision making methods with special reference to forest management and planning*. Ecological economics, 68(10) :2535–2548, 2009.
- [2] Ben Mrad, A., V. Delcroix, M. A. Maalej, S. Piechowiak et M. Abid: *Uncertain Evidence in Bayesian Networks : Presentation and Comparison on a Simple Example*. Dans *Proceedings of the 14th Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2012*, pages 39–48, Catania, Italy, July 2012.
- [3] Bessière, Pierre, Emmanuel Mazer, Juan Manuel Ahuactzin et Kamel Mekhnacha: *Bayesian Programming*. CRC Press, 2013.
- [4] Bouyssou, Denis: *Readings in Multiple Criteria Decision-Aid*, chapitre Building criteria : a prerequisite for MCDA, pages 58–80. Springer Verlag, Heidelberg, 1990.
- [5] Chulyadyo, Rajani et Philippe Leray: *A Personalized Recommender System from Probabilistic Relational Model and Users' Preferences*. Dans *18th International Conference in Knowledge Based and Intelligent Information and Engineering Systems, KES 2014, Gdynia, Poland, 15-17 September*, pages 1063–1072, 2014.
- [6] Delcroix, Véronique: *Modèles graphiques probabilistes pour les problèmes récurrents d'ADMC*. Dans *Actes des Journées d'Intelligence Artificielle Fondamentale*, Toulouse, France, 2012.
- [7] Delcroix, Véronique: *Réseau Bayésien versus Diagramme d'Influence pour modéliser les problèmes récurrents d'ADMC*. Dans *Actes des Journées Francophones sur les Réseaux Bayésiens*, Sfax, Tunisie, 2012.
- [8] Delcroix, Veronique, Karima Sedki et François Xavier Lepoutre: *A Bayesian network for recurrent multi-criteria and multi-attribute decision problems : Choosing a manual wheelchair*. Expert Systems with Applications, 40(7) :2541–2551, 2013.
- [9] Diehl, M. et Yacov Y. Haimes: *Influence diagrams with multiple objectives and tradeoff analysis*. IEEE Transactions on Systems, Man, and Cybernetics, Part A, 34(3) :293–304, 2004.
- [10] Fenton, Norman et Martin Neil: *Making Decisions : Bayesian Nets and MCDA*. Knowledge-Based Systems, 14(7) :307–325, 2001.
- [11] Huang, Yanliu et J. Wesley Hutchinson: *The roles of planning, learning, and mental models in repeated dynamic decision making*. Organizational Behavior and Human Decision Processes, 122(2) :163 – 176, 2013, ISSN 0749-5978.
- [12] Koller, D. et N. Friedman: *Probabilistic Graphical Models : Principles and Techniques*. MIT Press, 2009.
- [13] Lejarraga, Tomas et Cleotilde Gonzalez: *Effects of feedback and complexity on repeated decisions from description*. Organizational Behavior and Human Decision Processes, 116(2) :286 – 295, 2011.
- [14] Miravete, Eugenio J. et Ignacio Palacios-Huerta: *Consumer Inertia, Choice Dependence, and Learning from Experience in a Repeated Decision Problem*. The Review of Economics and Statistics, 96(3) :524–537, July 2014.
- [15] Mrad, Ali Ben, Véronique Delcroix, Sylvain Piechowiak, Philip Leicester et Mohamed Abid: *An explanation of uncertain evidence in Bayesian networks : likelihood evidence and probabilistic evidence - Uncertain evidence in Bayesian networks*. Appl. Intell., 43(4) :802–824, 2015.
- [16] Roy, B.: *Multicriteria Methodology for Decision Aiding*. Kluwer Academic, Dordrecht, 1996.
- [17] Sedki, K., V. Delcroix, F. X. Lepoutre, E. Adam, A. P. Maquinghen-Godillon et I. Ville: *Bayesian network model for decision problems*. Dans *Klopotek, M.a., M. Marciniak, A. Mykowiecka, W. Penczek et S.t.*

Wierzchon (Ed.) (rédacteurs) : *Intelligent Information Systems, new approaches*, pages 285–298, Publishing House of University of Podlasie, Siedlce, Poland, June 2010. ISBN 978-83-7051-580-5.

- [18] Sedki, Karima et Véronique Delcroix: *A Model Based on Influence Diagrams for Multi-Criteria Decision-Making*. *International Journal on Artificial Intelligence Tools*, 21(4), 2012.
- [19] Watthayu, Wiboonsak et Yun Peng: *A Bayesian network based framework for multi-criteria decision making*. Dans *Proceedings of the 17th International Conference on Multiple Criteria Decision Analysis*, Whistler, British Columbia CA, August 2004.

Tableau-based Revision for Expressive Description Logics

Thinh Dong Chan Le Duc Myriam Lamolle

LIASD - IUT de Montreuil, Université Paris 8, France
{dong, leduc, lamolle}@iut.univ-paris8.fr

Abstract

In this paper we present a tableau algorithm for building completion graphs of an ontology expressed in the description logic *SHIQ* with individuals. Based on a distance defined over completion graphs, we introduce a revision operation applied to a *SHIQ* ontology with a set of new semantic constraints. This revision operation computes the completion graphs that a revised ontology should admit. However, there does not always exist an ontology expressible in *SHIQ* from which a tableau algorithm generates exactly a given set of completion graphs. This leads us to introduce the notion of *upper approximation ontology* from which a tableau algorithm can generate the smallest set of completion graphs including a given set of completion graphs. This notion allows us to design an algorithm for constructing a revised ontology from an initial ontology with a set of new semantic constraints. We also implement the proposed algorithms with optimizations and report some experimental results to show that a model-based approach to revision of expressive ontologies is practicable.

1 Introduction

Formalisms based on Description Logics (DLs) such as OWL are widely used to represent ontologies encapsulated in semantics-based applications. An interesting feature of ontologies expressed in DLs (called DL ontologies) is to support automated inference services which allow ontology designers to detect eventual errors and allow users to entail new knowledge from ontologies with help of a reasoner. However, ontologies are not static but evolve over time. When changing ontologies, we are confronted with the problem of dealing with inconsistencies since new knowledge may contradict what exists in the ontology. The problem of revising a DL ontology is closely related to the problem of belief revision which has been widely discussed in the literature. Among early works on belief revision, Alchourrón, Gärdenfors and Makinson (AGM) [4] introduced intuitive and plausible constraints (namely AGM postulates) which

should be satisfied by any rational belief revision operator. Existing belief revision approaches can be classified into *syntax-based* and *model-based (semantic)* approaches [18].

Syntax-based approaches manipulate directly syntactical entities such as formulas occurring in a knowledge base (KB). To take into account a new formula in preserving consistency, these approaches try to identify other formulas which should be removed. A main advantage of syntax-based approaches is to allow for distinguishing between the relevance of different formulas [2]. For instance, one can affect a lower priority to formulas that can change and a higher priority to those that would be “protected”. The main issues are that the procedures resulting from these approaches heavily depend on the syntax of knowledge bases. Despite these issues, there have been some syntax-based belief revision operations developed for revising a DL ontology [10, 15].

Contrary to syntax-based approaches, semantic approaches investigate and manipulate models of ontologies rather than their syntactical entities. The main issues in adapting semantic approaches to DL ontologies are how to define a distance between models and how to compute a revised ontology from the models selected according to the defined distance. In addition, other problems may arise from dealing with models of DL ontologies. First, DL ontologies have infinitely many models which make impossible to construct directly a revised ontology from models. Second, models of a DL ontology have usually infinite complex structures, which may require a complex definition of distance between two models. Despite these problems, there have been several attempts to adapt classical model-based revision approaches to DL ontologies [9, 19, 21, 20, 22].

In this paper, we propose a new model-based approach for revising ontologies in *SHIQ* with individuals. A preliminary result of the present work for revising ontologies in *SHIQ* without individuals was published at LPAR forum [17]. We base the construction of our revision procedure

on the following points : (i) using completion graphs generated by a novel tableau algorithm to characterize the semantics of a *SHIQ* ontology. A completion graph for an ontology O consists of nodes and edges which are respectively labelled by sets of concepts and roles from the signature of O in such a way that each axiom from O is satisfied in each node and edge. This algorithm must build a set of completion graphs, denoted $FM(O)$, for an ontology O by considering all intrinsic non-deterministic cases instead of building one completion graph as existing tableau algorithms do ; (ii) defining a distance over a set of completion graphs for addressing the principle of minimal change. Given an ontology O' containing new axioms which should be taken into account when revising, this distance can help to choose completion graphs from $FM(O')$ that are semantically closest to those in $FM(O)$. A revised ontology of O by O' should admit the chosen completion graphs as models ; (iii) introducing the notion of approximation ontology to overcome inexpressibility issue. Our revision procedure returns an approximation ontology that is expressible in *SHIQ* and admits a given set of completion forests.

To illustrate the idea behind the construction, we consider the following running example.

Example 1 Given an ontology *UNI* consisting of the following axioms et assertion : α_1 : Professor \sqsubseteq Researcher \sqcup Expert (Professors are researchers or experts), α_2 : Professor $\sqsubseteq \exists$ supervises.Student (A professor supervises at least a student), α_3 : Professor $\sqsubseteq (\geq 2$ teaches.Course) (A professor teaches at least two courses), and β : Professor(Alex) (Alex is a professor).

Assume that researchers and experts do not supervise any student. We add to *UNI* the following axioms which express these semantic constraints : (δ_1) : Researcher $\sqsubseteq \forall$ supervises. $(\neg$ Student), (δ_2) : Expert $\sqsubseteq \forall$ supervises. $(\neg$ Student).

However, the presence of δ_1 and δ_2 will make *UNI* inconsistent, and this requires a revision to maintain consistency of *UNI*. One of the ways of revision is to remove a number of axioms from *UNI*. Intuitively, we can eliminate the axiom α_1 or α_2 to maintain consistency of *UNI*. In this case, if α_1 is chosen to remove then the obtained ontology $\widehat{O} = \{\alpha_2, \alpha_3, \beta, \delta_1, \delta_2\}$ is consistent. However, the knowledge “Professors are experts” in α_1 does not contradict the ontology \widehat{O} but it has been removed together with α_1 . In other words, the goal should be to build a new ontology O^* which is “compatible” with the axioms from *UNI* such that O^* is semantically as close as possible to *UNI*. We can check that the completion graphs $\mathcal{F}_1, \mathcal{F}_2$ in Figure 1 yield models of *UNI*. Similarly, the two completion graphs in Figure 2 yield models of $\{\delta_1, \delta_2\}$.

If we define a distance between completion graphs based on structural similarity, it would be plausible to say that \mathcal{F}'_1 is closer to \mathcal{F}_1 and \mathcal{F}_2 than \mathcal{F}'_2 . Therefore, a revised ontology O^* should admit \mathcal{F}'_1 rather than \mathcal{F}'_2 . Indeed, this intuition will be confirmed in Section 5 where we present a procedure for computing the revised ontology.

The present paper is organized as follows. Section 2 describes the DL *SHIQ*. In Section 3, we present a novel tableau algorithm for building a set of completion forests which represents all models of a *SHIQ* ontology. Section 4 introduces a revision operation which satisfies all revision postulates reformulated for DL ontologies. Based on the defined revision operation, we introduce in Section 5 the notion of upper approximation which allows us to propose a procedure for computing a revised ontology expressible in *SHIQ* from a set of completion forests. Section 6 describes some techniques for optimizing our procedure. We also describe an implementation of our algorithm and report some experimental results in Section 7. Finally, we summarize our work in Section 8.

2 Preliminaries

We begin by presenting the syntax and the semantics of *SHIQ*. Let \mathbf{R} be a non-empty set of role names and $\mathbf{R}_+ \subseteq \mathbf{R}$ be a set of transitive role names. We use $\mathbf{R}_\perp = \{R^- \mid R \in \mathbf{R}\}$ to denote a set of inverse roles. Each element of $\mathbf{R} \cup \mathbf{R}_\perp$ is called a *SHIQ*-role. To simplify notations for nested inverse roles, we define a function $\text{Inv}(S) = R^-$ if $S = R$; and $\text{Inv}(S) = R$ if $S = R^-$ where $R \in \mathbf{R}$. A role inclusion axiom is of the form $R \sqsubseteq S$ for two (possibly inverse) *SHIQ*-roles R and S . A role hierarchy \mathcal{R} is a finite set of role inclusion axioms. A sub-role relation \sqsubseteq is defined as the transitive-reflexive closure of \sqsubseteq on $\mathcal{R}^+ = \mathcal{R} \cup \{\text{Inv}(R) \sqsubseteq \text{Inv}(S) \mid R \sqsubseteq S \in \mathcal{R}\}$. We define a function $\text{Trans}(R)$ which returns true iff R is a transitive role. More precisely, $\text{Trans}(R) = \text{true}$ iff $R \in \mathbf{R}_+$ or $\text{Inv}(R) \in \mathbf{R}_+$. A role R is called *simple* with respect to (w.r.t.) \mathcal{R}^+ iff $R \notin \mathbf{R}_+$ and, for any $R' \sqsubseteq R$, R' is also a simple role. An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a non-empty set $\Delta^{\mathcal{I}}$ (domain) and a function $\cdot^{\mathcal{I}}$ which maps each role name to a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ such that $R^{-\mathcal{I}} = \{(x, y) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid \langle y, x \rangle \in R^{\mathcal{I}}\}$ for all $R \in \mathbf{R}$, and $\langle x, z \rangle \in S^{\mathcal{I}}, \langle z, y \rangle \in S^{\mathcal{I}}$ implies $\langle x, y \rangle \in S^{\mathcal{I}}$ for each $S \in \mathbf{R}_+$. An interpretation \mathcal{I} is a model of \mathcal{R} , written $\mathcal{I} \models \mathcal{R}$, if $R^{\mathcal{I}} \sqsubseteq S^{\mathcal{I}}$ for each $R \sqsubseteq S \in \mathcal{R}$.

Let \mathbf{C} be a non-empty set of concept names. The set of *SHIQ*-concepts is inductively defined as the smallest set containing all C in \mathbf{C} , \top , $C \sqcap D$, $C \sqcup D$, $\neg C$, $\exists R.C$, $\forall R.C$, $(\leq n.S.C)$ and $(\geq n.S.C)$ where n is a positive integer, C and D are *SHIQ*-concepts, R is a *SHIQ*-role and S is a simple role w.r.t. a role hierarchy. We write \perp for $\neg \top$. The interpretation function $\cdot^{\mathcal{I}}$ of an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$

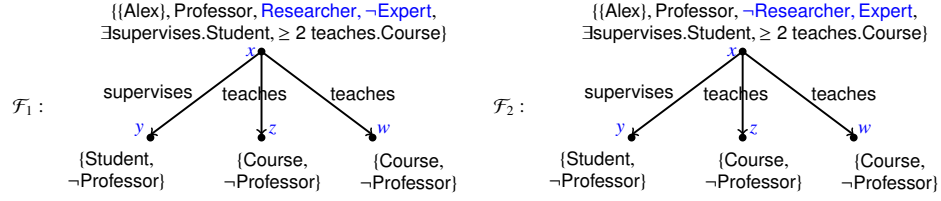


FIGURE 1 – Completion graphs yielding models of UNI

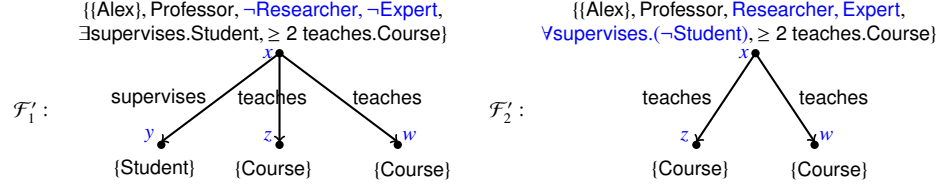


FIGURE 2 – Completion graphs yielding models of $\{\delta_1, \delta_2\}$

maps each concept name to a subset of Δ^I such that $\top^I = \Delta^I$, $(C \sqcap D)^I = C^I \cap D^I$, $(C \sqcup D)^I = C^I \cup D^I$, $(\neg C)^I = \Delta^I \setminus C^I$, $(\exists R.C)^I = \{x \in \Delta^I \mid \exists y \in \Delta^I, \langle x, y \rangle \in R^I \wedge y \in C^I\}$, $(\forall R.C)^I = \{x \in \Delta^I \mid \forall y \in \Delta^I, \langle x, y \rangle \in R^I \Rightarrow y \in C^I\}$, $(\geq n S.C)^I = \{x \in \Delta^I \mid |\{y \in C^I \mid \langle x, y \rangle \in S^I\}| \geq n\}$, $(\leq n S.C)^I = \{x \in \Delta^I \mid |\{y \in C^I \mid \langle x, y \rangle \in S^I\}| \leq n\}$ where $|S|$ stands for the cardinality of a set S . An axiom $C \sqsubseteq D$ is called a general concept inclusion (GCI) where C, D are (possibly complex) *SHIQ*-concepts, and a finite set of GCIs is called a terminology \mathcal{T} . An interpretation I satisfies a GCI $C \sqsubseteq D$, written $I \models (C \sqsubseteq D)$, if $C^I \subseteq D^I$. I is a model of \mathcal{T} , written $I \models \mathcal{T}$, if I satisfies each GCI in \mathcal{T} .

Let \mathbf{I} be a set of individual names. An assertion is of the form $C(a)$, $R(a, b)$, or $a \neq b$ for $a, b \in \mathbf{I}$, a *SHIQ*-role R and a *SHIQ*-concept C . An ABox consists of a finite set of assertions. For an interpretation $I = (\Delta^I, \cdot^I)$, an element $x \in \Delta^I$ is called an instance of a concept C iff $x \in C^I$. For ABoxes, the function \cdot^I of I maps each individual $a \in \mathbf{I}$ to some element $a^I \in \Delta^I$. An interpretation I satisfies an assertion $C(a)$ (resp. $R(a, b)$, and $a \neq b$) iff $a^I \in C^I$ (resp. $\langle a^I, b^I \rangle \in R^I$, and $a^I \neq b^I$). I satisfies an ABox \mathcal{A} if it satisfies each assertion in \mathcal{A} . Such an interpretation is called a *model* of \mathcal{A} , denoted by $I \models \mathcal{A}$.

We use $O = (\mathcal{T}, \mathcal{R}, \mathcal{A})$ to denote a *SHIQ* ontology, where \mathcal{T} is a *SHIQ* terminology, \mathcal{R} is a *SHIQ* role hierarchy, and \mathcal{A} is an ABox. An ontology $O = (\mathcal{T}, \mathcal{R}, \mathcal{A})$ is said to be consistent if there is a model I of \mathcal{T} , \mathcal{R} and \mathcal{A} , i.e., $I \models \mathcal{T}$, $I \models \mathcal{R}$ and $I \models \mathcal{A}$. Additionally, we use $\text{Mod}(O)$ to denote all the models, and $S(O) = \mathbf{R} \cup \mathbf{C} \cup \mathbf{I}$ to denote the signature of an ontology O .

For the ease of construction, we assume all concepts to be in *negation normal form* (NNF), i.e., negation occurs

only in front of concept names. Any *SHIQ*-concept can be transformed to an equivalent one in NNF by using De Morgan's laws and the duality between concepts [14]. For a concept C , we use $\text{nnf}(C)$ and $\neg C$ to denote respectively the NNF of C and $\neg C$. The function $\text{nnf}(C)$ can be computed in polynomial time in the size of C [7]. In the remaining of this section, we introduce some notations which will be used in the next sections.

Definition 1 (Subconcepts) Let $O = (\mathcal{T}, \mathcal{R}, \mathcal{A})$ be a *SHIQ* ontology with $S(O) = \mathbf{R} \cup \mathbf{C} \cup \mathbf{I}$. A set $\text{sub}(O)$ is inductively defined as follows :

$$\begin{aligned} \text{sub}(O) &= \text{sub}(\mathcal{T}) \cup \text{sub}(\mathcal{A}) \cup \{\neg C \mid C \in \text{sub}(\mathcal{T}) \cup \text{sub}(\mathcal{A})\} \\ \text{sub}(\mathcal{T}) &= \bigcup_{C \sqsubseteq D \in \mathcal{T}} \text{sub}(\text{nnf}(\neg C \sqcup D)) \\ \text{sub}(\mathcal{A}) &= \{\text{sub}(\text{nnf}(C)) \mid C(a) \in \mathcal{A}\} \end{aligned}$$

$$\text{sub}(C) = \begin{cases} \{C, \neg C\} & \text{if } C \in \mathbf{C} \\ \text{sub}(E) \cup \text{sub}(F) & \text{if } C \in \{E \sqcap F, E \sqcup F\} \\ \{C\} \cup \{\exists R'.E \mid R \sqsubseteq R'\} \cup \text{sub}(E) & \text{if } C = \exists R.E \\ \{C\} \cup \{\forall R'.E \mid R \sqsubseteq R'\} \cup \text{sub}(E) & \text{if } C = \forall R.E \\ \{C\} \cup \{\geq nR'.E \mid R \sqsubseteq R'\} \cup \text{sub}(E) & \text{if } C = (\geq nR.E) \\ \{C\} \cup \text{sub}(E) & \text{if } C = (\leq nR.E) \end{cases}$$

Note that $\text{sub}(O)$ contains no disjunctions or conjunctions since they are replaced with their disjuncts and conjuncts.

To characterize the semantics of an ontology we need to explore all intrinsic non-determinism arising from disjunctions and numbering restrictions when constructing a completion graph for the ontology. For this reason, we introduce a function $\text{Flat}(C)$ which makes explicit all disjunctions at top-level of a concept C (i.e. those that do not appear in the filler of a universal, existential, numbering restrictions occurring in C).

Definition 2 (Flattening) Let C be a $SHIQ$ concept. We define a function $\text{Flat}(C)$ which returns a set of subsets of $\text{sub}(C)$ as follows :

1. If C is a concept name or C is an existential, universal, number restriction, we define $\text{Flat}(C) = \{\{C\}\}$;
2. If $C = E \sqcup F$, we define $\text{Flat}(C) = \text{Flat}(E) \cup \text{Flat}(F)$;
3. If $C = E \sqcap F$, we define $\text{Flat}(C) = \{W \cup W' \mid W \in \text{Flat}(E), W' \in \text{Flat}(F)\}$

Applying the item 3 in Definition 2 to a concept C may make $\text{Flat}(C)$ increase exponentially. For instance, $\text{Flat}((A_1 \sqcup B_1) \sqcap (A_2 \sqcup B_2)) = \{W \cup W' \mid W \in \{\{A_1\}, \{B_1\}\}, W' \in \{\{A_2\}, \{B_2\}\}\} = \{\{A_1, A_2\}, \{A_1, B_2\}, \{B_1, A_2\}, \{B_1, B_2\}\}$. More general, if $C = (A_1 \sqcup B_1) \sqcap \dots \sqcap (A_n \sqcup B_n)$, $\text{Flat}(C)$ contains 2^n elements.

3 Novel Tableau-based algorithm

In this section we introduce a tableau-based algorithm for generating a finite set of completion graphs representing the infinite set of all models of a $SHIQ$ ontology. Horrocks and colleagues [12] have proposed a tableau algorithm for checking consistency of a $SHIQ$ ontology and have shown that there always exists a finite *completion forest* iff the ontology is consistent. This algorithm attempts to construct a completion forest, returns “YES” if it succeeds in building such a completion forest and “NO” if it fails after considering all possibly non-deterministic cases. To be able to characterize the semantics of an ontology, we need rather a set of completion forests which describes different models resulting from non-deterministic logical constructors than one completion forest. For this purpose, we adapt the tableau algorithm by Horrocks and colleagues [12] in such a way that it would explore all intrinsic non-deterministic cases.

Definition 3 (Completion forest) Let $O = (\mathcal{T}, \mathcal{R}, \mathcal{A})$ be a $SHIQ$ ontology. A completion forest \mathcal{F} for O is a tuple $\mathcal{F} = (G, T(\widehat{x}_1), \dots, T(\widehat{x}_n))$ where

- $G = (\mathbf{V}, \mathbf{E}, \mathbf{L})$ is a directed graph with \mathbf{V} a set of root nodes, \mathbf{E} a set of edges connecting root nodes and \mathbf{L} a labelling function which associates to each node $x \in \mathbf{V}$ a set $\mathbf{L}(x) \subseteq \text{sub}(O)$ and to each edge $\langle \widehat{x}, \widehat{y} \rangle \in \mathbf{E}$ a set $\mathbf{L}(\langle \widehat{x}, \widehat{y} \rangle) \subseteq \mathbf{R} \cup \mathbf{R}_i$. A node $\widehat{y} \in \mathbf{V}$ is called an R -neighbor of $\widehat{x} \in \mathbf{V}$ if $R \in \mathbf{L}(\langle \widehat{x}, \widehat{y} \rangle)$ or $\text{Inv}(R) \in \mathbf{L}(\langle \widehat{y}, \widehat{x} \rangle)$.
- Each $T(\widehat{x}_i) = (V_i, E_i, L_i)$ ($1 \leq i \leq n$) is a tree rooted by \widehat{x}_i belonging to G (i.e. $\widehat{x}_i \in \mathbf{V}$), V_i a set of nodes, E_i a set of edges, and L_i a labelling function which associates to each node $x \in V_i$ a set $L_i(x) \subseteq \text{sub}(O)$ and to each edge $\langle x, y \rangle \in E_i$ a set $L_i(\langle x, y \rangle) \subseteq \mathbf{R} \cup \mathbf{R}_i$.

If two nodes $x, y \in V_i$ (of some tree $T(\widehat{x}_i) = (V_i, E_i, L_i)$) connected by an edge $\langle x, y \rangle \in E_i$, then y is called a successor of x , and x is called a predecessor of y ; ancestor

is the transitive closure of predecessor. A node y is called an R -successor of x if, for some role R' with $R' \sqsubseteq R$, $R' \in L(\langle x, y \rangle)$; x is called an R -predecessor of y , if y is an R -successor of x . A node y is called an R -neighbor of x if y is an R -successor or x is an $\text{Inv}(R)$ -successor of y .

A node $x \in V_i$ is called blocked by a node $y \in V_i$ if x is not a root node and it has ancestors x', y and y' such that (i) y is not a root node, (ii) x is a successor of x' and y is a successor of y' , (iii) $L(x) = L(y)$, $L(x') = L(y')$, and (iv) $L(\langle x', x \rangle) = L(\langle y', y \rangle)$.

Furthermore, there are an inequality relation \neq and an equality relation \doteq defined over nodes in \mathcal{F} . In addition, \mathcal{F} is said to contain a clash if (i) there is some node x in \mathcal{F} such that either $\{A, \neg A\} \subseteq L(x)$ for some concept name $A \in \mathbf{C}$, or (ii) $(\leq nS.C) \in L(x)$ and there are $(n+1)$ S -neighbors y_1, \dots, y_{n+1} of x with $y_i \neq y_j$ and $X \subseteq L(y_i)$ for some $X \in \text{Flat}(C)$ and all $1 \leq i < j \leq (n+1)$.

Based on the Horrocks and colleagues' work [12], we design a tableau algorithm for building a completion forest by applying the expansion rules in Figure 3. There are two main differences between the rules in Figure 3 and those presented in a standard tableau algorithm : (i) the absence of conjunction and disjunction rules. According to Definition 2, applying the function Flat to a concept freshly added to the label of a node removes all conjunctions and disjunctions at top-level from that concept; (ii) the presence of the sat -rule (sat stands for *saturate*). A choice of a subset $S \subseteq \text{sub}(O)$ in the sat -rule must include all flattened concepts from GCI axioms (such as those added by the \sqsubseteq -rule [12]). In addition, the sat -rule adds to each node label either C or $\neg C$ for each $C \in \text{sub}(O)$. These behaviors may lead to an exponential blow-up but it is needed for constructing an approximation ontology from a set of completion forests and a set of subconcepts (Section 5). An optimization of this rule will be proposed and discussed in Section 6.

For an input ontology $O = (\mathcal{T}, \mathcal{R}, \mathcal{A})$, our tableau algorithm starts by initializing a completion forest \mathcal{F} with only root nodes and edges between them. This part of \mathcal{F} represents individuals and assertions defined in \mathcal{A} . The algorithm applies the rules from Figure 3 to each node until no rule is applicable to any node. In this case, \mathcal{F} is called *complete*. If \mathcal{F} contains no clash it is called *clash-free*.

Lemma 1 (Soundness and completeness). Let $O = (\mathcal{T}, \mathcal{R}, \mathcal{A})$ be a $SHIQ$ ontology.

1. The tableau algorithm with the expansion rules in Figure 3 terminates.
2. If the tableau algorithm with the expansion rules in Figure 3 yields a complete and clash-free completion forest from the input ontology O then O is consistent;
3. If O is consistent then the tableau algorithm with the

<p>\exists-rule : if (1) $\exists S.C \in L(x)$, x is not blocked, and (2) x has no S-neighbor y s.t. $X \subseteq L(y)$ for some $X \in \text{Flat}(C)$, then create a new node y with $L(\langle x, y \rangle) \leftarrow \{S\}$ and $L(y) \leftarrow X$ for some $X \in \text{Flat}(C)$.</p>
<p>\forall-rule : if (1) $\forall S.C \in L(x)$, and (2) there is an S-neighbor y of x s.t. $X \not\subseteq L(y)$ for all $X \in \text{Flat}(C)$, then $L(y) \leftarrow L(y) \cup X$ for some $X \in \text{Flat}(C)$.</p>
<p>\forall_+-rule : if (1) $\forall S.C \in L(x)$, (2) there is an R with $\text{Trans}(R)$ s.t. $R \sqsubseteq S$, and (3) there is an R-neighbor y of x s.t. $\forall R.C \notin L(y)$, then $L(y) \leftarrow L(y) \cup \{\forall R.C\}$.</p>
<p>\geq-rule : if 1. $(\geq nS.C) \in L(x)$, x is not blocked, and (2) x has no n S-neighbors y_1, \dots, y_n such that $X \subseteq L(y_i)$ for some $X \in \text{Flat}(C)$ and $y_i \neq y_j$ for $0 \leq i < j \leq n$, then (i) create n new nodes y_1, \dots, y_n with $L(\langle x, y_i \rangle) \leftarrow \{S\}$, (ii) $L(y_i) \leftarrow X$ for some $X \in \text{Flat}(C)$ and $y_i \neq y_j$ for $1 \leq i < j \leq n$.</p>
<p>\leq-rule : if (1) $(\leq nS.C) \in L(x)$, (2) x has $n+1$ S-neighbors y_0, \dots, y_n s.t. $X \subseteq L(y_i)$ for some $X \in \text{Flat}(C)$, (3) there are two S-neighbors y, z of x with $X_1 \subseteq L(y)$, $X_2 \subseteq L(z)$ for some $X_1, X_2 \in \text{Flat}(C)$, y is not an ancestor of z, and not $y \neq z$, then (i) $L(z) \leftarrow L(z) \cup L(y)$ and $L(\langle x, y \rangle) \leftarrow \emptyset$; (ii) if z is an ancestor of x then $L(\langle z, x \rangle) \leftarrow L(\langle z, x \rangle) \cup \{\text{Inv}(R) \mid R \in L(\langle x, y \rangle)\}$, else $L(\langle x, z \rangle) \leftarrow L(\langle x, z \rangle) \cup L(\langle x, y \rangle)$; (iii) add $u \neq z$ for all u such that $u \neq y$.</p>
<p>\leq_r-rule : if (1) $(\leq nS.C) \in L(x)$, (2) x has $n+1$ S-neighbors y_0, \dots, y_n s.t. $X \subseteq L(y_i)$ for some $X \in \text{Flat}(C)$, (3) $y_i \neq y_j$ does not hold for some $0 \leq i < j \leq n$ where y_i, y_j are root nodes, then (i) $L(y_i) \leftarrow L(y_i) \cup L(y_j)$, (ii) for all edges $\langle y_j, w \rangle$: if the edge $\langle y_i, w \rangle$ does not exist, create it with $L(\langle y_i, w \rangle) = \emptyset$; set $L(\langle y_i, w \rangle) \leftarrow L(\langle y_i, w \rangle) \cup L(\langle y_j, w \rangle)$, (iii) for all edges $\langle w, y_j \rangle$: if the edge $\langle w, y_i \rangle$ does not exist, create it with $L(\langle w, y_i \rangle) = \emptyset$; set $L(\langle w, y_i \rangle) \leftarrow L(\langle w, y_i \rangle) \cup L(\langle w, y_j \rangle)$, (iv) set $L(y_j) \leftarrow \emptyset$ and remove all edges to/from y_j, (v) set $u \neq y_i$ for all u with $u \neq y_j$, et (vi) set $y_j \doteq y_i$.</p>
<p>sat-rule : if sat-rule has never been applied to x then (i) choose a subset $S \subseteq \text{sub}(O)$ such that $L(x) \cup \bigcup_{X \in \text{Flat}(\text{nnf}(\neg C \sqcup D)), C \sqsubseteq D \in T} X \subseteq S$, and (ii) set $L(x) \leftarrow S \cup \bar{S}$ where $\bar{S} = \{\neg C \mid C \in \text{sub}(O) \setminus S\}$.</p>

FIGURE 3 – Expansion rules for *SHIQ*

expansion rules in Figure 3 yields a complete and clash-free completion forest from the input ontology O .

The tableau algorithm can build a completion forest whose depth is bounded by an exponential function in the size of O due to the blocking condition. Given a *SHIQ* ontology O , a complete and clash-free completion forest \mathcal{F} built by running the tableau algorithm with the expansion rules in Figure 3 over O is called a *forest-like model*. According to soundness of the tableau algorithm (Lemma 1), one can devise by unraveling a model from a complete and clash-free completion forest \mathcal{F} , denoted $\widehat{I}(\mathcal{F})$. Given an axiom $C \sqsubseteq D$, define $\widehat{I}(\mathcal{F}) \models (C \sqsubseteq D)$ if $C^{\widehat{I}(\mathcal{F})} \subseteq D^{\widehat{I}(\mathcal{F})}$. Given an assertion $C(a)$ (or $R(a, b)$), define $\widehat{I}(\mathcal{F}) \models C(a)$ (resp. $\widehat{I}(\mathcal{F}) \models R(a, b)$) if $a^{\widehat{I}(\mathcal{F})} \in C^{\widehat{I}(\mathcal{F})}$ (resp. $\langle a^{\widehat{I}(\mathcal{F})}, b^{\widehat{I}(\mathcal{F})} \rangle \in R^{\widehat{I}(\mathcal{F})}$). Conversely, according to completeness of the tableau algorithm (Lemma 1), it can build a complete and clash-free completion forest \mathcal{F} from a model $I \in \text{Mod}(O)$, denoted $\mathcal{F}(I)$. This remark allows us to introduce the following notation.

Notation 1 Let O be a *SHIQ* ontology.

- Let \mathcal{F} a forest-like model constructed by the tableau algorithm which takes O as input. We note $\widehat{I}(\mathcal{F})$ the model of O obtained by unraveling \mathcal{F} .

- Let I a model in $\text{Mod}(O)$. We note $\mathcal{F}(I)$ the forest-like model of O constructed by the tableau algorithm from I .

Contrary to standard tableau algorithms which terminate when a forest-like model is found, we design a new tableau algorithm which has to consider all non-deterministic cases and build all forest-like models for an ontology O . We use $\text{FM}(O)$ to denote the set of all forest-like models built by running the new tableau algorithm over a *SHIQ* ontology O . Given a set of concepts sub , we define $\text{FM}(O, \text{sub})$ to be the set of all forest-like models built by running the tableau algorithm on O such that the sat-rule operates on $\text{sub}(O) \cup \text{sub}$ (i.e. it chooses a subset $S \subseteq \text{sub}(O) \cup \text{sub}$). In particular, given an ontology O' the set $\text{FM}(O, \text{sub}(O'))$ can be built by running the new tableau algorithm over O with the sat-rule operating on $\text{sub}(O) \cup \text{sub}(O')$. In addition, we need to import to O roles which occur in role assertions from O' . This leads to add to the label of each root edge $\langle \widehat{x}, \widehat{y} \rangle$ of each forest $\mathcal{F} \in \text{FM}(O, \text{sub}(O'))$ a subset of roles

$S_{\mathcal{R}} \subseteq \mathbf{R}_{\mathcal{O}}$ where $\mathbf{R}_{\mathcal{O}}$ is the set of all roles occurring in \mathcal{O}' with their inverse. This new behavior can be formalized as follows :

- $\text{sat}_{\mathcal{R}}$ -rule : for each root edge $\langle \widehat{x}, \widehat{y} \rangle \in \mathbf{E}$ with $G = (\mathbf{V}, \mathbf{E}, \mathbf{L})$, $\mathcal{F} = (G, T\langle \widehat{x}_1 \rangle, \dots, T\langle \widehat{x}_n \rangle)$ and $\mathcal{F} \in \text{FM}(\mathcal{O}, \text{sub}(\mathcal{O}'))$, we set $L(\langle \widehat{x}, \widehat{y} \rangle) \leftarrow L(\langle \widehat{x}, \widehat{y} \rangle) \cup S_{\mathcal{R}}$ for some $S_{\mathcal{R}} \subseteq \mathbf{R}_{\mathcal{O}}$.

The construction of $\text{FM}(\mathcal{O}, \text{sub}(\mathcal{O}'))$ allows one to import the signature of an ontology \mathcal{O}' to \mathcal{O} when building completion forests for \mathcal{O} . Note that we do not import any new semantic constraint from \mathcal{O}' when building $\text{FM}(\mathcal{O}, \text{sub}(\mathcal{O}'))$. What we really perform in this construction is to import into \mathcal{O} concepts written in the signature of \mathcal{O}' . This importation may extend $\text{FM}(\mathcal{O})$ with new completion forests but never changes consistency of \mathcal{O} .

We now use the notation recently introduced to formulate the following properties on $\text{FM}(\mathcal{O})$ which characterizes the semantics of an ontology \mathcal{O} .

Corollary 1 *Let \mathcal{O} and \mathcal{O}' be two consistent SHIQ ontologies. Let α be a concept axiom or assertion written in $\mathcal{S}(\mathcal{O})$. $\widehat{I}(\mathcal{F}) \models \alpha$ for each forest-like model $\mathcal{F} \in \text{FM}(\mathcal{O}, \text{sub}(\alpha))$ iff $I \models \alpha$ for each model $I \in \text{Mod}(\mathcal{O})$.*

The Corollary 1 affirms the semantic equivalence between $\text{Mod}(\mathcal{O})$ and $\text{FM}(\mathcal{O})$ in the sense that each axiom/assertion which is satisfied by $\text{Mod}(\mathcal{O})$ is satisfied by $\text{FM}(\mathcal{O})$, and conversely. This result allows us to replace a possibly infinite set $\text{Mod}(\mathcal{O})$ with a finite set $\text{FM}(\mathcal{O})$ in constructions presented in the next sections.

4 Revision Operation

The main goal of the present section is to define a revision operation which allows for revising a consistent ontology \mathcal{O} by axioms from another consistent ontology \mathcal{O}' , and however $\mathcal{O} \cup \mathcal{O}'$ is inconsistent. Such a revision operation returns a set of completion forests of which a revised ontology should admit in order to take into account new knowledge from \mathcal{O}' and to be semantically as close as possible to \mathcal{O} . These properties on revision operation are captured by the AGM postulates rephrased for DL ontologies [9]. To reach this goal, we need to define a distance between two completion forests which yields a total pre-order over them and allows one to talk about similarity between two ontologies. This distance is an extension of that defined over completion trees [17].

Definition 4 (Isomorphism) *Let $\mathcal{F} = (G, T\langle \widehat{x}_1 \rangle, \dots, T\langle \widehat{x}_n \rangle)$ and $\mathcal{F}' = (G', T\langle \widehat{x}'_1 \rangle, \dots, T\langle \widehat{x}'_n \rangle)$ be two forest-like models with $G = (\mathbf{V}, \mathbf{E}, \mathbf{L})$, $G' = (\mathbf{V}', \mathbf{E}', \mathbf{L}')$, $T\langle \widehat{x}_i \rangle = \langle V_i, L_i, E_i \rangle$ and*

$T\langle \widehat{x}'_j \rangle = \langle V'_j, L'_j, E'_j \rangle$ ($1 \leq i, j \leq n$). Let $\mathcal{V} = \mathbf{V} \cup V_1 \cup \dots \cup V_n$ and $\mathcal{V}' = \mathbf{V}' \cup V'_1 \cup \dots \cup V'_n$. Let $\mathcal{E} = \mathbf{E} \cup E_1 \cup \dots \cup E_n$ and $\mathcal{E}' = \mathbf{E}' \cup E'_1 \cup \dots \cup E'_n$. We use $\text{succ}(x)$ to denote the set of successors of a node x in a tree $T\langle \widehat{x}_i \rangle$ or $T\langle \widehat{x}'_j \rangle$ with $1 \leq i, j \leq n$.

- *$T\langle \widehat{x}_i \rangle$ and $T\langle \widehat{x}'_j \rangle$ are isomorphic for $1 \leq i, j \leq n$ if there is a bijection π from V_i to V'_j such that (i) $\pi(\widehat{x}_i) = \widehat{x}'_j$; and (ii) for each node $x \in V_i$, we have $\pi(x') \in \text{succ}(\pi(x))$ for each $x' \in \text{succ}(x)$.*

- *\mathcal{F} and \mathcal{F}' are isomorphic if there is a bijection π from \mathcal{V} to \mathcal{V}' such that (i) $\pi(\widehat{x}_i) = \widehat{x}'_j$ for each $\widehat{x}_i \in \mathbf{V}$, (ii) for each $T\langle \widehat{x}_i \rangle \in \mathcal{F}$, two trees $T\langle \widehat{x}_i \rangle$ and $T\langle \pi(\widehat{x}_i) \rangle$ are isomorphic. In this case, we say that π is an isomorphism between \mathcal{F} and \mathcal{F}' .*

Note that if there exists a bijection π between two trees $T\langle \widehat{x}_i \rangle$ and $T\langle \widehat{x}'_j \rangle$ as described in Definition 4 then the restriction of π to $\text{succ}(x)$, denoted $\pi|_{\text{succ}(x)}$, is a bijection from $\text{succ}(x)$ to $\text{succ}(\pi(x))$.

Remark 1 *Let $\mathcal{F} \in \text{FM}(\mathcal{O}, \text{sub}(\mathcal{O}'))$ and $\mathcal{F}' \in \text{FM}(\mathcal{O}', \text{sub}(\mathcal{O}'))$ two forest-like models with the sets of root nodes \mathbf{V} and \mathbf{V}' . It is needed to import all individuals from \mathcal{O} (included in $\text{sub}(\mathcal{O})$) to \mathcal{O}' and reversely when revising \mathcal{O} by new axioms from \mathcal{O}' . Therefore, for each individual a there are a unique tree $T\langle \widehat{x}_i \rangle$ of \mathcal{F} and a unique tree $T\langle \widehat{x}'_j \rangle$ of \mathcal{F}' such that $a \in L(\widehat{x}_i) \cap L(\widehat{x}'_j)$. This implies that there is a bijection φ from \mathbf{V} to \mathbf{V}' such that $\varphi(\widehat{x}_i) = \widehat{x}'_j$ iff $a \in L(\widehat{x}_i) \cap L(\widehat{x}'_j)$ for some individual a .*

Since the notion of isomorphism refers only to the structure of completion forests, we can always obtain such an isomorphism between two any completion forests by adding empty nodes and edges to these completion forests. This is similar to what we have made between two completion trees [17]. In the following, we introduce a distance between two isomorphic completion forests.

Definition 5 (Distance) *Let $\mathcal{F} = (G, T\langle \widehat{x}_1 \rangle, \dots, T\langle \widehat{x}_n \rangle)$ and $\mathcal{F}' = (G', T\langle \widehat{x}'_1 \rangle, \dots, T\langle \widehat{x}'_n \rangle)$ two forest-like models with $G = (\mathbf{V}, \mathbf{E}, \mathbf{L})$, $G' = (\mathbf{V}', \mathbf{E}', \mathbf{L}')$, $T\langle \widehat{x}_i \rangle = \langle V_i, L_i, E_i \rangle$ and $T\langle \widehat{x}'_j \rangle = \langle V'_j, L'_j, E'_j \rangle$ for $1 \leq i, j \leq n$. Let φ be a bijection from \mathbf{V} to \mathbf{V}' such that $\varphi(\widehat{x}_i) = \widehat{x}'_j$ iff there is some individual a satisfying $a \in L(\widehat{x}_i) \cap L'(\widehat{x}'_j)$. The distance between \mathcal{F} and \mathcal{F}' , denoted $d(\mathcal{F}, \mathcal{F}')$, is defined as follows :*

$$d(\mathcal{F}, \mathcal{F}') = \sum_{i=1}^n d(T\langle \widehat{x}_i \rangle, T\langle \varphi(\widehat{x}_i) \rangle) + \max_{\langle x, y \rangle \in \mathbf{E}} (|L(\langle x, y \rangle) \Delta L'(\langle \varphi(x), \varphi(y) \rangle)|)$$

$$\text{where } d(T, T') = \min_{\pi \in \Pi(T, T')} \{ \max_{\langle x, y \rangle \in \mathbf{E}} (|L(x) \Delta L'(\pi(x))| + |L(y) \Delta L'(\pi(y))|) \}$$

with $S \Delta S' = (S \cup S') \setminus (S \cap S')$ for any two sets S and S' , and $\Pi(T, T')$ is the set of all isomorphisms between two

trees T and T' .

We have defined a distance over forest-like models whose sets of root nodes should be associated by a bijection φ according to Remark 1. As any distance, $d(\mathcal{F}, \mathcal{F}')$ should allow one to measure the difference between two forests \mathcal{F} and \mathcal{F}' —that is—it must satisfy identity, symmetry and triangle inequality properties. For this purpose, we use an operator, namely \max , which represents the greatest difference between two triples (composed of an edge and two nodes) associated by an isomorphism π between \mathcal{F} and \mathcal{F}' . This operation \max allows us to ensure the identity property but it is not sufficient for guaranteeing the triangle inequality property $d(\mathcal{F}, \mathcal{F}') \leq d(\mathcal{F}, \mathcal{F}'') + d(\mathcal{F}'', \mathcal{F}')$. For this reason, we must use a further operator, namely \min , which allows for choosing an isomorphism π from all isomorphisms between \mathcal{F} and \mathcal{F}' (one of which gets involved to determine $d(\mathcal{F}, \mathcal{F}'') + d(\mathcal{F}'', \mathcal{F}')$) such that the greatest difference between triples associated by π is smallest.

Lemma 2 *The function $d(\mathcal{F}, \mathcal{F}')$ in Definition 5 satisfies identity, symmetry and triangle inequality properties.*

To show that the distance in Definition 5 yields a total pre-order over a set of isomorphic completion forests, we define a relation “ $\mathcal{F} \leq \mathcal{F}'$ ” over isomorphic completion forests including a forest \mathcal{F}_0 containing only empty labels as follows : $\mathcal{F} \leq \mathcal{F}'$ iff $d(\mathcal{F}_0, \mathcal{F}) \leq d(\mathcal{F}_0, \mathcal{F}')$.

Lemma 3 *The relation “ \leq ” is a total pre-order over isomorphic completion forests.*

All of the above notions provide sufficiently elements to define a revision operation for a *SHIQ* ontology \mathcal{O} by another ontology \mathcal{O}' .

Definition 6 (Revision Operation) *Let \mathcal{O} and \mathcal{O}' be two consistent SHIQ ontologies. A set of forest-like models of the revision of \mathcal{O} by \mathcal{O}' , denoted $\text{FM}(\mathcal{O}, \mathcal{O}')$, is defined as follows :*

$$\begin{aligned} \text{FM}(\mathcal{O}, \mathcal{O}') = \{ & \mathcal{F} \in \text{FM}(\mathcal{O}', \text{sub}(\mathcal{O})) \mid \exists \mathcal{F}_0 \in \text{FM}(\mathcal{O}, \text{sub}(\mathcal{O}')), \\ & \forall \mathcal{F}' \in \text{FM}(\mathcal{O}', \text{sub}(\mathcal{O})), \mathcal{F}'' \in \text{FM}(\mathcal{O}, \text{sub}(\mathcal{O}')) : \\ & d(\mathcal{F}, \mathcal{F}_0) \leq d(\mathcal{F}', \mathcal{F}'') \} \end{aligned}$$

Intuitively, among the forest-like models in $\text{FM}(\mathcal{O}', \text{sub}(\mathcal{O}))$, $\text{FM}(\mathcal{O}, \mathcal{O}')$ retains only those which are closest to forest-like models from $\text{FM}(\mathcal{O}, \text{sub}(\mathcal{O}'))$ thanks to the distance $d(\mathcal{F}_1, \mathcal{F}_2)$ that characterizes the difference between \mathcal{F}_1 and \mathcal{F}_2 .

Example 2 *Consider again the ontology UNI of Example 1. As for simplification, assume that \mathcal{O} is an ontology obtained by adding into UNI the axioms from Table 1, and \mathcal{O}' consists of δ_1, δ_2 . By applying the new tableau algorithm over \mathcal{O} , the set $\text{FM}(\mathcal{O}, \text{sub}(\mathcal{O}'))$ contains 3 forest-like models $\mathcal{F}_1, \mathcal{F}_2$, and \mathcal{F}_3 in Figure 4. By running the new tableau algorithm over \mathcal{O}' , we obtain a forest-like model $\mathcal{F}'_1 \in \text{FM}(\mathcal{O}', \text{sub}(\mathcal{O}))$ illustrated in Figure 4 among*

other forest-like models. By applying the distance formula introduced in Definition 5, we obtain $d(\mathcal{F}'_1, \mathcal{F}_3) = 4$ and $d(\mathcal{F}'_1, \mathcal{F}_1) = d(\mathcal{F}'_1, \mathcal{F}_2) = 2$. According to Definition 8, $\text{FM}(\mathcal{O}, \mathcal{O}')$ contains a unique forest-like model \mathcal{F}'_1 . Note that $d(\mathcal{F}'_x, \mathcal{F}_y) > 2$ for all $\mathcal{F}_y \in \text{FM}(\mathcal{O}, \text{sub}(\mathcal{O}'))$ and $\mathcal{F}'_x \in \text{FM}(\mathcal{O}', \text{sub}(\mathcal{O}))$ with $\mathcal{F}'_x \neq \mathcal{F}'_1$.

As mentioned in Section 1, our goal is to propose a revision operation that ensures the principle of minimal change introduced by Alchourrón, Gärdenfors and Makinson [4] as postulates in belief revision framework. Katsuno and Mendelzon [11] have rephrased these postulates for propositional knowledge bases and shown that the existence of a total pre-order over models of a propositional knowledge base is equivalent to the satisfaction of the postulates. Inspired from Katsuno and Mendelzon’s work [11], we rephrase the postulates in our setting as follows.

- (P1) $\widehat{I}(\mathcal{F}) \models \alpha$ for each forest-like model $\mathcal{F} \in \text{FM}(\mathcal{O}, \mathcal{O}')$ and each axiom $\alpha \in \mathcal{O}'$.
- (P2) If $\text{FM}(\mathcal{O}, \text{sub}(\mathcal{O}')) \cap \text{FM}(\mathcal{O}', \text{sub}(\mathcal{O})) \neq \emptyset$ then $\text{FM}(\mathcal{O}, \mathcal{O}') = \text{FM}(\mathcal{O}, \text{sub}(\mathcal{O}')) \cap \text{FM}(\mathcal{O}', \text{sub}(\mathcal{O}))$.
- (P3) If \mathcal{O}' is consistent then $\text{FM}(\mathcal{O}, \mathcal{O}') \neq \emptyset$.
- (P4) If $\text{FM}(\mathcal{O}_1, \text{sub}(\mathcal{O}'_1)) = \text{FM}(\mathcal{O}_2, \text{sub}(\mathcal{O}'_2))$ and $\text{FM}(\mathcal{O}'_1, \text{sub}(\mathcal{O}_1)) = \text{FM}(\mathcal{O}'_2, \text{sub}(\mathcal{O}_2))$ then $\text{FM}(\mathcal{O}_1, \mathcal{O}'_1) = \text{FM}(\mathcal{O}_2, \mathcal{O}'_2)$.
- (P5) $\text{FM}(\mathcal{O}, \mathcal{O}') \cap \text{FM}(\mathcal{O}'', \text{sub}(\mathcal{O}) \cup \text{sub}(\mathcal{O}'')) \subseteq \text{FM}(\mathcal{O}, \mathcal{O}' \cup \mathcal{O}'')$.
- (P6) If $\text{FM}(\mathcal{O}, \mathcal{O}') \cap \text{FM}(\mathcal{O}'', \text{sub}(\mathcal{O}) \cup \text{sub}(\mathcal{O}'')) \neq \emptyset$ then $\text{FM}(\mathcal{O}, \mathcal{O}' \cup \mathcal{O}'') \subseteq \text{FM}(\mathcal{O}, \mathcal{O}') \cap \text{FM}(\mathcal{O}'', \text{sub}(\mathcal{O}) \cup \text{sub}(\mathcal{O}''))$.

Intuitively, (P1) guarantees that all axioms from \mathcal{O}' can be inferred from the revised ontology. (P2) says that the initial ontology \mathcal{O} is not changed if $\mathcal{O} \cup \mathcal{O}'$ is consistent. (P3) is a condition preventing a revision from introducing unwarranted inconsistency. (P4) says that the revision should be independent of the syntax of ontologies. In fact, if we replace \mathcal{O}_1 and \mathcal{O}'_1 with \mathcal{O}_2 and \mathcal{O}'_2 such that they admit the same forest-like models then revision ontologies admit the same forest-like models as well. (P5) and (P6) can ensure the principle of minimal change.

Theorem 1 *The revision operation $\text{FM}(\mathcal{O}, \mathcal{O}')$ described in Definition 6 satisfies the postulates (P1)-(P6).*

The equivalence between the existence of a total pre-order over forest-like models and the satisfaction of (P1)-(P6) also holds in our setting. Indeed, $\text{FM}(\mathcal{O}, \mathcal{O}')$ in Definition 6 retains only forest-like models from $\text{FM}(\mathcal{O}', \text{sub}(\mathcal{O}))$ which are closest to forest-like models from $\text{FM}(\mathcal{O}, \text{sub}(\mathcal{O}'))$ according to the distance between completion forests. This distance infers the total pre-order “ \leq ” over forest-like models. This observation allows us to get straightforwardly the result saying that the postulates imply a total pre-order over forest-like models since we consider only models such as forest-like models over which a total pre-order exists already. In addition, the clause that a total pre-order

$\alpha_4 : \neg\text{Professor} \sqsubseteq \forall\text{supervises}.\neg\text{Student} \sqcap (\leq 1 \text{ teaches.Course})$	Someone who is not a professor cannot supervise any student and does not teach more than a course
$\alpha_5 : \text{Student} \perp \text{Course}, \text{Student} \perp \text{Professor}, \text{Student} \perp \text{Researcher}, \text{Student} \perp \text{Expert}$	A student is not a course nor a professor nor a researcher nor an expert (note that $A \perp B$ is equivalent to $A \sqsubseteq \neg B$)
$\alpha_6 : \text{Course} \perp \text{Professor}, \text{Course} \perp \text{Researcher}, \text{Course} \perp \text{Expert}$	A course is not a professor nor a researcher nor an expert

TABLE 1 – Axioms added into the ontology UNI

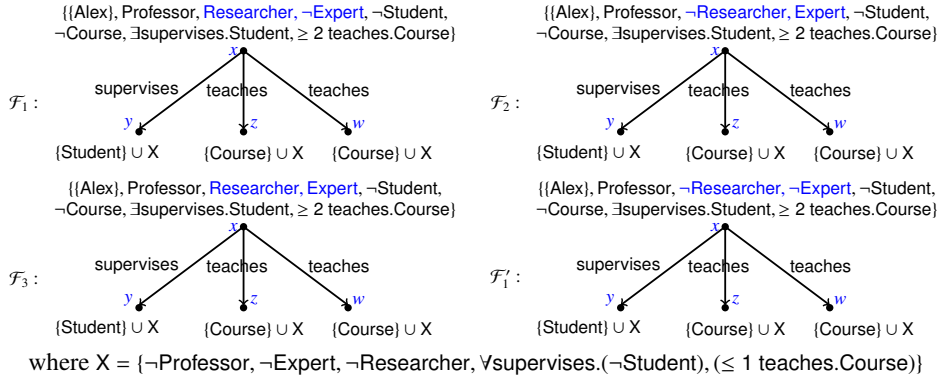


FIGURE 4 – Completion forests yielding models of UNI ($\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3$) and of $\{\delta_1, \delta_2\}$ (\mathcal{F}'_1)

over forest-like models implies the postulates is proved by Theorem 1. Therefore, the principle of minimal change is also ensured in our revision.

5 Computing The Revised Ontology

In this section, we present a procedure for constructing a *SHIQ* ontology \mathcal{O}^* that admits at least forest-like models in $\text{FM}(\mathcal{O}, \mathcal{O}')$. It has turned out [8] that there may not exist a DL-lite ontology which admits exactly a given set of models. It is also the case for *SHIQ* ontologies. To address this issue, we are borrowing the notion of *maximal approximation* from De Giacomo and colleagues' work [8] to define *upper approximation ontology* in our setting as follows.

Definition 7 (Upper approximation) Let \mathcal{O} and \mathcal{O}' be two consistent *SHIQ* ontologies with revision operation $\text{FM}(\mathcal{O}, \mathcal{O}')$. We use $\mathcal{S}(\mathcal{O}')$ to denote the signature of an ontology \mathcal{O}' . An ontology \mathcal{O}^* is an upper approximation from $\text{FM}(\mathcal{O}, \mathcal{O}')$ if (i) $\mathcal{S}(\mathcal{O}^*) \subseteq \mathcal{S}(\mathcal{O}) \cup \mathcal{S}(\mathcal{O}')$; (ii) $\text{FM}(\mathcal{O}, \mathcal{O}') \subseteq \text{FM}(\mathcal{O}^*)$; (iii) There does not exist any ontology \mathcal{O}'' such that $\text{FM}(\mathcal{O}, \mathcal{O}') \subseteq \text{FM}(\mathcal{O}'') \subset \text{FM}(\mathcal{O}^*)$.

Definition 7 characterizes the approximation ontology we should build such that it admits all models in $\text{FM}(\mathcal{O}, \mathcal{O}')$. An interesting point is that if such an upper approximation exists it is unique up to semantic equivalence. We show that such an upper approximation in Definition 7 actually exists and propose a procedure to build it.

Definition 8 (Revised ontology) Let $\mathcal{O} = (\mathcal{T}, \mathcal{R}, \mathcal{A})$ and $\mathcal{O}' = (\mathcal{T}', \mathcal{R}', \mathcal{A}')$ be two consistent *SHIQ* ontologies with $\text{FM}(\mathcal{O}, \mathcal{O}') = \{\mathcal{F}_1, \dots, \mathcal{F}_n\}$ for $1 \leq i \leq n$. For each $\mathcal{F}_i = (G_i, T(\bar{x}_1), \dots, T(\bar{x}_m))$ with $G_i = (\mathbf{V}_i, \mathbf{E}_i, \mathbf{L}_i)$ and $T(\bar{x}_j) = \langle V_j, L_j, E_j \rangle$ ($1 \leq j \leq m$), let $\mathcal{V}_i = \mathbf{V}_i \cup \mathbf{V}_1 \cup \dots \cup \mathbf{V}_m$. A revised ontology $\mathcal{O}^* = (\widehat{\mathcal{T}}, \widehat{\mathcal{R}}, \widehat{\mathcal{A}})$ of \mathcal{O} by \mathcal{O}' is defined as follows : $\widehat{\mathcal{R}} := \mathcal{R}'$, $\widehat{\mathcal{T}} := \mathcal{T}' \cup \{\top \sqsubseteq \bigsqcup_{1 \leq i \leq n} (\bigsqcup_{x \in \mathbf{V}_i} (\bigsqcup_{C \in L_i(x)} C))\}$,

and $\widehat{\mathcal{A}}$ contains a set of assertions as follows :

$$\begin{aligned} & \{C(x) \in \mathcal{A}'\} \cup \{R(x, y) \in \mathcal{A}'\} \cup \\ & \{C(x) \in \mathcal{A} \mid X \subseteq L_i(x), X \in \text{Flat}(C), 1 \leq i \leq n\} \cup \\ & \{R(x, y) \in \mathcal{A} \mid R \in L_i(\langle x, y \rangle), 1 \leq i \leq n\} \cup \\ & \{x \neq y \mid x, y \in \mathbf{I} \cup \mathbf{I}', x \neq y \in \mathcal{F}_i, 1 \leq i \leq n\}. \end{aligned}$$

The construction of revised ontology \mathcal{O}^* retains all concept and role axioms as well as assertions from \mathcal{O}' . It also adds to \mathcal{O}^* a new concept axiom which is built literally from $\text{FM}(\mathcal{O}, \mathcal{O}')$. When building a forest-like model \mathcal{F} by running the tableau algorithm on \mathcal{O}^* , this concept axiom forces to choose a node x from a forest-like model $\mathcal{F}_i \in \text{FM}(\mathcal{O}, \mathcal{O}')$ to add its label $L(x)$ to the current node of \mathcal{F} . Apart from the forest-like models in $\text{FM}(\mathcal{O}, \mathcal{O}')$, the tableau algorithm may build a forest-like model whose nodes have labels coming from different forest-like models from $\text{FM}(\mathcal{O}, \mathcal{O}')$. This is why $\text{FM}(\mathcal{O}^*)$ may be larger than $\text{FM}(\mathcal{O}, \mathcal{O}')$.

Note that the concept axiom built from $\text{FM}(\mathcal{O}, \mathcal{O}')$ allows for capturing semantic parts of only role and concept axioms from \mathcal{O} which should be propagated to \mathcal{O}^* . To transfer semantic parts of the assertions from \mathcal{O} to \mathcal{O}^* , it is needed to determine assertions from \mathcal{O} which remain to be sa-

tified in forest-like models from $\text{FM}(O, O')$. This means that there may be some assertion from O which cannot be propagated to O^* . For example, O contains assertions $R(a, b)$, $S(a, c)$ while O' contains axioms $\top \sqsubseteq \forall R.\perp$ and $\top \sqsubseteq \forall S.\top$. By construction, each forest-like model for O' has an empty edge between a and b but a is an S -neighbor of c . This implies that $S(a, c)$ but not $R(a, b)$ will be added to O^* .

Example 3 *To continue Example 2, we construct from $\text{FM}(O, O')$ an ontology O^* which admits a unique forest-like model \mathcal{F}'_1 according to Definition 8. Thus, O^* contains the following axioms : $\text{Expert} \sqsubseteq \forall \text{supervises} . (\neg \text{Student})$, $\text{Researcher} \sqsubseteq \forall \text{supervises} . (\neg \text{Student})$ (from O'), and*

$$\begin{aligned} \top \sqsubseteq & (\text{Professor} \sqcap \neg \text{Researcher} \sqcap \text{Expert} \sqcap \\ & \neg \text{Course} \sqcap \neg \text{Student} \sqcap \exists \text{supervises} . \text{Student} \sqcap \\ & (\geq 2 \text{ teaches} . \text{Course}) \sqcup (\text{Professor} \sqcap \text{Researcher} \sqcap \\ & \neg \text{Expert} \sqcap \neg \text{Course} \sqcap \neg \text{Student} \sqcap \exists \text{supervises} . \text{Student} \sqcap \\ & (\geq 2 \text{ teaches} . \text{Course}) \sqcup (\text{Student} \sqcap \\ & \forall \text{supervises} . (\neg \text{Student}) \sqcap (\leq 1 \text{ teaches} . \text{Course}) \sqcap \\ & \neg \text{Course} \sqcap \neg \text{Professor} \sqcap \neg \text{Researcher} \sqcap \\ & \neg \text{Expert}) \sqcup (\text{Course} \sqcap \forall \text{supervises} . (\neg \text{Student}) \sqcap \\ & (\leq 1 \text{ teaches} . \text{Course}) \sqcap \neg \text{Student} \sqcap \neg \text{Professor} \sqcap \\ & \neg \text{Researcher} \sqcap \neg \text{Expert}), \text{Professor}(\text{Alex}). \end{aligned}$$

We can now formulate an important result which affirms that the revised ontology O^* defined for two given ontologies O and O' according to Definition 8 is an upper approximation from $\text{FM}(O, O')$. Our argument relies heavily on the specific behavior of the *sat*-rule and the particularity of the concept axiom added to O^* . In fact, if one knows the result of application of the *sat*-rule (i.e. the subset S chosen from $\text{sub}(O)$) to each node of a forest-like model, she knows also the whole forest-like model.

Theorem 2 *Let O and O' be two consistent SHIQ ontologies. The revised ontology O^* of O by O' is an upper approximation from $\text{FM}(O, O')$. Additionally, the size of O^* is bounded by a triple exponential function in the size of O and O' .*

6 Optimizations

So far we have showed that the size of revised ontology is bounded by a triple exponential function in the size of initial ontology. This high complexity is not surprising and arises mainly from the following sources : (i) the characterization of the ontology semantics by using forest-like models obtained from exploring all non-deterministic branches ; (ii) the computation of the distance between two forest-like models may be exponential in the size of forest-like models, and (iii) the construction of an upper approximation ontology forces the tableau algorithm to use the *sat*-rule which considers exhaustively non-deterministic cases.

We present optimization techniques to reduce the complexity arisen from the mentioned sources.

6.1 Computing distance between two forests

According to the formula of the distance between two forest-like models (Definition 5), there is a unique isomorphism between two root nodes of two forest-like models. Therefore, it suffices to investigate optimization of distance computation between two tree-like structures. We present an algorithm for computing the distance $d(T\langle x_0 \rangle, T\langle z_0 \rangle)$ that runs in time polynomial in the size of two trees $T\langle x_0 \rangle$ and $T\langle z_0 \rangle$. For lack of space, we only give the main ideas of the algorithm which are founded on the following observations :

(i) Given an isomorphism π , we denote $h(\pi) = \max_{\langle x, y \rangle \in E_1} (|L_1(x) \Delta L_2(\pi(x))| + |L_1(\langle x, y \rangle) \Delta L_2(\langle \pi(x), \pi(y) \rangle)| + |L_1(y) \Delta L_2(\pi(y))|)$.

There are at most $O(\ell)$ different values of $h(\pi)$ where ℓ is the maximum size of O and O' . In fact, by construction we have $|L(x)| \leq O(\ell)$ and $|L(\langle x, y \rangle)| \leq O(\ell)$ for each node x and edge $\langle x, y \rangle$ of trees. This allows us to partition $\Pi(T\langle x_0 \rangle, T\langle z_0 \rangle)$ into groups each of which corresponds to a value $v_i \in \Delta$ where $v_{i-1} > v_i$ for all $2 \leq i \leq m$.

(ii) For each value $v_i \in \Delta$ from the greatest to the smallest value, it is possible to determine polynomially whether there exists an isomorphism $\pi \in \Pi(T\langle x_0 \rangle, T\langle z_0 \rangle)$ such that $v_i > h(\pi)$. If there does not exist such an isomorphism π , we obtain $d(T\langle x_0 \rangle, T\langle z_0 \rangle) = v_i$. Otherwise, the algorithm considers the value v_{i+1} .

6.2 Constructing $\text{FM}(O, O')$

According to Definition 6, $\text{FM}(O, O')$ is built from $\text{FM}(O, \text{sub}(O'))$ and $\text{FM}(O', \text{sub}(O))$ where O is much larger than O' . As described in Section 3, the tableau algorithm must find all completion forests for O to build $\text{FM}(O, \text{sub}(O'))$. This construction involves at least two sources of complexity : (i) exponential blow-up arising from disjunction and numbering restrictions occurring in O , (ii) exponential blow-up arising from behavior of the *sat*-rule.

To address the first source of complexity, we use various optimization techniques in the literature such as basic and binary absorptions [13, 1]. However, this complexity belongs to intrinsicness of our characterization of ontology semantics since we need a model for each intrinsic non-deterministic case. Therefore, the construction of $\text{FM}(O, \text{sub}(O'))$ is as complex as answering YES to a query such as $O \models C \sqsubseteq D$ since a reasoner must consider all non-deterministic cases.

To address the second source of complexity, we perform the construction $FM(O, O')$ in several stages : (i) Constructing $FM(O, sub(O'))$ without *sat*-rule. (ii) Applying indirectly the *sat*-rule by using absorption techniques to saturate the labels of nodes and edges in completion forests from $FM(O, sub(O'))$. (iii) Constructing $FM(O', sub(O))$ by propagating node and edge labels from completion forests in $FM(O, sub(O'))$. (iv) Choosing completion forests from $FM(O', sub(O))$ to build $FM(O, O')$ by computing distance between completion forests from $FM(O, sub(O'))$ and $FM(O', sub(O))$.

7 Implementation and Experiments

We have implemented a revision engine as prototype, called *ONTOREV*, which is based on the algorithms and definitions described in the previous sections. Similarly to DL reasoners such as *HermiT* [16], *Pellet* [6], *FaCT++* [5], we have implemented in *ONTOREV* various optimization techniques such as absorption, core/anywhere blockings. For instance, we have used basic and binary absorptions [13, 1] to reduce non-deterministic cases arising from disjunction. We have also applied the core blocking technique [3] beside the pairwise blocking technique to scale down the size of completion forests. Differently from existing tableau reasoners, we need to explore all intrinsic non-deterministic cases involved in ontologies to construct all completion forests. As a consequence, we always consider worst-case scenarios where all forests would be built to represent the semantics of an ontology. In the current version of *ONTOREV*, some optimization techniques such as pruning of backtracking points for dealing with intrinsic non-determinism have not been implemented. The lack of implementations of advanced optimizations may slow down *ONTOREV* when it runs on ontologies containing a numerous amount of non-determinism.

We have carried experiments on ontologies *GALEN*, *PIZZA* and *TRAINING* which are modified for simplification. The reason for this choice is that *PIZZA* is a small ontology with a numerous amount of non-determinism arising from disjunctions while *GALEN* would force a tableau algorithm to build completion forests with a sizeable depth. *TRAINING* has resulted from a *FUI* research project on e-learning¹ which involves a revision engine within its platform.

We present in Table 2 the ontology characteristics used for tests in revision of an initial ontology by a revising ontology, and in Table 3 the obtained results. We have run all tests on a *DELL* with 8 Intel 3.4GHz Processors and 32Gb RAM under *Ubuntu*. As mention in Section 6.2, the construction of $FM(O, sub(O'))$ without *sat*-rule allows us to explore only intrinsic non-deterministic

1. <http://www.omendo.com/plateforme-learning-cafe>

Initial Ontology	Ontology characteristics					
	Concepts	Roles	Assertions	Inclusion	Axioms	
					Equivalence	Disjointness
GALEN_1	2748	413	2	3238	699	2
GALEN_2	2748	413	1	3239	699	1
PIZZA	99	5	2	259	8	398
TRAINING	451	95	2	442	0	79

Revising Ontology	Ontology characteristics					
	Concepts	Roles	Assertions	Inclusion	Axioms	
					Equivalence	Disjointness
REV_GALEN_1	3	2	0	2	0	0
REV_GALEN_2	2	1	0	1	0	0
REV_PIZZA	4	1	0	2	0	0
REV_TRAINING	2	2	1	1	0	0

TABLE 2 – Ontologies for experiments with characteristics

Ontology	Revision result				
	$ FM(O, sub(O')) $	$ FM(O, O') $	Tree depth	N^0 of disjunctions	Times (sec.)
REV_GALEN_1	1	1	3	11	3
REV_GALEN_2	1	1	6	17	4
REV_PIZZA	4096	4096	2	18	165
REV_TRAINING	2	2	1	4	2

TABLE 3 – Results of experiments

cases in ontologies. The set $FM(O, sub(O'))$ can help to construct $FM(O', sub(O))$ since it contains forests candidates which have minimal distances to $FM(O, sub(O'))$. Therefore, the size of $FM(O, O')$ is greater than or equal to that of $FM(O, sub(O'))$. In addition, there may exist completion forests in $FM(O, O')$ which are equivalent. This explains why the number of disjunctions (N^0 of disjunctions in Table 3) in the axiom of the resulting ontology O^* is small. Moreover, we can also restore axioms from initial ontology by checking whether a concept name occurs in a node label of a completion forest and, if that is not the case, we add directly to the resulting ontology the axioms transformed by absorption.

8 Conclusion

We have presented in this paper a model-based approach for revising a *SHIQ* ontology with individuals. An interesting feature of our approach is to introduce finite structures, namely *completion forests*, for characterizing the semantics of a *SHIQ* ontology. Semantic distance between expressive ontologies can now be translated onto a distance between completion forests. This feature is crucial to define a revision operation which ensures minimal change. Thanks to that distance we are able to determine a set of completion forests that a revised ontology should admit. To deal with inexpressiveness issue, we have introduced the notion of upper approximation ontology. Finally, we have also proposed optimization techniques for addressing two main sources of complexity, and the presented algorithms have been implemented and tested on various *SHIQ* ontologies.

Références

- [1] Andreas Steigmiller, Thorsten Liebig et Birte Glimm: *Konclude : System description*. J. Web Sem., 27 :78–85, 2014.
- [2] Bernhard Nebel: *Belief Revision and Default Reasoning : Syntax-Based Approaches*. Dans *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*. Cambridge, MA, USA, April 22-25, 1991., pages 417–428, 1991.
- [3] Birte Glimm, Ian Horrocks et Boris Motik: *Optimized Description Logic Reasoning via Core Blocking*. Dans *Proceedings of the 5th International Conference on Automated Reasoning, IJCAR'10*, pages 457–471, 2010.
- [4] Carlos Alchourrón, Peter Gärdenfors et David Makinson: *On the logic of theory change : Partial meet contraction and revision functions*. Journal of symbolic Logic, 50 :510–530, 1985.
- [5] Dmitry Tsarkov et Ian Horrocks: *FaCT++ Description Logic Reasoner : System Description*. Dans *Proceedings of the Third International Joint Conference on Automated Reasoning, IJCAR'06*, pages 292–297, 2006.
- [6] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur et Yarden Katz: *Pellet : A practical OWL-DL reasoner*. Web Semantics : Science, Services and Agents on the World Wide Web, 5(2) :51 – 53, 2007.
- [7] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi et Peter F. Patel-Schneider (éditeurs): *The Description Logic Handbook : Theory, Implementation, and Applications*. Cambridge University Press, 2007.
- [8] Giuseppe De Giacomo, Maurizio Lenzerini, Antonella Poggi et Riccardo Rosati: *On the Approximation of Instance Level Update and Erasure in Description Logics*. Dans *Proceedings of AAAI*, pages 403–408, 2007.
- [9] Guilin Qi et Jianfeng Du: *Model-based Revision Operators for Terminologies in Description Logics*. Dans *Proceedings of the 21st International Joint Conference on Artificial Intelligence, IJCAI'09*, pages 891–897, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.
- [10] Guilin Qi, Weiru Liu et David A. Bell: *Knowledge Base Revision in Description Logics*. Dans *European Conference on Logics in Artificial Intelligence*, pages 386–398, 2006.
- [11] Hirofumi Katsuno et Alberto O. Mendelzon: *Propositional knowledge base revision and minimal change*. Artificial Intelligence, 53(3) :263–294, 1991.
- [12] Horrocks, Ian, Ulrike Sattler et Stephan Tobies: *Reasoning with Individuals for the Description Logic SHIQ*, pages 482–496. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000, ISBN 978-3-540-45101-3.
- [13] Ian Horrocks et Ulrike Sattler: *Optimised reasoning for SHIQ*. Dans *Proceedings of the 15th European Conference on Artificial Intelligence*, pages 277–281. Springer-Verlag, 2002.
- [14] Ian Horrocks, Ulrike Sattler et Stephan Tobies: *Practical reasoning for expressive description logics*. Dans *Proceedings of LPAR*. Springer, 1999.
- [15] Marcio Moretto Ribeiro et Renata Wassermann: *Base revision in description logics - preliminary results*. Proc. of IWOD, pages 69–82, 2007.
- [16] Rob Shearer, Boris Motik et Ian Horrocks: *Hermit : A Highly-Efficient OWL Reasoner*. Dans Ruttenberg, Alan, Ulrike Sattler et Cathy Dolbear (éditeurs) : *Proc. of the 5th Int. Workshop on OWL : Experiences and Directions (OWLED 2008 EU)*, Karlsruhe, Germany, October 26–27 2008.
- [17] Thinh Dong, Chan Le Duc, Philippe Bonnot et Myriam Lamolle: *Tableau-Based Revision over SHIQ TBoxes*. Dans *Logic for Programming, Artificial Intelligence, and Reasoning - 20th International Conference, LPAR-20 2015, Suva, Fiji, November 24-28, 2015, Proceedings*, pages 575–590, 2015.
- [18] Thomas Eiter et Georg Gottlob: *On the Complexity of Propositional Knowledge Base Revision, Updates, and Counterfactuals*. Artificial Intelligence, 57 :227–270, 1992.
- [19] Zhe Wang, Kewen Wang et Rodney Topor: *A New Approach to Knowledge Base Revision in DL-Lite*. Proc. of 24th AAAI, pages 369–374, 2010.
- [20] Zhe Wang, Kewen Wang, Zhiqiang Zhuang et Guilin Qi: *Instance-driven Ontology Evolution in DL-lite*. Dans *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI'15*, pages 1656–1662. AAAI Press, 2015, ISBN 0-262-51129-0.
- [21] Zhiqiang Zhuang, Zhe Wang, Kewen Wang et Guilin Qi: *Contraction and Revision over DL-Lite TBoxes*. Dans *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, AAAI'14*, pages 1149–1155. AAAI Press, 2014.
- [22] Zhiqiang Zhuang, Zhe Wang, Kewen Wang et Guilin Qi: *DL-Lite Contraction and Revision*. Journal of Artificial Intelligence Research, 56 :329–378, 2016.

Dynamique des systèmes d'argumentation : changement de sémantique et forçage d'extension

Sylvie Doutre¹ Jean-Guy Mailly²

¹ IRIT, Université Toulouse 1 Capitole

² LIPADE, Université Paris Descartes

doutre@irit.fr jean-guy.mailly@parisdescartes.fr

Résumé

La dynamique des systèmes d'argumentation a fait l'objet de plusieurs études ces dernières années. La plupart des travaux sur ce sujet concernent le changement de la structure du graphe d'argumentation (addition ou retrait d'arguments et d'attaques), ou le changement du résultat du système (les statuts d'acceptation des arguments). Le changement de la sémantique d'acceptabilité utilisée dans le système n'a pas reçu beaucoup d'attention jusqu'ici. Un tel changement peut être motivé par différentes raisons, notamment comme un moyen de modifier le résultat du système. Dans cet article, nous montrons comment le changement de sémantique peut être utilisé pour atteindre un but sur les statuts des arguments, dans une situation de forçage d'extension.

Abstract

Change in argumentation frameworks has been widely studied in the recent years. Most of the existing works on this topic are concerned with change of the structure of the argumentation graph (addition or removal of arguments and attacks), or change of the outcome of the framework (acceptance statuses of arguments). Change on the acceptability semantics that is used in the framework has not received much attention so far. Such a change can be motivated by different reasons, especially as a way to change the outcome of the framework. In this paper, it is shown how semantic change can be used as a way to reach a goal about acceptance statuses in a situation of extension enforcement.

1 Introduction

La dynamique des systèmes d'argumentation a reçu une attention particulière ces dernières années [10, 12, 6, 7, 5, 26, 11, 19, 15, 16, 17]. De façon générale, on peut distinguer deux types d'approches sur le changement des systèmes d'argumentation : certaines concernent la structure du système (l'ensemble d'arguments et la relation d'attaque), tandis que les autres concernent les

statuts des extensions (extensions, *labellings*, acceptation sceptique, . . .). Cependant, un troisième composant du processus argumentatif n'a reçu que peu d'attention : la sémantique qui relie la structure du système et les statuts des arguments. Même si certaines approches permettent de modifier la sémantique durant le processus (par exemple [6]), il n'est pas expliqué *pourquoi* la sémantique doit changer, ni *comment* la nouvelle sémantique est sélectionnée. Dans cet article, nous étudions ces questions, en nous intéressant plus particulièrement aux sémantiques basées sur les extensions ; ces sémantiques produisent un ensemble d'ensembles d'arguments acceptables appelés extensions, lorsqu'elles sont appliquées à un système d'argumentation.

Deux raisons principales peuvent motiver un changement de sémantique. D'abord, cela peut être nécessaire pour des raisons pratiques. En effet, un problème de certaines sémantiques d'argumentation est leur complexité élevée. Cette complexité théorique n'est pas un problème lorsque l'on considère des systèmes d'argumentation appartenant à une classe particulière, ou de taille modeste. Cependant, même si, jusqu'à un certain point, il semble raisonnable pour un agent d'utiliser une sémantique dont la complexité est élevée (sémantique qui lui garantit l'existence d'au moins une extension par exemple), l'évolution du système peut justifier un changement de sémantique. Par exemple, si l'agent interagit avec d'autres agents dans le contexte d'un débat, des arguments et des attaques peuvent être ajoutés au système. De tels ajouts augmentent la taille du système, et peuvent amener le système à quitter la classe structurelle à laquelle il appartenait. Cela peut rendre le calcul des extensions plus difficile en pratique. Un changement de sémantique peut alors être souhaitable.

Une deuxième motivation du changement de sémantique est d'avoir un moyen alternatif de forcer certaines contraintes sur les statuts des arguments. En effet, il peut y avoir des limitations dans le cadre de certaines applica-

tions, qui empêchent de modifier la relation d'attaque, et l'ensemble d'arguments. Dans ce cas, si l'agent doit satisfaire une contrainte sur l'acceptation des arguments, le seul composant qui peut être modifié est la sémantique. En fait, que le changement de la structure du système soit possible ou non, nous montrons qu'un changement de sémantique peut permettre de satisfaire ce but avec moins de modifications de la structure du système. Cela signifie que dans un cadre où les ajouts et retraites d'arguments ou d'attaques sont limités (par exemple parce qu'ils ont un coût), le changement de sémantique est un levier supplémentaire pour atteindre un objectif (dans cet article, le forçage d'extension) en minimisant ce coût.

Principales contributions

1. Nous définissons un cadre abstrait unifié pour décrire la dynamique des systèmes d'argumentation, qui généralise les approches existantes. Cela permet d'utiliser les mêmes outils pour analyser et étendre ces différentes approches.
2. Nous étendons des travaux existants sur les caractéristiques pour le forçage d'extension [5], c'est-à-dire que nous donnons de nouveaux résultats sur le changement minimal à effectuer sur un système d'argumentation pour garantir qu'un ensemble d'arguments est (inclus dans) une extension, étant donnée une sémantique.
3. Nous étudions le taux de succès du changement de sémantique pour le forçage d'extension, c'est-à-dire le pourcentage de systèmes d'argumentation pour lesquels le résultat est meilleur (concernant le changement minimal de la structure du système) quand le changement de sémantique est utilisé.

Organisation de l'article La section 2 présente les préliminaires formels sur l'argumentation abstraite, sur laquelle repose la contribution. La section 3 propose un cadre général pour le changement des systèmes d'argumentation, qui généralise les approches existantes. Dans la section 4, nous montrons comment le changement de sémantique peut être utilisé pour forcer une contrainte d'acceptabilité dans un système d'argumentation. La section 5 décrit notre analyse expérimentale du taux de succès du changement de sémantique. La dernière section conclut l'article et décrit des pistes intéressantes pour de futurs travaux.

2 Argumentation abstraite

[23] présente l'argumentation comme étant l'étude des relations entre arguments, sans prendre en compte l'origine des arguments ou leur structure interne. Dans ce contexte, un système d'argumentation est un graphe dirigé $\langle A, R \rangle$ où

les nœuds de A sont les *arguments* et les arcs dans R représentent des *attaques* entre arguments. $(a_i, a_j) \in R$ signifie que a_i attaque a_j ; a_i est appelé un *attaquant* de a_j . Un argument a_i (respectivement un ensemble d'arguments S) *défend* un argument a_j contre son attaquant a_k si a_i (respectivement, un argument dans S) attaque a_k . La *portée* d'un ensemble d'arguments S par rapport à R , notée S_R^+ , est le sous-ensemble de A qui contient S et les arguments attaqués par S ; formellement $S_R^+ = S \cup \{a_j \mid \exists a_i \in S \text{ tel que } (a_i, a_j) \in R\}$. Différentes méthodes permettent d'évaluer les arguments. Une approche habituelle est de calculer des *extensions*, qui sont des ensembles d'arguments acceptables conjointement. Différentes sémantiques ont été définies, chacune donnant lieu à différents types d'extensions [23, 2].

Définition 1. Soit $F = \langle A, R \rangle$ un système d'argumentation. Un ensemble $S \subseteq A$ est

- *sans conflit* dans F si $\nexists a_i, a_j \in S$ tel que $(a_i, a_j) \in R$;
- *admissible* dans F si S est sans conflit et S défend chaque $a_i \in S$;
- une *extension naïve* de F si S est un ensemble sans conflit maximal (pour \subseteq);
- une *extension complète* de F si S est admissible et S contient tous les arguments qu'il défend;
- une *extension préférée* de F si S est une extension complète maximale (pour \subseteq);
- une *extension stable* de F si S est sans conflit et $S_R^+ = A$;
- une *extension de base* de F si S est une extension complète minimale (pour \subseteq).

Nous utilisons les notations *cf*, *ad*, *na*, *co*, *pr*, *st*, *gr* pour ces sémantiques. Pour chaque sémantique σ , les σ -extensions de F sont notées $Ext_\sigma(F)$.

Exemple 1. Soit F_1 le système d'argumentation donné en figure 1. Nous illustrons ici certaines des sémantiques.

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

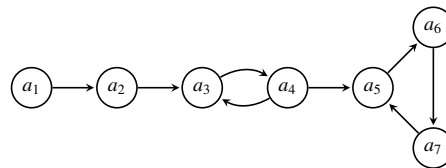


FIGURE 1 – Le système d'argumentation F_1

La notion de fonction de défense¹ d'un ensemble d'arguments dans un système d'argumentation est nécessaire pour la suite.

1. Cette fonction est appelée *characteristic function* par [23]. Nous l'appelons fonction de défense pour éviter la confusion avec les caractéristiques de [5].

Definition 2. Étant donné un système d'argumentation $F = \langle A, R \rangle$ et un ensemble d'arguments $E \subseteq A$, la *fonction de défense de E dans F* est la fonction qui associe à E et F l'ensemble d'arguments $f(E, F)$ défini par :

$$f(E, F) = \{a \in A \mid E \text{ défend } a \text{ contre tous ses attaquants}\}$$

Nous introduisons maintenant un moyen de mesurer la différence entre sémantiques. Cette distance entre sémantiques a été proposée dans [21]. Elle est basée sur les relations entre ensembles d'extensions produits par les différentes sémantiques.

Definition 3. Soit $\Sigma = \{\sigma_1, \dots, \sigma_n\}$ un ensemble de sémantiques, le *graphe d'inclusion des extensions* de Σ est défini par $Inc(\Sigma) = \langle \Sigma, D \rangle$ avec $D \subseteq \Sigma \times \Sigma$ tel que $(\sigma_i, \sigma_j) \in D$ si et seulement si

- pour tout système d'argumentation F , $Ext_{\sigma_i}(F) \subseteq Ext_{\sigma_j}(F)$;
- il n'existe pas $\sigma_k \in \Sigma$ ($k \neq i, k \neq j$) tel que $Ext_{\sigma_i}(F) \subseteq Ext_{\sigma_k}(F)$ et $Ext_{\sigma_k}(F) \subseteq Ext_{\sigma_j}(F)$.

Étant données $\sigma_i, \sigma_j \in \Sigma$, la *mesure de différence basée sur les inclusions* de Σ entre sémantiques est la longueur du plus court chemin non orienté entre σ_i et σ_j dans $Inc(\Sigma)$, notés $\delta_{Inc, \Sigma}(\sigma_i, \sigma_j)$.

Example 2. La figure 2 décrit le graphe d'inclusion des extensions de $\Sigma = \{cf, ad, na, st, pr, co, gr\}$. Nous observons par exemple que $\delta_{Inc, \Sigma}(st, ad) = 3$, $\delta_{Inc, \Sigma}(pr, gr) = 2$, $\delta_{Inc, \Sigma}(co, pr) = 1$.

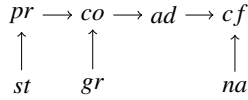


FIGURE 2 – Graphe d'inclusion des extensions $Inc(\Sigma)$

3 Cadre général pour le changement en argumentation

Nous proposons ici une définition abstraite qui inclut toutes les méthodes pour le changement dans les systèmes d'argumentation, en une famille globale.

Definition 4. Un *opérateur de changement* est une fonction χ d'un multi-ensemble de systèmes d'argumentation $\mathcal{F} = \{\{F_1, \dots, F_n\}\}$, une formule φ d'un langage logique et une sémantique σ , vers un multi-ensemble $\mathcal{F}' = \{\{F'_1, \dots, F'_k\}\}$ et une sémantique σ' . Formellement,

$$\chi(\mathcal{F}, \varphi, \sigma) = (\mathcal{F}', \sigma')$$

La plupart des opérations existantes considèrent un unique système comme entrée et sortie des opérateurs, ce

qui est évidemment un cas particulier de multi-ensemble. C'est similaire pour les approches qui considèrent des ensembles de systèmes. [18] considère un profil de systèmes comme l'entrée d'opérateurs de fusion, ce qui peut également être considéré comme un multi-ensemble, puisque l'ordre des systèmes dans le tuple n'est pas pris en compte. À l'exception de [6], les travaux existants ne prennent pas en compte la possibilité de changer de sémantique, ce qui signifie que $\sigma' = \sigma$ pour ces approches. Le langage auquel appartient la formule φ n'est pas le même selon les approches (par exemple, chacun des travaux présentés dans [26, 19, 20, 15, 16] a son propre langage). Certaines approches n'utilisent pas directement une formule d'un langage logique, mais elles peuvent être traduites en une formule d'un langage particulier. Par exemple, ajouter ou retirer un argument ou une attaque [10, 12] est équivalent à certaines formules du langage défini dans [16]. De façon similaire, les ensembles d'arguments utilisés pour le forçage d'extension [6, 5, 17] sont des cas particuliers de formules définies dans [15, 18].

Parmi ces approches, certaines considèrent une notion de minimalité, comme le changement minimal de la relation d'attaque [5, 26, 19, 16], le changement minimal des statuts des arguments [15, 16, 18], ou la minimalité du cardinal [15, 18]. Nous pouvons donner une définition générale de la minimalité dans le processus de changement.

Definition 5. Un *critère de minimalité* est une fonction d'un tuple $\langle \mathcal{F}, \varphi, \sigma, \mathcal{F}', \sigma' \rangle$ vers un tuple de nombres réels positifs $d(\langle \mathcal{F}, \varphi, \sigma, \mathcal{F}', \sigma' \rangle)$.

Étant donné deux tels tuples t_1, t_2 , on définit $t_1 < t_2$ si le i^e élément de t_1 est plus petit que le i^e élément de t_2 , où i est le plus petit indice tel que t_1 et t_2 diffèrent.

Étant donné un multi-ensemble de systèmes $\mathcal{F} = \{\{F_1, \dots, F_n\}\}$, une formule φ et une sémantique σ , un opérateur de changement χ satisfait le critère de minimalité d ssi $\chi(\mathcal{F}, \varphi, \sigma) = (\mathcal{F}', \sigma')$ et $d(\langle \mathcal{F}, \varphi, \sigma, \mathcal{F}', \sigma' \rangle)$ est minimal.

Évidemment, les critères de minimalité les plus simples peuvent être définis avec un nombre unique, donc $d(\langle \mathcal{F}, \varphi, \sigma, \mathcal{F}', \sigma' \rangle)$ est un tuple de longueur 1. Par exemple, nousinstancions cette définition avec les opérateurs de forçage d'extension [6, 5, 17].

Definition 6. Étant donné un système d'argumentation $F = \langle A, R \rangle$ et un ensemble d'arguments $E \subseteq A$, un *opérateur de forçage strict* (respectivement *non strict*) est une fonction de $\mathcal{F} = \{\{F\}\}$, une formule $\varphi_E = \bigwedge_{a_i \in E} a_i$ et une sémantique σ , vers $\mathcal{F}' = \{\{F'\}\}$ et σ' tel que $E \in Ext_{\sigma'}(F')$ (respectivement $\exists \epsilon \in Ext_{\sigma'}(F')$ avec $E \subseteq \epsilon$).

Un forçage est *minimal* ssi il satisfait le critère de minimalité

$$d(\langle \mathcal{F}, \varphi, \sigma, \mathcal{F}', \sigma' \rangle) = \langle d_H(\mathcal{F}, \mathcal{F}') \rangle$$

où d_H est la distance de Hamming entre graphes.^{2 3}

On dit que F' est un forçage de E dans F . On utilise $\varphi_E = \bigwedge_{a_i \in E} a_i$ pour spécifier que l'ensemble E est la requête de forçage ; cela rappelle les encodages logiques utilisés dans [17, 27].

Certains opérateurs de changement utilisent des critères de minimalité plus complexes, qui combinent m critères simples. On peut représenter de tels critères avec un tuple de longueur m ; c'est le cas par exemple de [15, 16, 18].

4 Forçage d'extension et changement de sémantique

Dans cette section, nous étudions l'utilisation du changement de sémantique pour le forçage d'extension. Nous rappelons d'abord la définition de cinq approches pour le forçage d'extension. Ensuite nous montrons sur des exemples intuitifs que modifier la sémantique peut permettre de réaliser un forçage avec moins de changement de la structure du graphe (voire même, *sans aucun changement* de la structure). Finalement, nous étendons l'étude de Baumann [6, 5] sur le changement minimal en fonction de la sémantique, et nous définissons une classe plus générale d'opérateurs de forçage qui atteignent notre but : réaliser le forçage d'extension avec un changement minimal de la structure *via un changement de sémantique*.

4.1 Opérateurs de forçage d'extension

Dans les premiers travaux sur le forçage d'extension [6], il est considéré que ce qui apparaît dans le système d'argumentation ne peut pas être modifié. Les seules modifications autorisées sont les ajouts d'arguments, et éventuellement d'attaques qui concernent au moins un nouvel argument. Ce type de changement est appelé *expansion normale*. Des cas particuliers de l'expansion normale sont appelés *expansion forte* et *expansion faible*. Une expansion forte (respectivement une expansion faible) est une expansion normale qui ajoute uniquement des arguments forts (respectivement faibles), c'est-à-dire des arguments qui ne peuvent pas être attaqués par (respectivement qui ne peuvent pas attaquer) les arguments pré-existants.

Definition 7. Soient F, F' deux systèmes d'argumentation tels que F' est un forçage strict (respectivement non strict) d'un ensemble E dans F .

2. La distance de Hamming entre deux graphes $F_1 = \langle A_1, R_1 \rangle$ et $F_2 = \langle A_2, R_2 \rangle$ est égale à la cardinalité de la différence symétrique entre R_1 et R_2 ; autrement dit, dans le cas présent, c'est le nombre d'attaques qu'il est nécessaire d'ajouter/d'enlever à un graphe pour obtenir l'autre.

3. Ici $\mathcal{F}, \mathcal{F}'$ sont des singletons, la distance de Hamming entre graphes peut donc être utilisée directement. Pour d'autres types d'opérateurs de changement, il faudrait utiliser une version généralisée aux multi-ensembles.

- Si F' est une expansion normale de F , alors le changement de F à F' est un *forçage normal strict* (respectivement *non strict*).
- Si F' est une expansion forte de F , alors le changement de F à F' est un *forçage fort strict* (respectivement *non strict*).
- Si F' est une expansion faible de F , alors le changement de F à F' est un *forçage faible strict* (respectivement *non strict*).

[17] propose ensuite de nouvelles approches qui, à l'opposé, remettent en question la relation d'attaque entre les arguments existants. Deux opérateurs sont définis.

Definition 8. Soient $F = \langle A, R \rangle, F' = \langle A', R' \rangle$ deux systèmes d'argumentation tels que F' est un forçage strict (respectivement non strict) de l'ensemble d'arguments E dans F .

- Si $A = A'$ et $R \neq R'$, alors le changement de F à F' est un *forçage à arguments fixés strict* (respectivement *non strict*).
- Si $A \subseteq A'$, alors le changement de F à F' est un *forçage général strict* (respectivement *non strict*).

Dans toutes ces approches, il est considéré que :

- soit la sémantique ne change pas durant le processus de forçage ;
- soit la nouvelle sémantique est donnée en paramètre de l'opérateur : il n'est pas expliqué pourquoi la sémantique change, ni pourquoi une sémantique en particulier est choisie pour être la nouvelle sémantique du système.

Nous notons ces méthodes $Nor_x, Str_x, Weak_x, Fix_x$ et Gen_x , avec $x \in \{s, ns\}$ qui correspond à strict et non strict.

4.2 Changement de structure minimal grâce au changement de sémantique

Example 3. Considérons à nouveau le système F_1 décrit à la figure 1. Nous voulons forcer l'ensemble $E = \{a_1, a_3\}$ en tant qu'extension. Nous considérons que l'agent utilise pour l'instant la sémantique stable. De toute évidence, pour forcer E , un changement de structure est requis si l'agent ne change pas de sémantique. Mais nous avons vu que E est déjà une extension de F_1 si l'on considère, par exemple, la sémantique préférée ou la sémantique complète. Donc si l'agent est autorisé à changer de sémantique, le forçage peut être réalisé sans modifier la structure du graphe.

Bien sûr, dans certaines situations, changer uniquement la sémantique peut ne pas être suffisant pour atteindre le but, si aucune des sémantiques possibles ne permet de générer des extensions qui correspondent à l'objectif. Dans ce cas, même si le changement de structure est permis, le changement de sémantique peut malgré tout être un moyen de minimiser les changements de structure à effectuer pour

atteindre le but. Même si des changements de structure sont autorisés (ou nécessaires), il peut être coûteux de réaliser de tels changements, et de telles modifications de l'ensemble d'arguments et de l'ensemble d'attaques peuvent être limitées.

Le problème du changement minimal pour le forçage d'extensions a déjà été étudié par [5], pour un sous-ensemble des approches de forçage possibles. Tout d'abord, cela concerne uniquement certaines sémantiques cibles (stable, préférée, complète et admissible). De plus, les opérateurs de forçage à arguments fixés ne sont pas considérés. Par ailleurs, seuls les opérateurs de forçage non stricts sont caractérisés. Pour chaque couple composé d'une de ces sémantiques et d'un opérateur de forçage, le nombre minimal de changements (ajout ou retrait d'attaques) à effectuer pour atteindre un forçage est appelé la *caractéristique*. Cette caractéristique est un entier naturel lorsque le forçage est possible, $+\infty$ signifie que le forçage de cet ensemble est impossible avec la sémantique considérée.

Nous poursuivons cette étude des caractéristiques et donnons ici des résultats pour le forçage à arguments fixés. Nous avons d'abord besoin d'introduire certaines notations.

Definition 9. Étant donné un système d'argumentation $F = \langle A, R \rangle$, et $X \subseteq A$,

- $R_{\downarrow}(F, X) = R \cap (X \times X)$ pour tout $X \subseteq A$;
- $na(F, X) = \{a_i \in A \setminus X \mid \forall a_j \in X, (a_i, a_j) \notin R \text{ et } (a_j, a_i) \notin R\}$
- $ad(F, X) = \{a_i \in A \setminus X \mid \exists a_j \in X, (a_i, a_j) \in R \text{ et } \forall a_j \in X, (a_j, a_i) \notin R\}$
- $st(F, X) = \{a_i \in A \setminus X \mid \forall a_j \in X, (a_j, a_i) \notin R\}$.

Proposition 1. Soit $F = \langle A, R \rangle$ un système d'argumentation, et $E \subseteq A$. La caractéristique du forçage à arguments fixés strict pour $\sigma \in \{cf, ad, st, co, pr, na\}$ est définie par la fonction $V_{\sigma, Fix_s}^F(E)$:

$$\begin{aligned} V_{cf, Fix_s}^F(E) &= |R_{\downarrow}(F, E)| \\ V_{na, Fix_s}^F(E) &= |R_{\downarrow}(F, E)| + |na(F, E)| \\ V_{ad, Fix_s}^F(E) &= |R_{\downarrow}(F, E)| + |ad(F, E)| \\ V_{st, Fix_s}^F(E) &= |R_{\downarrow}(F, E)| + |st(F, E)| \\ V_{co, Fix_s}^F(E) &= \min\{|R' \Delta R| + |R_{\downarrow}(F', E)| \mid f(E, F') = E, \\ &\quad F' = \langle A, R' \rangle\} \\ V_{pr, Fix_s}^F(E) &= \min\{|R' \Delta R| + |R_{\downarrow}(F', E)| \mid E \subseteq f(E, F'), \\ &\quad \forall E \subset E' \subseteq A, E' \not\subseteq f(E', F'), F' = \langle A, R' \rangle\} \end{aligned}$$

Nous constatons que ces résultats sont en lien avec les résultats de complexité de [27]. En effet, ces caractéristiques suggèrent des algorithmes en temps polynomial pour calculer le forçage minimal de E avec les sémantiques cf, na, ad et st .

Proposition 2. Soit $F = \langle A, R \rangle$ un système d'argumentation, et $E \subseteq A$. La caractéristique du forçage à arguments fixés non strict pour $\sigma \in \{cf, ad, st, co, pr, na\}$ est définie

par la fonction $V_{\sigma, Fix_{ns}}^F(E)$:

$$\begin{aligned} V_{na, Fix_{ns}}^F(E) &= V_{cf, Fix_{ns}}^F(E) = |R_{\downarrow}(F, E)| \\ V_{ad, Fix_{ns}}^F(E) &= \min(|R_{\downarrow}(F, E')| + |ad(F, E')| \mid E \subseteq E' \subseteq A) \\ V_{st, Fix_{ns}}^F(E) &= \min(|R_{\downarrow}(F, E')| + |st(F, E')| \mid E \subseteq E' \subseteq A) \\ V_{pr, Fix_{ns}}^F(E) &= V_{co, Fix_{ns}}^F(E) = V_{ad, Fix_{ns}}^F(E) \end{aligned}$$

Nous observons que ces résultats rappellent les caractéristiques pour le forçage général [5].

Observation 1. Pour $Op \in \{Nor, Str, Weak\}$, la caractéristique est triviale pour les sémantiques naïve et sans conflit : soit l'ensemble E est sans conflit, et la caractéristique est alors 0 ; soit E n'est pas sans conflit, et la caractéristique est alors $+\infty$.

Nous généralisons à présent la définition des opérateurs de forçage pour prendre en compte le changement de sémantique.

Definition 10. Soient $F = \langle A, R \rangle$ un système d'argumentation, σ une sémantique, Σ un ensemble de sémantiques, et $E \subseteq A$. Étant donné $Op \in \{Nor, Str, Weak, Fix, Gen\}$ et $x = s$ (respectivement $x = ns$), le forçage minimal de E dans F par Op_x est défini par $\chi(\langle \{F\}, \varphi_E, \sigma \rangle) = (\langle \{F'\}, \sigma' \rangle)$ avec $\sigma' \in \Sigma$, tel que $E \in Ext_{\sigma'}(F')$ (respectivement $\exists \epsilon \in Ext_{\sigma'}(F')$ tel que $E \subseteq \epsilon$), et le critère $\langle V_{\sigma, Op_x}^F, \delta_{Inc, \Sigma}(\sigma, \sigma') \rangle$ est satisfait.

Cela signifie que, contrairement aux travaux existants, la sémantique cible n'est pas un paramètre de l'opérateur de forçage. Elle est choisie pour garantir que :

- la caractéristique (*i.e.* le nombre de modifications du graphe) est minimale ;
- dans le cas où plusieurs sémantiques ont la même caractéristique, celle qui est choisie minimise le changement sémantique.

Nous utilisons ici $\delta_{Inc, \Sigma}$ pour illustrer notre approche, mais d'autres mesures de différence entre sémantiques peuvent être utilisées pour définir le changement sémantique minimal. Le graphe d'inclusion que nous utilisons ici est un cas particulier du graphe de relation défini dans [21]. D'autres notions intéressantes de graphes de relation peuvent être utilisées pour définir des distances entre sémantiques, comme les graphes d'inter-transductabilité [24] ou les relations de scepticisme [3]. [21] mentionne également d'autres approches, basées sur les propriétés des sémantiques, ou basées sur l'ensemble d'extensions d'un système d'argumentation pour les différentes sémantiques. Cela offre une grande diversité des alternatives pour définir le changement de sémantique.

Observation 2. Notre approche ne peut pas donner un résultat « pire », par rapport au changement de structure, que les approches de forçage classiques (c'est-à-dire les approches sans changement de sémantique, ou avec une sémantique spécifiée en paramètre). De plus, nous pouvons identifier quelques cas basiques pour lesquels notre

approche est sûre de donner un meilleur résultat que les approches classiques. Par exemple, lorsque l'ensemble E à forcer n'est pas une extension selon σ du système d'argumentation F (où σ est la sémantique actuelle), mais E est une extension de F selon σ' (avec σ' une des sémantiques alternatives possibles). Dans cette situation, il est garanti que forcer E dans F avec notre approche par changement de sémantique est faisable sans changement de la structure du graphe, tandis que les approches classiques ne le permettent pas.

5 Étude expérimentale

Dans cette section, nous présentons une étude expérimentale du succès du changement de sémantique pour le forçage d'extension. Nous avons calculé le résultat de requêtes de forçage pour un large ensemble de systèmes d'argumentation (en utilisant l'approche de forçage strict à arguments fixés), pour différentes sémantiques ($\Sigma = \{ad, st, co, na\}$), et pour chaque couple $(\sigma_1, \sigma_2) \in \Sigma \times \Sigma$, nous avons comparé V_{σ_1, Fix_s}^F et V_{σ_2, Fix_s}^F . Quand V_{σ_1, Fix_s}^F est significativement plus élevé que V_{σ_2, Fix_s}^F , cela indique que le changement de sémantique est pertinent pour ce couple de sémantiques et cet opérateur de forçage : passer de σ_1 à σ_2 permet de réduire le coût du forçage (en terme de changement sur le graphe). Dans la suite, nous présentons en détail notre protocole expérimentale et nous analysons les résultats.

5.1 Protocole

Nous avons utilisé les systèmes d'argumentation et les requêtes de forçage de [27], qui sont disponibles en ligne.

Ces auteurs fournissent des systèmes d'argumentation de différentes tailles $|A| \in \{50, 100, 150, 200, 250, 300\}$. Les systèmes sont générés suivant le modèle d'Erdős-Rényi [25]. Pour $p \in \{0.05, 0.1, 0.2, 0.3\}$, chaque couple $(a_i, a_j) \in A \times A$ a une probabilité p d'appartenir à la relation d'attaque R . Pour chaque $|A|$ et chaque p , cinq systèmes ont été générés. Finalement, pour chaque système, cinq ensembles d'arguments $E \subset A$ ont été générés aléatoirement pour chaque $|E|/|A| \in \{0.05, 0.1, 0.2, 0.3\}$. Cela signifie que pour chaque $|A|$, 400 instances de problème de forçage ($F = \langle A, R \rangle, E \subset A$) ont été générées.

Pour toutes ces requêtes, nous avons calculé le résultat du forçage strict à arguments fixés pour $\sigma \in \{na, ad, st, co\}$. Le forçage sous la sémantique naïve a été fait via un logiciel que nous avons développé en Java. Pour les autres sémantiques, nous avons utilisé Pakota, le solveur de forçage fournit par [27].⁴

4. Pakota a aussi la possibilité d'effectuer le forçage sous la sémantique préférée. En raison de la complexité élevée du problème de forçage sous cette sémantique, nos expérimentations ont fait face à un nombre élevé de *timeouts*. Pour cette raison, nous excluons pour l'instant la sé-

Les expérimentations ont été faites sur un système Ubuntu 16.04 64bits, équipé de 8Go de RAM et d'un CPU Intel Core i5 cadencé à 3.20GHz. Le temps limite pour chaque instance était de 10 minutes.

5.2 Analyse des résultats

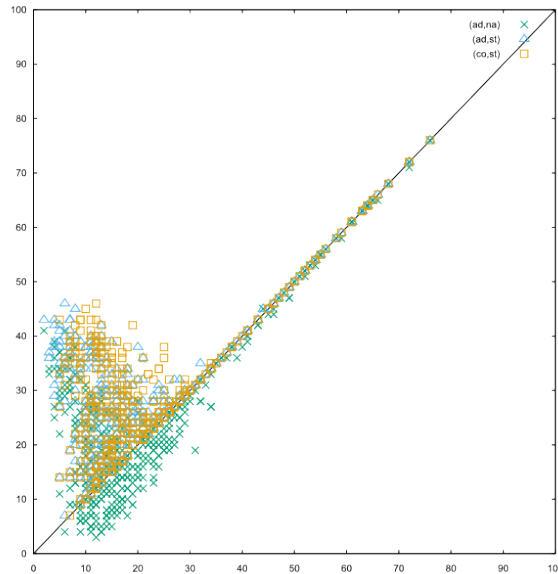


FIGURE 3 – Comparaison du changement minimal en fonction de la sémantique, pour les systèmes à 50 arguments

La figure 3 présente nos résultats pour un sous-ensemble des instances, à savoir les systèmes d'argumentation avec $|A| = 50$ et les requêtes de forçages associées $E \subset A$. Nous présentons uniquement les résultats pour cette classe de systèmes pour des raisons de lisibilité. En effet, pour les autres valeurs de $|A|$, les résultats sont similaires. Par ailleurs, nous présentons uniquement trois des six combinaisons de sémantiques possibles : (ad, st) (représentée par Δ), (ad, na) (représentée par \times) et (co, st) (représentée par \square). Pour chacune de ces combinaisons (σ_1, σ_2) , chaque point représente une instance (c'est-à-dire un couple $(F = \langle A, R \rangle, E \subseteq A)$) de telle façon que son abscisse est le changement minimal pour forcer E dans F sous la sémantique σ_1 , et l'ordonnée est la valeur pour le forçage sous la sémantique σ_2 . Ainsi, un point qui se situe sous la diagonale représente une instance pour laquelle le changement nécessaire sous la sémantique σ_1 est supérieur au changement à apporter sous la sémantique σ_2 , et vice-versa pour les points situés au-dessus de la diagonale. On observe qu'une proportion importante de points sont significativement éloignés de la diagonale, ce qui signifie qu'il y a une proportion élevée d'instances qui peuvent tirer partie du changement de sémantique préférée de notre analyse.

ment de sémantique pour effectuer le forçage. Au contraire, les points situés sur la diagonale représentent des instances pour lesquelles le forçage ne peut pas être amélioré par le changement de sémantique.

Mentionnons le fait que nous avons des résultats similaires pour les couples de sémantiques (st, na) et (co, na) . Seul le couple (ad, co) exhibe des points proches de la diagonale pour une proportion élevée des instances. Pour $|A| \in \{100, 150, 200, 250, 300\}$, nous observons des résultats similaires. Il est tout de même intéressant de mentionner que plus la valeur de $|A|$ est élevée, plus la proportion d'instances avec un faible succès du changement de sémantique diminue. Cependant, même pour $|A| = 300$, il y a toujours un nombre suffisamment important d'instances qui tirent partie du changement de sémantique (c'est-à-dire avec une différence significative entre V_{σ_1, Fix_s}^F et V_{σ_2, Fix_s}^F).⁵ Nous décrivons à la figure 4, pour chaque $|A|$ et chaque couple de sémantiques, le pourcentage d'instances pour lesquelles le rapport $V_{\sigma_1, Fix_s}^F / V_{\sigma_2, Fix_s}^F$ est plus petit que 0.9 ou plus grand que 1.1, c'est-à-dire le pourcentage d'instances pour lesquelles le changement de sémantique apporte une différence significative.

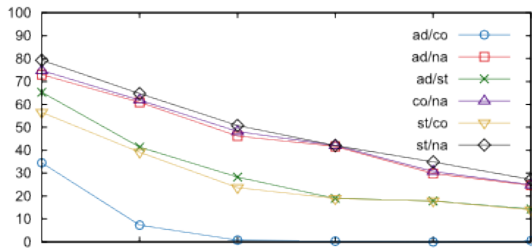


FIGURE 4 – Pourcentage de succès pour différentes situations de changement de sémantique

6 Conclusion

Cet article traite d'un aspect particulier de la *dynamique des systèmes d'argumentation*. Les approches existantes dans ce domaine concernent soit un changement de la structure d'un système, soit un changement des statuts des arguments (les deux étant liés). Nous défendons l'idée que dans certaines applications, cela a du sens de permettre à un agent de modifier son processus de raisonnement, qui est représenté par la sémantique d'acceptabilité. Ce changement peut être motivé par des raisons calculatoires (le besoin d'une complexité moins élevée), ou par certaines propriétés qui doivent être forcées dans l'ensemble d'extensions (par exemple le fait que certains arguments doivent

5. Une description et une analyse complètes de nos expérimentations, y compris les instances, le système de forçage et les courbes pour toutes les valeurs de $|A|$ et tous les couples (σ_1, σ_2) seront disponibles en ligne.

	<i>cf</i>	<i>ad</i>	<i>st</i>	<i>co</i>	<i>pr</i>	<i>na</i>
<i>Nor_s</i>						
<i>Str_s</i>	T	?	?	?	?	T
<i>Weak_s</i>						
<i>Nor_{ns}</i>	T	<i>ad</i>	<i>st</i>	<i>ad</i>	<i>ad</i>	T
<i>Str_{ns}</i>	T	<i>ad</i>	<i>st</i>	<i>ad</i>	<i>ad</i>	T
<i>Fix_s</i>	<i>cf</i>	<i>ad</i>	<i>st</i>	<i>co</i>	<i>pr</i>	<i>st</i>
<i>Fix_{ns}</i>	<i>cf</i>	<i>ad</i>	<i>st</i>	<i>ad</i>	<i>ad</i>	<i>cf</i>
<i>Gen_s</i>	?	?	?	?	?	?
<i>Gen_{ns}</i>	?	<i>ad</i>	<i>st</i>	<i>ad</i>	<i>ad</i>	?

TABLE 1 – Résumé des caractéristiques en fonction de σ et Op . Les cellules grises correspondent aux résultats pré-existants. **T** représente les caractéristiques triviales. σ signifie $V_{\sigma, Op}^F$, avec Op donné dans la première case de la ligne.

être acceptés), avec un changement minimal de la structure.

Un tel changement du processus de raisonnement est lié à ce qui est mentionné dans [8, 9]. Brièvement, l'idée est qu'un agent peut être à même d'utiliser différents processus de raisonnement, par exemple un qui est plus difficile à calculer et plus rationnel, et un autre qui est plus facile à calculer et basé sur des concepts moins rationnels (par exemple, il peut y avoir un biais dû à la perception de sources d'informations par l'agent). Le changement de sémantique en argumentation peut être conduit par une idée similaire.

Dans cet article, nous avons défini un cadre abstrait général pour décrire le changement en argumentation. Ce cadre est utile pour décrire et analyser, avec les mêmes outils, les différentes approches de la dynamique des systèmes d'argumentation. Nous avons ensuite instancié ce cadre pour une famille d'opérateurs de changements particuliers : les opérateurs de forçage d'extension. Nous avons montré que permettre à un agent de modifier la sémantique lorsqu'il réalise un forçage d'extension est utile dans certaines situations, car ce changement de sémantique ne peut pas donner un résultat moins bon (vis-à-vis du nombre de modifications du graphe) que le forçage « classique », et peut même donner de meilleurs résultats. Cette affirmation est basée sur notre étude des caractéristiques, résumée en table 1. Les cases grises correspondent aux résultats donnés dans [5]. Nous avons mené une étude expérimentale qui montre l'impact du changement de sémantique sur un large ensemble d'instances.

Plusieurs questions intéressantes sont apparues suite à ce travail. Naturellement, nous souhaitons compléter notre étude des caractéristiques et nos expérimentations avec plus de sémantiques. Les sémantiques idéales [22], prudente [14] ou les sémantiques SCC-récurrentes [4] sont des candidates potentielles. Déterminer les caractéristiques manquantes (c'est-à-dire remplacer les points d'interrogation

de la table 1 par leurs valeurs) est aussi un travail important. Comme les différences entre sémantiques sont évaluées dans le cadre des sémantiques à extensions, étendre cette approche aux sémantiques à *labellings* semble immédiat. Au contraire, le changement de sémantique pour les sémantiques graduées [1] nécessite un travail plus approfondi. Concernant notre étude expérimentale, nous souhaitons déterminer plus en détail l'impact de différents paramètres sur le changement de sémantique, comme la taille du système d'argumentation, celle de l'ensemble d'arguments à forcer, et la probabilité d'attaques. Nous avons utilisé le modèle de génération de graphes d'Erdős-Rényi, qui capture une structure de graphes intéressantes et a déjà été l'objet d'autres études [27]. Nous prévoyons de conduire une étude similaire pour d'autres familles de graphes [13] pour déterminer si l'impact du changement de sémantique est différent pour ces familles. Nous voulons également étendre les systèmes de forçage d'extension pour tirer profit de l'étude des caractéristiques : il est possible de calculer les caractéristiques pour une liste d'opérateurs de forçage et une liste de sémantiques, et de choisir le meilleur opérateur et la meilleure sémantique pour forcer un ensemble avec le moins possible de changements du graphe.

Pour finir, nous souhaitons étudier l'impact du changement de sémantique sur des opérations qui retournent un ensemble [15, 18]. Dans ces articles, le résultat de l'opération représente un résultat incertain (intuitivement, l'ensemble est interprété comme une « disjonction » de systèmes d'argumentation). Notre but est de déterminer si le changement de sémantique peut aider à réduire le cardinal de cet ensemble (c'est-à-dire réduire l'incertitude du résultat).

Références

- [1] Amgoud, Leila et Jonathan Ben-Naim: *Ranking-Based Semantics for Argumentation Frameworks*. Dans *Proc. SUM'13*, pages 134–147, 2013.
- [2] Baroni, Pietro, Martin Caminada et Massimiliano Giacomin: *An introduction to argumentation semantics*. Knowledge Eng. Review, 2011.
- [3] Baroni, Pietro et Massimiliano Giacomin: *Skepticism relations for comparing argumentation semantics*. *Int. J. Approx. Reasoning*, 50(6) :854–866, 2009.
- [4] Baroni, Pietro, Massimiliano Giacomin et Giovanni Guida: *SCC-recursiveness : a general schema for argumentation semantics*. *Artif. Intell.*, 2005.
- [5] Baumann, Ringo: *What Does it Take to Enforce an Argument? Minimal Change in abstract Argumentation*. Dans *Proc. ECAI'12*, pages 127–132, 2012.
- [6] Baumann, Ringo et Gerhard Brewka: *Expanding Argumentation Frameworks : Enforcing and Monotonicity Results*. Dans *Proc. COMMA'10*, pages 75–86, 2010.
- [7] Bisquert, Pierre, Claudette Cayrol, Florence Dupin de Saint-Cyr et Marie Christine Lagasquie-Schiex: *Change in argumentation systems : exploring the interest of removing an argument*. Dans *Proc. SUM 2011*, pages 275–288, 2011.
- [8] Bisquert, Pierre, Madalina Croitoru et Florence Dupin de Saint-Cyr: *Four Ways to Evaluate Arguments According to Agent Engagement*. Dans *Proc. BIH'15*, pages 445–456, 2015.
- [9] Bisquert, Pierre, Madalina Croitoru et Florence Dupin de Saint-Cyr: *Towards a Dual Process Cognitive Model for Argument Evaluation*. Dans *Proc. SUM'15*, pages 298–313, 2015.
- [10] Boella, Guido, Souhila Kaci et Leendert van der Torre: *Dynamics in Argumentation with Single Extensions : Attack Refinement and the Grounded Extension*. Dans *Proc. AAMAS'09*, pages 1213–1214, 2009, ISBN 978-0-9817381-7-8.
- [11] Booth, Richard, Souhila Kaci, Tjitze Rienstra et Leendert van der Torre: *A Logical Theory about Dynamics in Abstract Argumentation*. Dans *Proc. SUM 2013*, pages 148–161, 2013.
- [12] Cayrol, Claudette, Florence Dupin de Saint-Cyr et Marie Christine Lagasquie-Schiex: *Change in Abstract Argumentation Frameworks : Adding an Argument*. *J. Artif. Intell. Res.*, 38 :49–84, mai 2010.
- [13] Cerutti, Federico, Massimiliano Giacomin et Mauro Vallati: *Generating Structured Argumentation Frameworks : AFBenchGen2*. Dans *Proc. COMMA'16*, pages 467–468, 2016.
- [14] Coste-Marquis, Sylvie, Caroline Devred et Pierre Marquis: *Prudent Semantics for Argumentation Frameworks*. Dans *Proc. ICTAI 2005*, pages 568–572, 2005.
- [15] Coste-Marquis, Sylvie, Sébastien Konieczny, Jean-Guy Mailly et Pierre Marquis: *On the Revision of Argumentation Systems : Minimal Change of Arguments Statuses*. Dans *Proc. KR'14*, 2014.
- [16] Coste-Marquis, Sylvie, Sébastien Konieczny, Jean-Guy Mailly et Pierre Marquis: *A Translation-based Approach for Revision of Argumentation Frameworks*. Dans *Proc. JELIA'14*, 2014.
- [17] Coste-Marquis, Sylvie, Sébastien Konieczny, Jean-Guy Mailly et Pierre Marquis: *Extension Enforcement in Abstract Argumentation as an Optimization Problem*. Dans *Proc. IJCAI'15*, 2015.
- [18] Delobelle, Jérôme, Adrian Haret, Sébastien Konieczny, Jean-Guy Mailly, Julien Rossit et Stefan Woltran: *Merging of Abstract Argumentation Frameworks*. Dans *Proc. KR'16*, pages 33–42, 2016.

- [19] Doutre, Sylvie, Andreas Herzig et Laurent Perrussel: *A Dynamic Logic Framework for Abstract Argumentation*. Dans *Proc. KR'14*, pages 62–71, 2014.
- [20] Doutre, Sylvie, Faustine Maffre et Peter McBurney: *A Dynamic Logic Framework for Abstract Argumentation : Adding and Removing Arguments*. Dans *Proc. IEA/AIE'17*, 2017.
- [21] Doutre, Sylvie et Jean-Guy Mailly: *Quantifying the Difference Between Argumentation Semantics*. Dans *Proc. COMMA'16*, 2016.
- [22] Dung, Phan Mihn, Paolo Mancarella et Francesca Toni: *Computing Ideal Sceptical Argumentation*. *Artif. Intell.*, 171(10-15) :642–674, 2007.
- [23] Dung, Phan Minh: *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.
- [24] Dvorák, Wolfgang et Christof Spanring: *Comparing the Expressiveness of Argumentation Semantics*. Dans *Proc. COMMA'12*, pages 261–272, 2012.
- [25] Erdős, Paul et Alfréd Rényi: *On Random Graphs I*. *Publicationes Mathematicae*, pages 290–297, 1959.
- [26] Saint-Cyr, Florence Dupin de, Pierre Bisquert, Claudette Cayrol et Marie-Christine Lagasquie-Schiex: *Argumentation update in YALLA (Yet Another Logic Language for Argumentation)*. *Int. J. Approx. Reasoning*, 75 :57–92, 2016.
- [27] Wallner, Johannes Peter, Andreas Niskanen et Matti Järvisalo: *Complexity Results and Algorithms for Extension Enforcement in Abstract Argumentation*. Dans *Proc. AAAI'16*, pages 1088–1094, 2016.

Vers une nouvelle méthode de calcul de modèles stables et extensions en programmation logique

Tarek Khaled¹

Belaïd Benhamou¹

Pierre Siègel²

¹ Université Aix Marseille, LSIS UMR 7296, Marseille, France

² Université Aix-Marseille, LIF UMR 7279, Marseille, France

{tarek.khaled,belaid.benhamou}@lsis.org pierre.siegel@lif.univ-mrs.fr

Résumé

La programmation par ensembles réponses (Answer set programming (ASP)) est une approche très connue en programmation déclarative. Elle est très utilisée dans le domaine de la représentation des connaissances et le raisonnement. L'ASP est un sujet de recherche très prisé ces dernières années. Les solveurs ASP sont devenus très performants et compétitifs, ils ont gagné leurs galons dans les applications industrielles. Le travail que nous présentons dans cet article porte sur l'élaboration d'un algorithme de recherche d'ensembles réponses basé sur une nouvelle sémantique (Benhamou et al. 2012) des programmes logiques qui capture et qui étend celle des modèles stables (Gelfond et al. 1988). Cet algorithme est basé sur une énumération au sens de la procédure DPLL adaptée au cadre de l'ASP et à la nouvelle sémantique utilisée. L'avantage de cette méthode est qu'elle opère sur un système de clauses de Horn ayant la même taille que le programme logique source et à espace constant. Elle évite ainsi la lourdeur de la complétion de Clark et la gestion des boucles souvent utilisées dans les solveurs ASP basés sur DPLL. De plus, l'énumération est faite sur une restriction de l'ensemble des variables représentant le strong backdoor (STB) du programme logique considéré. La complexité de l'algorithme est calculée en fonction de cet ensemble (STB) et son efficacité en dépend. Nous avons introduit quelques nouvelles règles d'inférence dans ce cadre que l'algorithme utilise pour faire des coupures dans l'arbre de recherche et réduire ainsi sa taille. Cette méthode permet en plus des modèles stables, de générer des extra-modèles exprimant l'extension apportée à la sémantique de Gelfond et al. Nous avons implémenté une première version de cette méthode qui permet de calculer les modèles stables d'un programme logique que nous avons testée dans un premier temps sur des instances de coloriage de graphes aléatoires.

Abstract

Answers set programming (ASP) is a well-known approach in declarative programming. It is widely used in the field of knowledge representation and reasoning. ASP is a highly sought research topic in recent years. ASP solvers have become very efficient and competitive, they have gained their stripes in industrial applications. The work we present in this paper focuses on the development of an algorithm for finding answer sets that is based on a new semantics (Benhamou et al., 2012) of logical programs that captures and extends that one of stable models (Gelfond 1988). This algorithm is based on an enumeration in the sense of the DPLL procedure adapted to the framework of the ASP and to the new semantics used. The advantage of this method is that it operates on a Horn clause system having the same size as the source logic program. It thus, avoids the heaviness of the Clark completion and the loop management often used in ASP solvers based on DPLL. Moreover, the enumeration is done on a restriction of the set of variables of the set of clauses representing the strong backdoor (STB) of the considered logic program. The complexity of the algorithm is calculated as a function of this set (STB) on which its efficiency depends. We have introduced some new inference rules in this framework that the algorithm uses to make cuts in the search tree and thus reduce its size. This method also allows in addition to the stable models, to generate extra-models expressing the extension given to the semantics of Gelfond and al. We have implemented a first version of this method which allows to calculate the stable models of a logic program that we tested on random graph coloring instances.

1 Introduction

Un programme logique π est un ensemble de faits et de règles se terminant par un point ('.'). Les connecteurs ':-' et ',' apparaissent dans ces règles et peuvent être interprétés comme étant 'si' et 'et' respectivement. Précisément, le

programme logique π est un ensemble de règles de la forme $r : tête(r) \leftarrow corps(r)$, où $corps(r)$ représente l'ensemble des prémisses de la règle r et $tête(r)$ sa partie conclusion ou conséquence. Cet ensemble de prémisses est en général une conjonction de littéraux où il peut y avoir des négations classiques et des négations par échec. Cette conjonction est donnée sous forme de liste de littéraux séparée par des virgules. La partie $tête(r)$ est exprimée soit par un seul littéral, ou par une disjonction de littéraux. La lecture intuitive de ces règles nous amène à dire que la partie $tête(r)$ peut être inférée si la partie $corps(r)$ a été déjà prouvée.

La Programmation par ensembles réponses (ASP) [17] est une forme de programmation déclarative non monotone qui offre un cadre idéal pour la formulation de problèmes en Intelligence Artificielle. Cela peut être du raisonnement de sens commun, du web sémantique, du raisonnement causal, ou de la résolution de problèmes combinatoires tels que la planification, des problèmes de théorie des graphes, la configuration, ou les problèmes de bio-informatique [5]. Le paradigme de programmation ASP fournit un cadre général pour la résolution de problème de décision et d'optimisation [16]. L'approche a été rendue populaire grâce au développement de plusieurs solveurs performants dédiés à l'ASP comme *smodels* [13], *DLV* [11] et dernièrement *Clasp* [15] et notamment ceux basés sur la procédure DPLL et les solveurs SAT comme *ASSAT* [7].

L'idée de base de l'ASP est d'exprimer un problème donné sous forme d'un programme logique pour lequel on doit chercher les modèles stables qui représentent les solutions du problème originel [10]. Pour avoir une expression concise du problème, on utilise dans un premier temps, des règles construites à base de la logique du premier ordre. En conséquence, le problème sera exprimé par un programme logique dit souvent programme non terminal contenant des prédicats avec des variables. Des systèmes appelés en anglais "grounders" ont été conçus pour calculer l'ensemble des instances terminales des règles de ce programme non terminal donnant en conséquence un programme logique terminal équivalent. La figure 1 illustre le fonctionnement général d'un système ASP.

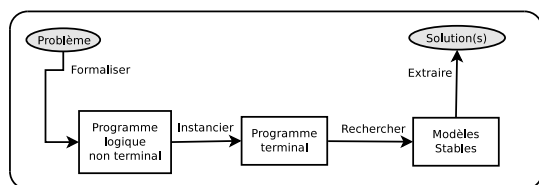


FIGURE 1 – Modèle conceptuel du paradigme ASP

La sémantique des modèles stables [8] est l'une des plus utilisée dans le cadre ASP. Il existe dans la littérature plusieurs sémantiques pour les programmes logiques. La première sémantique introduite dans ce cadre, est la complétion de Clark [4]. Puis, d'autres sémantiques telles que celle

de "well-founded" [1], celle des modèles stables [8] ou celle d'ensembles réponses [12] ont été introduites. Ces sémantiques ont toutes pour but de donner une signification aux programmes logiques considérés. Notamment, elles essaient de donner un sens à la négation par échec présente dans les règles du programme. C'est souvent ce sens qui les définit et qui les distingue.

Dans le cadre de ce travail nous utilisons la nouvelle sémantique introduite dans [2]. Nous avons choisi cette sémantique pour sa simplicité et les avantages qu'elle offre. Cette sémantique est basée sur une notion d'extension qu'elle définit pour les programmes logiques. Dans cette sémantique, les programmes sont représentés par un ensemble de clauses de Horn ayant la même taille que le programme d'entrée. Elle a l'avantage de caractériser les modèles stables en utilisant cette représentation Horn clause. Cette représentation clause permet d'avoir des algorithmes de résolution avec de bonnes propriétés de complexité. L'algorithme que nous allons présenter utilise cette représentation et tire profit de tous les avantages qu'elle offre un ensemble de clauses de Horn. Elle évite ainsi la lourdeur de la complétion de Clark [4] souvent utilisées dans les solveurs ASP basés sur DPLL et la gestion des boucles qui les rend non constant en complexité spatiale. C'est une adaptation de la procédure DPLL au cadre de l'ASP sous la nouvelle sémantique. Cet algorithme est basé sur une énumération faite sur une restriction de l'ensemble des variables représentant le strong backdoor¹ (STB) [14] du programme logique considéré. La complexité de l'algorithme est calculée en fonction de cet ensemble (STB) et son efficacité en dépend. Nous utilisons quelques nouvelles règles d'inférence que l'algorithme exploite pour élaguer l'arbre de recherche. Cette méthode dans sa globalité, permet de calculer des extensions, à partir desquelles on peut générer des modèles stables ou des extra-modèles exprimant ainsi l'extension apportée à la sémantique de Gelfond et al. [8].

Le reste de l'article est organisé comme suit : dans la section 2, nous rappelons quelques notions de base sur la programmation ASP et les fondements de la nouvelle sémantique [2] que nous utilisons dans la nouvelle méthode de résolution. Nous présenterons cette méthode dans la section 3. Dans la section 4, nous donnons les premiers résultats expérimentaux de notre approche et enfin nous concluons le travail en section 5.

2 Préliminaires

Un programme logique π est composé par un ensemble de règles de la forme $r : tête(r) \leftarrow corps(r)$. Il existe différentes classes de programmes logiques. Ils se distinguent par la présence ou l'absence de la négation classique et

1. Intuitivement, un ensemble de variables d'un problème forme un strong backdoor si le problème restant après l'instanciation de ces variables est résolu en temps polynomial

de la négation par échec dans l'ensemble des règles qui les composent. Un programme logique positif π est un ensemble de règles de la forme : $r = A_0 \leftarrow A_1, A_2, \dots, A_m$, avec ($m \geq 0$) et où $A_{i \in \{0, \dots, m\}}$ est un atome. Un programme positif ne contient pas de négation par échec ou de négation classique. Il est connu que tout programme positif admet un seul modèle stable. Il est le plus petit ensemble d'atomes obtenu par l'application des règles du programme jusqu'à un point fixe. C'est le modèle minimal de Herbrand de l'ensemble de clauses de Horn formant le programme π ; il est noté $Cn(\pi)$.

Dans la suite de cet article, nous ne nous intéresserons qu'aux programmes logiques généraux (normaux). Un programme logique général π est un ensemble de règles de la forme : $r = A_0 \leftarrow A_1, A_2, \dots, A_m, \text{not } A_{m+1}, \dots, \text{not } A_n$, ($0 \leq m < n$) où $A_{i \in \{0, \dots, m\}}$ est un atome et *not* le symbole exprimant la négation par échec. Le corps positif de r est dénoté par $\text{corps}^+(r) = \{A_1, A_2, \dots, A_m\}$ et le corps négatif par $\text{corps}^-(r) = \{A_{m+1}, \dots, A_n\}$. Intuitivement, la règle r est interprétée comme suit : si on prouve tous les atomes du corps positif $\text{corps}^+(r) = \{A_1, A_2, \dots, A_m\}$ de la règle r , et qu'on n'arrive à prouver aucun des atomes du corps négatif $\text{corps}^-(r) = \{A_{m+1}, \dots, A_n\}$, alors on infère A_0 . Le réduit d'un programme π par rapport à un ensemble d'atomes X est le programme positif π^X obtenu à partir de $\text{ground}(\pi)$ en supprimant :

- chaque instance de règle ayant un atome *not* A_i dans son corps négatif tel que $A_i \in X$ et
- tous les atomes *not* A_j tels que $A_j \notin X$, dans les corps négatifs des instances des règles restantes.

La sémantique la plus utilisée pour les programmes logiques généraux est celle des modèles stables [8]. Un ensemble d'atomes X est un modèle stable de π si et seulement si $X = Cn(\pi^X)$. En d'autres mots, l'ensemble d'atomes X est un modèle stable de programme π ssi X est identique au modèle minimal de Herbrand du réduit π^X du programme π par rapport à X .

On trouve d'autres sémantiques pour les programmes logiques. Pour gérer la notion de "négation par échec", Clark a proposé le concept de complétion d'un programme logique (notation $\text{comp}(\pi)$). Il est connu que tout modèle stable de π est un modèle de sa complétion mais la réciproque ne s'applique que si le programme est sans boucle (tight en anglais) [6]. Afin d'établir l'équivalence entre la sémantique d'un programme et sa complétion, des formules de traitement des boucles sont ajoutées à la complétion (Fangzhen Lin et Yuting Zhao [7]). Le nombre de boucles dans le pire des cas peut varier d'une façon exponentielle, ce qui rend leur traitement infaisable[18].

Dans le cas de notre étude, nous utilisons la sémantique introduite dans [2] qui offre plusieurs avantages. Elle est basée sur une notion d'extension d'un ensemble de clauses représentant le programme d'entrée. Cet ensemble a l'avantage d'avoir la même taille que le programme lo-

gique qu'il représente. C'est une très bonne alternative à la complétion de Clark [4] utilisée par la majorité des solveurs basés sur l'énumération à la DPLL. La méthode que nous présentons ici est dispensée de la lourdeur qu'implique l'utilisation de la complétion. A sa place, elle manipule un ensemble de clauses de Horn sur lequel elle effectue une énumération pour chercher les modèles stables. Pour un programme logique donné, la méthode calcule une extension en ajoutant à l'ensemble de clauses de Horn qui le représentent un ensemble maximal consistant de littéraux (*not* $A_i \in STB$) de l'ensemble *strong backdoor* STB . Chaque modèle stable correspond à une extension du programme considéré; la caractérisation de ce modèle se fait grâce à une condition discriminante facilement vérifiable [2].

La nouvelle sémantique est basée sur un langage propositionnel classique L ayant deux types de variables : un sous-ensemble de variables classiques $V = \{A_i : A_i \in L\}$ et un autre $nV = \{\text{not } A_i : \text{not } A_i \in L\}$. Pour chaque variable $A_i \in V$, il existe une variable correspondante *not* $A_i \in nV$ désignant la négation par échec de A_i . La nouvelle sémantique sémantique donne un lien entre les deux types de variables (celles de V et celles de nV). Ce lien est exprimé par l'ajout au langage propositionnel L d'un axiome exprimant l'exclusion mutuelle entre chaque littéral $A_i \in V$ et son littéral négatif correspondant *not* $A_i \in nV$. Cet axiome d'exclusion mutuelle est exprimé par l'ensemble de clauses $ME = \{(\neg A_i \vee \neg \text{not } A_i) : A_i \in V\}$.

Un programme logique $\pi = \{r : A_0 \leftarrow A_1, A_2, \dots, A_m, \text{not } A_{m+1}, \dots, \text{not } A_n\}$ où ($0 \leq m < n$) est exprimé dans le langage propositionnel L par un ensemble de clauses de Horn propositionnelles $CR = \{A_0 \vee \neg A_1 \vee \dots \vee \neg A_m \vee \neg \text{not } A_{m+1}, \dots, \neg \text{not } A_n\}$ où ($0 \leq m < n$) qui représente l'ensemble des règles du programme logique. Chaque règle $r \in \pi$ est traduite par une clause de Horn. Pour compléter la représentation du programme π dans cette nouvelle sémantique, on rajoute à l'ensemble des clauses exprimant les règles l'ensemble de clauses $ME = \{(\neg A_i \vee \neg \text{not } A_i) : A_i \in V\}$ exprimant l'axiome d'exclusion mutuelle. La représentation logique du programme π dans le langage propositionnel L est donnée par l'ensemble de clauses $CR \cup ME$. Un programme logique est donc exprimé par un ensemble de clauses de Horn :

$$L(\pi) = \left\{ \bigcup_{r \in \pi} (A_0 \vee \neg A_1 \vee \dots \vee \neg A_m \vee \neg \text{not } A_{m+1}, \dots, \neg \text{not } A_n) \right. \\ \left. \bigcup_{A_i \in V} (\neg A_i \vee \neg \text{not } A_i) \right\}.$$

Nous pouvons remarquer que la taille du codage $L(\pi)$ est de l'ordre de $\text{taille}(\pi) + 2n$, n étant le nombre de variables propositionnelles de π appartenant à V . Le facteur $2n$ dans la taille de $L(\pi)$ correspond à l'ensemble de clauses ME représentant l'axiome d'exclusion mutuelle. Mais ce dernier peut être implémenté comme une règle d'inférence sans avoir besoin de mémoriser l'ensemble de clauses ME .

La méthode que nous allons décrire travaillera alors avec la forme Horn clausale $L(\pi)$ ayant exactement la même taille que le programme π . Elle énumérera sur un sous-ensemble de variables jouant le rôle d'un strong backdoor (STB). Ce sous-ensemble est défini par $STB = \{not A_i : \exists r \in \pi, not A_i \in corps^-(r)\} \subseteq nV$. L'ensemble STB est le sous-ensemble des littéraux positifs de type $not A_i$ qui apparaissent dans π .

Étant donné un programme π et son ensemble STB. Une extension de $L(\pi)$ par rapport à STB (une extension de $(L(\pi), STB)$) est l'ensemble de clauses consistant obtenu à partir de $L(\pi)$ en ajoutant un ensemble maximal de littéraux $not A_i \in STB$. Formellement :

Definition 1 Soit $L(\pi)$ le codage CNF d'un programme logique π , STB son strong backdoor et un sous-ensemble $S' \subseteq STB$, l'ensemble $E = L(\pi) \cup S'$ de clauses est alors une extension de $(L(\pi), STB)$ si les conditions suivantes sont vérifiées :

1. E est consistant,
2. $\forall not A_i \in STB - S', E \cup \{not A_i\}$ est inconsistant.

Exemple 1 On considère le programme logique :

$$\pi = \left\{ \begin{array}{l} a \leftarrow c, not b \\ b \leftarrow a \\ c \leftarrow not d \\ a \leftarrow \end{array} \right\}$$

sa représentation Horn clausale est $L(\pi) = CR \cup ME$ où :

$$CR = \{a \vee \neg c \vee \neg not b, b \vee \neg a, c \vee \neg not d, a\}$$

$$ME = \{\neg a \vee \neg not a, \neg b \vee \neg not b, \neg c \vee \neg not c, \neg d \vee \neg not d\}$$

Son ensemble strong backdoor est $STB = \{not b, not d\}$ et $(L(\pi), STB)$ a une seule extension $E = L(\pi) \cup \{not d\}$. E est maximale consistante sur l'ensemble STB. En effet, si on rajoute $not b$ à l'extension E , l'ensemble devient inconsistant.

Il a été démontré dans [2] que chaque modèle stable d'un programme logique π est représenté par une extension E de sa forme logique $L(\pi)$ qui vérifie la condition discriminante $(\forall A_i \in V, E \models \neg not A_i \Rightarrow E \models A_i)$. Certaines extensions de $L(\pi)$ ne correspondent à aucun modèle stable. Ces extra-extensions coïncident avec des extra-modèles qui représentent une extension à la sémantique des modèles stables [8]. La caractérisation d'un modèle stable se fait de manière très simple grâce à une condition simple à vérifier que doivent satisfaire les extensions correspondant aux modèles stables. Cette condition est appelée condition discriminante dans [2]. Formellement, nous avons :

Théorème 1 Si X est un modèle stable d'un programme logique π , alors il existe une extension E de $(L(\pi), STB)$ telle que $X = \{A_i \in V : E \models A_i\}$. D'autre part, E vérifie la condition dite discriminante : $(\forall A_i \in V, E \models \neg not A_i \Rightarrow E \models A_i)$.

Preuve 1 La preuve est donnée dans [2].

Théorème 2 Si E est une extension de $(L(\pi), STB)$, qui vérifie la condition discriminante $(\forall A_i \in V, E \models \neg not A_i \Rightarrow E \models A_i)$, alors $X = \{A_i : E \models A_i\}$ est un modèle stable de π .

Preuve 2 La preuve est donnée dans [2].

Remarque 1 Comme toute extension E est formée par un ensemble de clauses de Horn, alors la résolution unitaire est suffisante pour déduire tout atome A_i à partir de E ($E \models A_i$).

Exemple 2 Soit le programme logique π composé de deux règles :

$$\pi = \left\{ \begin{array}{l} a \leftarrow not b \\ b \leftarrow not a \end{array} \right\}$$

Sa représentation clausale est : $L(\pi) = CR \cup ME$ où $CR = \{a \vee \neg not b, b \vee \neg not a\}$ et où $ME = \{\neg a \vee \neg not a, \neg b \vee \neg not b\}$. Son ensemble strong backdoor est $STB = \{not a, not b\}$. La paire $(L(\pi), STB)$ a deux extensions $E_1 = L(\pi) \cup \{not a\}$ et $E_2 = L(\pi) \cup \{not b\}$ qui vérifient la condition discriminante. On peut facilement voir que $E_1 \models \{b, \neg a, \neg not b\}$ et que $E_2 \models \{a, \neg b, \neg not a\}$. A partir de E_1 resp. E_2 , on obtient, par résolution unitaire, les ensembles de littéraux positifs impliqués $X_1 = \{b\}$ resp. $X_2 = \{a\}$ qui sont les deux modèles stables du programme π .

Comme π est un programme logique général, toute extension de $(L(\pi), STB)$ est un ensemble consistant de clause de Horn. Donc, tous les littéraux positifs (A_i) seront inférés par résolution unitaire. Ce qui réduit la complexité de l'algorithme de calcul de modèles stables résultant de la nouvelle sémantique.

3 Présentation de la nouvelle méthode

Nous présenterons ici un algorithme de recherche de modèles stables basé sur la nouvelle sémantique [2]. Étant donné un programme logique π , cet algorithme calcule toutes les extensions de $(L(\pi), STB)$ à partir desquelles les modèles stables seraient déduits par résolution unitaire. Intuitivement, la recherche d'extension de $(L(\pi), STB)$ se fait par l'ajout progressive de littéraux $not A_i$ du STB en vérifiant à chaque ajout la consistance de l'ensemble obtenu. Ensuite, si on ne s'intéresse qu'aux modèles stables, il suffit de retenir que les extensions qui vérifient la condition discriminante. En d'autres mots, on élimine les extra-extensions qui ne vérifient pas celle-ci.

Il existe deux grandes approches dans la conception de systèmes ASP pour le calcul de modèles stables. L'une d'entre elles s'appuie sur les propriétés de la sémantique

utilisée pour concevoir le système, et l'autre transforme le programme logique considéré en un problème de satisfaisabilité booléenne (SAT) selon la complétion de Clark pour lequel on pourrait utiliser des solveurs SAT dans la résolution. Notre algorithme fait partie de la première approche. Nous avons adapté la procédure DPLL au cadre de l'ASP traité dans la nouvelle sémantique. Le principe de notre méthode de calcul est le suivant : on construit incrémentalement une extension en alternant dans l'arbre de recherche des nœuds déterministes correspondant aux propagations unitaires et des nœuds non déterministes appelés points de choix définis par l'affectation d'une valeur de vérité (vrai ou faux) à un littéral de l'ensemble strong backdoor STB . Nous introduisons quelques nouvelles règles d'inférence qui permettent d'augmenter le nombre de propagations unitaires et, par conséquent, réduire l'espace de recherche. L'algorithme que nous présentons permet de calculer tous les modèles stables d'un programme logique donné.

3.1 Fondements théoriques de la méthode

Nous allons maintenant introduire quelques règles d'inférence que la méthode utilisera par la suite dans le processus d'énumération de modèles stables.

Definition 2 Soit un programme π et $L(\pi) = CR \cup ME$ sa forme clausale. On définit sur $L(\pi)$ les deux règles d'inférence suivantes :

$$\frac{A_i}{\neg not A_i} \text{ et } \frac{not A_i}{\neg A_i}.$$

Tout programme logique général $\pi = \{r : A_0 \leftarrow A_1, A_2, \dots, A_m, not A_{m+1}, \dots, not A_n\}$, ($0 \leq m < n$) est exprimé par l'ensemble de clauses propositionnelles de Horn $L(\pi) = \{\bigcup_{r \in \pi} (r : A_0 \vee \neg A_1 \vee, \dots, \neg A_m \vee \neg not A_{m+1}, \dots, \neg not A_n) \bigcup_{A_i \in V} (\neg A_i \vee \neg not A_i)\}$ qui représente son codage *CNF* dans le langage propositionnel L . Les deux règles d'inférence se justifient par la présence de l'ensemble de clauses $ME = \bigcup_{A_i \in V} \{(\neg A_i \vee \neg not A_i)\}$ exprimant l'exclusion mutuelle entre toute paire d'atomes A_i et $\neg not A_i$.

Avec la considération de ces deux règles dans le raisonnement de la méthode, on pourrait supprimer de $L(\pi)$ le sous-ensemble de clauses ME des exclusions mutuelles. On obtient ainsi une taille de $L(\pi)$ identique à celle du programme π . La forme clausale du programme logique π serait réduite à : $L(\pi) = \{\bigcup_{r \in \pi} (r : A_0 \vee \neg A_1 \vee, \dots, \neg A_m \vee \neg not A_{m+1}, \dots, \neg not A_n)\}$

L'utilisation et l'implémentation des deux règles d'inférence introduites précédemment dans le processus de résolution permettraient d'accroître le nombre de propagations unitaires et élagueraient l'arbre de recherche.

La complexité d'une méthode d'énumération dans un espace de recherche représenté par un arbre binaire est souvent calculée en fonction du nombre de points de choix effectués. Dans le cas de notre méthode, l'énumération est faite sur le sous-ensemble de variables $STB = \{not A_i : \exists r \in \pi, A_i \in r^-\}$ qui représente le strong backdoor. Soit $C_{STB} = \{c_i = \neg not A_{i_1} \vee, \dots, \vee \neg not A_{i_k} / | c_i | \geq 1, \forall j \in \{1..k\}, not A_{i_j} \in STB\}$ l'ensemble de clauses négatives possibles formées par les littéraux du STB ayant au moins un littéral. L'assignation non déterministe d'un point de choix correspondant à la variable $not A_j$ est faite en lui affectant en premier lieu la valeur de vérité vrai pour favoriser la maximalité de l'extension en cours. L'exploration de la branche correspondant à l'assignation de la valeur de vérité faux à $not A_j$ (où en affectant $\neg not A_j$ à vrai) n'est nécessaire que dans le cas où la première branche aurait produit au moins une sous clause $c_i \in C_{STB}$. Cette propriété, que nous allons démontrer par la suite, permettrait de réduire considérablement la complexité de l'algorithme étudié. L'algorithme vérifie à chaque affectation d'un nouveau littéral la consistance du système de clauses présent en ce nœud. Comme nous avons un ensemble de clauses de Horn, une propagation par résolution unitaire assure ce teste de consistance.

Proposition 1 Soit π un programme logique, $L(\pi)$ sa forme Horn clausale, $L(\pi)_I$ sa forme Horn clausale simplifiée par l'instanciation partielle I correspondant au nœud courant n de l'arbre de recherche, $STB = \{not A_i : \exists r \in \pi, not A_i \in r^-\}$ son strong backdoor, et C_{STB} l'ensemble de clauses négatives possibles construites sur les littéraux de l'ensemble STB . Si $not A_j \in STB$ est le littéral courant à affecter au nœud n et que $\forall c_i \in C_{STB}, L(\pi)_I \wedge not A_j \not\models c_i$, alors toute extension de $L(\pi)_I \wedge \neg not A_j$ est aussi une extension de $L(\pi)_I \wedge not A_j$.

Preuve 3 Le sous-ensemble de clauses $L(\pi)_I$ est le système de clauses simplifié, obtenu à partir de $L(\pi)$ par la considération des littéraux interprétés dans l'instanciation partielle I . L'ensemble de clauses $L(\pi)_I$ représente le sous-problème correspondant au nœud courant n de l'arbre de recherche. Par hypothèse $not A_j$ est le prochain littéral du STB à affecter en ce point n de l'arbre. Le système de clauses simplifié $L(\pi)_I$ correspondant au nœud n contient deux sortes de clauses : le sous ensemble de clauses de la forme $\neg not A_j \vee C_1$ contenant le littéral $\neg not A_j$ et où C_1 représente un ensemble de bouts de clauses, et le sous ensemble clauses de C_2 ne contenant pas le littéral $not A_j$. Soit $e = not A_{i_1} \wedge \dots \wedge not A_{i_k}$, avec $not A_{i_j} \in STB$ une extension de $L(\pi)_I \wedge \neg not A_j$, montrons que e est aussi une extension de $L(\pi)_I \wedge not A_j$. On peut remarquer que $L(\pi)_I \wedge \neg not A_j \equiv C_2$ et que $L(\pi)_I \wedge not A_j \equiv C_1 \wedge C_2$. L'ensemble e est une extension de $L(\pi)_I \wedge \neg not A_j$, donc $C_2 \wedge e$ est consistant. Pour montrer que e est aussi une extension de $L(\pi)_I \wedge not A_j$, il suffit de montrer que $C_1 \wedge$

$C_2 \wedge e$ est consistant. Procédons par l'absurde, en supposant que $C_1 \wedge C_2 \wedge e$ est inconsistent. Il en résulte que $C_1 \wedge C_2 \wedge e \models \square$ et donc $C_1 \wedge C_2 \models \neg e$. Ce qui veut dire que $C_1 \wedge C_2 \models \neg \text{not } A_{i_1} \vee \dots \vee \neg \text{not } A_{i_k} \in C_{STB}$. Il en résulte que $L(\pi)_I \wedge \text{not } A_j \models \neg \text{not } A_{i_1} \vee \dots \vee \neg \text{not } A_{i_k} \in C_{STB}$. Donc $L(\pi)_I \wedge \text{not } A_j \models c_i \in C_{STB}$ et cela contredit l'hypothèse.

En d'autres mots, si aucune clause $c_i \in C_{STB}$ n'a été produite en un point de choix de l'arbre où on a interprété à vrai un littéral $\text{not } A_j \in STB$, il est alors inutile d'explorer la branche correspondant au littéral négatif $\neg \text{not } A_j$. Cela éviterait à la méthode d'explorer des branches redondantes et inutiles. En conséquence, cette propriété permet de réduire le nombre de points de choix de l'arbre de recherche. On a introduit ainsi une nouvelle coupure dans l'arbre de recherche qui pourrait réduire la complexité de l'algorithme et qui pourrait augmenter son efficacité dans la pratique.

Nous allons maintenant montrer comment exploiter l'apparition de certains littéraux purs (monotones) dans la forme clausale $L(\pi)$ du programme logique π . Nous rappelons qu'un littéral pur (monotone) est un littéral qui apparaît dans une seule forme, soit dans sa parité positive ou dans sa parité négative. Ces littéraux sont souvent ignorés dans les implémentations des solveurs SAT basé sur DPLL, mais, pour les solveur ASP, ils jouent un rôle très important.

Proposition 2 Soit un programme logique π , $L(\pi)$ sa forme clausale et $\neg A_i$ un littéral pur de $L(\pi)$, si X est un modèle stable ou un extra-modèle de π alors $\neg A_i \in X$.

Preuve 4 Le littéral $\neg A_i$ est pur dans $L(\pi)$, ce qui implique que A_i n'a pas d'occurrences dans $L(\pi)$, donc le littéral A_i ne pourra jamais être inféré. Par conséquent, A_i ne pourra faire partie d'un modèle stable / extra-modèle X . Par l'application de l'hypothèse du monde clos, nous avons $\neg A_i \in X$.

Cette proposition permet de traiter les littéraux purs comme des mono-littéraux à propager. La propagation de ces derniers contribuent à la réduction des points de choix de l'arbre et, en conséquence, son élagation.

La propriété qui va suivre est restreinte au cas de modèles stables. Elle n'est pas vérifiée dans le cas d'extra-modèles.

Proposition 3 Soit un programme logique π et $L(\pi)$ sa forme clausale, si $\neg A_i$ est vrai dans un modèle stable X de π alors $\text{not } A_i$ doit être vrai dans X .

Preuve 5 Si $\neg A_i$ est vrai dans le modèle stable X , alors le seul cas où $\neg \text{not } A_i$ pourrait être produit est l'existence d'une sous-clause $A_i \vee \neg \text{not } A_i$ de $L(\pi)_X$. Mais dans ce cas, l'extension correspondant à X ne vérifie pas la condition discriminante. En conséquence, X ne serait pas un modèle stable, ce qui est contradictoire avec l'hypothèse.

La dernière proposition peut se traduire dans l'algorithme par la règle d'inférence ($\frac{\neg A_i}{\text{not } A_i}$) qui permet d'élaguer l'arbre de recherche des modèles stables. cette règle garantit la complétude de l'algorithme sur les modèles stables mais pas sur les extra-modèles. En effet, l'application de cette dernière pourrait supprimer des extra-modèles correspondant à des extra-extensions. Nous ne l'appliquons que pour la recherche de modèles stables. Toutes les propositions énoncées précédemment s'expriment par des règles d'inférence que nous avons implémentées dans la nouvelle méthode. Ces différentes règles induisent des coupures dans l'arbre de recherche et réduisent de manière considérable l'espace de recherche.

Proposition 4 Si $L(\pi)$ est la forme clausale d'un programme logique π et I l'instanciation partielle courante, alors la résolution unitaire est suffisante pour produire toute clause $c_i = \neg \text{not } A_{i_1} \vee \dots \vee \neg \text{not } A_{i_k} \in C_{STB}$ à partir de $L(\pi)_I$.

Preuve 6 D'après le théorème de déduction automatique, montrer que $L(\pi)_I \models c_i$ est équivalent à $L(\pi)_I \wedge \neg c_i \models \perp$. Comme $L(\pi)$ est à la base un ensemble de clauses de Horn, il en résulte que l'ensemble de clauses simplifié $L(\pi)_I \wedge \neg c_i$ l'est aussi car $L(\pi)_I \wedge \text{not } A_{i_1} \wedge \dots \wedge \text{not } A_{i_k}$ est trivialement un ensemble de clauses de Horn. Comme la résolution unitaire est suffisante pour décider la consistance de tout ensemble de clauses de Horn, alors elle est en particulier pour $L(\pi)_I \wedge \neg c_i$. En d'autres mots, la résolution unitaire est suffisante pour montrer $L(\pi)_I \wedge \neg c_i \models \perp$ et, par conséquent, suffisante pour montrer $L(\pi)_I \models c_i$.

Pour appliquer la coupure induite par la proposition 2 en un point de choix donné de l'arbre de recherche, notre méthode doit prouver qu'aucune clause $c_i \in C_{STB}$ n'est produite en ce nœud. Pour ce faire, la méthode essaye de produire une telle clause en utilisant la résolution unitaire (Proposition 4).

3.2 Description de l'algorithme

Nous présentons dans ce qui va suivre le nouvel algorithme de recherche de modèles stables. Le processus d'énumération de ce dernier est basé sur la procédure DPLL que nous avons adaptée au cas des ASP traités par la nouvelle sémantique [2]. Nous avons aussi introduit et implémenté plusieurs nouvelles règles d'inférences pour booster la méthode. Dans ce nouvel algorithme, la recherche de modèle stable alterne des phases de propagation unitaire déterministes et des phases de points de choix non déterministes où le processus de production de clauses $c_i \in C_{STB}$ est lancé sur la première branche du point de choix où un littéral $\text{not } A_i$ du STB est interprété à vrai. Le processus de production est inutile sur la deuxième branche du point de choix où le littéral $\text{not } A_i$ du STB est interprété à

faux. Durant les deux phases alternées, l'algorithme affecte des valeurs de vérité aux littéraux à la manière DPLL. Si un conflit est rencontré au cours de la recherche, alors l'algorithme explore la branche correspondant à la deuxième valeur de vérité de la variable représentant le point de choix courant uniquement si une clause $c_i \in C_{STB}$ est produite. Sinon, un rebroussement (backtrack) est effectué.

Une extension est trouvée quand toutes les clauses sont satisfaites ou bien quand tous les littéraux du STB ont été affectés sans falsifier aucune clause. Dans les deux cas, l'algorithme exécute une phase dite de complétion. Dans le premier cas, la méthode complète l'interprétation courante par l'assignation à vrai de l'ensemble des variables *not* A_i restant du STB et par l'assignation à faux de toutes les autres variables non encore affectés (hypothèse du monde clos). Dans le deuxième cas, la complétion consiste selon l'hypothèse du monde clos, à mettre à faux les variables non affectées. Dans les deux cas, un modèle minimal candidat est trouvé. L'algorithme vérifie alors si le modèle candidat est bien un modèle stable.

Algorithm 1 Schéma général de la nouvelle méthode de recherche de modèles stables

Entrées La forme clausale $l(\pi)$ d'un programme logique π
Sorties Tous les modèles stables de π

```

1: répéter
2:   tant que  $STB \neq \emptyset$  et 'pas de conflit' Faire
3:     tant que  $Lmonos \neq \emptyset$  ou  $Lpurs \neq \emptyset$  Faire
4:       propagation-unitaire( $L(\pi), Lmonos, I$ );
5:       inférence( $L(\pi), Lpurs, I$ );
6:       production-clause( $L(\pi)$ );
7:     fin tant que
8:     choisir littéral;
9:   fin tant que
10:  si pas de conflit alors
11:     $E = L(\pi)_I$  est une extension;
12:     $E =$  complétion( $E$ );
13:    Condition-discriminante( $E$ )
14:  sinon
15:    backtrack
16:  fin si
17: jusqu'à

```

L'algorithme commence par un premier appel à la procédure propagation-unitaire qui propage tous les mono-littéraux jusqu'à ce que la liste de ces derniers $Lmono$ se vide. Puis un appel est fait pour traiter les littéraux purs qui à leur tour peuvent induire des mono-littéraux. Quand il n'y a plus de mono-littéraux ou des littéraux purs à assigner, l'algorithme essaye de produire une clause $c_i \in C_{STB}$ en utilisant la proposition 4.

Si on arrive à produire une clause $c_i \in C_{STB}$, alors la seconde branche du point de choix courant sera explorée. Si, par contre, aucune clause n'est produite et qu'il ne reste

pas de mono-littéraux ou littéraux purs à propager, alors la seconde branche de la variable point de choix devient inutile et donc coupée. L'énumération continue par le choix du prochain littéral dans STB à assigner et ce processus est réitéré jusqu'à la satisfaction de toutes les clauses ou l'affectation de tous les littéraux du STB sans apparition de la clause vide. Dans ce cas, une extension $E = L(\pi)_I$ est obtenue. Il ne reste plus qu'à effectuer la phase de complétion, ensuite vérifier si l'extension obtenue satisfait la condition dite discriminante pour induire un modèle stable. Le pseudo-code du schéma général de la méthode est donné dans l'algorithme 1.

Algorithm 2 Procédure propagation-unitaire

Entrées La forme clausale $L(\pi)$ du programme π , la liste de clauses unitaires $Lmonos$, l'interprétation partielle courante I

Sorties Une interprétation étendue I ou la détection d'un conflit,

```

1: tant que ( $Lmonos = \emptyset$ ) et non (conflit) Faire
2:    $v \leftarrow next(Lmonos)$ ;
3:    $I \leftarrow I \cup \{v\}$ ;
4:    $L(\pi) \leftarrow L(\pi) \setminus \{c_i, v \in c_i\}$ ;
5:    $c_i \leftarrow c_i \setminus \{\neg v, \neg v \in c_i\}$ ;
6:   si longueur( $c_i$ ) == 1 alors
7:      $Lmonos \leftarrow Lmonos \cup \{c_i\}$ 
8:   fin si
9:    $Lmonos = Lmonos \setminus \{v\}$ ;
10:   $v' \leftarrow inference(v)$ ;
11:   $Lmonos = Lmonos \cup \{v'\}$ ;
12: fin tant que
13: Si no(conflit) alors retourner  $I$ ;
14: else retourner conflit;

```

La procédure de propagation-unitaire (Algorithme 2) prends en entrée la forme clausale $L(\pi)$, la liste de clauses unitaires $Lmonos$, et l'interprétation partielle courante I . Elle renvoie au retour, soit une interprétation I étendue par la propagation des mono-littéraux, soit un message de conflit si une clause vide est rencontrée. La procédure commence par satisfaire toutes les clauses où apparaît le mono-littéral v et rajoute v à l'interprétation partielle I . Puis, elle réduit les clauses dans lesquelles se trouve l'opposé de v ($\neg v$). Si une clause unitaire est créée, elle est alors rajoutée à la liste $Lmonos$ qui représente l'ensemble des mono-littéraux. Si une clause vide est détectée, la procédure signale le conflit.

Ensuite, l'algorithme fait appel à la fonction *inférence* (Algorithme 3) qui implémente certaines règles d'inférences basées sur des propriétés théoriques que nous avons démontrées dans la sous-section précédente. Ces règles d'inférence permettent de réduire l'ensemble des points de choix de l'arbre de recherche. Le littéral v' retourné par la procédure *inférence* sera traité comme un mono-littéral.

Algorithm 3 Procédure inférence

Entrées Un littéral v **Sorties** Un littéral v'

- 1: **si** $v == A_i$ **alors**
 - 2: $v' \leftarrow \neg not A_i$
 - 3: **fin si**
 - 4: **si** $v == \neg A_i$ **alors**
 - 5: $v' \leftarrow not A_i$
 - 6: **fin si**
 - 7: **si** $v == not A_i$ **alors**
 - 8: $v' \leftarrow \neg A_i$
 - 9: **fin si**
-

Enfin, La procédure *production-clauses* utilise la unit résolution pour produire les clauses $c_i \in C_{STB}$ selon la proposition 4.

3.3 Complexité de l'algorithme

Si n est le nombre de variables de la forme clauseuse $L(\pi)$ du programme π , k le cardinal de son STB et m le nombre de clauses de $L(\pi)$, alors la complexité dans le pire des cas de l'algorithme est de l'ordre de $O(knm2^k)$.

Quant à la complexité spatiale, l'algorithme a l'avantage de travailler à espace constant. Il utilise la forme clauseuse $L(\pi)$ dont la taille est identique à la taille du programme d'entrée π . La complexité spatiale est constante, elle est de l'ordre de $O(|L(\pi)|) = O(|\pi|)$. Cet algorithme peut être utilisé pour des programmes logiques avec boucles et permet de calculer tous les modèles stables d'un programme général.

4 Expérimentation

Nous allons dans ce qui va suivre expérimenter notre première implémentation de la méthode sur des instances du problème de coloriage de graphes générées d'une façon aléatoire. Nous évaluons la performance de l'algorithme par rapport au temps de réponse pour trouver le premier modèle stable ou bien pour prouver qu'il n'existe aucun modèle stable. Nous donnons aussi le nombre de point de choix de l'arbre de recherche car il reflète la taille de l'arbre de recherche.

Il ne s'agit ici que d'une première implémentation de la méthode de base utilisant la nouvelle sémantique. Nous sommes encore loin des performances des méthodes optimisées comme *Clasp* [15]. Cette première implémentation et ces premières expérimentation vont juste servir pour étudier le comportement de la méthode et montrer une nouvelle façon de concevoir un système ASP. La comparaison avec les autres méthodes est nécessaire, nous envisageons donc de la faire après avoir introduit les optimisations nécessaires à la méthode.

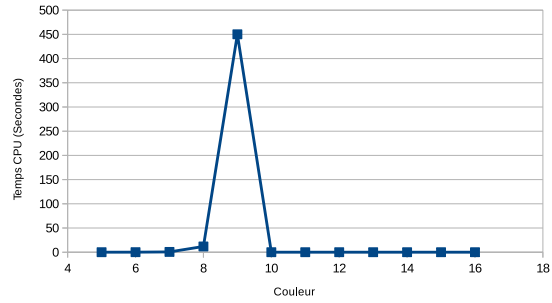


FIGURE 2 – Courbe des temps CPU d'instances de coloration de graphe générés aléatoirement avec $n=30$ et $d=0.5$

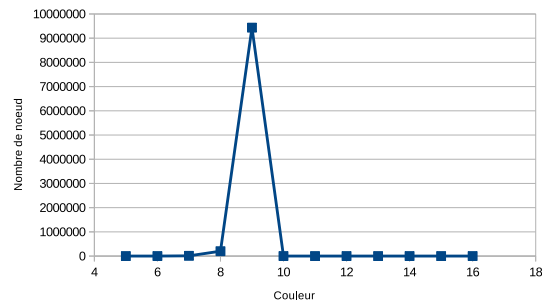


FIGURE 3 – Courbe des nœuds d'instances de coloration de graphe générés aléatoirement avec $n=30$ et $d=0.5$

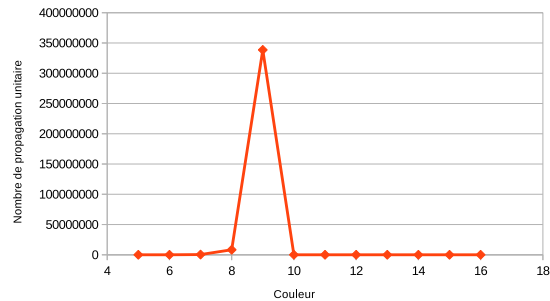


FIGURE 4 – Courbe des propagations d'instances de coloration de graphe générés aléatoirement avec $n=30$ et $d=0.5$

Les instances de coloriage de graphes que nous avons testées sont générées à base de trois paramètres d'entrée nécessaires pour le générateur : n le nombre de sommets du graphe, c le nombre de couleurs et d la densité du graphe qui est un nombre entre 0 et 1 exprimant le ratio du nombre d'arêtes dans le graphe sur le nombre de toutes les arêtes possibles.

La figure 2 montre les courbes moyennes représentant les temps CPU pour l'algorithme par rapport à une varia-

Couleurs	Temps(sec)	Points de choix	Propagations unitaires
5	0,024509	239	8600
6	0,114816	1439	52795
7	0,587897	10079	381716
8	11,816976	201599	8297582
9	450,134003	9434879	338575324
10	0,001291	94	230
11	0,001665	126	228
12	0,001691	156	228
13	0,002051	186	228
14	0,002315	216	228
15	0,00267	246	228
16	0,003071	276	228

FIGURE 5 – Résultats détaillés sur des instances de coloriage de graphes générés aléatoirement avec $n=30$ et $d=0.5$

tion du nombre de couleurs. On peut observer l'existence d'un pic de difficulté du problème de coloriage de graphes aléatoire. Ce pic se situe en général dans la zone du nombre chromatique (nombre minimal de couleurs pour colorier un graphe). Il correspond à une valeur critique du nombre de couleurs. Cette valeur représente en général le dernier nombre de couleurs avant le nombre chromatique.

La figure 3 et la figure 4 montrent les courbes moyennes représentant le nombre de points de choix ainsi que le nombre de propagations unitaires de l'algorithme par rapport à une variation du nombre de couleurs. Nous pouvons constater une augmentation significative du nombre de propagations unitaires dans la zone du pic de difficulté. Le nombre de points de choix est aussi plus élevé dans la zone du pic de difficulté. Nous pouvons remarquer que la méthode de base arrive à résoudre des instances de coloriage de graphe aléatoires dans la zone difficile. La figure 5 résume les résultats du problème de coloriage de graphes et donne des détails, la ligne en gras représente le pic de difficulté.

5 Conclusion et perspectives

Nous avons étudié dans cet article une méthode de calcul de modèles stables basée sur la nouvelle sémantique [2]. Cette méthode utilise une forme logique clauseuse ayant une taille identique au programme logique d'entrée. Nous évitons ainsi à la méthode toute lourdeur qu'aurait causée une représentation par la complétion de Clark implémentée dans la plupart des solveurs basés sur DPLL. L'autre avantage est que l'énumération est faite sur un sous-ensemble de variables STB représentant le strong backdoor du programme logique d'entrée. Nous avons proposé un ensemble de règles d'inférence qui permettent de réduire le nombre de points de choix et, par conséquent, la taille de l'arbre de recherche. L'algorithme de base a été implémenté et testé sur des instances de coloration de

graphes. Les premiers résultats obtenus sont satisfaisants. Notre but dans cet article est de montrer une nouvelle façon de programmer des systèmes ASP. Nous souhaitons avoir les premiers retours sur cette nouvelle méthode.

En perspectives, nous comptons tout d'abord optimiser notre implémentation par l'introduction du matériel utilisé dans les solveurs SAT modernes tel que l'apprentissage de clauses, le redémarrage, les structures paresseuses. Nous envisageons par la suite tester la méthode sur une grande variété de problèmes et la comparer par rapport aux méthodes connues dans le domaine.

Notre approche peut être étendue à d'autres classes de la programmation logique ou à des fragments de logiques non-monotone plus générales telle que la logique des hypothèses [3]. C'est un point que nous envisageons d'étudier dans le futur.

Références

- [1] A Van Gelder, KA Ross, JS Schlipf: *The well-founded semantics for general logic programs*. Journal of the ACM (JACM), 38:619–649, 1991.
- [2] Belaïd Benhamou, Pierre Siegel: *A new semantics for logic programs capturing and extending the stable model semantics*. Tools with Artificial Intelligence (ICTAI), pages 25–32, 2012.
- [3] Camilla Schwind, Pierre Siegel: *A modal logic for hypothesis theory*. Fundamenta Informaticae, 21:89–101, 1994.
- [4] Clark, Keith L: *Negation as failure*. Logic and data bases, pages 293–322, 1978.
- [5] Esra Erdem, Michael Gelfond, Nicola Leone: *Applications of answer set programming*. AI Magazine, 37:53–68, 2016.
- [6] Fages, Francois: *Consistency of clark's completion and existence of stable models*. Methods of Logic in Computer Science, 1:51–60, 1994.
- [7] Fangzhen Lin, Yuting Zhao: *Assat: Computing answer sets of a logic program by sat solvers*. Artificial Intelligence, pages 115–137, 2004.
- [8] Gelfond, Vladimir Lifschitz: *The stable model semantics for logic programming*. ICLP/SLP, 50:1070–1080, 1988.
- [9] Gilles Audemard, Belaïd Benhamou, Pierre Siegel: *Aval: An enumerative method for sat*. Computational Logic—CL 2000, pages 373–383, 2000.
- [10] Kaufmann, Benjamin et al: *Grounding and solving in answer set programming*. AI Magazine, 37:25–32, 2016.
- [11] Leone, Nicola et al: *The dlvs system for knowledge representation and reasoning*. ACM Transactions on Computational Logic (TOCL), 7:499–562, 2006.

- [12] Michael Gelfond, Vladimir Lifschitz: *Classical negation in logic programs and disjunctive databases*. *New generation computing*, 9:365–385, 1991.
- [13] Patrik Simons, Ilkka Niemelä, Timo Soininen: *Extending and implementing the stable model semantic*. *Artificial Intelligence*, 138:181–234, 2002.
- [14] Ryan Williams, Carla P Gomes, Bart Selman: *Backdoors to typical case complexity*. *INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE*, 18:1173–1178, 2003.
- [15] Schaub, Torsten et al: *Conflict-driven answer set solving*. *IJCAI*, 7:386–392, 2007.
- [16] Tomi Janhunen, Ilkka Niemelä: *The answer set programming paradigm*. *AI Magazine*, 37:13–24, 2016.
- [17] Victor W. Marek, Mirosław Truszczynski: *Stable models and an alternative logic programming paradigm*. *The Logic Programming Paradigm*, pages 375–398, 1999.
- [18] Vladimir Lifschitz, Alexander Razborov: *Why are there so many loop formulas?* *ACM Transactions on Computational Logic (TOCL)*, 7:261–268, 2006.

Sur la contraction itérée

Sébastien Konieczny¹

Ramon Pino Pérez²

¹ CRIL, CNRS - Université d'Artois - France

² Facultad de Ciencias - Universidad de los Andes - Venezuela

konieczny@cril.fr

pino@ula.ve

Résumé

Dans cet article nous nous intéressons aux opérateurs de contraction itérée, afin de fournir une caractérisation comarable à celle obtenue par Darwiche et Pearl pour la révision itérée. Nous proposons des postulats pour la contraction itérée, et donnons un théorème de représentation pour ces opérateurs. Nos résultats permettent de mettre en lumière les liens entre révision itérée et contraction itérée. Nous montrons en particulier que les opérateurs de révision itérée forment une classe plus large que les opérateurs de contraction itérée. Une conséquence importante de cela est que, dans le cadre des états épistémiques, les identités de Levi et de Harper ne sont pas aussi puissantes que dans le cadre AGM classique, et le lien entre révision itérée et contraction itérée est différent que celui que l'on obtient dans le cadre AGM.

Abstract

In this paper we study iterated contraction in the epistemic state framework, offering a counterpart of the work of Darwiche and Pearl for iterated revision. We provide pure syntactical postulates for iterated contraction, that is, the postulates are expressed only in terms of the contraction operator. We establish a representation theorem for these operators. Our results allow to highlight the relationships between iterated contraction and iterated revision. In particular we show that iterated revision operators form a larger class than that of iterated contraction operators. As a consequence of this, we have that, in the epistemic state framework, the Levi identity has limitations; namely, it doesn't allow to define all iterated revision operators.

1 Introduction

La théorie du changement de croyances [1, 8, 11, 12, 9, 7] a pour objet la modélisation de l'évolution des croyances logiques d'un agent lorsque celui-ci reçoit de nouvelles informations.

Les deux principales classes d'opérateurs de changements sont les opérateurs de révision, qui permettent de

corriger des croyances fausses d'un agent, et les opérateurs de contraction, qui permettent de retirer certaines informations des croyances de l'agent.

Contraction et révision, bien qu'étant des opérations différentes, sont étroitement liées. Deux identités permettent de définir la contraction à partir de la révision et vice-versa.

Il est possible de définir un opérateur de révision à partir d'un opérateur de contraction grâce à l'identité de Levi, qui dit que pour définir une révision par la formule α il est possible de d'abord effectuer une contraction par $\neg\alpha$, suivie d'une expansion¹ par α [15, 16].

Réciproquement, il est possible de définir la contraction à partir d'un opérateur de révision en utilisant l'identité de Harper : ce qui est vrai après une contraction par α est ce qui est vrai dans l'état de croyances courant et dans celui qui est le résultat de la révision par $\neg\alpha$ [10] (donc, intuitivement, ce qui est vrai indépendamment du fait que α soit vrai).

Afin de donner une définition formelle à ces identités, définissons K comme une théorie (en ensemble déductivement clos de formules logiques) et prenons une formule α . On note \star un opérateur de révision, \div un opérateur de contraction, et \oplus un opérateur d'expansion

$$\begin{array}{ll} \text{Identité de Levi} & K \star \alpha = (K \div \neg\alpha) \oplus \alpha \\ \text{Identité de Harper} & K \div \alpha = K \cap (K \star \neg\alpha) \end{array}$$

La connection obtenu avec ces identités est très forte, puisque l'on obtient en fait une bijection entre l'ensemble des opérateurs de révision et l'ensemble des opérateurs de contraction [8]. Donc, dans le cadre AGM, ces deux classes d'opérateurs sont les deux faces d'une même pièce de monnaie, et on peut donc choisir d'étudier soit la révision, soit la contraction, suivant l'opération que l'on considère être la plus naturelle ou élémentaire.

Bien que cette théorie du changement est intrinséquement un processus dynamique, les travaux initiaux ne s'in-

1. Voir [8] pour une définition exacte, mais il est possible d'identifier l'expansion avec la conjonction (ou l'union) dans la plupart des cas.

téressent qu'au cas (statique) d'une seule étape de changement [1, 8, 11], et ne permettent pas de capturer le changement itéré de manière convaincante.

Après de nombreuses tentatives infructueuses, la solution pour la modélisation de la révision itérée a été proposée par Darwiche et Pearl dans [6]. Ils ont proposé des postulats supplémentaires pour la révision itérée. Ces contraintes additionnelles pour l'itération ne peuvent pas se satisfaire de la représentation usuelle (théories) pour les croyances de l'agent, utilisée dans le cadre AGM classique. Il est nécessaire d'utiliser une représentation plus expressive, les états épistémiques (*epistemic states*), qui permettent d'encoder également la stratégie de révision de l'agent, et qui permettent d'assurer la cohérence de la dynamique du changement.

Il est intéressant de noter que le travail de Darwiche et Pearl a été consacré à la révision itérée. On pourrait espérer que la caractérisation de la contraction itérée pourrait être obtenue directement à partir de généralisation des identités. Mais, malheureusement, ce n'est pas le cas. Tout d'abord, jusqu'ici, il n'y a eu aucune proposition de postulats, ni de théorème de représentation pour la contraction itérée. Il y a eu quelques travaux sur la contraction itérée, que nous discuterons à la section sur les travaux connexes, mais il n'y a eu aucune contrepartie du travail de Darwiche et Pearl pour la contraction. C'est ce que nous proposons dans cet article.

La classe des opérateurs de contraction itérée obtenue est très intéressante à différents titres. Cela permet d'obtenir une meilleure compréhension de la théorie du changement de croyances. En fait, certaines des conséquences de notre théorème de représentation sont assez surprenantes.

Tout d'abord, lorsque l'on compare les deux théorèmes de représentation (celui pour la contraction itérée et celui pour la révision itérée), il est clair qu'ils utilisent la même classe d'assignements (fidèles). Cela signifie que la différence entre les opérateurs de contraction itérée et de révision itérée n'est pas une question de nature, mais une question de degré (la révision étant un changement plus important que la contraction, mais pas un autre type de changement).

La conséquence la plus importante et la plus surprenante est que nous montrons qu'il y a plus d'opérateurs de révision itérée que d'opérateurs de contraction itérée, contrairement à la bijection obtenue dans le cadre AGM classique.

Et une conséquence de cela est qu'il ne faut pas espérer obtenir des généralisations de l'identité de Levi dans le cas itéré : certains opérateurs de révision itérée ne peuvent être obtenus à partir d'opérateurs de contraction itérée.

Cela semble donc suggérer que les opérateurs de révision itérée sont plus élémentaires que les opérateurs de contraction itérée.

Dans la section suivante nous donnons les préliminaires formels nécessaires pour cet article. Dans la section 3 on

donne les postulats logiques pour modéliser la contraction itérée. Dans la section 4 nous donnons un théorème de représentation pour ces opérateurs de contraction itérée. Dans la section 5 nous étudions les liens entre opérateurs de contraction itérée et de révision itérée. Enfin, dans la section 6 nous discutons des travaux connexes, avant de conclure section 7.

2 Préliminaires

On considère un langage propositionnel \mathcal{L} , défini à partir d'un alphabet fini de variables propositionnelles \mathcal{P} et les connecteurs standards. On notera \mathcal{L}^* l'ensemble des formules cohérentes de \mathcal{L} .

Une interprétation ω est une fonction totale de \mathcal{P} dans $\{0, 1\}$. L'ensemble de toutes les interprétations est noté \mathcal{W} . Une interprétation ω est un modèle d'une formule $\phi \in \mathcal{L}$ si et seulement si elle la rend vraie (sous l'interprétation usuelle des connecteurs).

On notera $[[\alpha]]$ l'ensemble des modèles d'une formule α , i.e., $[[\alpha]] = \{\omega \in \mathcal{W} \mid \omega \models \alpha\}$.

Si \leq est un pré-ordre sur \mathcal{W} (i.e., une relation réflexive et transitive), alors $<$ dénote l'ordre strict associé et défini par $\omega < \omega'$ si et seulement si $\omega \leq \omega'$ et $\omega' \not\leq \omega$. Un pré-ordre est *total* si $\forall \omega, \omega' \in \mathcal{W}$, $\omega \leq \omega'$ ou $\omega' \leq \omega$. Si $A \subseteq \mathcal{W}$, alors l'ensemble des éléments minimaux de A pour un pré-ordre total \leq , noté $\min(A, \leq)$, est défini par $\min(A, \leq) = \{\omega \in A \mid \nexists \omega' \in A \text{ tel que } \omega' < \omega\}$. Et $\min(\leq)$ représente l'ensemble $\min(\mathcal{W}, \leq)$.

Nous utiliserons les états épistémiques pour représenter les croyances de l'agent, comme usuellement pour la révision itérée [6]. Un état épistémique Ψ représente les croyances actuelles de l'agent, mais également des informations conditionnelles additionnelles, qui guident le processus de révision (habituellement représentées par un pré-ordre sur les interprétations, un ensemble de conditionnels, une séquence de formules, etc.). Nous noterons \mathcal{E} l'ensemble de tous les états épistémiques. La fonction de projection $B : \mathcal{E} \rightarrow \mathcal{L}^*$ associe à chaque état épistémique Ψ une formule cohérente $B(\Psi)$, qui représente les croyances courantes de l'agent dans l'état épistémique Ψ . On appellera modèles de l'état épistémiques les modèles de ses croyances, i.e. $[[\Psi]] = [[B(\Psi)]]$.

Une représentation concrète très utile des états épistémiques est celle utilisant des pré-ordres sur les interprétations. Avec cette représentation, si $\Psi = \leq$, alors $B(\Psi)$ est une formule propositionnelle telle que $[[B(\Psi)]] = \min(\leq)$. On appellera cette représentation concrète d'états épistémiques, la *représentation canonique*.

Pour éviter des détails techniques inutiles, on ne considérera dans cet article que des états épistémiques cohérents et des nouvelles informations cohérentes.

Donc, nous nous intéressons à des fonctions \circ qui associent à un état épistémique et une formule cohérente, un

nouvel état épistémique, $\circ : \mathcal{E} \times \mathcal{L}^* \rightarrow \mathcal{E}$. L'image d'une paire (Ψ, α) sous \circ sera noté $\Psi \circ \alpha$.

3 Contraction Itérée

Donnons tous d'abord les postulats basiques pour la contraction d'états épistémiques. Nous utilisons les postulats de contraction proposés pour la contraction en logique propositionnelle dans [4], qui sont équivalents aux postulats AGM originaux [1]. Nous les adaptons simplement pour les états épistémiques :

- (C1) $B(\Psi) \vdash B(\Psi \div \alpha)$
- (C2) Si $B(\Psi) \not\vdash \alpha$, alors $B(\Psi \div \alpha) \vdash B(\Psi)$
- (C3) Si $B(\Psi \div \alpha) \vdash \alpha$, alors $\vdash \alpha$
- (C4) $B(\Psi \div \alpha) \wedge \alpha \vdash B(\Psi)$
- (C5) Si $\alpha_1 \equiv \alpha_2$ alors $B(\Psi \div \alpha_1) \equiv B(\Psi \div \alpha_2)$
- (C6) $B(\Psi \div (\alpha \wedge \beta)) \vdash B(\Psi \div \alpha) \vee B(\Psi \div \beta)$
- (C7) Si $B(\Psi \div (\alpha \wedge \beta)) \not\vdash \alpha$, alors $B(\Psi \div \alpha) \vdash B(\Psi \div (\alpha \wedge \beta))$

(C1) impose que la contraction ne peut que retirer de l'information, donc les croyances de l'état épistémique résultat sont plus faibles que celles de l'état courant. (C2) dit que si l'état épistémique courant n'implique déjà pas la formule par laquelle on contracte, alors l'état épistémique résultat aura exactement les mêmes croyances que l'état courant. (C3) est le postulat de succès, il exprime le fait que le seul cas où la contraction échoue à retirer une formule des croyances de l'agent est celui où cette formule est une tautologie. (C4) est le postulat de restauration, il dit que si on effectue une contraction par une formule, suivie par la conjonction par cette formule, on doit retrouver les croyances initiales. Cela assure qu'aucune information n'a inutilement été rejetée lors de la contraction. (C5) est le postulat d'indépendance de la syntaxe, il exprime le fait que la syntaxe des formules n'a pas d'impact sur le résultat de la contraction. (C6) dit que la contraction par une conjonction implique la disjonction des contractions des éléments de cette conjonction. (C7) dit que, si α n'est pas retirée lors d'une contraction par la conjonction $\alpha \wedge \beta$, alors la contraction par α implique la contraction par la conjonction.

Maintenant nous pouvons donner les postulats pour l'itération :

- (C8) Si $\neg \alpha \vdash \gamma$ alors $B(\Psi \div (\alpha \vee \beta)) \vdash B(\Psi \div \alpha) \Leftrightarrow B(\Psi \div \gamma \div (\alpha \vee \beta)) \vdash B(\Psi \div \gamma \div \alpha)$
- (C9) Si $\gamma \vdash \alpha$ alors $B(\Psi \div (\alpha \vee \beta)) \vdash B(\Psi \div \alpha) \Leftrightarrow B(\Psi \div \gamma \div (\alpha \vee \beta)) \vdash B(\Psi \div \gamma \div \alpha)$
- (C10) Si $\neg \beta \vdash \gamma$ alors $B(\Psi \div \gamma \div (\alpha \vee \beta)) \vdash B(\Psi \div \gamma \div \alpha) \Rightarrow B(\Psi \div (\alpha \vee \beta)) \vdash B(\Psi \div \alpha)$
- (C11) Si $\gamma \vdash \beta$ alors $B(\Psi \div \gamma \div (\alpha \vee \beta)) \vdash B(\Psi \div \gamma \div \alpha) \Rightarrow B(\Psi \div (\alpha \vee \beta)) \vdash B(\Psi \div \alpha)$

(C8) exprime le fait que si une contraction par une disjonction implique la contraction par un des éléments de la disjonction, alors ce sera également le cas si on commence par une contraction par une formule qui est conséquence de cette négation. (C9) capture le fait que si une contraction par une disjonction implique la contraction par un des éléments de cette disjonction, alors ce sera également le cas si on commence par une contraction par une formule qui implique cet élément. (C10) exprime le fait que si une contraction par une disjonction implique la contraction par un des éléments de cette disjonction après la contraction par une formule qui est une conséquence de la négation de l'autre élément de la disjonction, alors c'était déjà le cas avant cette contraction. (C11) capture le fait que si une contraction par une disjonction implique la contraction par un des éléments de cette disjonction après la contraction par une formule qui implique l'autre élément de la disjonction, alors c'était déjà le cas avant cette contraction.

Les opérateurs qui satisfont (C1)-(C7) seront appelés opérateurs de contractions. Et les opérateurs satisfaisant (C1)-(C11) seront appelés opérateurs de contraction itérée.

4 Théorème de Représentation

Nous allons à présent donner un théorème de représentation pour les opérateurs de contraction itérée en termes d'assignements fidèles. Ces assignements associent à chaque état épistémique un pré-ordre total sur les interprétations. Ce pré-ordre représente la plausibilité relative des différentes interprétations, et les croyances actuelles de l'agent sont définies par les interprétations les plus plausibles.

Commençons par rappeler la définition des assignements fidèles [6] :

Définition 1 *Un assignement fidèle est une fonction qui associe à chaque état épistémique Ψ un pré-ordre total \leq_Ψ sur les interprétations tel que :*

1. Si $\omega \models B(\Psi)$ et $\omega' \models B(\Psi)$, alors $\omega \simeq_\Psi \omega'$
2. Si $\omega \models B(\Psi)$ et $\omega' \not\models B(\Psi)$, alors $\omega <_\Psi \omega'$
3. Si $\Psi_1 = \Psi_2$, alors $\leq_{\Psi_1} = \leq_{\Psi_2}$

Nous pouvons à présent donner le théorème basique pour les opérateurs de contraction dans le cadre des états épistémiques :

Théorème 1 *Un opérateur \div satisfait les postulats (C1) - (C7) si et seulement si il existe un assignement fidèle qui associe à chaque état épistémique Ψ un pré-ordre total \leq_Ψ sur les interprétations tel que*

$$[[\Psi \div \alpha]] = [[\Psi]] \cup \min([[\neg \alpha]], \leq_\Psi)$$

On dira que l'assignement fidèle donné par le théorème ci-dessus *représente* l'opérateur \div . La preuve de ce théorème est similaire à celle pour le théorème de représentation dans le cas de la logique propositionnelle [4], nous ne la recopions donc pas ici et le lecteur pourra se référer à cet article.

La partie la plus importante ici est celle concernant l'itération.

Définition 2 *Un assignement fidèle itéré est un assignement fidèle qui satisfait également les conditions suivantes :*

4. Si $\omega, \omega' \in \llbracket \gamma \rrbracket$, alors $\omega \leq_{\Psi} \omega' \Leftrightarrow \omega \leq_{\Psi \div \gamma} \omega'$
5. Si $\omega, \omega' \in \llbracket \neg \gamma \rrbracket$, alors $\omega \leq_{\Psi} \omega' \Leftrightarrow \omega \leq_{\Psi \div \gamma} \omega'$
6. Si $\omega \in \llbracket \neg \gamma \rrbracket$ et $\omega' \in \llbracket \gamma \rrbracket$, alors $\omega <_{\Psi} \omega' \Rightarrow \omega <_{\Psi \div \gamma} \omega'$
7. Si $\omega \in \llbracket \neg \gamma \rrbracket$ et $\omega' \in \llbracket \gamma \rrbracket$, alors $\omega \leq_{\Psi} \omega' \Rightarrow \omega \leq_{\Psi \div \gamma} \omega'$

La condition 4 capture le fait que la plausibilité entre les modèles de γ sont exactement les mêmes avant et après la contraction par γ . Similairement, la condition 5 capture le fait que la plausibilité entre les modèles de la négation de γ sont exactement les mêmes avant et après la contraction par γ . On appellera les conditions 4 et 5, conditions de rigidité. La condition 6 capture le fait que si un modèle de $\neg \gamma$ est strictement plus plausible qu'un modèle de γ avant la contraction par γ , alors il le sera toujours après la contraction par γ . La condition 7 assure que la plausibilité des modèles de $\neg \gamma$ ne s'empirera pas par rapport aux modèles de γ après la contraction par γ .

Il est important de remarquer que les assignements fidèles itérés sont directement liés à ceux de la révision itérée [6], il y a juste une inversion dans les conditions 6 et 7, qui est due au fait que l'on doit considérer une contraction par $\neg \alpha$ comme un changement (un *improvement* [14]) plus restreint qu'une révision par α (voir la discussion au début de la section 6).

Nous pouvons à présent donner le théorème de représentation pour la contraction itérée.

Théorème 2 *Un opérateur \div est un opérateur de contraction itérée, i.e. il satisfait les postulats (C1) - (C11), si et seulement si il existe un assignement fidèle itéré qui associe à chaque état épistémique Ψ un pré-ordre total \leq_{Ψ} sur les interprétations tel que*

$$\llbracket \Psi \div \alpha \rrbracket = \llbracket \Psi \rrbracket \cup \min(\llbracket \neg \alpha \rrbracket, \leq_{\Psi})$$

5 Contraction itérée versus révision itérée

Nous voudrions à présent étudier les liens entre contraction itérée et révision itérée.

Une tentation naturelle serait de tenter de généraliser les identités de Levy et de Harper dans le cas itéré. En fait quelques travaux ont tenté de suivre cette voix [18, 5, 2].

Nous allons tout d'abord expliquer en quoi ce n'est pas si simple. Et nous allons également montrer qu'il y a de vrai problèmes si on tente de définir une connexion formelle. Plus exactement, nous allons montrer que dans le cas itéré révision et contraction ne sont plus les deux faces d'une même pièce de monnaie (i.e. deux classes d'opérateurs liées par une bijection), mais qu'ils doivent plutôt être vus comme deux instances d'une même classes d'opérateurs de changement (les opérateurs d'*improvement* [14]), et que la différence est seulement une question de degré de changement.

5.1 Identités dans le cas général

Commençons par rappeler les identités de Levy et de Harper :

$$\begin{array}{ll} \text{Identité de Levy} & \Psi \star \alpha = (\Psi \div \neg \alpha) \oplus \alpha \\ \text{Identité de Harper} & \Psi \div \alpha = \Psi \sqcap (\Psi \star \neg \alpha) \end{array}$$

Et voyons donc à présent les problèmes de l'utilisation de ces identités pour la contraction et la révision itérée dans le cadre des états épistémiques. Tout d'abord, dans le cadre AGM, ces deux identités sont définitionnelles, c'est-à-dire qu'en utilisant l'identité de Levy, par exemple, il est possible d'obtenir l'opérateur de révision \star qui définit la théorie $\Psi \star \alpha$ à partir de la partie droite de l'identité en utilisant un opérateur de contraction et l'opérateur d'expansion. Mais dans le cadre général, les états épistémiques sont des objets abstraits, que l'on ne peut caractériser au niveau logique qu'à partir de la fonction de projection B . Donc dans ce cadre général, on ne connaît pas exactement $\Psi \div \neg \alpha$, et donc on ne peut pas utiliser cela pour définir ce qu'est $\Psi \star \alpha$. Donc utiliser cette identité pour définir un opérateur est compliqué. La seule possibilité semble être de se restreindre à une représentation particulière des états épistémiques (comme les pré-ordres sur les interprétations par exemple), mais ces résultats ne vaudront alors que pour cette représentation, et pas dans le cas général.

Deuxième problème, alors que \oplus et \sqcap ont un sens clair dans le cadre AGM, il faut trouver des définitions adéquates à ces opérations dans le cadre des états épistémiques. Ce seul problème est déjà loin d'être aisé. Par exemple l'étude des différentes définitions possible de \sqcap est le but principal de [2].

Une possibilité serait de restreindre ces identités aux croyances associées à chaque état épistémique :

$$\begin{array}{ll} \text{Equivalence de Levy} & B(\Psi \star \alpha) \equiv B(\Psi \div \neg \alpha) \oplus \alpha \\ \text{Equivalence de Harper} & B(\Psi \div \alpha) \equiv B(\Psi) \vee B(\Psi \star \neg \alpha) \end{array}$$

Mais ce qui signifie que nous n'avons plus des identités, ces équivalences ne permettant plus de définir ces opérateurs. Il est donc nécessaire de partir de deux opérateurs \star

et \div , et de vérifier si ils sont liées par ces équivalences.

5.2 Identités sous la représentation canonique

Donc, pour tenter d'aller plus loin, nous devons nous focaliser sur une représentation particulière des états épistémiques. La représentation canonique, utilisant des pré-ordres totaux, peut-être utilisée pour définir totalement l'état épistémique après la contraction (ou la révision). Supposons donc que l'on dispose d'un assignement fidèle itéré $\Psi \mapsto \leq_{\Psi}$ (Définition 2). On identifie Ψ avec \leq_{Ψ} , et donc, par le théorème 2, l'opérateur défini par $\Psi \div \gamma = \leq_{\Psi \div \gamma}$ est un opérateur de contraction itérée. Dans ce cas on dit que l'opérateur \div est donné par l'assignement. On peut procéder de la même manière pour définir un opérateur de révision itéré [6], l'opérateur défini à partir de l'assignement itéré (pour la révision [6]) par $\Psi \star \alpha = \leq_{\Psi \star \alpha}$ est un opérateur de révision itérée.

On peut alors définir des identités sur les pré-ordres totaux associés aux états épistémiques par les opérateurs (tpo signifiant pré-ordres totaux) :

$$\begin{aligned} \text{Identité de Levy tpo} \quad & \leq_{\Psi \star \alpha} = \leq_{\Psi \div \neg \alpha} \oplus \alpha \\ \text{Identité de Harper tpo} \quad & \leq_{\Psi \div \alpha} = \leq_{\Psi} \sqcap \leq_{\Psi \star \neg \alpha} \end{aligned}$$

Donc maintenant il est possible de définir ces pré-ordres à partir des identités et de vérifier que les opérateurs ainsi obtenus satisfont les propriétés attendues. Le seul problème restant est alors de définir les opérateurs \oplus et \sqcap dans ce cadre.

Pour \oplus nous allons montrer qu'utiliser l'opérateur de révision naturelle de Boutilier \star_N [3] est une bonne solution.

Commençons par rappeler la définition de cet opérateur sur les pré-ordres totaux. Cela revient à trouver les modèles de la nouvelle information les plus plausibles, et d'en faire les interprétations les plus plausibles, alors que rien d'autre ne change dans le pré-ordre : Soit \leq_{Ψ} le pré-ordre associé à l'état épistémique Ψ par l'assignement fidèle, et soit une nouvelle information α , on définit alors $\leq_{\Psi \star_N \alpha}$ (on utilisera également la notation $\leq_{\Psi} \star_N \alpha$) par :

- Si $\omega \models \min([\alpha], \leq_{\Psi})$ et $\omega' \not\models \min([\alpha], \leq_{\Psi})$, alors $\omega <_{\Psi \star_N \alpha} \omega'$
- Dans tous les autres cas $\omega \leq_{\Psi \star_N \alpha} \omega'$ ssi $\omega \leq_{\Psi} \omega'$

On peut alors montrer que l'identité de Levi tpo fonctionne pour le cas itéré :

Proposition 1 *Soit un opérateur de contraction itérée \div donné par son assignement $\Psi \mapsto \leq_{\Psi}$. Alors l'assignement défini par $\leq_{\Psi \star \alpha} = \leq_{\Psi \div \neg \alpha} \star_N \alpha$ satisfait les propriétés (CRI-CR4) [6], et peut donc être utilisé pour définir un opérateur de révision itérée Darwiche et Pearl (avec cette représentation canonique).*

Cette proposition implique en particulier qu'à chaque opérateur de contraction correspond un opérateur de révision. Cela signifie donc que la cardinalité de la classe des

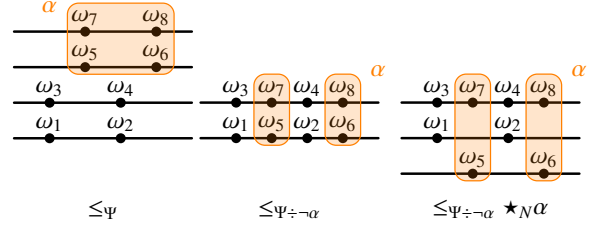


FIGURE 1 – De la contraction à la révision

opérateurs de révision itérée obtenus à partir de l'identité de Levi tpo est au plus la cardinalité de la classe des opérateurs de révision itérée. Notons également que cette observation ne dépend pas de l'interprétation de l'opérateur \oplus utilisé.

L'exemple suivant illustre cette utilisation concrète de l'identité de Levi tpo.

Exemple 1 *Considérons le pré-ordre total \leq_{Ψ} représenté figure 1. Dans cette figure plus une interprétation apparaît à un niveau bas, plus elle est plausible. Par exemple dans \leq_{Ψ} on constate que $\omega_1 <_{\Psi} \omega_3$. Une contraction itérée par $\neg \alpha$ est un changement qui augmente (improve) la plausibilité des modèles de α , avec la condition que les modèles les plus plausibles de α dans \leq_{Ψ} rejoignent (i.e. deviennent aussi plausibles que) les modèles les plus plausibles de \leq_{Ψ} . La relation entre les modèles de α ne peut pas changer après la contraction, ainsi que la relation entre les modèles de $\neg \alpha$. A partir de celà, pour définir une révision, il suffit de prendre ces modèles les plus plausibles de α et de les rendre les plus plausibles (en utilisant l'opérateur de révision naturelle de Boutilier [3]).*

Le processus inverse, c'est à dire la définition des opérateurs de contraction itérée à partir d'opérateurs de révision itérée en utilisant l'identité de Harper tpo, réclame quand à lui, de trouver une définition correcte pour \sqcap . Ce problème est le sujet de [2], où il est montré qu'il n'y a pas une unique façon canonique de procéder.

On peut voir cela comme une richesse supplémentaire offerte par le cadre itéré. Néanmoins cette liberté supplémentaire a un prix. Et en fait, dans le cadre itéré, il y a plus d'opérateurs de révision que d'opérateurs de contraction. L'exemple suivant illustre ce fait.

Exemple 2 *Soient \leq_{Ψ} et α comme illustrés figure 2. A cause des conditions de rigidité pour la contraction itérée, il n'y a que trois résultats possibles pour la contraction de \leq_{Ψ} par $\neg \alpha$. Mais il y a cinq résultats possibles pour la révision. Trois de ces résultats peuvent être obtenus à partir de la contraction suivie de la révision naturelle, comme pour l'exemple précédent. Les deux derniers résultats possibles sont illustrés sur la figure.*

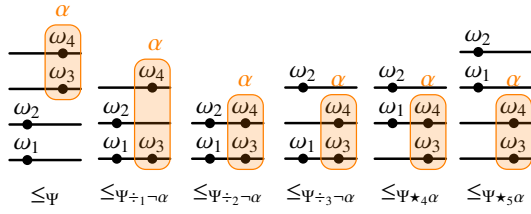


FIGURE 2 – Toutes les contractions possibles par $\neg\alpha$ et les revisions possibles par α à partir de \leq_{Ψ}

Observation 1 Il y a plus d'opérateurs de révision que d'opérateurs de contraction. En particulier cela implique qu'il est impossible d'obtenir une interprétation de l'expansion \oplus dans l'identité de Levi *tpo* permettant d'obtenir tous les opérateurs de révision itérée via cette identité.

Cette remarque est importante car elle met en lumière une distinction importante avec le cadre AGM classique, où il y a une bijection entre opérateurs de contraction et opérateurs de révision. La contraction est souvent considéré comme un opérateur plus élémentaire que la révision puisque la révision peut être définie comme une contraction suivie par une expansion (conjonction). Dans le cadre itéré, il y a plus d'opérateurs de révision que d'opérateurs de contraction, donc les opérateurs les plus élémentaires semblent être les opérateurs de révision.

Dans des travaux précédents nous avons proposé une classe d'opérateurs de changement plus générale, les opérateurs d'amélioration (improvement) [14, 13], dont on peut avoir comme sous-classes les opérateurs de révision itérée, ainsi que ceux de contraction itérée. Nous revenons plus en détail sur ce point dans la section suivante.

6 Travaux connexes

Dans [14, 13] les opérateurs d'amélioration (improvement) sont définis comme une classe générale d'opérateurs de changement itéré, qui contiennent les opérateurs de révision itérée de Darwiche et Pearl comme cas particulier. Une classe plus générale, des opérateurs de *basic improvement* a également été introduite dans [17]. Les postulats caractérisant ces opérateurs induisent qu'au moins une partie de la nouvelle information est améliorée, et que la nouvelle information n'est pas dégradée (cela correspond aux postulats C3, C4 et DP).

Les opérateurs d'améliorations sont définis sémantiquement sur les assignements fidèles comme une augmentation de la plausibilité des modèles de la nouvelles information. Cette augmentation de plausibilité peut être plus ou moins restreinte, menant à différentes familles d'opérateurs [13]. Mais clairement l'augmentation de plausibilité pour les opérateurs de contraction est limitée à cause du fait que

les modèles les plus plausibles de la nouvelle information ne peuvent pas devenir (strictement) plus plausibles que les modèles des croyances précédentes de l'agent. Alors que pour la révision, il n'y a pas de telle frontière, et donc on dispose de plus de liberté pour le changement.

La proposition suivante illustre le fait que les opérateurs de contraction itérée sont également des opérateurs d'amélioration faible (*weak improvement operators*) (par la négation de la formule).

Proposition 2 Soit un opérateur de contraction itérée \div , alors l'opérateur $\hat{\div}$ défini par $\Psi \hat{\div} \alpha = \Psi \div \neg\alpha$ est un *weak improvement operator* [14]. C'est également un *basic improvement operator*.

Cette proposition, avec la remarque que les opérateurs de révision itérée sont également des opérateurs de *basic improvement*, permettent d'illustrer le fait que ces opérateurs appartiennent à une même famille d'opérateurs de changement, que les opérateurs de contraction sont principalement différenciés des opérateurs de révision par la limite imposée au degré de changement, et que les opérateurs d'amélioration (*improvement*) sont bien la classe la plus primitive d'opérateurs de changement de croyances itérée.

Dans [5] les auteurs proposent également des postulats pour la contraction itérée, mais ils utilisent des opérateurs de révision itérés dans leurs postulats, donc les opérateurs ainsi définis ne le sont pas indépendamment, puisqu'ils apparaissent plutôt comme un produit des opérateurs de révision. Plus exactement, leur point de départ est un couple d'opérateurs $*$ et \div qui satisfont les équivalences de Levi et de Harper. Ils caractérisent ensuite les quatre propriétés sémantiques de l'itération de la définition 2 en termes de postulats syntaxiques, mixant les opérateurs $*$ et \div . Dans notre travail nous proposons une caractérisation directe des opérateurs de contraction itérée (qui ne dépendent d'aucun opérateur de révision itérée).

[2] étudie le problème de la définition d'opérateurs de contraction itérée, à partir de l'identité de Harper, dans le cas concret d'une représentation utilisant des pré-ordres sur les interprétations. Ce travail montre la non-trivialité de cette question, et l'étendue des possibilités pour la définition de \sqcap . Dans cet article nous expliquons cette difficulté par le fait qu'il y a plus d'opérateurs de révision itérée que d'opérateurs de contraction itérée, ce qui implique que différents opérateurs de révision correspondent, à travers de l'identité de Harper, au même opérateur de contraction.

Enfin le travail détaillé dans [19] est également intéressant. Il concerne la caractérisation de trois opérateurs de contraction itérée dans le cadre de la représentation canonique des états épistémiques. Ils donnent des caractérisation syntaxique de ces trois opérateurs, mais ils ne proposent pas de théorème de représentation pour la classe générale des opérateurs de contraction itérée.

7 Conclusion

En conclusion, cet article propose la première caractérisation logique directe de la classe des opérateurs de contraction itérée, et propose un théorème de représentation en terme de pré-ordres totaux sur les interprétations. Nous discutons le fait qu'il n'y a pas de généralisation directe des identités de Levi et de Harper dans le cas itéré, mais surtout, qu'il est vain d'aller à la quête de ce genre d'identités dans le cadre itéré, pour deux raisons. Tout d'abord, contrairement au cadre AGM classique, ces deux classes d'opérateurs ne sont pas liées par une bijection. Il y a plus d'opérateurs de révision itérée que d'opérateurs de contraction itérée. Et enfin, ces deux familles d'opérateurs ne sont pas des opérations différentes, mais deux cas particuliers d'une famille plus générale, les opérateurs d'*improvement*. Et la différence entre ces deux familles n'est donc pas une différence de nature, mais une différence de degré du changement attendu.

Références

- [1] Alchourrón, C. E., P. Gärdenfors et D. Makinson: *On the logic of theory change : Partial meet contraction and revision functions*. Journal of Symbolic Logic, 50 :510–530, 1985.
- [2] Booth, Richard et Jake Chandler: *Extending the Harper Identity to Iterated Belief Change*. Dans Kambhampati, Subbarao (rédacteur) : *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 987–993. IJCAI/AAAI Press, 2016, ISBN 978-1-57735-770-4.
- [3] Boutilier, C.: *Iterated Revision and Minimal Change of Conditional Beliefs*. Journal of Philosophical Logic, 25(3) :262–305, 1996.
- [4] Caridroit, T., S. Konieczny et P. Marquis: *Contraction in Propositional Logic*. Dans *Thirteenth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQA-RU'15)*, pages 186–196, 2015.
- [5] Chopra, Samir, Aditya Ghose, Thomas Andreas Meyer et Ka-Shu Wong: *Iterated Belief Change and the Recovery Axiom*. J. Philosophical Logic, 37(5) :501–520, 2008.
- [6] Darwiche, A. et J. Pearl: *On the logic of iterated belief revision*. Artificial Intelligence, 89 :1–29, 1997.
- [7] Fermé, E. et S. O. Hansson: *AGM 25 Years*. Journal of Philosophical Logic, 40(2) :295–331, 2011.
- [8] Gärdenfors, P.: *Knowledge in flux*. MIT Press, 1988.
- [9] Hansson, S. O.: *A Textbook of Belief Dynamics. Theory Change and Database Updating*. Kluwer, 1999.
- [10] Harper, William L.: *Rational conceptual change*. Dans *PSA : Proceedings of the Biennial Meeting of the Philosophy of Science Association*, pages 462–494, 1976.
- [11] Katsuno, H. et A. O. Mendelzon: *Propositional knowledge base revision and minimal change*. Artificial Intelligence, 52 :263–294, 1991.
- [12] Katsuno, H. et A. O. Mendelzon: *On the Difference Between Updating a Knowledge Base and Revising It*. Dans *Belief Revision*, pages 183–203. Cambridge University Press, 1992.
- [13] Konieczny, S., M. Medina Grespan et R. Pino Pérez: *Taxonomy of improvement operators and the problem of minimal change*. Dans *Proceedings of the Twelfth International Conference on Principles of Knowledge Representation And Reasoning (KR 2010)*., pages 161–170, 2010.
- [14] Konieczny, S. et R. Pino Pérez: *Improvement operators*. Dans *Proceedings of the Eleventh International Conference on Principles of Knowledge Representation And Reasoning (KR 2008)*., pages 177–187, 2008.
- [15] Levi, Isaac: *Decisions and revisions*. Cambridge University Press, 1984.
- [16] Levi, Isaac: *For the sake of the argument*. Cambridge University Press, 1996.
- [17] Medina Grespan, Mattia et Ramón Pino Pérez: *Representation of basic improvement operators*. Dans *Trends in Belief Revision and Argumentation Dynamics*, pages 195–227. College Publications, 2013.
- [18] Nayak, Abhaya C., Randy Goebel, Mehmet A. Orgun et Tam Pham: *Taking Levi Identity Seriously : A Plea for Iterated Belief Contraction*. Dans Lang, Jérôme, Fangzhen Lin et Ju Wang (rédacteurs) : *Knowledge Science, Engineering and Management, First International Conference, KSEM 2006, Guilin, China, August 5-8, 2006, Proceedings*, tome 4092 de *Lecture Notes in Computer Science*, pages 305–317. Springer, 2006, ISBN 3-540-37033-1.
- [19] Ramachandran, Raghav, Abhaya C. Nayak et Mehmet A. Orgun: *Three Approaches to Iterated Belief Contraction*. J. Philosophical Logic, 41(1) :115–142, 2012.

Un raccourci récursif pour CEGAR : Application au problème de satisfiabilité en logique modale K

Jean-Marie Lagniez¹ Daniel Le Berre¹ Tiago de Lima¹ Valentin Montmirail¹

¹ CRIL, Univ Artois et CNRS, France
{lagniez, leberre, delima, montmirail}@cril.fr

Résumé

L'approche Counter-Example Guided Abstraction Refinement (CEGAR) a été un grand succès dans la vérification de modèle. Depuis lors, elle a été appliquée à de nombreux problèmes différents. Il s'avère qu'il s'agit d'une approche pratique très efficace pour résoudre le problème QBF qui est PSPACE-complet. Dans cet article, nous proposons une nouvelle approche semblable à CEGAR pour aborder des problèmes PSPACE, approche que nous appelons RECAR (Recursive Explore and Check Abstraction Refinement). Nous montrons que cette approche générique est correcte et complète. Ensuite, nous proposons une instantiation du framework RECAR pour résoudre le problème de satisfiabilité en logique modale K. Nous implémentons les deux approches CEGAR et RECAR pour déterminer la cohérence d'une formule en logique modale K au sein du solveur MoSaiC. Nous comparons expérimentalement ces approches face aux solveurs de l'état de l'art. L'approche RECAR surpasse l'approche CEGAR sur ces problèmes et se compare favorablement aux solveurs de l'état de l'art sur les benchmarks considérés.

Abstract

Counter-Example Guided Abstraction Refinement (CEGAR) has been very successful in model checking. Since then, it has been applied to many different problems. It is especially proved to be a highly successful practical approach for solving the PSPACE complete QBF problem. In this paper, we propose a new CEGAR-like approach for tackling PSPACE complete problems that we call RECAR (Recursive Explore and Check Abstraction Refinement). We show that this generic approach is sound and complete. Then we propose a specific implementation of the RECAR approach to solve the modal logic K satisfiability problem. We implemented both CEGAR and RECAR approaches for the modal logic K satisfiability problem within the solver MoSaiC. We compared experimentally those approaches to the state-of-the-art solvers for that problem. The RECAR approach outperforms the CEGAR one for that problem and also compares favorably against the state-of-the-art on the benchmarks considered.

1 Introduction

L'idée d'utiliser un solveur SAT s'est révélée être une approche pratique très efficace pour résoudre certains problèmes NP-complets [3]. Un des principaux problèmes est de trouver le "bon" encodage pour le problème, c'est-à-dire de trouver la réduction polynomiale du problème original vers une formule de la logique propositionnelle en Forme Normal Conjonctive (CNF, un ensemble de clauses) qui peut être efficacement résolue par un solveur SAT [23]. Le solveur SAT est un moteur générique de résolution de problème, dont le paramètre d'entrée est une CNF équicohérent au problème original. Il arrive souvent que soit le solveur SAT résout efficacement la CNF soit pas du tout (voir les résultats de la compétition SAT [14]). Un cas particulier est quand la CNF générée est trop grande : le temps pour générer et lire la CNF est plus grand que le temps pour la résoudre. Ceci est due à une quantité limitée de mémoire allouée à la résolution et pas au solveur SAT directement. Nous abordons l'un de ces cas particuliers dans cet article.

Pour les CNF possédant énormément de clauses, des approches spécifiques ont été créées dans le passé, où le solveur SAT est utilisé tel un oracle dans une procédure plus complexe. Une de ces procédures est appelée Counter-Example-Guided Abstraction Refinement (CEGAR) [6]. Le solveur SAT est alimenté avec une abstraction du problème original qui permet plus de modèles (ce que nous appellerons ici une sous-abstraction). Si cette sous-abstraction est incohérent, alors le problème original l'est aussi (court-circuit UNSAT). Sinon la procédure est capable de vérifier si le modèle trouvé pour la sous-abstraction est une solution pour le problème original. Dans ce cas, nous avons un court-circuit SAT additionnel capable de décider la cohérence de la formule. Si ce n'est pas le cas, de nouvelles contraintes sont ajoutées pour prévenir le solveur de trouver de tel exemple fallacieux (étape de raffinement) et la procédure se répète. Éventuellement une formule équi-

cohérente est détectée, et le solveur SAT peut décider ainsi le problème. Une des raisons pour utiliser CEGAR est qu'en pratique, la formule complète est trop grande pour ne serait-ce qu'être générée, par conséquent, le seul espoir pour résoudre ce problème est "d'avoir de la chance" (court-circuit SAT) ou d'être capable de prendre en compte une structure spécifique du problème (court-circuit UNSAT). Cette approche est élégante et a été appliquée à de nombreux problèmes : la cohérence Modulo Théorie (SMT) [4], la Planification [27] ou plus récemment à QBF [13]. Il apparaît que CEGAR est la meilleur approche en pratique pour résoudre des problèmes QBF (selon la dernière compétition QBF [24]). Ce résultat étonnant a motivé notre travail. Notre but est d'appliquer une approche basée sur SAT à un autre problème PSPACE-complet, déterminer la cohérence d'une formule en logique modale K. De très nombreuses approches basées sur SAT ont déjà été proposées dans le cadre de la logique modale [25], d'aucun pourrait même argumenter que *SAT [8] est déjà une approche CEGAR pour la logique modale K.

Dans cet article, nous introduisons une extension à l'approche CEGAR qui propose une étape récursive pour introduire un nouveau court-circuit dans la procédure CEGAR originale. Dans notre contexte, la boucle CEGAR principale contient un court-circuit SAT (\hat{s} at), pendant que l'étape récursive permet à la procédure de fournir un court-circuit UNSAT (\hat{s} unsat). Nous appelons cette extension "Recursive Explore and Check Abstraction Refinement" (RECAR). L'idée de mixer les court-circuits SAT et UNSAT dans une procédure CEGAR n'est pas nouvelle : cela a déjà été employée pour SMT [4] et pour la détection de bug [29]. Ici, la nouveauté est que nous utilisons une abstraction du problème original dans la boucle, ce qui est rendu possible grâce à l'appel récursif dans la boucle principale.

L'approche RECAR est générique, elle n'est pas spécifique à un domaine. Dans cet article, nous présentons les conditions requises sur les abstractions utilisées et nous prouvons la correction et la complétude de l'approche. Ensuite, nousinstancions notre approche pour déterminer la cohérence d'une formule en logique modale K, en fournissant des fonctions d'abstractions pour ce problème et des résultats expérimentaux de cette approche face aux solveurs de l'état de l'art. Nous pensons que les bons résultats pratiques que nous obtenons en utilisant RECAR sur ce domaine particulier sont prometteurs pour d'autres domaines. Cet article présente, dans ce but, une approche générique avec un cas d'utilisation réussi.

Le reste de l'article est organisé comme suit : tout d'abord, nous présentons l'approche CEGAR ; Puis, nous proposons notre framework appelé RECAR et nous montrons sa correction et sa complétude ; Nous fournissons une implémentation de l'approche RECAR pour la logique modale K ; Et finalement, nous comparons l'efficacité de l'approche RECAR face aux solveurs de l'état de l'art pour la

logique modale K.

2 Préliminaires sur CEGAR

Counter-Example Guided Abstraction Refinement, CEGAR, est une méthode incrémentale pour décider de la cohérence d'une formule.

Il a été conçu à l'origine pour la vérification du modèle [6], c'est-à-dire de répondre aux questions du type "Est-ce que $S \models P$?" ou de manière équivalente, "Est-ce que $(S \wedge \neg P)$ est incohérent?", où S décrit un système et P une propriété. Dans les problèmes très structurés, il est souvent le cas que seul une petite partie de la formule soit nécessaire pour répondre à la question. L'idée derrière CEGAR est de remplacer $\phi = (S \wedge \neg P)$ par une abstraction $\hat{\phi}$, où $\hat{\phi}$ est plus simple à résoudre en pratique que ϕ . Il existe deux types d'abstractions :

- Une sur-abstraction de ϕ est une formule $\hat{\phi}$ telle que $\hat{\phi} \models \phi$, $\hat{\phi}$ a au plus autant de modèles que ϕ ;
- Une sous-abstraction de ϕ est une formule $\check{\phi}$ telle que $\phi \models \check{\phi}$, $\check{\phi}$ a au moins autant de modèles que ϕ .

Typiquement, ϕ est une CNF et sous/sur doivent être lu dans le sens "sous-contraint" et "sur-contraint". Ensuite, une méthode classique pour sous-approximer ϕ est "d'oublier" des clauses, c'est-à-dire $\check{\phi}$ est un sous-ensemble de clauses de ϕ . Un modèle de $\check{\phi}$ peut potentiellement, avec de la chance, satisfaire ϕ . De plus, si $\check{\phi}$ est montrée incohérent, alors ϕ l'est aussi. Cette double possibilité de conclure plus rapidement fait que les approches CEGAR basées sur les sous-abstractions sont très populaire. Une méthode classique pour sur-approximer ϕ est de borner la génération de la formule $\hat{\phi}$ en lui donnant un n plus petit que celui nécessaire pour atteindre l'équi-cohérence avec le problème original (comme dans la Vérification de Modèle Bornée (BMC) [6] ou encore en planification [27]). Ainsi un modèle de $\hat{\phi}$ peut être étendu pour devenir un modèle de ϕ mais l'incohérence de $\hat{\phi}$ signifie que la borne n a besoin d'être augmentée et la procédure doit être répétée.

Un exemple de CEGAR utilisant des sur-abstractions est donnée en Figure 1. Il reçoit en entrée une formule ϕ et calcule une sur-abstraction ψ . Ensuite, il utilise un solveur SAT pour vérifier si ψ est cohérent, si c'est le cas, il conclut que ϕ est cohérent. Sinon ψ est raffinée, c'est-à-dire, qu'elle se rapproche de ϕ , jusqu'à ce qu'elle soit cohérente ou jusqu'à ce que le raffinement de la sur-abstraction est détecté comme étant équi-cohérent à ϕ , noté $\psi \stackrel{?}{\equiv}_{\text{sat}} \phi$ (c-à-d. $\exists M, M \models_1 \psi$ iff $\exists M', M' \models_2 \phi$)¹ où il conclut que ϕ est incohérent. Dans la suite, $\phi \stackrel{?}{\equiv}_{\text{sat}} \psi$ signifie un test incomplet mais efficace pour l'équi-cohérence, test qui répond 'oui' ou 'inconnu'.

Récemment, les solveurs SAT sont devenus capable de vérifier la cohérence "sous hypothèses" [7], c'est-à-dire

1. \models_1 et \models_2 désignent éventuellement différentes relations de conséquence (en logique propositionnelle et en logique modale par exemple).

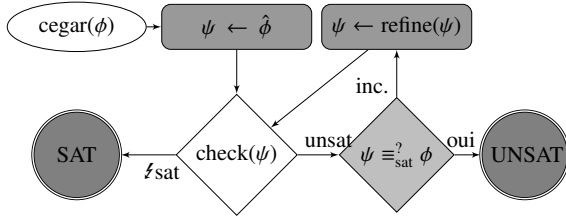


FIGURE 1 – Le framework CEGAR avec des sur-abstractions

donner la cohérence d’un ensemble de littéraux appelés hypothèses et de fournir dans le cas d’une incohérence, une “raison” de l’incohérence de la formule en fonction de ces littéraux.

Définition 1 (Coeur incohérent avec hypothèse). Soit ϕ une CNF et A un ensemble cohérent de littéraux venant de ϕ . Prenons ϕ cohérent et $(\phi \wedge \bigwedge_{a \in A} a)$ incohérent. $L \subseteq A$ est un coeur incohérent de ϕ si et seulement si $(\phi \wedge \bigwedge_{l \in L} l)$ est incohérent.

Par conséquent, un oracle SAT pour ϕ , sachant A , peut être vu comme une procédure fournissant une paire (d, ψ) avec $d \in \{\text{SAT}, \text{UNSAT}\}$ et ψ est un modèle de ϕ si $d = \text{SAT}$ ou un coeur incohérent de ϕ si $d = \text{UNSAT}$.

Les solveurs SAT modernes sont capables de prendre ψ en compte dans les deux cas. Les coeurs incohérents ont été utilisés par exemple dans une approche CEGAR pour décider la cohérence du fragment propositionnelle de la logique du premier ordre [16].

3 Recursive Explore and Check Abstraction Refinement

Une approche CEGAR classique avec une sur-abstraction et un court-circuit SAT fonctionne bien quand l’entrée est cohérent. Généralement, il ne fonctionne pas aussi bien sur les problèmes incohérents. La raison à cela est qu’il est nécessaire de continuer à raffiner jusqu’à obtenir l’équicohérence avec le problème original.

Une manière d’aborder ce problème est de mixer les court-circuits SAT et UNSAT, comme dans [4] et [29]. Dans ces approches, les méthodes alternent les sur et sous abstractions. L’approche RECAR, illustrée dans la Figure 2 et l’Algorithme 1 intercale les deux types d’abstractions : chaque abstraction est réalisée avec les informations retournées par le solveur sur la précédente. Le court-circuit UNSAT est réalisé en utilisant un appel récursif à la procédure principale quand une sous-abstraction stricte $\check{\phi}$ peut être construite. Il convient également de noter que l’approche proposée permet des abstractions sur deux niveaux : une utilisée pour simplifier le problème au niveau du domaine (l’appel récursif), tandis que l’autre permet de simplifier le problème au niveau de l’oracle.

Pour pouvoir appliquer l’approche RECAR, la sous-abstraction $\check{\phi}$ et la sur-abstraction $\hat{\phi}$ doivent satisfaire certaines conditions. Dans ce qui suit estSAT(ϕ) signifie que ϕ est cohérent ($\models_1 \neg\phi$) et estUNSAT(ϕ) signifie ($\models_2 \neg\phi$), mais possiblement avec différentes relations de conséquence. $RC(\phi, \check{\phi})$ désigne une fonction booléenne décidant si un appel récursif (Recursive Call) doit être effectué.

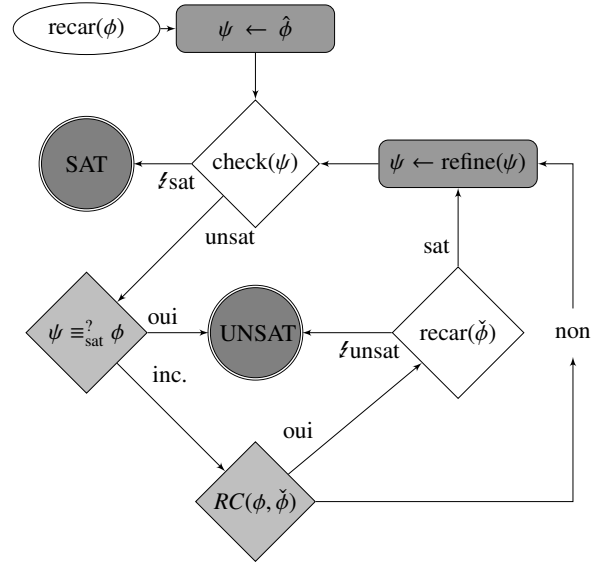


FIGURE 2 – L’approche RECAR

3.1 Conditions pour utiliser RECAR

1. La fonction ‘check’ est une implémentation de ‘estSAT’ qui est correcte, complète et qui se termine.
2. estSAT($\hat{\phi}$) implique estSAT(refine($\hat{\phi}$)).
3. Il existe $n \in \mathbb{N}$ tel que refineⁿ($\hat{\phi}$) $\equiv_{\text{sat}}^? \phi$.
4. estUNSAT($\check{\phi}$) implique estUNSAT(ϕ).
5. Soit under(ϕ) = $\check{\phi}$. Il existe $n \in \mathbb{N}$ tel que $RC(\text{under}^n(\phi), \text{under}^{n+1}(\phi))$ évalue à faux.

Notons que nous avons estSAT($\hat{\phi}$) implique ϕ est cohérent avec les Conditions 2 et 3 ensemble. Dans ce qui suit, nous montrons que, sous ces conditions, l’approche RECAR est correcte, complète et termine. Pour ce faire, nous présentons l’algorithme recar(ϕ) dans l’Algorithme 1.

Théorème 1 (Justesse). Si recar(ϕ) retourne SAT alors ϕ est cohérente.

Démonstration. Supposons que recar(ϕ) retourne SAT. Cela arrive seulement si check(ψ) retourne SAT, soit à la ligne 3 soit à la ligne 7. Ainsi, nous savons que estSAT(ψ) est vérifiée (Condition 1) mais ψ vaut $\hat{\phi}$ ou refineⁿ(ϕ) pour

Algorithme 1 : $\text{recar}(\phi)$

```
1  $\psi \leftarrow \hat{\phi}$ 
2 tant que  $\psi \equiv_{\text{sat}}^? \phi$  retourne "inconnu" faire
3   si  $\text{check}(\psi) = \text{SAT}$  alors retourner SAT ;
4   si  $\text{RC}(\phi, \check{\phi})$  alors
5     si  $\text{recar}(\check{\phi}) = \text{UNSAT}$  alors retourner
6       UNSAT ;
7    $\psi \leftarrow \text{refine}(\psi)$  ;
8 retourner  $\text{check}(\psi)$ 
```

un certain $n \in \mathbb{N}$. Par conséquent, ϕ est cohérente (par les Conditions 2 et 3). \square

L'intuition derrière la preuve du Théorème 2 est qu'il y a deux manières de conclure que ϕ est incohérente. Dans le premier cas, $\hat{\phi}$ est raffinée un nombre fini de fois jusqu'à ce que l'on détecte l'équi-cohérence avec ϕ et que check retourne UNSAT et donc que ϕ est incohérente. Dans le second cas, une des sous-abstractions est montrée UNSAT et donc par construction ϕ est UNSAT (Condition 4).

Théorème 2 (Complétude). *Si $\text{recar}(\phi)$ retourne UNSAT alors $\text{estUNSAT}(\phi)$.*

Démonstration. Par induction sur la taille k de la pile d'appels récursifs à recar (ligne 5). Supposons que $\text{recar}(\phi)$ retourne UNSAT après k appels récursifs. Dans la base d'induction ($k = 0$) (pas d'appel récursif). Nous devons avoir quitté la boucle ($\psi \equiv_{\text{sat}}^? \phi$) et $\text{check}(\psi)$ a retourné UNSAT. Cela signifie que ψ est incohérent (par la Condition 1) et par conséquent $\text{estUNSAT}(\phi)$ est vraie (à cause de l'équi-cohérence). L'hypothèse d'induction est la suivante : pour tous $k \leq n$, si $\text{recar}(\phi)$ retourne UNSAT après k appel récursif alors $\text{estUNSAT}(\phi)$. Dans l'étape d'induction $k = n + 1$. Les conditions des lignes 4 et 5 de l'algorithme sont vraies. Cela signifie que $\text{recar}(\check{\phi})$ retourne UNSAT après k appels récursifs à recar . $\text{estUNSAT}(\check{\phi})$ (par l'hypothèse d'induction). Donc $\text{estUNSAT}(\phi)$ (par la Condition 4). \square

L'intuition derrière la preuve du Théorème 3 est que l'algorithme va effectuer un nombre fini d'appels récursifs (Condition 5). De plus, chacun de ces appels récursifs a un nombre fini d'étapes de raffinement avant de terminer (Condition 3).

Théorème 3 (Terminaison). *RECAR termine pour n importe quelle entrée ϕ .*

Démonstration. Nous avons que (1) Pour toutes les formules ϕ , il existe $n \in \mathbb{N}$ tel que $\text{RC}(\text{under}^n(\phi), \text{under}^{n+1}(\phi))$ évalue à faux (Condition. 5) et (2) Pour chaque $i \leq n$ il existe $m_i \in \mathbb{N}$ tel que $\text{refine}^{m_i}(\hat{\phi}) \equiv_{\text{sat}}^? \phi$ (Condition. 3). Par conséquent, pour chaque entrée ϕ , l'appel récursif à la ligne 5 de l'Algorithme va s'exécuter au plus n fois avant

que la condition à la ligne 4 devienne fausse. Pour chacun de ces appels récursifs, la boucle tant-que de l'algorithme va être effectuée au plus m_i fois avant que la condition à la ligne 2 devienne fausse. Donc, pour toute les entrées, recar se termine après au plus $n + \sum_{i=0}^n (m_i)$ appels récursifs. \square

4 Résoudre K-SAT avec RECAR

Dans cette section, nous montrons comment RECAR est utilisé pour résoudre le problème de satisfiabilité en logique modale K, qui est un problème PSPACE-complet [11, 18].

4.1 La logique modale K

Avant de montrer comment nous appliquons l'approche RECAR, définissons formellement ce qu'est la logique modale K. Soit un ensemble non-vide fini de variables propositionnelles $\mathbb{P} = \{p_1, p_2, \dots\}$ et un ensemble de m opérateurs modaux unaires $\mathbb{M} = \{\Box_1, \dots, \Box_m\}$. Le langage de K (noté \mathcal{L}) est l'ensemble des formules contenant \mathbb{P} , fermé sous l'ensemble des connecteurs propositionnels $\{\neg, \wedge\}$ et l'ensemble des opérateurs modaux dans \mathbb{M} . Nous utiliserons les abréviations standard pour \top, \perp, \vee et \diamond . Par exemple $\diamond_a \phi = \neg \Box_a \neg \phi$.

Définition 2 (Modèle de Kripke). *un modèle de Kripke est un triplet $M = \langle W, \{R_a \mid \Box_a \in \mathbb{M}\}, V \rangle$, où W est un ensemble non-vide de "mondes possibles"; Chaque $R_a \subseteq W \times W$ est une relation d'accessibilité binaire sur W ; $V : \mathbb{P} \rightarrow 2^W$ est une fonction de valuation qui associe, à chaque $p \in \mathbb{P}$, l'ensemble des mondes possibles de W où p est vraie. Un modèle de Kripke pointé est une paire $\langle M, w \rangle$, où M est un modèle de Kripke et w est un monde possible dans W .*

Par la suite, chaque fois que nous utilisons le terme 'modèle', nous nous référons au 'modèle de Kripke pointé'

Dans ce qui suit, la taille d'un modèle $\langle M, w \rangle$ est le nombre d'éléments dans W , noté $|M|$.

Définition 3 (Relation de satisfiabilité). *La relation \models entre les modèles et les formules est définie récursivement de la façon suivante :*

$$\begin{aligned} \langle M, w \rangle \models p & \quad \text{ssi } w \in V(p) \\ \langle M, w \rangle \models \neg \phi & \quad \text{ssi } \langle M, w \rangle \not\models \phi \\ \langle M, w \rangle \models \phi_1 \wedge \phi_2 & \quad \text{ssi } \langle M, w \rangle \models \phi_1 \text{ et } \langle M, w \rangle \models \phi_2 \\ \langle M, w \rangle \models \Box_a \phi & \quad \text{ssi } (w, w') \in R_a \text{ implique } \langle M, w' \rangle \models \phi \end{aligned}$$

Définition 4 (Validité). *Comme d'habitude, une formule $\phi \in \mathcal{L}$ est valide (noté $\models \phi$) si et seulement si elle est satisfiable par tous les modèles $\langle M, w \rangle$. Une formule $\phi \in \mathcal{L}$ est satisfiable en K (noté $\text{estKSAT}(\phi)$) si et seulement si $\not\models \neg \phi$. Nous utilisons aussi $\text{estKUNSAT}(\phi)$ pour signifier $\models \neg \phi$.*

5 Les fonctions d'abstractions

Dans la suite, nous définissons une traduction de la logique modale K à la logique propositionnelle classique.

Définition 5 (Traduction).

$$\begin{aligned} \text{tr}(\phi, n) &= \text{tr}'(\text{nmf}(\phi), 0, n) \\ \text{tr}'(p, i, n) &= p_i \\ \text{tr}'(\neg p, i, n) &= \neg p_i \\ \text{tr}'(\phi \wedge \psi, i, n) &= \text{tr}'(\phi, i, n) \wedge \text{tr}'(\psi, i, n) \\ \text{tr}'(\phi \vee \psi, i, n) &= \text{tr}'(\phi, i, n) \vee \text{tr}'(\psi, i, n) \\ \text{tr}'(\Box_a \phi, i, n) &= \bigwedge_{j=0}^n (r_{i,j}^a \rightarrow \text{tr}'(\phi, j, n)) \\ \text{tr}'(\Diamond_a \phi, i, n) &= \bigvee_{j=0}^n (r_{i,j}^a \wedge \text{tr}'(\phi, j, n)) \end{aligned}$$

La traduction ajoute de nouvelles variables p_i et $r_{i,j}^a$ à la formule. p_i signifie que la variable p est vraie dans le monde w_i alors que $r_{i,j}^a$ signifie que w_j est accessible depuis w_i par la relation a .

Théorème 4. $\text{estKSAT}(\phi)$ si et seulement si $\text{estSAT}(\text{tr}(\phi, \text{nm}(\phi) + 1))$.

Démonstration. Le résultat présenté dans le Théorème 4 provient directement de [25], §25.2.2 (où $\text{nm}(\phi)$ est le nombre d'opérateurs modaux dans la formule ϕ). \square

Pour décider la cohérence de la formule $\phi \in \mathcal{L}$, on peut simplement alimenter le solveur SAT avec $\text{tr}(\phi, \text{nm}(\phi) + 1)$. En fait, c'est ce qui est globalement proposé dans Km2SAT [26].

Le principal problème avec cela est que la traduction peut générer une formule exponentiellement plus grande. Dans cet article, nous essayons de contourner ce problème en utilisant RECAR. Pour des raisons de simplicité, nous utiliserons maintenant $\text{tr}(\phi)$ pour signifier $\text{tr}(\phi, \text{nm}(\phi) + 1)$.

Dans [5], les auteurs utilisent aussi ce type de traduction pour la logique modale S5 vers la logique propositionnelle, mais en remplaçant $\text{nm}(\phi)$ par le diamond-degree $\text{dd}(\phi)$ qui est généralement plus petit. Malheureusement, ceci ne peut pas être reproduit pour la logique modale K, voici un contre-exemple :

Contre-Exemple 1. Soit $\phi = (p_1 \wedge p_2 \wedge p_3) \wedge (\Diamond_a(p_1 \wedge p_2 \wedge \neg p_3 \wedge \Box_a(p_1 \wedge \neg p_2 \wedge p_3))) \wedge (\Diamond_a(p_1 \wedge \neg p_2 \wedge \neg p_3 \wedge \Box_a(\neg p_1 \wedge \neg p_2 \wedge p_3))) \wedge (\Box_a \Diamond_a p_3)$, avec $\text{dd}(\phi) = 3$. Cette formule est satisfaite par le modèle illustré dans la Figure 3. Cependant, il est facile de voir qu'il n'existe pas de modèle satisfaisant ϕ avec moins de 5 mondes possibles.

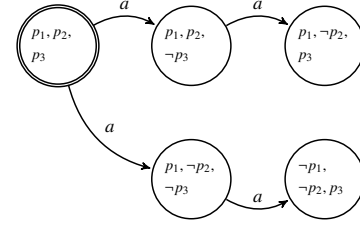


FIGURE 3 – $M \models \phi$

5.1 Sur-abstraction

Maintenant, dans le but d'appliquer le framework RECAR, nous avons besoin de trouver une sur-abstraction qui respecte les conditions présentées dans la Section 3.

Définition 6 (Sur-abstraction). Soit $\phi \in \mathcal{L}$. La sur-abstraction de ϕ , noté $\hat{\phi}$, est la formule $\text{tr}(\phi, 1)$.

Définition 7 (Raffinement). Soit $1 \leq n \leq \text{nm}(\phi) + 1$. Le raffinement de $\text{tr}(\phi, n)$, noté $\text{refine}(\text{tr}(\phi, n))$ est la formule $\text{tr}(\phi, n + 1)$.

Théorème 5. Si $\text{estSAT}(\text{tr}(\phi, n))$ alors $\text{estSAT}(\text{tr}(\phi, n + 1))$, pour tous $1 < n \leq \text{nm}(\phi) + 1$. (RECAR Condition. 2)

Esquisse de preuve. L'idée est que si ϕ est satisfaite par un modèle M avec n mondes, alors nous pouvons trouver un modèle M' avec $n + 1$ mondes satisfaisant ϕ . Le monde additionnel n'est juste pas accessible depuis les mondes déjà dans M . \square

Ce dernier résultat nous permet d'utiliser cette sur-abstraction et cette fonction de raffinement dans l'approche RECAR. Il est facile de voir que les Conditions 2 et 3 de RECAR sont satisfaites.

5.2 Sous-abstraction

Pour faire comprendre l'intuition derrière la fonction de sous-abstraction, nous utilisons un exemple. Soit $\phi = (\Diamond p \wedge \Box \neg p \wedge \chi)$ pour un $\chi \in \mathcal{L}$, où $\text{nm}(\chi)$ est grand. ϕ est clairement incohérente car $(\Diamond p \wedge \Box \neg p)$ est incohérente. N'importe qui peut le voir directement sans même savoir à quoi χ ressemble. Cependant, une approche CEGAR utilisant la sur-abstraction et le raffinement définis plus tôt va prendre un temps considérable avant d'être capable de conclure à l'incohérence. La raison étant que chaque raffinement $\text{tr}(\phi, n + 1)$ de la formule originale va être montré incohérente et la procédure ne va s'arrêter que lorsque la borne $\text{nm}(\phi) + 1$ sera atteinte.

Pour éviter ces cas pathologiques, l'approche RECAR effectue aussi des sous-abstractions. Pour voir comment cela fonctionne, reprenons notre exemple ϕ . Tout d'abord, nous ajoutons à chaque conjonction dans ϕ , une nouvelle variable s_i (un sélecteur) qui va être supposée vraie par le

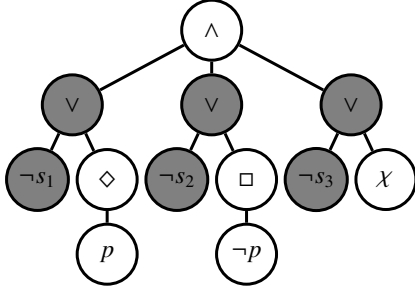


FIGURE 4 – Comment les sélecteurs sont appliqués à ϕ

solveur SAT, comme illustré par la Figure 4. Ensuite, nous construisons notre première sur-abstraction $\text{tr}(\phi, 1)$ et nous alimentons le solveur SAT avec. Le solveur va répondre incohérent en fournissant un coeur incohérent exprimé avec les s_i . De ce coeur incohérent, nous pouvons donc extraire un ensemble de sélecteurs *core*. Supposons que, dans notre exemple, $\text{core} = \{s_1, s_2\}$. Cela signifie que la formule $\check{\phi} = (\diamond p \wedge \square \neg p)$, étiquetée par les sélecteurs est suffisante pour montrer l'incohérence de ϕ avec seulement 1 monde possible. Prouver l'incohérence de $\check{\phi}$ va impliquer que ϕ est incohérente. Il est à noter que, dans ce cas, $\text{nm}(\check{\phi})$ est bien plus petit que $\text{nm}(\phi)$. Ainsi, l'approche CEGAR appliquée à $\check{\phi}$ va répondre bien plus rapidement, alors qu'elle aurait probablement échoué sur l'ensemble de la formule ϕ . Formellement, nous avons :

Définition 8 (Sous-Abstraction).

$$\text{under}(p, \text{core}) = p$$

$$\text{under}(\neg p, \text{core}) = \neg p$$

$$\text{under}(\square_a \phi, \text{core}) = \square_a(\text{under}(\phi, \text{core}))$$

$$\text{under}(\diamond_a \phi, \text{core}) = \diamond_a(\text{under}(\phi, \text{core}))$$

$$\text{under}((\phi \wedge \psi), \text{core}) = \text{under}(\phi, \text{core}) \wedge \text{under}(\psi, \text{core})$$

$$\text{under}((\psi \vee \chi), \text{core}) = \begin{cases} \text{under}(\chi, \text{core}) & \text{si } (\psi = \neg s_i, \\ & s_i \in \text{core}) \\ \top & \text{si } (\psi = \neg s_i, \\ & s_i \notin \text{core}) \\ (\text{under}(\psi, \text{core}) \\ \vee \text{under}(\chi, \text{core})) & \text{sinon} \end{cases}$$

Théorème 6. $\text{estKUNSAT}(\text{under}(\phi, \text{core}))$ implique $\text{estKUNSAT}(\phi)$.

L'intuition de la preuve qui suit est que chaque sélecteur s_i active une branche dans une conjonction de la formule. A chaque fois que la fonction 'under' est appelée avec un *core* non-vide, les branches non-activées avec un sélecteur venant de *core* vont être supprimées de la formule.

Démonstration. Soit ϕ une formule en NNF. Nous montrons que $\text{estKSAT}(\phi)$ implique $\text{estKSAT}(\text{under}(\phi, \text{core}))$ par induction sur la structure de ϕ . Supposons $\text{estKSAT}(\phi)$. Alors $\exists M, w$ t.q. $\langle M, w \rangle \models \phi$. Il y a deux cas possible dans la base d'induction : (1) $\phi = p$ et (2) $\phi = \neg p$. Dans ces deux cas, $\text{under}(\phi, \text{core}) = \phi$. Il y a quatre cas dans les étapes d'induction :

(1) $\phi = \diamond_a(\psi)$. $\exists M, w$ t.q. $\langle M, w \rangle \models \diamond_a(\psi)$. Alors $\exists M, w'$ t.q. $(w, w') \in R, \langle M, w' \rangle \models \psi$. Alors $\langle M, w' \rangle \models \text{under}(\psi, \text{core})$ par hypothèse d'induction. Ainsi $\langle M, w \rangle \models \text{under}(\phi, \text{core})$;

(2) $\phi = \square_a(\psi)$. Ce cas est similaire à (1).

(3) $\phi = (\psi \wedge \chi)$. $\exists \langle M, w \rangle \models (\psi \wedge \chi)$. Alors $\langle M, w \rangle \models \psi$ et $\langle M, w \rangle \models \chi$. Alors $\langle M, w \rangle \models \text{under}(\psi, \text{core})$ et $\langle M, w \rangle \models \text{under}(\chi, \text{core})$ par hypothèse d'induction. Ainsi $\langle M, w \rangle \models \text{under}(\phi, \text{core})$;

(4) $\phi = (\psi \vee \chi)$. Nous considérons trois cas :

(4.a) $\psi = \neg s_i$ et $s_i \in \text{core}$. Alors $\exists \langle M, w \rangle \models (\neg s_i \vee \chi)$ mais $s_i \in V(w)$, alors $\langle M, w \rangle \models \chi$. Alors $\langle M, w \rangle \models \text{under}(\chi, \text{core})$ par hypothèse d'induction. Ainsi $\langle M, w \rangle \models \text{under}(\phi, \text{core})$;

(4.b) $\psi = \neg s_i$ et $s_i \notin \text{core}$. $\exists \langle M, w \rangle \models (\neg s_i \vee \chi)$. mais nous avons toujours $\langle M, w \rangle \models \top$. Ainsi $\langle M, w \rangle \models \text{under}(\phi, \text{core})$.

(4.c) Ce cas est similaire à (3). \square

Le Théorème 6 montre que la fonction 'under' satisfait la Condition 4 de RECAR. Pour montrer qu'elle satisfait aussi la Condition 5, notons que la taille de $\text{under}^{n+1}(\phi, \text{core})$ est plus petit ou égale à celle de $\text{under}^n(\phi, \text{core}')$ (même si généralement les ensembles *core* et *core'* sont différents).

6 MoSaiC : RECAR pour K-SAT

Nous avons implémenté l'approche RECAR pour le problème de cohérence en logique modale dans le solveur MoSaiC, en utilisant les fonctions de sur et sous abstractions définies dans les sections précédentes. MoSaiC possède différentes fonctionnalités trouvées dans les solveurs de l'état-de-l'art. Comme dans Km2SAT [26], il simplifie la formule en entrée en effectuant différentes règles : le Box Lifting, le Flattening et la règle de Truth Propagation sur les opérateurs booléens et modaux (voir [26] pour plus d'informations). MoSaiC utilise aussi le solveur Glucose en mode incrémental [7, 1] pour décider la cohérence de chaque sur-abstraction (ψ).

Notons que l'implémentation diffère légèrement de la Figure 5. Par exemple, nous n'appelons pas Glucose sur ψ mais sur ψ sur laquelle nous avons appliqué les sélecteurs ; nous n'avons pas besoin de générer la sous-abstraction $\check{\phi}$ pour tester la condition ($\check{\phi} = \phi$) : il suffit de connaître le nombre de sélecteurs impliqués dans l'incohérence de la formule. Nous retournons aussi un modèle de Kripke dans la procédure principale, pas simplement SAT/UNSAT.

Nous tirons avantage d'une telle information pour fournir une nouvelle borne l . Et finalement, notons que dans

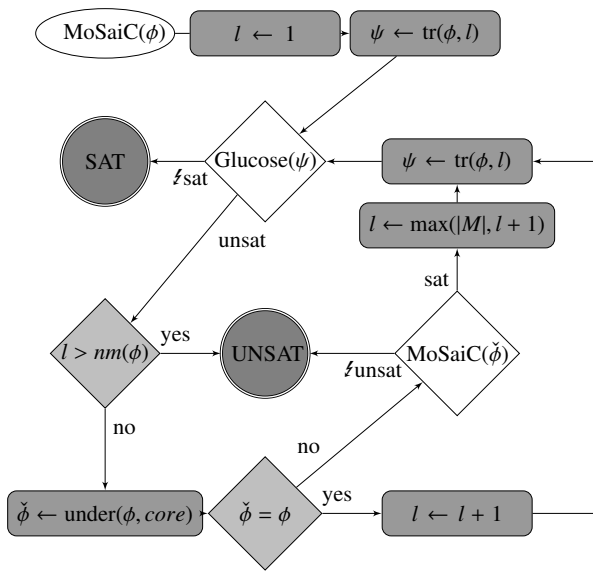


FIGURE 5 – MoSaiC : RECAR pour la logique modale K

notre cas $\max(|M|, l + 1)$ retourne toujours $|M|$ car il n'est pas possible de trouver un plus petit modèle de Kripke que M par construction de $\check{\phi}$. L'approche CEGAR du solveur présentée dans la Section 7 utilise un schéma similaire, mais sans l'étape récursif de RECAR.

7 Expérimentations

Nous comparons notre approche aux solveurs de l'état de l'art considérés dans [21], c'est-à-dire à : $K_S P$ 0.1 [21], BDDTab 1.0 [9], FaCT++ 1.6.4 [28], InKreSAT 1.0 [15], *SAT 1.3 [8], Km2SAT 1.0 [26] combiné avec le même Glucose que dans MoSaiC, Spartacus 1.0 [10] et une combinaison de l'Optimized Functional Translation [12] et Vampire 4.0 [17]. MoSaiC est configuré comme une approche CEGAR classique et comme une approche RECAR. Nous avons choisi d'exécuter ces solveurs sur les benchmarks suivant : L'ensemble complet des formules "TANCS-2000 modalised QBF (MQBF)" [20] complété avec les formules MQBF fournit dans [15], 1016 formules, 617 cohérents, 399 incohérents ; Les formules LWB [2], avec 56 formules choisit parmi les 18 classes, générées par le script fournit par les auteurs de [21], 1008 formules, 504 cohérents, 504 incohérents ; Les formules $3CNF_{KSP}$ générées aléatoirement [22] de profondeur modale égale à 1 ou 2, 1000 formules, 457 cohérents, 464 incohérents. Peu des solveurs considérés peuvent gérer des modalités multiples comme MoSaiC, de plus, à notre connaissance, il n'existe pas de benchmarks pour la logique modale K contenant plusieurs modalités. Nous avons utilisé une limite de mémoire de 32 Go et une limite de temps de 900

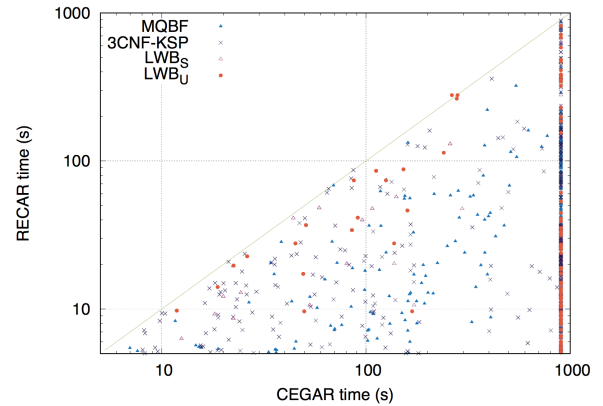


FIGURE 6 – Scatter-plot CEGAR contre RECAR

secondes par solveur par benchmark. Notons que pour des soucis d'espace, les résultats présentés ici sont globaux². Le comportement des solveurs varie beaucoup en fonction des familles de benchmarks. Nous croyons cependant que ces résultats fournissent une vision intéressante des capacités de l'approche proposée.

7.1 RECAR contre CEGAR

Nous pouvons voir sur la Figure 6 que dans la plus grande partie des benchmarks, l'approche RECAR surpasse l'approche CEGAR. La sous-abstraction fournit souvent une formule avec une borne plus petite, ce qui produit des CNF d'une taille raisonnable que le solveur SAT peut gérer. Notons que sur cette Figure, les memory-out des approches CEGAR et RECAR sont affichés comme des time-out, c'est-à-dire des points à 900 secondes. La différence dans les résultats vient majoritairement de la capacité de RECAR de résoudre des benchmarks incohérents : (1118/1367) benchmarks résolus par RECAR contre (155/1367) benchmarks résolus par CEGAR. Pour les benchmarks cohérents, l'amélioration de la borne venant de la sortie d'un appel récursif permet d'atteindre plus rapidement une sur-abstraction cohérent.

7.2 Comparaison avec les solveurs de l'état-de-l'art

Nous pouvons voir sur la Figure 7 que l'approche CEGAR utilisant notre sur-abstraction est le pire solveur alors que notre approche RECAR surpasse les autres solveurs. Km2SAT effectue un raisonnement spécifique capable de détecter rapidement certains des benchmarks incohérents sans avoir besoin de générer la CNF complètement, ce qui explique pourquoi il réussit mieux que notre approche CEGAR. *SAT alterne des raisonnements au niveau propositionnelle et au niveau modale, il peut donc être considéré

2. Des résultats additionnels peuvent être trouvés à l'adresse suivante : <http://www.cril.univ-artois.fr/~montmirail/mosaic>

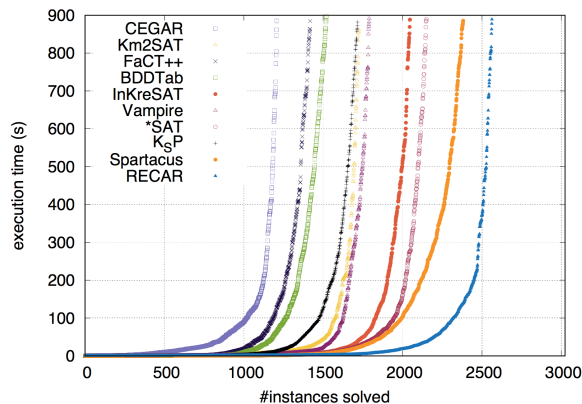


FIGURE 7 – Distribution des temps de résolutions

comme une approche CEGAR utilisant une sous-abstraction. Il obtient de bons résultats, malgré le fait qu’il est lié à un vieux solveur, SATO³. Notre meilleur compétiteur, Spartacus, est basé sur une méthode des tableaux, pas sur une approche utilisant un solveur SAT : les approches basées sur SAT n’étaient pas les meilleures approches pour attaquer des problèmes de cohérence en logique modale K jusqu’à présent. Spartacus atteint la plus part du temps la limite de temps sur ses benchmarks non-résolus alors que nous épuisons la mémoire disponible : les solveurs se comportent différemment et ont différentes limites.

8 Conclusion

Nous avons proposé ici une nouvelle approche pour résoudre des problèmes de décision en utilisant une approche basée sur du raffinement d’abstractions récursives. Nous avons montré la correction et la complétude de cette approche et nous l’avons instancié pour le problème de la cohérence d’une formule en logique modale K. Nous avons comparé notre approche face aux solveurs représentant, à notre connaissance, l’état-de-l’art pour la résolution pratique du problème K-SAT, sur une large gamme de benchmarks de logique modale K. Une simple approche CEGAR utilisant une sur-abstraction n’est pas compétitive du tout, due aux nombreux benchmarks disponibles qui sont incohérents. Notre approche RECAR, mixant les court-circuits SAT et UNSAT, surpasse les autres solveurs sur les benchmarks considérés. Ces résultats prometteurs sont une première étape vers des solveurs de logique modale plus efficaces : MoSaiC peut être étendu à d’autres logiques modales tel que KT, S4, S5 et KD45, en adaptant la traduction vers la logique propositionnelle. Nous croyons aussi que RECAR est une approche prometteuse pour attaquer des problèmes de décision au dessus de NP dans la hiérarchie

3. *SAT est profondément lié à SATO, ce qui rend très difficile une mise-à-jour avec un solveur SAT plus récent

polynomiale. Il n’est pas encore clair pour nous si les approches CEGAR existantes pour QBF telles que [13] peuvent être exprimées dans notre approche. C’est un futur travail passionnant.

Remerciements

Les auteurs remercient les relecteurs d’IJCAI et de JIAF pour leurs remarques éclairées. Une partie de ce travail est supportée par l’ANR SATAS (ANR-15-CE40-0017), le Ministère de l’Enseignement supérieur, de la Recherche et de l’Innovation et le conseil régional des Hauts-De-France à travers le “Contrat de Plan État Région (CPER)” et les financements FEDER.

Remarques

Cet article sera présenté à IJCAI 2017 [19]

Références

- [1] Audemard, Gilles, Jean-Marie Lagniez et Laurent Simon: *Improving glucose for incremental SAT solving with assumptions: Application to MUS extraction*. Dans *Proc. of SAT’13*, pages 309–317, 2013. http://dx.doi.org/10.1007/978-3-642-39071-5_23.
- [2] Balsiger, Peter, Alain Heuerding et Stefan Schwendimann: *A Benchmark Method for the Propositional Modal Logics K, KT, S4*. *J. Autom. Reasoning*, 24(3):297–317, 2000. <http://dx.doi.org/10.1023/A:1006249507577>.
- [3] Biere, Armin, Marijn Heule, Hans van Maaren et Toby Walsh (rédacteurs): *Handbook of Satisfiability*, tome 185 de *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2009, ISBN 978-1-58603-929-5.
- [4] Brummayer, Robert et Armin Biere: *Effective bit-width and under-approximation*. Dans *Proc. of EUROCAST’09*, pages 304–311, 2009.
- [5] Caridroit, Thomas, Jean Marie Lagniez, Daniel Le Berre, Tiago de Lima et Valentin Montmirail: *A SAT-based Approach For Solving The Modal Logic S5-Satisfiability Problem*. Dans *Proc. of AAI’17*, 2017.
- [6] Clarke, Edmund M., Orna Grumberg, Somesh Jha, Yuan Lu et Helmut Veith: *Counterexample-guided abstraction refinement for symbolic model checking*. *J. ACM*, 50(5):752–794, 2003. <http://doi.acm.org/10.1145/876638.876643>.
- [7] Eén, Niklas et Niklas Sörensson: *An Extensible SAT-solver*. Dans *Proc. of SAT’03*, pages 502–518, 2003. http://dx.doi.org/10.1007/978-3-540-24605-3_37.

- [8] Giunchiglia, Enrico, Armando Tacchella et Fausto Giunchiglia: *SAT-Based Decision Procedures for Classical Modal Logics*. J. Autom. Reasoning, 28(2):143–171, 2002. <http://dx.doi.org/10.1023/A:1015071400913>.
- [9] Goré, Rajeev, Kerry Olesen et Jimmy Thomson: *Implementing Tableau Calculi Using BDDs: BDDTab System Description*. Dans *Proc. of IJCAR'14*, pages 337–343, 2014. http://dx.doi.org/10.1007/978-3-319-08587-6_25.
- [10] Götzmann, Daniel, Mark Kaminski et Gert Smolka: *Spartacus: A Tableau Prover for Hybrid Logic*. Electr. Notes Theor. Comput. Sci., 262:127–139, 2010. <http://dx.doi.org/10.1016/j.entcs.2010.04.010>.
- [11] Halpern, Joseph Y. et Yoram Moses: *A Guide to Completeness and Complexity for Modal Logics of Knowledge and Belief*. Artif. Intell., 54(2):319–379, 1992. [http://dx.doi.org/10.1016/0004-3702\(92\)90049-4](http://dx.doi.org/10.1016/0004-3702(92)90049-4).
- [12] Horrocks, Ian, Ullrich Hustadt, Ulrike Sattler et Renate A. Schmidt: *4 Computational modal logic*. Studies in Logic and Practical Reasoning, 3:181–245, 2007.
- [13] Janota, Mikolás, William Klieber, Joao Marques-Silva et Edmund M. Clarke: *Solving QBF with counterexample guided refinement*. Artif. Intell., 234:1–25, 2016.
- [14] Järvisalo, Matti, Daniel Le Berre, Olivier Roussel et Laurent Simon: *The international SAT solver competitions*. AI Magazine, 33(1), 2012. <http://www.aaai.org/ojs/index.php/aimagazine/article/view/2395>.
- [15] Kaminski, Mark et Tobias Tebbi: *InKreSAT: Modal Reasoning via Incremental Reduction to SAT*. Dans *Proc. of CADE'13*, pages 436–442, 2013. http://dx.doi.org/10.1007/978-3-642-38574-2_31.
- [16] Khasidashvili, Zurab, Konstantin Korovin et Dmitry Tsarkov: *EPR-based k-induction with Counterexample Guided Abstraction Refinement*. Dans *Proc. of GCAI'15*, tome 36, pages 137–150, 2015.
- [17] Kovács, Laura et Andrei Voronkov: *First-Order Theorem Proving and Vampire*. Dans *Proc. of CAV'13*, pages 1–35, 2013. http://dx.doi.org/10.1007/978-3-642-39799-8_1.
- [18] Ladner, Richard E.: *The Computational Complexity of Provability in Systems of Modal Propositional Logic*. SIAM J. Comput., 6(3):467–480, 1977.
- [19] Lagniez, Jean Marie, Daniel Le Berre, Tiago de Lima et Valentin Montmirail: *A Recursive Shortcut for CEGAR: Application To The Modal Logic K Satisfiability Problem*. Dans *Proc. of IJCAI'17*, 2017.
- [20] Massacci, Fabio et Francesco M. Donini: *Design and results of TANCS-2000 non-classical (modal) systems comparison*. Dans *Proc. of TABLEAUX'00*, pages 52–56, 2000. http://dx.doi.org/10.1007/10722086_4.
- [21] Nalon, Cláudia, Ullrich Hustadt et Clare Dixon: *K_SP : A Resolution-Based Prover for Multimodal K*. Dans *Proc. of IJCAR'16*, pages 406–415, 2016. http://dx.doi.org/10.1007/978-3-319-40229-1_28.
- [22] Patel-Schneider, Peter F. et Roberto Sebastiani: *A new general method to generate random modal formulae for testing decision procedures*. CoRR, abs/1106.5261, 2011. <http://arxiv.org/abs/1106.5261>.
- [23] Prestwich, Steven David: *CNF encodings*. Dans *Handbook of Satisfiability*, pages 75–97. IOS Press, 2009. <http://dx.doi.org/10.3233/978-1-58603-929-5-75>.
- [24] Pulina, Luca: *The ninth QBF solvers evaluation - preliminary report*. Dans *Proc. of the 4th International Workshop (QBF 2016) co-located with (SAT 2016)*, pages 1–13, 2016. <http://ceur-ws.org/Vol-1719/paper0.pdf>.
- [25] Sebastiani, Roberto et Armando Tacchella: *SAT techniques for modal and description logics*. Dans *Handbook of Satisfiability*, pages 781–824. IOS Press, 2009. <http://dx.doi.org/10.3233/978-1-58603-929-5-781>.
- [26] Sebastiani, Roberto et Michele Vescovi: *Automated Reasoning in Modal and Description Logics via SAT Encoding: the Case Study of K(m)/ALC-Satisfiability*. J. Artif. Intell. Res. (JAIR), 35:343–389, 2009.
- [27] Seipp, Jendrik et Malte Helmert: *Counterexample-guided cartesian abstraction refinement*. Dans *Proc. of ICAPS'13*, 2013. <http://www.aaai.org/ocs/index.php/ICAPS/ICAPS13/paper/view/6034>.
- [28] Tsarkov, Dmitry et Ian Horrocks: *FaCT++ Description Logic Reasoner: System Description*. Dans *Proc. of IJCAR'06*, pages 292–297, 2006. http://dx.doi.org/10.1007/11814771_26.
- [29] Wang, Chao, Aarti Gupta et Franjo Ivancic: *Induction in CEGAR for detecting counterexamples*. Dans *Proc. of FMCAD'07*, pages 77–84, 2007. <http://dx.doi.org/10.1109/FAMCAD.2007.21>.

Les oiseaux picorant artificiels : une nouvelle méta-heuristique inspirée du comportement des pigeons

Jean-Baptiste Lamy¹

¹ LIMICS, Université Paris 13, Sorbonne Paris Cité, 93017 Bobigny
France, INSERM UMRS 1142, UPMC Université Paris 6, Sorbonne Universités, Paris, France
jean-baptiste.lamy@univ-paris13.fr

Résumé

De nombreux algorithmes et méta-heuristiques pour l'optimisation s'inspirent de la nature, par exemple du comportement social des abeilles pour l'algorithme ABC (Artificial Bee Colony). Ces algorithmes permettent souvent de résoudre des problèmes d'optimisation de nature très diverse. Dans cet article, nous proposons une nouvelle méta-heuristique, les oiseaux picorant artificiels (OPA), inspirée du comportement des pigeons lorsqu'ils sont à la recherche de nourriture. Cet algorithme est très simple, il donne de bons résultats et peut facilement s'adapter à des problèmes d'optimisation divers. Nous l'appliquons à l'optimisation globale non linéaire (y compris l'apprentissage de réseaux de neurones artificiels) et à la résolution du problème du voyageur de commerce, et nous montrons que OPA donne des résultats équivalents ou meilleurs que ABC sur plusieurs fonctions de test. Enfin, nous discuterons des différences entre les deux algorithmes et du choix des sources d'inspiration issues de la nature.

Abstract

Many optimization algorithms and metaheuristics have been inspired by nature, *e.g.* by the social behavior of honey bees for Artificial Bee Colony (ABC). These algorithms often permit solving a wide range of optimization problems. In this paper, we present Artificial Feeding Birds (AFB), a new metaheuristic inspired by the behavior of pigeons searching for food. AFB is very simple, yet efficient, and easy to adapt to various optimization problems. We apply it to unconstrained global nonlinear optimization (including the training of artificial neural networks) and to the resolution of traveling salesman problem. We also show that AFB gives results equivalent or better than ABC on several benchmark functions. Finally, we discuss the differences between the two algorithms and the choice of inspiration sources from nature.

1 Introduction

De nombreux algorithmes s'inspirent de la nature. Deux exemples bien connus sont les réseaux de neurones artificiels [1] et les algorithmes génétiques [5]. Plus récemment, les chercheurs se sont inspirés du comportement des animaux pour inventer de nouveaux algorithmes : l'organisation sociale des abeilles et des fourmis, la communication lumineuse chez les lucioles, la cohésion au sein d'un essaim d'oiseaux en vol ont chacun donné lieu à des algorithmes pour résoudre des problèmes d'optimisation tels que la recherche du minimum global d'une fonction numérique (optimisation globale non linéaire) ou la résolution du problème du voyageur de commerce. Ces algorithmes s'appuient généralement sur l'intelligence distribuée (*Swarm intelligence*), c'est-à-dire qu'ils font intervenir une population d'agents qui interagissent entre eux et avec leur environnement [8]. Ces agents sont très simples mais permettent d'accomplir des tâches complexes, et notamment de résoudre des problèmes d'optimisation [2]. Ces algorithmes sont appelés *méta-heuristiques* lorsqu'ils proposent une stratégie générale de « haut-niveau » qui peut être utilisée pour guider une recherche locale heuristique de plus « bas niveau » [16]. Par conséquent, une méta-heuristique n'est pas spécifique à un type donné de problème mais permet au contraire de résoudre des problèmes très différents, en fonction de la recherche de « bas niveau » utilisée.

Dans cet article, nous proposons les oiseaux picorant artificiels (OPA), une nouvelle méta-heuristique inspirée du comportement des pigeons lorsqu'ils sont à la recherche de nourriture. Ces oiseaux sont très communs et, en Europe, tout le monde a eu l'occasion de les observer en quête de nourriture sur les trottoirs. Notre hypothèse est la suivante : si le pigeon est aussi largement répandu, c'est que sa technique de recherche de nourriture est efficace et

donc qu'elle ferait une bonne source d'inspiration pour un algorithme. La méta-heuristique OPA résultante présente plusieurs avantages : elle est très simple, elle donne de bons résultats et elle peut facilement s'adapter à des problèmes d'optimisation divers. Nous montrerons l'adaptabilité d'OPA en l'appliquant à l'optimisation globale non linéaire, y compris l'entraînement de réseaux de neurones, et au problème du voyageur de commerce.

La suite de l'article est organisée de la manière suivante. La section 2 présente un état de l'art sur les algorithmes d'optimisation inspirés de la nature. La section 3 décrit le comportement que nous avons observé chez les pigeons en quête de nourriture. La section 4 décrit la méta-heuristique et son adaptation à l'optimisation globale non linéaire et à la résolution du problème du voyageur de commerce. La section 5 présente plusieurs expérimentations visant à fixer les valeurs des paramètres et à tester l'algorithme OPA. Enfin, la section 6 discute les principaux résultats, les différences entre OPA et ABC, le choix des sources d'inspiration issues de la nature, et propose des perspectives.

2 État de l'art

De nombreux algorithmes d'optimisation se sont inspirés de la nature [19]. La plupart s'inspirent des comportements sociaux des insectes [3] et de leurs capacités à communiquer entre eux par le biais de substances chimiques, de mouvements ou de signaux lumineux. Un premier exemple est le comportement des fourmis [6]. Les fourmis explorent leur environnement à la recherche de nourriture et laissent des traces sur leur passage sous forme de phéromones. Ces phéromones sont ensuite utilisées par d'autres fourmis comme des signaux leur indiquant les directions à suivre ou ne pas suivre. Le comportement des lucioles et leur utilisation de l'émission de lumière pour attirer leurs partenaires ont aussi donné lieu à un algorithme [18], qui a par la suite été appliqué à l'entraînement de réseaux de neurones artificiels [4]. L'optimisation par essaims particuliers (*particle swarm optimization*, PSO) s'inspire du comportement des animaux évoluant en formation (essaims d'insectes, vols d'oiseaux, bancs de poissons) [14]. Le déplacement de chaque individu de l'essaim dépend de la position des autres individus et de la qualité des solutions associées aux positions des éléments de l'essaim.

Le comportement des abeilles a également fait l'objet d'un algorithme qui repose sur le comportement des abeilles à la recherche de nectar et la communication entre abeilles par le biais de « danses » effectuées en vol [12]. Ces danses permettent à une abeille qui a trouvé un endroit riche en nectar de « recruter » d'autres abeilles afin de les conduire à cet endroit. L'algorithme ABC (*Artificial Bee Colony*) proposé par D Karaboga [12] considère 3 types d'abeilles : les ouvrières, les spectatrices et les scoutes. Chaque ouvrière est associée à une source de nectar, qui

correspond à une solution du problème d'optimisation. Les meilleures solutions correspondent à des sources de nectar plus riches. À chaque itération, l'ouvrière essaie d'améliorer sa solution en essayant une solution voisine, puis elle communique aux spectatrices la qualité et la position de sa solution. Les abeilles spectatrices obtiennent les informations des ouvrières ; à chaque itération, chaque spectatrice choisit d'aider une ouvrière, le choix étant aléatoire mais avec une probabilité plus grande de choisir les ouvrières qui ont des solutions plus intéressantes. La spectatrice essaie alors d'améliorer la solution de l'ouvrière, de la même manière que celle-ci. Lorsqu'une solution n'a pas pu être améliorée après un nombre défini de tentatives, celle-ci est abandonnée et l'abeille scoute est chargée de la remplacer par une solution aléatoire. L'algorithme ABC a été adapté à l'optimisation globale non linéaire [12], à l'optimisation de problèmes contraints [9], à l'entraînement de réseaux de neurones artificiels [10] et au *clustering* [11].

Plus récemment, plusieurs algorithmes se sont inspirés du comportement des oiseaux. XS Yang *et al.* se sont inspirés du comportement parasitique du coucou pour l'optimisation de fonctions numériques [17]. H Duan *et al.* ont proposé un algorithme inspiré du vol des pigeons et de leurs capacités à s'orienter dans l'espace en fonction de la position du soleil et à déterminer l'orientation du pôle Nord [7]. Cet algorithme a permis d'optimiser le parcours d'un robot volant.

3 Description du comportement des pigeons

Le pigeon est un oiseau très commun dans les villes européennes et facile à observer. Il se nourrit en picorant des graines ou des miettes de nourriture présentes sur le sol. Lorsqu'il n'a pas de nourriture à sa portée immédiate, le pigeon explore son environnement à la recherche de nourriture, en utilisant les deux modes de déplacement à sa disposition : la marche et le vol.

Nous avons observé que le pigeon accomplit quatre types de déplacement différents afin de chercher sa nourriture (figure 1) : (1) marcher vers une position proche de sa position actuelle, (2) s'envoler et se poser à une nouvelle position choisie de manière arbitraire ou aléatoire par le pigeon, (3) s'envoler et se poser à côté d'un autre pigeon, et (4) s'envoler et retourner à une position mémorisée riche en nourriture (nous avons constaté que les pigeons tendent à visiter préférentiellement certains endroits, comme les aires de pique-nique, et nous ferons l'hypothèse qu'ils gardent en mémoire ce genre d'emplacement). Typiquement, le pigeon marche à la recherche de nourriture (déplacement 1 effectué de manière répétitive). Au bout d'un certain temps, s'il ne trouve rien, il s'envole et se pose soit à un endroit aléatoire (déplacement 2), soit à côté d'un autre pigeon (déplacement 3), soit vers une position mémorisée (déplacement 4). Puis il recommence à marcher (retour au

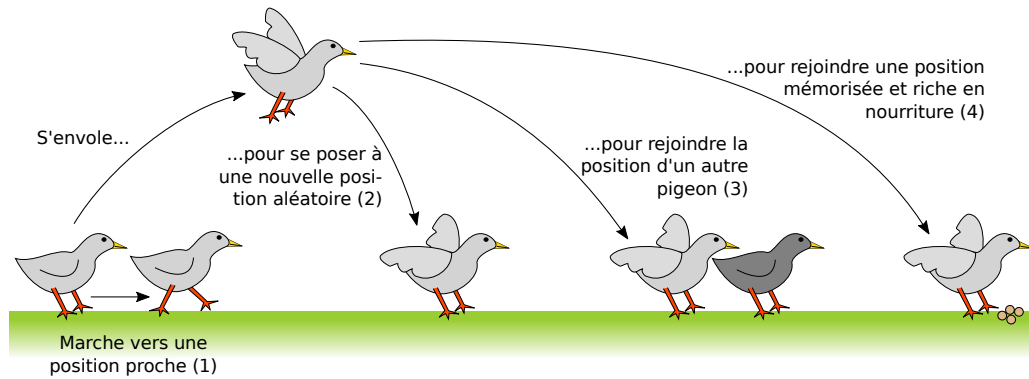


FIGURE 1 – Les quatre types de déplacement observés chez un pigeon lorsqu’il recherche de la nourriture.

déplacement 1), et ainsi de suite.

Ce comportement simple permet d’optimiser la recherche de nourriture. Le déplacement 1 (marche) permet une recherche locale. Ce comportement est censé car une position avec de la nourriture a de fortes chances d’être placée à proximité d’une autre position avec de la nourriture : par exemple si une miette de sandwich est présente à une position x , il est fort probable que d’autres miettes du même sandwich soient présentes dans des positions voisines ou proches x' , x'' , etc. Le déplacement 2 (nouvelle position aléatoire) permet l’exploration aléatoire. Le déplacement 3 (rejoindre un autre pigeon) permet de profiter de la nourriture qu’aurait éventuellement trouvée le pigeon rejoint. Cette action peut conduire à former de grands groupes de pigeons lorsqu’une quantité importante de nourriture est disponible à un endroit donné. Le déplacement 4 (retour à une position mémorisée) permet de retrouver de la nourriture, ou de continuer à en chercher aux alentours.

Nos observations ont porté sur les pigeons, cependant de nombreux autres oiseaux (moineaux par exemple) semblent adopter un comportement similaire.

4 Traduction en algorithmes

4.1 Méta-heuristique

Le comportement du pigeon optimise la recherche de nourriture. En nous inspirant de ce comportement, nous pouvons aboutir à une méta-heuristique pour minimiser une fonction de coût. Nous considérerons un système multi-agents où chaque agent est un « oiseau artificiel ». Chaque oiseau possède une position qui est une solution candidate pour la fonction de coût ; il dispose également d’une mémoire qui lui permet de retenir la meilleure position qu’il a trouvée. La position initiale de l’oiseau est aléatoire. Lorsque la position actuelle de l’oiseau minimise autant ou davantage la fonction de coût que la meilleure

position qu’il a en mémoire, la position actuelle devient la meilleure position et nous considérons que l’oiseau « a trouvé de la nourriture » et « a mangé ».

La méta-heuristique est un processus cyclique. À chaque itération, l’oiseau artificiel effectue l’un des quatre déplacements décrits à la section précédente : (1) il « marche », c’est-à-dire qu’il modifie légèrement sa position actuelle, (2) il « s’envole » vers une position aléatoire, c’est-à-dire qu’il réinitialise sa position actuelle avec des valeurs aléatoires, (3) il « s’envole » et rejoint un autre oiseau, c’est-à-dire qu’il choisit un oiseau j au hasard et réinitialise sa position actuelle avec la position actuelle de j , ou bien (4) il « s’envole » pour retourner à la meilleure position qu’il a mémorisée. Le choix du déplacement se fait de manière suivante : si l’oiseau vient de se poser au sol, il marche. Si l’oiseau vient de « manger », il marche. Sinon, le choix du déplacement est aléatoire, avec des probabilités spécifiques à chaque type de déplacement.

Pour définir l’algorithme général, nous considérons un problème d’optimisation défini par une fonction de coût dont les solutions appartiennent à l’ensemble admissible A . La recherche de nourriture par les pigeons telle que décrite en section 3 peut être considérée comme la recherche d’une solution optimale dans un espace à deux dimensions puisque les graines sont posées au sol ; dans ce cas, $A = \mathbb{R}^2$. Deux des quatre déplacements (3 et 4) sont génériques et indépendants du problème d’optimisation. Au contraire les deux autres (1 et 2) dépendent du problème. Nous définirons donc le problème d’optimisation à résoudre par un triplet de trois fonctions (*coût*, *vol*, *marche*) définies comme suit :

- *coût* : $A \rightarrow \mathbb{R}$, la fonction de coût à minimiser
- *vol* : $\emptyset \rightarrow A$, une fonction sans argument générant une position (ou solution) aléatoire
- *marche* : $\mathbb{N} \rightarrow A$, une fonction générant une nouvelle position aléatoire proche de la position de l’oiseau dont l’index est passé en paramètre

La méta-heuristique des oiseaux picorant artificiels (OPA)

que nous proposons prend 4 paramètres :

- n , le nombre d'oiseaux artificiels
- p_2 , la probabilité qu'un oiseau choisisse le déplacement 2
- p_3 , la probabilité qu'un oiseau choisisse le déplacement 3
- p_4 , la probabilité qu'un oiseau choisisse le déplacement 4

La probabilité de choisir le déplacement 1 (marche) est donc $p_1 = 1 - p_2 - p_3 - p_4$.

L'algorithme définit les 5 variables suivantes pour chaque oiseau ($1 \leq i \leq n$) :

- $x_i \in A$ est la position actuelle de l'oiseau i
- $f_i \in \mathbb{R}$ est la valeur de la fonction de coût pour la position actuelle de l'oiseau i
- $X_i \in A$ est la meilleure position trouvée et gardée en mémoire par l'oiseau i
- $F_i \in \mathbb{R}$ est la valeur de la fonction de coût pour la meilleure position de l'oiseau i
- $d_i \in \{1, 2, 3, 4\}$ est le type de déplacement effectué par l'oiseau à l'itération précédente (1 marche, 2 vol aléatoire, 3 vol vers un autre oiseau, 4 vol vers la position mémorisée)

L'algorithme 1 permet d'initialiser les variables, d'effectuer les itérations et de déterminer la meilleure solution trouvée en fin du processus. Lors de l'initialisation, la position x_i de chaque oiseau est définie de manière aléatoire avec la fonction $vol()$, le coût actuel f_i est calculé et la variable d_i est initialisée à 2 (car la position initiale correspond à celle à l'issue du déplacement 2).

À chaque itération, pour chaque oiseau i , l'algorithme choisit l'un des quatre déplacements possibles, effectue le déplacement choisi, met à jour la variable d_i avec le type de déplacement effectué, et met à jour la meilleure position mémorisée si nécessaire. Le choix du déplacement effectué se fait de la manière suivante : si le déplacement précédent était un vol (déplacement 2, 3 ou 4, c'est-à-dire $d_i \in \{2, 3, 4\}$) ou si le coût actuel f_i est égal au meilleur coût mémorisé F_i , l'oiseau fera le déplacement 1 (marche), sinon, l'oiseau choisira aléatoirement le déplacement 1, 2, 3 ou 4 avec les probabilités p_1 , p_2 , p_3 et p_4 , respectivement.

Si le déplacement 1 est choisi, la position de l'oiseau x_i est modifiée avec la fonction $marche()$ et le coût f_i est recalculé en appelant la fonction $coût()$. Si le déplacement 2 est choisi, la position de l'oiseau x_i est modifiée avec la fonction $vol()$ et le coût f_i est recalculé en appelant la fonction $coût()$. Si le déplacement 3 est choisi, la position de l'oiseau x_i et son coût actuel f_i sont remplacés par les valeurs x_j et f_j d'un autre oiseau choisi au hasard. Si le déplacement 4 est choisi, la position de l'oiseau x_i et son coût actuel f_i sont remplacés par les valeurs X_i et F_i mémorisées précédemment. Les deux derniers mouvements ne testent pas une nouvelle solution, donc aucun appel n'est fait à la fonction de coût.

Algorithme 1 Algorithme des oiseaux picorant artificiels (OPA) en pseudo-code.

Pour $1 \leq i \leq n$:

$x_i = X_i = vol()$
 $f_i = F_i = coût(x_i)$
 $d_i = 2$

Répéter :

Pour $1 \leq i \leq n$:

Si $d_i \in \{2, 3, 4\}$ ou $f_i = F_i$:

$p = 1$

Sinon :

$p =$ nombre réel aléatoire entre 0 et 1

Si $p > p_2 + p_3 + p_4$:

$d_i = 1$

$x_i = marche(i)$

$f_i = coût(x_i)$

Sinon, si $p < p_2$:

$d_i = 2$

$x_i = vol()$

$f_i = coût(x_i)$

Sinon, si $p < p_2 + p_3$:

$d_i = 3$

$j =$ nombre entier aléatoire entre 1 et n

$x_i = x_j$

$f_i = f_j$

Sinon :

$d_i = 4$

$x_i = X_i$

$f_i = F_i$

Si $f_i \leq F_i$:

$X_i = x_i$

$F_i = f_i$

Vérifier la condition d'arrêt

La meilleure solution trouvée est X_k , avec $1 \leq k \leq n$ tel que $F_k = \min(F_i)$ pour $1 \leq i \leq n$

Si le nouveau coût actuel f_i est inférieur ou égal au meilleur coût mémorisé F_i , alors la position et le coût actuels sont mémorisés (la condition « inférieur **ou égal** » permet de modifier la valeur mémorisée afin de garder une solution qui n'est pas meilleure que la précédente, mais qui est différente ; cela permet d'augmenter la « diversité » des solutions et de tester d'autres solutions).

Enfin, il est nécessaire d'ajouter une condition d'arrêt à l'algorithme. Nous suggérons de limiter l'algorithme à un nombre prédéfini de solutions testées (c'est-à-dire d'appels à la fonction $coût()$). Comme notre méta-heuristique effectue un nombre variable d'appels à la fonction $coût()$ par cycle, cette condition d'arrêt facilite la comparaison entre différents algorithmes ou entre différents résultats obtenus avec un même algorithme pour des paramètres différents.

Algorithme 2 Fonctions $vol()$ et $marche()$ pour résoudre des problèmes d’optimisation dans \mathbb{R}^d .

Fonction $vol()$:

$x' \in \mathbb{R}^d$

Pour $1 \leq k \leq d$:

$x'_k =$ nombre réel aléatoire entre x_{min} et x_{max}

Retourne x'

Fonction $marche(i)$:

$x' \in \mathbb{R}^d$, $x'_k = x_{ik}$ for $1 \leq k \leq d$

$j =$ nombre entier aléatoire entre 1 et n , $j \neq i$

$k =$ nombre entier aléatoire entre 1 et d

$r =$ nombre réel aléatoire entre -1 et 1

$x'_k = x'_k + r \times |x_{ik} - x_{jk}|$

If $x'_k < x_{min}$: $x'_k = x_{min}$

If $x'_k > x_{max}$: $x'_k = x_{max}$

Retourne x'

En fin de processus, la meilleure solution correspond à la « meilleure meilleure position » parmi l’ensemble des oiseaux (c’est-à-dire X_k pour k conduisant à la plus petite valeur F_k).

4.2 Adaptation à l’optimisation globale non linéaire non contrainte

Dans cette section, nous appliquerons la méta-heuristique OPA lorsque $A = \mathbb{R}^d$, c’est-à-dire pour des problèmes d’optimisation de fonctions numériques avec d dimensions, dont les solutions sont des séquences de d nombres réels compris entre x_{min} et x_{max} .

L’algorithme 2 décrit les fonctions $vol()$ et $marche()$ que nous proposons pour ce type d’optimisation. La fonction $vol()$ génère simplement une solution aléatoire. La fonction $marche()$ modifie l’une des coordonnées k (choisie au hasard) de la position x_i qui lui est passée en argument. La modification se fait avec une amplitude maximale qui est égale à la valeur absolue de la différence des coordonnées x_{ik} et x_{jk} , où x_i est la position actuelle de l’oiseau qui marche et x_j est la position d’un oiseau artificiel choisi aléatoirement. Cette fonction de marche est directement inspirée du déplacement des abeilles dans l’algorithme ABC [12].

4.3 Adaptation au problème du voyageur de commerce

Le problème du voyageur de commerce est un problème d’optimisation bien connu dans lequel un voyageur de commerce doit visiter un ensemble de villes V puis revenir à son point de départ. L’objectif est de trouver l’ordre optimal dans lequel visiter les villes de sorte à minimiser la distance totale parcourue. Dans cette section, nous appliquerons l’algorithme méta-heuristique OPA au problème du voyageur de commerce.

Algorithme 3 Fonctions $vol()$ et $marche()$ pour résoudre des problèmes d’optimisation combinatoire.

Fonction $vol()$:

$x' =$ séquence composée des éléments de V dans un ordre aléatoire

Retourne x'

Fonction $marche(i)$:

Répéter :

$j =$ nombre entier aléatoire entre 1 et n , $j \neq i$

$k =$ nombre entier aléatoire entre 1 et $|V|$

$\Delta =$ position de x_{ik} dans x_j

– position de $x_{i(k-1)}$ dans x_j

Si $abs(\Delta) > 1$: Quitter la boucle

$l = (k + \Delta)$ modulo $|V|$

Si $k > l$: Échanger les valeurs de k et l

$x' =$ clone de la séquence x_j

Inverser l’ordre des éléments entre x'_k et x'_l

Retourne x'

L’algorithme 3 décrit les fonctions $vol()$ et $marche()$ que nous proposons pour ce problème. La fonction $vol()$ génère simplement une séquence avec un ordre aléatoire. La fonction $marche()$ correspond à une variante de l’algorithme de recherche heuristique locale « 2-opt » [13], dans lequel une sous-séquence est choisie de manière aléatoire dans la séquence de ville, et l’ordre des éléments dans cette sous-séquence est inversé. Nous avons modifié l’algorithme « 2-opt » pour prendre en compte la similarité locale avec la position d’un autre oiseau aléatoire. Cette similarité est estimée (de manière grossière) par Δ , le nombre de villes séparant la ville localisée en position k (correspondant à la première extrémité de la sous-séquence) et la ville précédente dans x_j . Δ correspondra ensuite à la longueur de la sous-séquence.

Notons que ces deux fonctions ne sont *a priori* pas spécifiques du problème du voyageur de commerce, mais qu’elles pourraient être utilisées pour d’autres problèmes d’optimisation combinatoire.

5 Expérimentation

5.1 Implémentation

Les algorithmes décrits à la section précédente ont été implémentés en langage Python et exécutés avec l’interpréteur PyPy2 (une version de Python incluant un compilateur *juste-à-temps*, JIT). L’algorithme ABC a également été implémenté pour servir de référence.

5.2 Fonctions de tests

Nous avons utilisé quatre fonctions couramment utilisées pour tester les algorithmes d’optimisation : une fonc-

$$\begin{aligned}
Sphère(x_1, \dots, x_n) &= \sum_{i=1}^n x_i^2, & n = 5 \text{ et } -100 < x_i < 100 \\
Rosenbrock(x, y) &= (1 - x^2) + 100 \times (y - x^2)^2, & -2,048 < x < 2,048 \text{ et } -2,048 < y < 2,048 \\
Rastrigin(x_1, \dots, x_n) &= \sum_{i=1}^n x_i^2 - 10 \times \cos(2\pi x_i) + 10, & n = 10 \text{ et } -600 < x_i < 600 \\
Eggholder(x, y) &= -(y + 47) \sin\left(\sqrt{|y + \frac{x}{2} + 47|}\right) - x \sin\left(\sqrt{|x - (y + 47)|}\right) + 959,640662720851, & -512 < x, y < 512
\end{aligned}$$

FIGURE 2 – Les quatre fonctions de test utilisées pour l’expérimentation.

tion de sphère en 5 dimensions, la fonction de Rosenbrock, la fonction de Rastrigin avec 10 dimensions et la fonction d’Eggholder (à laquelle nous avons ajouté une constante, de sorte que son minimum global soit de 0 comme pour les trois autres fonctions, Figure 2).

Pour tester l’entraînement de réseaux de neurones artificiels, nous utiliserons le problème « Xor6 ». Il s’agit d’un réseau comprenant 2 neurones d’entrée E_1 et E_2 , 2 neurones intermédiaires I_1 et I_2 et 1 neurone de sortie S . Les neurones n’ont pas de biais, il y a donc 6 coefficients à optimiser en tout : E_1-I_1 , E_1-I_2 , E_2-I_1 , E_2-I_2 , I_1-S et I_2-S . Les 4 exemples d’apprentissage (E_1, E_2, S) sont (0, 0, 0), (0, 1, 1), (1, 0, 1), (1, 1, 0) et correspondent à un « ou exclusif » logique. La fonction de coût prend 6 paramètres correspondant aux 6 coefficients du réseau, et retourne l’erreur quadratique moyenne. Nous testerons le réseau Xor6 avec deux fonctions d’activation : la fonction sinus et la fonction sigmoïde, ce qui aboutit à deux fonctions à optimiser, $Xor6_{sin}$ et $Xor6_{sig}$ avec $x_{min} = -100$ et $x_{max} = 100$.

5.3 Études des paramètres

Pour étudier et fixer les valeurs des paramètres n , p_2 , p_3 et p_4 , d’une part nous avons observé les déplacements des agents au fil du temps à l’aide d’une interface graphique présentée en figure 3, et d’autre part nous avons généré des tableaux montrant l’évolution des résultats en fonction de deux paramètres. Le tableau 1 montre l’impact des paramètres p_2 et p_3 sur l’optimisation des fonctions $Sphère()$ et $Rastrigin()$. Pour la fonction $Sphère()$, nous constatons que le déplacement 2 (vol aléatoire) impacte négativement les résultats, en revanche le déplacement 3 (vol vers un autre oiseau) améliore les résultats. Minimiser la fonction $Sphère()$ peut se faire en « descendant la pente », par conséquent repartir vers une position aléatoire à de fortes chances de provoquer un retour en arrière. Au contraire, pour la fonction $Rastrigin()$, le déplacement 2 semble améliorer les résultats, contrairement au déplacement 3. Dans ce cas, le déplacement 3, s’il est trop fréquent, risque d’amener tous les oiseaux au même endroit, et ceux-ci risquent alors de rester bloqués dans un minimum local.

Nous proposons les valeurs suivantes, déterminées de manière empirique, qui semblent convenir à différents

$Sphère()$		p_3				
p_2		0,0	0,001	0,002	0,003	0,005
0,0		4,13e-73	1,80e-73	1,14e-73	2,13e-73	3,9e-74
0,015		1,36e-69	1,40e-69	1,87e-69	1,26e-70	8,30e-71
0,03		1,60e-66	1,03e-66	4,58e-67	2,18e-67	1,69e-67
0,05		6,38e-63	6,16e-62	5,93e-62	1,65e-62	2,59e-64
0,1		1,68e-54	3,90e-55	5,75e-55	4,88e-55	7,71e-56

$Rastrigin()$		p_3				
p_2		0,0	0,001	0,002	0,003	0,005
0,0		8,26e-17	1,51e-11	3,31e-11	3,41e-06	3,98e-03
0,015		0	7,11e-18	1,70e-14	1,07e-09	9,95e-04
0,03		1,24e-17	8,81e-16	8,11e-12	1,51e-07	2,85e-05
0,05		8,45e-15	8,26e-17	7,37e-17	7,64e-16	2,17e-08
0,1		4,38e-14	4,48e-12	1,79e-14	5,30e-12	1,16e-08

TABLE 1 – Résultats obtenus pour l’optimisation de deux fonctions de test pour différentes valeurs de p_2 et p_3 , avec 20 oiseaux et une condition d’arrêt de 40 000 solutions testées. Les résultats sont des moyennes obtenues sur 250 exécutions, et les plus petites valeurs sont les meilleures.

types de problèmes : $n = 20$, $p_2 = 0,03$, $p_3 = 0,003$, $p_4 = 0,6$ (et donc $p_1 = 0,367$). Ces valeurs seront systématiquement utilisées dans la suite du papier.

5.4 Optimisation globale non linéaire et comparaison avec l’algorithme ABC

Nous avons testé la méta-heuristique OPA sur les fonctions de test décrites en section 5.2, et nous avons comparé les résultats avec ceux obtenus en utilisant l’algorithme ABC. Les paramètres de l’algorithme ABC ont été fixés comme suit : $n = 20$ (nombre d’abeilles), $limite = 100$ (une source de nourriture est considérée comme épuisée si elle n’a pas pu être améliorée pendant $limite$ itérations). Pour l’algorithme OPA, nous avons utilisé les valeurs données à la section précédente. Pour les deux algorithmes, la condition d’arrêt a été fixée à 40 000 solutions testées (ce qui, pour l’algorithme ABC, correspond approximativement à 2000 cycles pour 20 abeilles, c’est-à-dire aux conditions expérimentales utilisées par D. Karaboga).

Le tableau 2 donne les résultats. Les résultats obte-

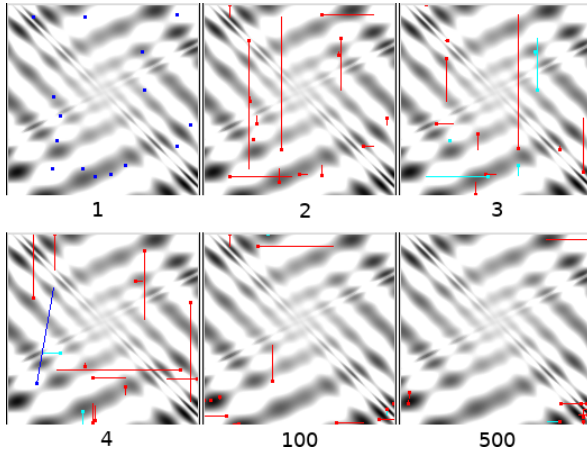


FIGURE 3 – Copie d’écran d’une interface graphique montrant les déplacements des oiseaux artificiels au fil du temps. Les copies d’écran correspondent aux itérations 1, 2, 3, 4, 100 et 500. L’image en noir et blanc montre la fonction *Eggholder()*, les valeurs les plus basses correspondant aux pixels les plus sombres. Chaque point coloré indique la position d’un oiseau et chaque ligne son déplacement. La couleur indique le type de déplacement effectué (rouge : marche, bleu : vol aléatoire, cyan : vol vers la meilleure position, vert : vol vers un autre oiseau). À l’itération 100, nous constatons que les oiseaux se sont regroupés en deux principaux *clusters*, en bas à gauche et en bas à droite. À l’itération 500, la quasi-totalité des oiseaux sont localisés en bas à droite, dans la région où se situe le minimum global de la fonction.

nus pour l’algorithme ABC sont similaires à ceux publiés par les auteurs [12] ou retrouvés dans la littérature [4]. Nous constatons que l’algorithme OPA donne de meilleurs résultats pour toutes les fonctions sauf pour la fonction *Xor6_{sig}()*, où l’écart reste faible. En revanche, l’algorithme OPA donne des résultats nettement meilleurs sur les fonctions *Rosenbrock()* et *Eggholder()*. Cela peut s’expliquer car, dans l’algorithme ABC, une seule coordonnée de la solution est modifiée à la fois puis, si la nouvelle solution n’est pas meilleure que la précédente, elle n’est pas conservée. Or la fonction *Rosenbrock()* décrit une « vallée » étroite en diagonale, il est donc nécessaire d’effectuer un déplacement simultané dans les deux dimensions. Dans l’algorithme OPA, nous modifions également une seule coordonnée de la solution à la fois, cependant la méta-heuristique permet d’effectuer plusieurs déplacements avant de retourner à la meilleure position (lorsque le déplacement 1 est choisi plusieurs fois de suite). Un problème similaire est rencontré dans l’optimisation de la fonction *Eggholder()*. Par ailleurs, le temps de calcul est comparable entre les deux algorithmes, avec un temps légèrement plus court pour OPA.

		OPA	ABC
<i>Sphère()</i>	Résultat	3,82e-67	6,23e-17
	Écart-type	2,21e-66	3,05e-17
	Temps (ms)	14	15
<i>Rosenbrock()</i>	Résultat	5,44e-06	8,73e-03
	Écart-type	3,49e-05	1,40e-02
	Temps (ms)	10	10
<i>Rastrigin()</i>	Résultat	1,42e-17	7,94e-15
	Écart-type	1,58e-16	1,02e-13
	Temps (ms)	26	28
<i>Eggholder()</i>	Résultat	0	0.48
	Écart-type	0	2.86
	Temps (ms)	15	17
<i>Xor6_{sin}()</i>	Résultat	4,54e-06	9,31e-06
	Écart-type	7,77e-06	1,13e-05
	Temps (ms)	37	39
<i>Xor6_{sig}()</i>	Résultat	6,63e-02	4,05e-02
	Écart-type	3,46e-02	3,10e-02
	Temps (ms)	37	39

TABLE 2 – Comparaison des résultats obtenus avec les oiseaux picorant artificiels (OPA) et les colonies d’abeilles artificielles (ABC) pour l’optimisation de différentes fonctions numériques. Les résultats sont des moyennes sur 250 exécutions, les valeurs les plus faibles sont les meilleures.

5.5 Test sur le problème du voyageur de commerce

Pour tester le problème du voyageur de commerce, nous avons utilisé le jeu de donnée FRI26 comprenant 26 villes. Nous avons fixé la condition d’arrêt à 40 000 solutions testées et nous avons effectué la moyenne de 250 exécutions. Nous obtenons un résultat de 940,7 (écart-type 6,9, temps moyen 144ms), ce qui représente une marge d’erreur de 0,4% par rapport à la meilleure solution possible correspondant à une distance de 937.

À titre de comparaison, nous avons également testé un algorithme génétique utilisant la technique des clefs aléatoires (*random-key*) proposée par Snyder *et al.* [15], sans ajouter d’amélioration locale (l’algorithme OPA n’en utilisant pas). Nous avons obtenu un résultat moyen de 1 059,0 (écart type 52,8, temps moyen 564ms).

6 Discussion

Dans cet article, nous avons présenté une nouvelle méta-heuristique, les oiseaux picorant artificiels (OPA), inspirée du comportement des pigeons à la recherche de nourriture. Nous avons montré que cette méta-heuristique était capable de résoudre des problèmes variés en l’adaptant à l’optimisation globale non linéaire (y compris l’entraînement de réseaux de neurones) et à la résolution du problème du voyageur de commerce. Nous avons également montré que OPA

conduit à des résultats proches voire meilleurs que l'algorithme ABC inspiré du comportement social des abeilles.

Les algorithmes inspirés de la nature sont réputés pour leur simplicité souvent surprenante au regard de leur performance [19]. Cela est particulièrement vrai pour la méta-heuristique OPA que nous proposons : l'algorithme 1 est très simple et, en particulier, il ne fait pas appel à des calculs complexes contrairement à l'algorithme ABC (pour le calcul des probabilités associées au choix de chaque source de nourriture par les spectatrices) ou à celui inspiré des lucioles. Par ailleurs OPA ne fait aucune supposition sur le problème d'optimisation en lui-même ou sur l'espace des solutions, en particulier il ne nécessite pas de calcul de distances. Les calculs de distance peuvent être complexes à mettre en oeuvre dans certaines situations, par exemple dans le problème du voyageur de commerce, calculer la distance entre deux solutions n'est pas trivial.

L'algorithme OPA présente certaines similarités avec l'algorithme ABC. Les agents des deux algorithmes réalisent les tâches suivantes : l'exploration aléatoire de l'espace des solutions, la recherche locale, le retour aux meilleures solutions trouvées et la concentration de plusieurs agents sur les solutions les plus prometteuses. Dans OPA, ces différentes tâches sont associées aux quatre déplacements des oiseaux : le vol aléatoire (déplacement 2) permet l'exploration aléatoire, la marche (déplacement 1) permet la recherche locale, le vol vers la position mémorisée (déplacement 4) permet de retourner à la meilleure position rencontrée, et le vol vers un autre oiseau (déplacement 3) permet à un oiseau de « joindre ses forces » à un autre oiseau et éventuellement d'adopter sa meilleure solution. Dans ABC, les différentes tâches sont associées aux différents types d'abeilles : ouvrières, spectatrices et scoutes. Les scoutes sont chargées de l'exploration aléatoire, les ouvrières de la recherche locale avec un retour à la meilleure position mémorisée en cas d'échec, et les spectatrices permettent de concentrer davantage de moyens sur les solutions les plus prometteuses. Cependant, une différence importante entre les deux algorithmes est que, dans ABC, l'amélioration itérative et le retour à la meilleure position sont associés en un seul comportement chez les ouvrières alors que dans OPA nous les avons dissociés en deux déplacements (1 et 4). Nous avons vu dans la section 5.4 que cette séparation permet d'améliorer nettement les résultats dans certaines situations.

La source d'inspiration de notre algorithme est particulière à deux titres. Premièrement, dans la section 2, nous avons constaté que la plupart des algorithmes d'optimisation issus de la nature se sont inspirés de comportements animaux jugés « exceptionnels » ou « extraordinaires » : l'organisation sociale des abeilles ou des fourmis, la capacité des lucioles à émettre de la lumière, celle des pigeons à déterminer la direction du pôle Nord, ou des coucous à pondre leurs oeufs dans les nids des autres oiseaux, *etc.*

Dans ce travail, nous nous sommes au contraire inspirés avec succès d'un comportement extrêmement « trivial », celui des oiseaux à la recherche de nourriture. D'un point de vue évolutionnaire, les comportements les plus performants conduisent à une plus grande survie des individus et sont donc les plus fréquemment rencontrés. Par conséquent, il pourrait être intéressant de s'inspirer des comportements largement répandus parmi de nombreuses espèces, plutôt qu'à des comportements exceptionnels. Cette hypothèse reste cependant à vérifier sur d'autres comportements.

Deuxièmement, la plupart des sources d'inspiration font intervenir la communication entre les animaux, que ce soit par le biais de signaux chimiques (fourmis), de mouvements (abeilles) ou de lumière (lucioles). Au contraire, en ce qui concerne les pigeons en quête de nourriture, nous n'avons pas observé de communication entre oiseaux : par exemple, lorsqu'un pigeon trouve à manger, il ne semble pas appeler d'autres pigeons à le rejoindre. Lorsqu'un pigeon en rejoint un autre, ce n'est pas suite à un appel, mais plutôt suite à une observation. De même, les deux fonctions marche que nous avons proposées ne sont pas entièrement aléatoires puisqu'elles prennent en compte la position actuelle d'un autre oiseau. Cela permet de réduire l'amplitude de la marche lorsqu'un autre oiseau est proche, ce qui peut simuler l'encombrement spatial : un oiseau marche plus vite lorsqu'il est seul que lorsqu'il est pris dans une « foule ». Cependant, l'obtention de cette position ne nécessite pas une vraie communication entre oiseaux. Par conséquent, dans OPA, l'observation remplace la communication. Dans l'algorithme que nous avons proposé (algorithme 1), chaque agent n'accède qu'à ses propres informations, ainsi qu'à la position des autres agents (pour le déplacement 1 et 3), laquelle peut être déduite par simple observation. Au contraire, un agent n'accède jamais à la meilleure position mémorisée par un autre agent. De manière assez surprenante, si nous modifions l'algorithme pour intégrer la communication, en faisant en sorte que le déplacement 3 amène l'oiseau sur la meilleure position mémorisée par un autre oiseau plutôt que sur sa position actuelle ($x_i = X_j$ et $f_i = F_j$), cela n'améliore pas les résultats, voire les détériore.

Dans la présentation de l'algorithme OPA, nous avons séparé la méta-heuristique de son adaptation aux deux problèmes (optimisation globale non linéaire et voyageur de commerce). Cette séparation facilite l'adaptation à de nouveaux problèmes, puisqu'il suffit de définir les deux fonctions *vol()* et *marche()*. Dans la plupart des algorithmes méta-heuristiques existants, la séparation n'est pas aussi claire et l'adaptation plus complexe.

Les perspectives de ce travail incluent (1) une évaluation plus poussée de la méta-heuristique et sa comparaison à d'autres méta-heuristiques ou des approches différentes (solveur numérique non linéaire par exemple), (2)

l'application de la méta-heuristique à d'autres problèmes d'optimisation, tels que le *clustering* et l'optimisation sous contraintes, ainsi que (3) l'amélioration de l'algorithme. Pour cela, une piste serait l'ajout de la biodiversité, en considérant plusieurs populations d'oiseaux aux caractéristiques différentes. La simplicité extrême de l'algorithme OPA pourrait aussi le rendre intéressant dans l'éducation.

Références

- [1] Abraham A: *Handbook of Measuring System Design*, chapitre Artificial neural networks. John Wiley & Sons, 2005.
- [2] Blum, C et X Li: *Natural computing series, Swarm intelligence : introduction and applications*, tome 43-85, chapitre Swarm intelligence in optimization. 2008.
- [3] Bonabeau, E, M Dorigo et G Theraulaz: *Inspiration for optimization from social insect behaviour*. Nature, 406 :39–42, 2000.
- [4] Brajevic, I et M Tuba: *Training feed-forward neural networks using firefly algorithm*. Dans *Recent advances in knowledge engineering and systems science*, 2013.
- [5] Darrell W: *A genetic algorithm tutorial*. Statistics and Computing, 4 :65–85, 1994.
- [6] Dorigo, M, M Birattari et T Stutzle: *Ant Colony Optimization - Artificial ants as a computational intelligence technique*. IEEE Comput. Intell. Mag, 1 :28–39, 2006.
- [7] Duan, H et P Qiao: *Pigeon-inspired optimization : a new swarm intelligence optimizer for air robot path planning*. International journal of intelligent computing and cybernetics, 7(1) :24–37, 2014.
- [8] Garnier, S, J Gautrais et G Theraulaz: *The biological principles of swarm intelligence*. Swarm intelligence, 1(1) :3–31, 2007.
- [9] Karaboga, D et B Basturk: *Artificial Bee Colony (ABC) optimization algorithm for solving constrained optimization problems*. Lecture Notes in Computer Science, 4529 :789–798, 2007.
- [10] Karaboga, D et C Ozturk: *Neural networks training by artificial bee colony algorithm on pattern classification*, 2009.
- [11] Karaboga, D et C Ozturk: *A novel clustering approach : Artificial Bee Colony (ABC) algorithm*. Applied Soft Computing, 11(1) :652–657, 2011.
- [12] Karaboga D: *An idea based on honey bee swarm for numerical optimization*. Technical report, 2005.
- [13] Marinakis Y: *Encyclopedia of optimization*, tome 1498-1506, chapitre Heuristic and metaheuristic algorithms for the traveling salesman problem. Springer-Verlag, 2009.
- [14] Poli, R, J Kennedy et T Blackwell: *Particle swarm optimization - An overview*, 2007.
- [15] Snyder, L V et M S Daskin: *A random-key genetic algorithm for the generalized traveling salesman problem*. European Journal of Operational Research, 174(1) :38–53, 2015.
- [16] Voss F: *Encyclopedia of optimization*, tome 2061-2075, chapitre Metaheuristics. Springer-Verlag, 2009.
- [17] Yang, X S et S Deb: *Cuckoo search via Levy flights*. Dans *World Congress on Nature & Biologically Inspired Computing*, tome 210-214, 2009.
- [18] Yang XS: *Firefly algorithms for multimodal optimization*. Stochastic Algorithms : Foundations and Applications - Lecture Notes in Computer Sciences, 5792 :169–178, 2009.
- [19] Yang XS: *Nature-inspired metaheuristic algorithms (second edition)*. Luniver Press, 2010.

Une logique modale normale de la confiance

Christopher Leturc

Grégory Bonnet

Normandie Université, UNICAEN, GREYC, CNRS UMR 6072, France
{prénom.nom}@unicaen.fr

Résumé

Nous proposons dans cet article une logique modale permettant de raisonner sur la confiance d'un agent envers le discours d'un autre agent. Notre modèle considère une modalité de confiance doxastique similaire à celle initialement introduite par Dundua et Uridia. En combinant cette formulation générale avec un formalisme STIT, nous exprimons aussi la confiance qu'un agent accorde à un autre lorsque ce dernier annonce qu'il veille à réaliser un énoncé. Nous retrouvons alors la notion de confiance dispositionnelle formalisée par Herzig *et al.* et caractérisons certaines propriétés qui avaient été axiomatisées par Singh. Enfin, nous montrons que notre modèle permet d'exprimer des notions de confiance collective.

Abstract

We propose in this paper a modal logic to reason on the trust of an agent towards a statement formulated by another agent. Our model considers a doxastic trust modality inspired from the work of Dundua and Uridia. We combine this modality with a STIT formalism that allows us to express that an agent trust another one when this latter states it will sees to it that a statement is true. Furthermore we can express a dispositional trust like Herzig *et al.* and we show that our formalism allows to deduce some axioms given by Singh. Finally, we propose some ways to express collective trust.

1 Introduction

Dans le domaine des systèmes multi-agents, une attention particulière a été accordée à la notion de confiance. En effet, au sein d'un tel système, les agents doivent coopérer les uns avec les autres pour satisfaire leurs buts. Toutefois, tous les agents ne sont pas nécessairement fiables ou coopératifs et une des techniques privilégiées pour déterminer si un agent est de confiance ou non est l'utilisation de systèmes de réputation [17]. Un système de réputation agrège des valeurs de *confiance* individuelles entre les agents afin de définir une valeur collective appelée *réputation*.

Dans la littérature, si de nombreuses approches s'appuient sur une représentation statistique de la confiance [11, 12, 16], d'autres travaillent à représenter les aspects socio-cognitifs de la confiance [2]. Certains travaux reposent sur des logiques floues [8, 13, 21] et introduisent une notion subjective de degré de confiance et de vraisemblance d'une source d'information. D'autres travaux s'intéressent à une confiance modélisée par des logiques modales [7, 10, 18, 19]. Ces logiques modales permettent d'exprimer la confiance par le biais d'une ou plusieurs modalités comme une intention, une croyance, une action ou un but qu'un agent possède. Si ces approches permettent d'exprimer facilement certains aspects de la confiance comme *la délégation*, elles s'intéressent principalement à la confiance dans l'action des autres agents. Or, dans le cadre des systèmes de réputation, les agents sont amenés à communiquer avec d'autres pour les informer, par exemple, de la qualité des services proposés par des agents tiers. Si certains travaux sont plutôt consacrés aux aspects de révision des connaissances fondés sur la confiance [15, 9], d'autres sont consacrés à la modélisation de la confiance qu'exprime un agent envers le discours d'un autre agent [4, 6, 14, 5].

Dans cet article, nous proposons une logique modale de la confiance permettant d'exprimer *la confiance accordée par un agent i envers un énoncé ϕ proposé par un autre agent j* . Cette formulation est générale car, en combinant notre modèle avec un formalisme STIT, nous pouvons aussi exprimer la confiance qu'un agent accorde à un autre lorsque ce dernier annonce qu'il veille à réaliser une action $[stit]_j\phi$. Nous montrons alors que nous retrouvons une notion de confiance dispositionnelle comme [10, 18] puis nous étendons notre formalisme à la confiance de groupe modélisant une notion de réputation. Cet article est structuré comme suit. Dans la section 2, nous présentons les logiques modales de la confiance et, dans la section 3, le formalisme STIT [1, 3] sur lequel repose notre approche. Notre axiomatique de la confiance est présentée en section 4 et nous montrons quelques propriétés de la

confiance en section 5. Enfin, nous présentons en section 6 la confiance collective.

2 Logiques modales de la confiance

Castelfranchi et Falcone [2] ont étudié la hiérarchie des différents composants fondamentaux de la confiance mais aussi ses aspects dynamiques, en particulier dans le cadre de la décision, de la construction d'intentions, de l'acte de faire confiance en soi ou de s'autoriser à déléguer des actions. Modéliser la confiance dans son entièreté est très complexe et la plupart des approches logiques se restreignent à des aspects précis de la confiance. Nous distinguons ici deux types d'approches : les logiques modales qui utilisent un prédicat pour représenter la confiance et les logiques qui s'appuient sur une modalité de confiance.

2.1 La confiance comme un prédicat

Herzig *et al.* [10] considèrent la confiance comme un prédicat signifiant qu'un agent i fait confiance à un autre agent j par rapport à une action α aboutissant à une proposition ϕ , si et seulement si toutes les propriétés suivantes sont vraies :

1. i a le but que ϕ soit réalisé,
2. i croit que :
 - (a) j est capable de réaliser l'action α ,
 - (b) j , après avoir réalisé α assurera ϕ ,
 - (c) j a l'intention de faire α .

Ceci permet de définir un prédicat de *confiance immédiate*¹. Cette notion traduit un aspect de la confiance dans le présent, à savoir qu'un agent j s'apprête bien à réaliser l'action pour laquelle i lui fait confiance. Une seconde notion de confiance, la *confiance dispositionnelle*², est la confiance accordée par un agent i à un agent j que cet agent j réalisera le but ϕ de i dans un contexte spécifique.

Smith *et al.* [19] considèrent aussi une confiance immédiate signifiant qu'un agent i fait confiance à un autre agent j pour ϕ si, et seulement si toutes les propriétés suivantes sont vraies :

1. i a le but ϕ ,
2. i croit que j réalise ϕ ,
3. i a l'intention que :
 - (a) j réalise ϕ ,
 - (b) i ne fasse pas ϕ .
4. i a le but que j ait l'intention de faire ϕ ,
5. i croit que j a l'intention de ϕ .

1. Occurrent trust.

2. Dispositional trust.

Remarquons que ces deux manières de définir la confiance ne suffisent pas pour considérer, par exemple, les mécanismes de transmission de la confiance car, ici, un agent ne peut pas avoir confiance dans *la confiance d'autrui*. De plus, nous ne pouvons pas caractériser un autre aspect de la confiance qui est la *confiance dans le discours d'un agent*. En effet, supposons qu'un agent j communiquant ψ à un agent i , notée $\alpha_{j,i}$, a pour conséquence $\phi = B_i\psi$ sur l'agent i . Si nous appliquons la définition de Herzig *et al.*, dire que i fait confiance à j pour l'action $\alpha_{j,i}$ implique que i a pour objectif de croire la proposition ψ . Cela exprime le fait que l'agent a confiance parce qu'il croit mais ne permet pas d'exprimer qu'il croit parce qu'il a confiance. Des travaux comme ceux de Christianson et Harbison [4] ou Demolombe [6] ont proposés de capturer d'autres aspects de la confiance en définissant des propriétés qui doivent être satisfaites pour voir apparaître une confiance. Par exemple, Demolombe modélise la confiance en la sincérité (un agent a confiance en la sincérité de son interlocuteur s'il croit que ce dernier croit son discours), la confiance en la crédibilité ou l'honnêteté [6]. Christianson et Harbison ont appliqué cette approche à la confiance en un discours [4]. Par exemple, leur confiance en l'honnêteté se modélise comme la confiance en la sincérité de Demolombe. Ainsi, il existe plusieurs aspects de la confiance pouvant être modélisés de façon indépendante. Dans cet article, nous considérons de manière abstraite la confiance en un énoncé produit par un autre agent sans pour autant préciser de quel aspect il s'agit (cela dépend uniquement de l'application considérée).

2.2 La confiance comme une modalité

Une modalité de confiance d'un agent i envers un agent j peut être considérée soit relativement à un contexte [18], soit relativement à des croyances sur l'état du monde [7] ou encore comme un moyen pour inférer de la croyance [14, 5].

La première approche, proposée par Singh [18], exprime une confiance dispositionnelle par l'intermédiaire d'une modalité $T_{i,j}(\phi, \psi)$ signifiant que, dans un contexte ϕ , un agent i fait confiance à un autre agent j pour réaliser ψ . Si ϕ est vraie, la confiance de l'agent i envers j est active. Une forme de confiance immédiate peut alors être exprimée par $T_{i,j}(\top, \psi)$ signifiant qu'à tout instant (et donc dans l'instant présent) i fait confiance à j pour réaliser ψ . Singh utilise pour cela une vingtaine d'axiomes. Par exemple, si ψ est déjà vraie alors l'agent i n'a pas confiance en j pour que, dans le contexte ϕ, ψ soit vraie.

La seconde approche, proposée par Dundua et Uridia [7], introduit une modalité $T_{i,j}$ représentant la confiance comme une forme de croyance qu'un agent entretient sur un autre agent. La confiance est alors axiomatisée au regard d'une modalité de croyance comme par exemple :

$$T_{i,j}\phi \Leftrightarrow B_i T_{i,j}\phi.$$

Toutefois, ces travaux n'expriment que la capacité d'un agent à effectuer une tâche [18] ou encore le lien qu'il peut exister entre une notion de confiance dispositionnelle ou immédiate et la croyance d'un agent à bien effectuer l'action [7]. De plus, ces dernières approches ne s'intéressent pas aux discours des agents. Les travaux de Dastani *et al.* et Liau se sont intéressés à l'influence de la confiance dans l'assimilation de nouvelles informations au travers du formalisme BIT [5, 14]. Ce formalisme est une logique non-normale qui propose une modalité de confiance $T_{i,j}$ entre deux agents, en y incorporant la notion de thème (ou *topics*) modélisant une transitivité entre les confiances accordées à un discours relevant du même thème. Cette approche est pertinente mais, en tant que logique non-normale, elle se passe de certaines propriétés, comme $T_{i,j}(\phi \wedge \psi) \Rightarrow T_{i,j}\phi$, nécessaires pour exprimer les propriétés intuitives des confiances dispositionnelles.

Nous proposons alors une *logique normale* avec une modalité permettant de représenter *une confiance abstraite dans un énoncé produit par des agents*. Par exemple, un agent peut avoir confiance en un autre lorsque celui-ci déclare qu'il *veille à ce que quelque chose soit vrai lorsqu'un contexte est vérifié* ou encore qu'il a *confiance dans l'énoncé d'un tiers*. Cette formulation générale nous permet d'exprimer une notion de confiance dispositionnelle mais aussi les aspects doxastiques de la confiance. Comme certains aspects de la notion de confiance peuvent s'appuyer sur la notion d'action, nous considérons le formalisme STIT.

3 Formalisme STIT

3.1 Langage et règles d'inférences

Présentons le langage de STIT [1, 3].

$$\phi ::= p \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \phi \Rightarrow \phi \mid B_i\phi \mid [stit]_i\phi \mid X\phi \mid Y\phi$$

Ce langage utilise plusieurs modalités et règles d'inférence. La modalité $[stit]_i\phi$ signifie que l'agent i veille à ce que ϕ soit vraie. Les modalités X et Y sont des modalités temporelles représentant respectivement le futur et le passé. Enfin, B_i est la modalité de croyance associée à un agent i .

Le formalisme STIT considère les règles d'inférence classiques suivantes : **PC** (toutes les tautologies du calcul propositionnel notée **R₁** à **R₈** et rappelée en Figure 1), le modus ponens **MP**, la substitution **Sub** et la nécessité **Nec** pour toutes les modalités. De plus, le formalisme STIT considère **KD45** pour les modalités B_i et **S5** pour les $[stit]_i$, **KD** pour X et **K** pour Y de type \square . Les règles d'inférence spécifiques à ce formalisme sont données en Figure 2.

$$\begin{aligned} \phi &\Rightarrow (\psi \Rightarrow \phi) & (R_1) \\ (\phi \Rightarrow (\psi \Rightarrow \theta)) &\Rightarrow ((\phi \Rightarrow \psi) \Rightarrow (\phi \Rightarrow \theta)) & (R_2) \\ (\neg\psi \Rightarrow \neg\phi) &\Rightarrow (\phi \Rightarrow \psi) & (R_3) \\ (\neg\phi \Rightarrow \neg\psi) &\Rightarrow ((\neg\phi \Rightarrow \psi) \Rightarrow \phi) & (R_4) \\ \phi &\Rightarrow (\phi \vee \psi), \psi \Rightarrow (\phi \vee \psi) & (R_5) \\ (\phi \vee \psi) &\Rightarrow ((\phi \Rightarrow \theta) \Rightarrow ((\psi \Rightarrow \theta) \Rightarrow \theta)) & (R_6) \\ \phi &\Rightarrow (\psi \Rightarrow \phi \wedge \psi) & (R_7) \\ \phi \wedge \psi &\Rightarrow \phi, \phi \wedge \psi \Rightarrow \psi & (R_8) \end{aligned}$$

FIGURE 1 – Tautologies du calcul propositionnel

$$\begin{aligned} (\mathbf{Alt}_X) & \quad \neg X\phi \Rightarrow X\neg\phi \\ (\mathbf{Alt}_Y) & \quad \neg Y\phi \Rightarrow Y\neg\phi \\ (\mathbf{Conv}_{X,Y}) & \quad \phi \Rightarrow XY\phi \\ (\mathbf{Conv}_{Y,X}) & \quad \phi \Rightarrow YX\phi \\ (\mathbf{Rel}_{B_i,[stit]_i}) & \quad B_i\phi \Rightarrow [stit]_j\phi \\ (\mathbf{AIA}) & \quad \neg B_i\neg[stit]_j\phi_1 \wedge \neg B_i\neg[stit]_k\phi_2 \\ & \quad \Rightarrow \neg B_i\neg([stit]_j\phi_1 \wedge [stit]_k\phi_2) \\ (\mathbf{Rel}_{[stit]_i,[stit]_N}) & \quad \bigwedge_{n \in N} [stit]_n\phi_n \Rightarrow [stit]_N \bigwedge_{n \in N} \phi_n \\ (\mathbf{NCUH}) & \quad X\neg B_i\neg\phi \Rightarrow \langle stit \rangle_N X\phi \end{aligned}$$

FIGURE 2 – Règles d'inférence de STIT

4 Une logique de la confiance

Nous proposons une logique modale de la confiance où cette dernière est représentée par un ensemble de modalités $T_{i,j}\phi$ exprimant qu'un agent i accorde sa confiance à un agent j pour l'énoncé ϕ . Cette proposition ϕ peut exprimer le pouvoir de l'agent j à agir sur le monde, un état mental de cet agent, ou simplement un état du monde. Par exemple, une proposition ϕ peut être *l'agent j veille à ψ* que nous pouvons écrire à l'aide du formalisme STIT, $T_{i,j}[stit]_j\psi$. Cette formule signifie alors que l'agent i a confiance en j lorsque j déclare veiller à ce que ψ soit vraie. Nous pouvons aussi exprimer une confiance envers l'énoncé d'un état de croyance d'un agent avec la formule $T_{i,j}B_j\phi$ signifiant que l'agent i a confiance en j lorsque j déclare croire ϕ . L'intérêt de cette modalité doublement indicée est donc d'exprimer clairement la disposition d'un agent à faire confiance à un agent tiers donné.

4.1 Langage de la confiance

Précisons le langage de notre modèle. Soit un ensemble fini de variables propositionnelles $\mathcal{A} = \{a, b, c, \dots\}$, deux agents $i, j \in N$, un sous-ensemble d'agents $N \subseteq N$, $p \in \mathcal{A}$ une variable propositionnelle et ϕ une formule du langage

STIT. Le langage que nous employons est le suivant :

$$\psi ::= p \mid \neg\psi \mid \psi \wedge \psi \mid \psi \vee \psi \mid \psi \Rightarrow \psi \mid T_{i,j}\psi \mid \phi$$

Remarquons que la modalité B_i est bien différente de $T_{i,i}$ car elle signifie que l'agent i a confiance en lui pour l'énoncé ϕ .

4.2 Une logique modale normale

Nous considérons les axiomes suivants : les tautologies du calcul propositionnel (\mathbf{R}_1 à \mathbf{R}_7), les règles d'inférence classiques des logiques modales (\mathbf{K} , \mathbf{Nec} , \mathbf{Sub}) et une règle de non-inconsistance entre confiances (\mathbf{D}). Notre logique de la confiance est donc une logique normale qui satisfait la nécessité, la substitution, le modus ponens et l'axiome de Kripke \mathbf{K} .

La **nécessitation** signifie que si une proposition ϕ est un théorème ($\vdash \phi$) alors n'importe quel agent i peut avoir confiance en n'importe quel autre agent j pour ce théorème ($\vdash T_{i,j}\phi$). La **substitution** signifie que si un théorème est prouvé et que nous substituons uniformément une formule quelconque à une lettre de proposition, la formule résultante est aussi un théorème. Le **modus ponens** signifie que si une proposition ϕ est prouvée et qu'il est aussi prouvé que $\phi \Rightarrow \psi$ alors la formule ψ est prouvée.

Enfin, notre modalité de confiance satisfait l'axiome \mathbf{K} : si un agent i a confiance en un agent j sur ϕ qui implique ψ alors, si i a confiance en j pour ϕ alors i a aussi confiance en j pour ψ .

$$T_{i,j}(\phi \Rightarrow \psi) \Rightarrow T_{i,j}\phi \Rightarrow T_{i,j}\psi \quad (\mathbf{K})$$

4.3 Non-inconsistance de la confiance

Nous voulons exprimer le fait que si un agent i fait confiance à j pour un énoncé, il ne peut pas lui faire confiance pour son contraire pour des raisons de cohérence de discours : il n'est pas possible de faire confiance à un agent qui raisonne de manière inconsistante.

$$T_{i,j}\phi \Rightarrow \neg T_{i,j}\neg\phi \quad (\mathbf{D})$$

Ceci traduit le fait que si un agent i a confiance en un agent j pour ϕ alors cet agent i n'a pas confiance en j pour $\neg\phi$. Cependant, nous ne pouvons généraliser cet axiome pour tout autre agent $k \in \mathcal{N}$, $T_{i,j}\phi \Rightarrow \neg T_{i,k}\neg\phi$. En effet, considérons un agent i qui a confiance en j lorsque j déclare veiller à ce que ϕ soit vrai, c'est-à-dire $T_{i,j}[stit]_j\phi$, et qui a aussi confiance en un agent k lorsque ce dernier déclare veiller à ce que $\neg\phi$ soit vrai, c'est-à-dire $T_{i,k}[stit]_k\neg\phi$. Ici, la confiance ne préjuge pas de la vérité de la formule sur laquelle elle s'applique : deux agents peuvent croire être dans deux mondes différents et donc tenir deux discours inconsistants entre eux tout en étant, par exemple, sincères et honnêtes. Nous désirons alors capturer le fait qu'un agent

puisse tout de même accorder une forme de confiance à ces deux agents, tout en sachant que les formules ϕ et $\neg\phi$ ne pourront pas toutes les deux être vraies. Or, comme $[stit]$ est une modalité $\mathbf{S5}$ alors – selon l'axiome \mathbf{T} – $[stit]_j\phi \Rightarrow \phi$ et $[stit]_k\neg\phi \Rightarrow \neg\phi$. En appliquant la **nécessitation** sur ces deux dernières formules qui sont des théorèmes, nous obtenons $T_{i,j}([stit]_j\phi \Rightarrow \phi)$ et $T_{i,k}([stit]_k\neg\phi \Rightarrow \neg\phi)$ puis par \mathbf{K} , nous déduisons $T_{i,j}[stit]_j\phi \Rightarrow T_{i,j}\phi$ et $T_{i,k}[stit]_k\neg\phi \Rightarrow T_{i,k}\neg\phi$. Enfin, par **modus ponens**, nous déduisons alors $T_{i,j}\phi$ et $T_{i,k}\neg\phi$. Ainsi, si $T_{i,j}\phi \Rightarrow \neg T_{i,k}\neg\phi$ était un axiome, nous déduirions aussi $\neg T_{i,k}\neg\phi$ ce qui est contradictoire avec $T_{i,k}\neg\phi$.

4.4 Lien entre confiance et croyance

Dundua et Uridia [7] ont axiomatisé un lien entre la confiance et la croyance. Notre modalité de confiance étant une forme de croyance, nous posons que pour tout agent $i, j \in \mathcal{N}$, si un agent i croit qu'il a confiance en j sur l'énoncé ϕ alors il est vrai que l'agent i a confiance en l'agent j sur l'énoncé ϕ :

$$B_i T_{i,j}\phi \Rightarrow T_{i,j}\phi \quad (\mathbf{C4}_{T,B})$$

De la même manière, nous posons la réciproque signifiant que si un agent i a confiance en un agent j sur l'énoncé ϕ alors l'agent i croit qu'il a confiance en l'agent j sur l'énoncé ϕ :

$$T_{i,j}\phi \Rightarrow B_i T_{i,j}\phi \quad (\mathbf{4}_{T,B})$$

De plus, nous remarquons que nous ne pouvons pas considérer un axiome tel que si un agent croit que quelque chose est vrai alors il ne peut pas faire confiance dans un discours qui annonce le contraire : $\forall i, j \in \mathcal{N}, B_i\phi \Rightarrow \neg T_{i,j}\neg\phi$ ($\mathbf{D}_{B,T}$). En effet, lorsque nous considérons la confiance en la sincérité, il est tout à fait possible de croire quelque chose en contradiction avec la confiance qu'on accorde à quelqu'un.

5 Propriétés

Comme indiqué en Section 2, plusieurs travaux se sont déjà intéressés à modéliser les aspects de la confiance, comme la notion de *confiance immédiate* ou encore la notion de *confiance dispositionnelle* [10, 19]. D'autres se sont intéressés aux propriétés de la confiance [18] ou son lien avec la notion de croyance [7]. Dans cette section, nous montrons que notre modalité de confiance permet d'exprimer ces aspects.

5.1 Distributivité de la confiance

Proposition 5.1. Soit $i, j \in \mathcal{N}$,

$$i) T_{i,j}\phi \wedge T_{i,j}\psi \equiv T_{i,j}(\phi \wedge \psi) \quad (\wedge_T)$$

$$ii) (T_{i,j}\phi \vee T_{i,j}\psi) \Rightarrow T_{i,j}(\phi \vee \psi) \quad (\vee_T)$$

Démonstration. i) Soit $i, j \in \mathcal{N}$,

(\Rightarrow) Supposons $T_{i,j}\phi \wedge T_{i,j}\psi$, c'est-à-dire $T_{i,j}\phi$ et $T_{i,j}\psi$. Or, $\phi \Rightarrow (\psi \Rightarrow (\phi \wedge \psi))$ est une tautologie et donc un théorème. Par conséquent, en appliquant la nécessité, nous déduisons $T_{i,j}(\phi \Rightarrow (\psi \Rightarrow (\phi \wedge \psi)))$. Par application de K, nous déduisons $T_{i,j}\phi \Rightarrow T_{i,j}(\psi \Rightarrow (\phi \wedge \psi))$. Par modus ponens, nous déduisons $T_{i,j}(\psi \Rightarrow (\phi \wedge \psi))$ puis, de nouveau par application de K, nous obtenons $T_{i,j}\psi \Rightarrow T_{i,j}(\phi \wedge \psi)$. Enfin, par modus ponens, nous prouvons $T_{i,j}(\phi \wedge \psi)$ et, par synthèse, nous prouvons $T_{i,j}\phi \wedge T_{i,j}\psi \Rightarrow T_{i,j}(\phi \wedge \psi)$

(\Leftarrow) Supposons $T_{i,j}(\phi \wedge \psi)$. Or, $\phi \wedge \psi \Rightarrow \phi$ et $\phi \wedge \psi \Rightarrow \psi$ sont des tautologies et donc des théorèmes. En appliquant la nécessité et l'axiome K, nous déduisons $T_{i,j}(\phi \wedge \psi) \Rightarrow T_{i,j}\phi$ et $T_{i,j}(\phi \wedge \psi) \Rightarrow T_{i,j}\psi$. Or, $(T_{i,j}(\phi \wedge \psi) \Rightarrow T_{i,j}\phi) \wedge (T_{i,j}(\phi \wedge \psi) \Rightarrow T_{i,j}\psi) \Rightarrow (T_{i,j}(\phi \wedge \psi) \Rightarrow T_{i,j}\phi \wedge T_{i,j}\psi)$. Enfin, par modus ponens, nous prouvons $T_{i,j}\phi \wedge T_{i,j}\psi$ et, par synthèse, nous prouvons $T_{i,j}\phi \wedge T_{i,j}\psi \Leftarrow T_{i,j}(\phi \wedge \psi)$.

ii) Soit $i, j \in \mathcal{N}$. Supposons $T_{i,j}\phi \vee T_{i,j}\psi$. Or, $\phi \Rightarrow (\phi \vee \psi)$ et $\psi \Rightarrow (\phi \vee \psi)$ sont des tautologies et donc des théorèmes. En appliquant la nécessité et l'axiome K, nous obtenons $T_{i,j}\phi \Rightarrow T_{i,j}(\phi \vee \psi)$ et $T_{i,j}\psi \Rightarrow T_{i,j}(\phi \vee \psi)$. Or, $(T_{i,j}\phi \vee T_{i,j}\psi) \Rightarrow ((T_{i,j}\phi \Rightarrow T_{i,j}(\phi \vee \psi)) \Rightarrow (T_{i,j}\psi \Rightarrow T_{i,j}(\phi \vee \psi))) \Rightarrow T_{i,j}(\phi \vee \psi)$ est aussi une tautologie du calcul propositionnel. Par modus ponens, nous prouvons $T_{i,j}(\phi \vee \psi)$ et, par synthèse, nous prouvons $(T_{i,j}\phi \vee T_{i,j}\psi) \Rightarrow T_{i,j}(\phi \vee \psi)$. \square

Ces propositions se fondent sur le fait que nous utilisons une logique normale : un agent i ne peut pas faire confiance en un discours inconsistant (l'ensemble des propositions énoncées par l'agent j) car cela le conduirait à avoir confiance en toute proposition émanant de j . Par exemple, un agent ne peut pas faire confiance à un autre pour l'enrichir et l'appauvrir (en admettant que les deux propositions soient contraires) mais il n'exprimera pas pour autant une non confiance à son égard. Un agent peut bien entendu faire confiance à un autre pour l'enrichir ou l'appauvrir. Toutefois, il s'agit d'une tautologie et un agent a toujours confiance dans une tautologie, par nécessité.

5.2 Confiance dispositionnelle

Rappelons que la confiance dispositionnelle présentée dans [10] exprime la confiance en un autre agent pour réaliser une action lorsqu'un contexte donné est vérifié. Exprimons cette notion en utilisant le formalisme STIT : la confiance dispositionnelle est la confiance (dans le présent) que, lorsque le contexte ϕ est vérifié, l'agent veille bien à ce que ψ soit réalisé.

$$\forall i, j \in \mathcal{N}, T_{s_{i,j}}(\phi, \psi) \stackrel{\Delta}{=} T_{i,j}(\phi \Rightarrow [stit]_j\psi)$$

Contrairement à [10], nous n'introduisons pas la notion d'intention car celle-ci est déjà exprimée dans la modalité

STIT. En effet, si un agent veille à ce que quelque chose devienne vrai, c'est qu'il a l'intention de rendre cette chose vraie. Mais intéressons-nous désormais aux propriétés algébriques liées aux opérateurs \wedge et \vee de notre confiance dispositionnelle. Que pouvons-nous déduire si un agent i a confiance en un agent j pour que, lorsque le contexte ϕ est réalisé, l'agent j veille à ce que $\psi \wedge \theta$ soit vrai, ou que dans un contexte $\phi \vee \psi$, l'agent j veille à ce que θ soit vrai.

Proposition 5.2. Soit $i, j \in \mathcal{N}$,

- i) $T_{s_{i,j}}(\phi, \psi \wedge \theta) \equiv T_{s_{i,j}}(\phi, \psi) \wedge T_{s_{i,j}}(\phi, \theta)$ (TS_1)
- ii) $(T_{s_{i,j}}(\phi, \theta) \vee T_{s_{i,j}}(\psi, \theta)) \Rightarrow T_{s_{i,j}}(\phi \wedge \psi, \theta)$ (TS_2)
- iii) $T_{s_{i,j}}(\phi \vee \psi, \theta) \equiv T_{s_{i,j}}(\phi, \theta) \wedge T_{s_{i,j}}(\psi, \theta)$ (TS_3)
- iv) $(T_{s_{i,j}}(\phi, \psi) \vee T_{s_{i,j}}(\phi, \theta)) \Rightarrow T_{s_{i,j}}(\phi, \psi \vee \theta)$ (TS_4)

Démonstration. Soit $i, j \in \mathcal{N}$,

i) Supposons $T_{s_{i,j}}(\phi, \psi \wedge \theta)$. Par définition, $T_{i,j}(\phi \Rightarrow [stit]_j(\psi \wedge \theta)) \equiv T_{i,j}(\phi \Rightarrow ([stit]_j\psi \wedge [stit]_j\theta)) \equiv T_{i,j}((\phi \Rightarrow [stit]_j\psi) \wedge (\phi \Rightarrow [stit]_j\theta)) \equiv T_{i,j}(\phi \Rightarrow [stit]_j\psi) \wedge T_{i,j}(\phi \Rightarrow [stit]_j\theta) \equiv T_{s_{i,j}}(\phi, \psi) \wedge T_{s_{i,j}}(\phi, \theta)$.

ii) Supposons $T_{s_{i,j}}(\phi, \theta) \vee T_{s_{i,j}}(\psi, \theta)$. Par définition, $T_{i,j}(\phi \Rightarrow [stit]_j\theta) \vee T_{i,j}(\psi \Rightarrow [stit]_j\theta)$. Donc, par modus ponens sur \vee_T , nous déduisons $T_{i,j}((\phi \Rightarrow [stit]_j\theta) \vee (\psi \Rightarrow [stit]_j\theta)) \equiv T_{i,j}((\phi \wedge \psi \Rightarrow [stit]_j\theta)) \equiv T_{s_{i,j}}(\phi \wedge \psi, \theta)$. Par synthèse, nous prouvons alors $(T_{s_{i,j}}(\phi, \theta) \vee T_{s_{i,j}}(\psi, \theta)) \Rightarrow T_{s_{i,j}}(\phi \wedge \psi, \theta)$.

iii) Supposons $T_{s_{i,j}}(\phi \vee \psi, \theta)$. Par définition, $T_{i,j}((\phi \vee \psi) \Rightarrow [stit]_j\theta) \equiv T_{i,j}((\phi \Rightarrow [stit]_j\theta) \wedge (\psi \Rightarrow [stit]_j\theta)) \equiv T_{i,j}(\phi \Rightarrow [stit]_j\theta) \wedge T_{i,j}(\psi \Rightarrow [stit]_j\theta)$.

iv) Supposons $T_{s_{i,j}}(\phi, \psi) \vee T_{s_{i,j}}(\phi, \theta)$. Par définition, $T_{i,j}(\phi \Rightarrow [stit]_j\psi) \vee T_{i,j}(\phi \Rightarrow [stit]_j\theta)$. Donc, par modus ponens sur \vee_T , nous déduisons $T_{i,j}((\phi \Rightarrow [stit]_j\psi) \vee (\phi \Rightarrow [stit]_j\theta)) \equiv T_{i,j}(\phi \Rightarrow ([stit]_j\psi \vee [stit]_j\theta))$. Par nécessité, nous obtenons $T_{i,j}(\phi \Rightarrow (([stit]_j\psi \vee [stit]_j\theta) \Rightarrow [stit]_j(\psi \vee \theta))) \Rightarrow ((\phi \Rightarrow ([stit]_j\psi \vee [stit]_j\theta)) \Rightarrow (\phi \Rightarrow [stit]_j(\psi \vee \theta)))$. Par K et modus ponens, nous déduisons $T_{i,j}(\phi \Rightarrow [stit]_j(\psi \vee \theta))$. Enfin, par synthèse, nous obtenons $(T_{s_{i,j}}(\phi, \psi) \vee T_{s_{i,j}}(\phi, \theta)) \Rightarrow T_{s_{i,j}}(\phi, \psi \vee \theta)$ \square

Ceci nous permet de prouver quelques déductions logiques naturelles comme, par exemple, lorsqu'un agent i a confiance dans un agent j pour ϕ et que cet agent i a aussi confiance dans l'agent j pour que, dans le contexte ϕ , l'agent j veille à ψ alors l'agent i a confiance en l'agent j pour ψ . Nous pouvons déduire aussi que si l'agent i a confiance en j qu'il veille à $\psi \Rightarrow \theta$ et si i a une confiance dispositionnelle en j pour que, dans le contexte ϕ , j veille à ψ et i a aussi une confiance dispositionnelle en j pour que, dans le contexte θ , j veille à θ alors i a aussi une confiance dispositionnelle en j pour que, dans le contexte ϕ , j veille à θ . Enfin, si i a confiance en j pour ϕ et qu'il a aussi une confiance dispositionnelle en j pour que, dans le contexte $\phi \wedge \psi$, j veille à θ alors i a aussi une confiance dispositionnelle envers j pour que, dans le contexte ψ , j veille à θ .

Proposition 5.3. Soit $i, j \in \mathcal{N}$,

- i) $(T_{i,j}\phi \wedge Ts_{i,j}(\phi, \psi)) \Rightarrow T_{i,j}\psi$
- ii) $(T_{i,j}[stit]_j(\psi \Rightarrow \theta) \wedge Ts_{i,j}(\phi, \psi) \wedge Ts_{i,j}(\theta, \vartheta)) \Rightarrow Ts_{i,j}(\phi, \vartheta)$
- iii) $T_{i,j}\phi \wedge Ts_{i,j}(\phi \wedge \psi, \theta) \Rightarrow Ts_{i,j}(\psi, \theta)$

Démonstration. Soit $i, j \in \mathcal{N}$,

i) Supposons $(T_{i,j}\phi \wedge Ts_{i,j}(\phi, \psi))$. Par définition, $T_{i,j}\phi \wedge T_{i,j}(\phi \Rightarrow [stit]_j\psi)$. Par application de K, nous déduisons $T_{i,j}\phi \wedge (T_{i,j}\phi \Rightarrow T_{i,j}[stit]_j\psi)$. Par modus ponens, $T_{i,j}[stit]_j\psi$. Or, par nécessité sur $T_{[stit]}$ et K, $T_{i,j}[stit]_j\psi \Rightarrow T_{i,j}\psi$. Ainsi, par modus ponens, nous prouvons $T_{i,j}\psi$. Par synthèse, nous prouvons alors $(T_{i,j}\phi \wedge Ts_{i,j}(\phi, \psi)) \Rightarrow T_{i,j}\psi$.

ii) Supposons $T_{i,j}[stit]_j(\psi \Rightarrow \theta) \wedge Ts_{i,j}(\phi, \psi) \wedge Ts_{i,j}(\theta, \vartheta)$, c'est-à-dire $T_{i,j}([stit]_j(\psi \Rightarrow \theta) \wedge (\phi \Rightarrow [stit]_j\psi) \wedge (\theta \Rightarrow [stit]_j\vartheta))$. En appliquant l'axiome K sur $[stit]$, nous avons $T_{i,j}([stit]_j\psi \Rightarrow [stit]_j\theta) \wedge (\phi \Rightarrow [stit]_j\psi) \wedge (\theta \Rightarrow [stit]_j\vartheta)$. Or, $([stit]_j\psi \Rightarrow [stit]_j\theta) \Rightarrow (\phi \Rightarrow ([stit]_j\psi \Rightarrow [stit]_j\theta))$ est une tautologie et donc, par modus ponens, nous déduisons $\phi \Rightarrow ([stit]_j\psi \Rightarrow [stit]_j\theta)$. De plus, $(\phi \Rightarrow ([stit]_j\psi \Rightarrow [stit]_j\theta)) \Rightarrow ((\phi \Rightarrow [stit]_j\psi) \Rightarrow (\phi \Rightarrow [stit]_j\theta))$ est aussi une tautologie. Ainsi, par application de la nécessité, K, puis modus ponens, nous prouvons $T_{i,j}(\phi \Rightarrow [stit]_j\theta)$.

À ce point, nous avons donc $T_{i,j}(\phi \Rightarrow [stit]_j\theta) \wedge T_{i,j}(\theta \Rightarrow [stit]_j\vartheta)$. Or, l'axiome T sur $[stit]$ indique que $[stit]_j\theta \Rightarrow \theta$ est un théorème. Par un raisonnement analogue, nous avons $T_{i,j}(\phi \Rightarrow \theta) \wedge T_{i,j}(\theta \Rightarrow [stit]_j\vartheta)$ et donc $T_{i,j}(\phi \Rightarrow [stit]_j\vartheta)$, c'est-à-dire, $Ts_{i,j}(\phi, \vartheta)$. Par synthèse, nous concluons $(T_{i,j}[stit]_j(\psi \Rightarrow \theta) \wedge Ts_{i,j}(\phi, \psi) \wedge Ts_{i,j}(\theta, \vartheta)) \Rightarrow Ts_{i,j}(\phi, \vartheta)$.

iii) Supposons $T_{i,j}\phi \wedge Ts_{i,j}(\phi \wedge \psi, \theta)$. Par définition, $T_{i,j}\phi \wedge T_{i,j}(\phi \wedge \psi \Rightarrow [stit]_j\theta)$, c'est-à-dire $T_{i,j}(\phi \wedge (\phi \wedge \psi \Rightarrow [stit]_j\theta))$. En substituant dans la tautologie $\alpha \wedge (\alpha \wedge \beta \Rightarrow \gamma) \Rightarrow (\beta \Rightarrow \gamma)$ ³, nous déduisons $T_{i,j}(\psi \Rightarrow [stit]_j\theta)$. Par synthèse, nous prouvons alors $T_{i,j}\phi \wedge Ts_{i,j}(\phi \wedge \psi, \theta) \Rightarrow Ts_{i,j}(\psi, \theta)$. \square

5.3 Similarités avec la théorie de Singh

Nous pouvons faire un lien direct entre notre axiomatique et celle proposée par Singh [18]. Ces derniers utilisent vingt-et-un axiomes, notés de T_1 à T_{21} . Nous ne considérons pas T_1 , T_4 , T_5 et T_{19} qui s'appuient sur le fait que la confiance est une modalité qui s'active ou non selon le contexte. Nous ne considérons pas non plus les axiomes T_{10} à T_{13} et T_{20} et T_{21} qui font appel à des modalités ou des prédicats supplémentaires comme l'engagement, le scepticisme ou l'espérance. Nos propriétés algébriques et déductions logiques nous permettent de retrouver les autres axiomes : T_2 , T_3 , T_6 , T_7 , T_8 , T_9 . Nous traitons les axiomes T_{14} à T_{18} relatifs à la confiance collective dans la section suivante.

3. Rappelons que $\alpha \wedge (\alpha \wedge \beta \Rightarrow \gamma) \equiv \alpha \wedge (\neg\alpha \vee \neg\beta \vee \gamma) \equiv \alpha \wedge \neg\alpha \vee \alpha \wedge (\neg\beta \vee \gamma) \equiv \alpha \wedge (\beta \Rightarrow \gamma)$. Et $\alpha \wedge (\beta \Rightarrow \gamma) \Rightarrow (\beta \Rightarrow \gamma)$.

Notre Proposition 5.3.iii est similaire à l'axiome T_2 ($T_{i,j}(\phi \wedge \phi', \psi) \wedge \phi \Rightarrow T_{i,j}(\phi', \psi)$) qui indique que si un agent i a confiance envers j pour ψ dans le contexte $\phi \wedge \phi'$ et que ϕ est vrai alors l'agent i a confiance en j pour ψ dans le contexte ϕ' .

Notre Proposition 5.2.i permet de déduire l'axiome T_3 ($T_{i,j}(\phi, \psi \wedge \psi') \wedge \neg\psi \Rightarrow T_{i,j}(\phi, \psi)$) qui indique que si un agent i a confiance envers j pour $\psi \wedge \psi'$ dans le contexte ϕ et que $\neg\psi$ est vrai alors l'agent i a confiance envers j pour ψ dans le contexte ϕ . En effet, en appliquant l'augmentation d'hypothèse R_1 sur notre Proposition 5.2.i, nous déduisons la même propriété. En effet, si $Ts_{i,j}(\phi, \psi \wedge \psi') \Rightarrow Ts_{i,j}(\phi, \psi)$ alors, par R_1 , nous avons $Ts_{i,j}(\phi, \psi) \Rightarrow (Ts_{i,j}(\phi, \psi \wedge \psi') \wedge \neg\psi \Rightarrow Ts_{i,j}(\phi, \psi))$. Donc, si $Ts_{i,j}(\phi, \psi \wedge \psi')$ est prouvé alors, par modus ponens, nous prouvons $Ts_{i,j}(\phi, \psi \wedge \psi') \wedge \neg\psi \Rightarrow Ts_{i,j}(\phi, \psi)$.

L'axiome T_6 (De $T_{i,j}(\phi, \psi)$, $\phi' \vdash \phi$ et $\phi \not\vdash \psi$, l'agent infère $T_{i,j}(\phi', \psi)$) est naturel par syllogisme dans notre formalisme. En effet, il exprime que si un agent i a confiance envers j pour ψ dans un contexte ϕ et que $\phi' \Rightarrow \phi$ est un théorème alors i a confiance envers j pour ψ dans le contexte ϕ' .

Nos Propositions 5.2.iii et 5.2.i sont similaires aux axiomes T_7 ($T_{i,j}(\phi, \psi) \wedge T_{i,j}(\phi', \psi) \Rightarrow T_{i,j}(\phi \vee \phi', \psi)$) et T_8 ($T_{i,j}(\phi, \psi) \wedge T_{i,j}(\phi, \psi') \Rightarrow T_{i,j}(\phi, \psi \wedge \psi')$) indiquant respectivement (T_7) que si un agent i a confiance envers j pour ψ dans un contexte ϕ et si i a confiance envers j pour ψ dans un contexte ϕ' alors l'agent i a confiance envers j pour ψ dans le contexte $\phi \vee \phi'$, et (T_8) que si un agent i a confiance envers j pour ψ dans le contexte ϕ et qu'il a confiance envers j pour ψ' dans le contexte ϕ alors l'agent i a confiance envers j pour $\psi \wedge \psi'$ dans le contexte ϕ .

Enfin, notre Proposition 5.3.ii est similaire à l'axiome T_9 (De $T_{i,j}(\phi, \psi)$, $\psi \vdash \phi'$ et $T_{i,j}(\phi', \psi')$, l'agent infère $T(\phi, \psi')$) indiquant que si un agent i a confiance en j pour ψ dans le contexte ϕ , qu'il a aussi confiance en j pour ψ' dans le contexte ϕ' et que $\psi \Rightarrow \phi'$ alors l'agent i a confiance en j pour ψ' dans le contexte ϕ .

6 Vers une confiance collective

Jusqu'à présent, nous nous sommes intéressés à la confiance qu'il pouvait y avoir entre deux agents. Dans cette section, nous étendons la notion de confiance à des groupes d'agents afin d'exprimer de la *confiance collective* avec deux notions : la *confiance commune* et la *confiance par transitivité*. Remarquons que, si nous nous restreignons dans ce travail à ces deux notions, nous sommes bien conscients de l'intérêt d'autres aspects de la confiance collective, comme la *confiance réciproque* ou encore la *confiance mutuelle*.

6.1 Confiance commune

Afin de définir la *confiance commune*, nous nous appuyons sur la définition de Smith *et al.* [19] : un groupe d'agents I a une confiance commune envers un autre groupe d'agents J si, et seulement si, tous les agents de I ont confiance en tous les agents de J .

$$\forall I, J \subseteq \mathcal{N} : Tc_{I,J}\phi \triangleq \bigwedge_{(i,j) \in I \times J} T_{i,j}\phi$$

Il s'agit d'un consensus en ce sens que tous les agents du groupe I doivent avoir confiance dans tous les agents du groupe J à propos du même énoncé. De plus, nous considérons une notion duale au prédicat $Tc_{I,J}$, notée $Tc_{I,J}^*$, comme étant le prédicat suivant.

$$\forall I, J \subseteq \mathcal{N} : Tc_{I,J}^*\phi \triangleq \bigvee_{(i,j) \in I \times J} T_{i,j}\phi$$

Ce prédicat exprime qu'il existe au moins un agent de I faisant confiance à un agent de J . En effet, si aucun agent de I ne fait confiance aux agents de J pour ϕ alors nous pouvons écrire $\neg Tc_{I,J}^*\phi$.

Remarquons qu'il existe d'autres définitions dans la littérature. Par exemple, Herzig *et al.* [10] considèrent un prédicat de *réputation* indiquant qu'il y a consensus si une *majorité* d'agents de I a une confiance dispositionnelle dans les agents de J . Pour raison de simplicité, nous n'introduisons pas de notion de majorité et ne considérons donc pas cette notion de réputation.

6.2 Confiance par transitivité

Notre notion de confiance considère de manière abstraite chaque aspect de la confiance, seule l'application de notre logique va spécifier l'aspect considéré. Lorsque nous considérons, par exemple, l'aspect de la confiance comme une croyance en la sincérité et la fiabilité d'un agent à produire un énoncé, nous pouvons caractériser certaines propriétés spécifiques à cet aspect comme la *transitivité*. En effet, il semble naturel de dire que si un agent i a confiance en la confiance qu'accorde un agent j à propos d'un troisième agent k alors i a confiance en k si et seulement si i pense que j est sincère et fiable. Nous présentons alors dans cette section une confiance collective qui pourrait être construite sur cette notion. Celle-ci repose sur l'idée que si deux agents i et j ont mutuellement confiance dans la confiance que l'autre accorde à un agent k pour ϕ alors les deux agents font confiance à l'agent k pour ϕ .

$$Ta_{\{i,j\},\{k\}}\phi \triangleq (T_{j,i}T_{i,k}\phi \Rightarrow T_{j,k}\phi) \wedge (T_{i,j}T_{j,k}\phi \Rightarrow T_{i,k}\phi)$$

Nous pouvons par ailleurs nous demander pourquoi nous ne considérons pas la transitivité de la confiance

$T_{i,j}T_{j,k}\phi \Rightarrow T_{i,k}\phi$ comme un principe de raisonnement naturel à la confiance ? En effet, cette notion de transitivité s'applique très bien à certains aspects de la confiance comme la croyance en la fiabilité d'un agent. En revanche ceci ne peut pas s'appliquer pour d'autres aspect de la confiance comme la confiance en l'honnêteté d'un discours. Considérons par exemple qu'un agent i a confiance en l'honnêteté d'un agent j à propos de la confiance que j accorde à un agent k . Supposons que i sait que k n'est pas fiable et qu'il sait que j ne le sait pas. Dans ces conditions, i peut faire confiance à j quand à l'honnêteté de sa déclaration mais il ne peut pas faire confiance à k par transitivité. C'est pour cette raison que nous ne pouvons pas considérer comme principe général la transitivité de la confiance. Toutefois, il est possible d'utiliser ce principe de manière locale lorsque les agents s'associent avec d'autres agents et pensent que ces agents sont justement fiables. C'est l'idée fondatrice de notre *confiance par transitivité*. Ainsi, nous généralisons cette confiance par transitivité à un groupe d'agents $I \subseteq \mathcal{N}$ envers un autre groupe $K \subseteq \mathcal{N}$.

$$Ta_{I,K}\phi \triangleq \bigwedge_{(i,j) \in I \times I} \bigwedge_{k \in K} (T_{i,j}T_{j,k}\phi \Rightarrow T_{i,k}\phi)$$

De la même manière, nous exprimons la *non confiance par transitivité*. Si un agent i a confiance en j dans le fait que ce dernier n'accorde pas sa confiance à k pour ϕ alors l'agent i ne fait pas confiance à k pour ϕ . Pour tout $I, K \subseteq \mathcal{N}$:

$$Tb_{I,K}\phi \triangleq \bigwedge_{(i,j) \in I \times I} \bigwedge_{k \in K} (T_{i,j}\neg T_{j,k}\phi \Rightarrow \neg T_{i,k}\phi)$$

6.3 Propriétés usuelles

La confiance commune a les propriétés usuelles.

Proposition 6.1. *Pour tout $I, J, K \subseteq \mathcal{N}$,*

- i) $Tc_{I,J}\phi \wedge Tc_{I,J}\psi \equiv Tc_{I,J}(\phi \wedge \psi)$
- ii) $(Tc_{I,J}\phi \vee Tc_{I,J}\psi) \Rightarrow Tc_{I,J}(\phi \vee \psi)$
- iii) $Tc_{I,J}\phi \Rightarrow \neg Tc_{I,K}^*\neg\phi$

Démonstration. Pour tout $I, J, K \subseteq \mathcal{N}$,

i) Supposons $Tc_{I,J}\phi \wedge Tc_{I,J}\psi$. Or pour tout $i, j \in I \times J$, $T_{i,j}\phi \wedge T_{i,j}\psi \equiv T_{i,j}(\phi \wedge \psi)$. Donc :

$$\begin{aligned} Tc_{I,J}\phi \wedge Tc_{I,J}\psi &\equiv \bigwedge_{(i,j) \in I \times J} T_{i,j}\phi \wedge \bigwedge_{(i,j) \in I \times J} T_{i,j}\psi \\ &\equiv \bigwedge_{(i,j) \in I \times J} (T_{i,j}\phi \wedge T_{i,j}\psi) \\ &\equiv \bigwedge_{(i,j) \in I \times J} T_{i,j}(\phi \wedge \psi) \\ &\equiv Tc_{I,J}(\phi \wedge \psi) \end{aligned}$$

ii) Supposons $T_{C_{I,J}}\phi \vee T_{C_{I,J}}\psi$. Donc :

$$\bigwedge_{(i,j) \in I \times J} T_{i,j}\phi \vee \bigwedge_{(i,j) \in I \times J} T_{i,j}\psi$$

Or pour tout $(k, l) \in I \times J$, nous avons :

$$(T_{k,l}\phi \vee T_{k,l}\psi) \Rightarrow T_{k,l}(\phi \vee \psi)$$

Par soucis de lisibilité, notons Δ la formule $(i, j) \in (I \times J) \setminus \{(k, l)\}$, Δ' la formule $(i, j) \in (I \times J)$ et Δ'' la formule $(k, l) \in (I \times J)$. Ainsi :

$$\bigwedge_{\Delta} T_{i,j}\phi \wedge T_{k,l}\phi \vee \bigwedge_{\Delta} T_{i,j}\psi \wedge T_{k,l}\psi$$

En appliquant (R₈) :

$$\begin{aligned} & \left(\bigwedge_{\Delta} T_{i,j}\phi \vee \left(\bigwedge_{\Delta} T_{i,j}\psi \wedge T_{k,l}\psi \right) \right) \wedge \left(T_{k,l}\phi \vee \bigwedge_{\Delta} T_{i,j}\psi \right) \\ & \wedge \left(T_{k,l}\phi \vee T_{k,l}\psi \right) \Rightarrow T_{k,l}(\phi \vee \psi) \end{aligned}$$

Donc :

$$\begin{aligned} & \bigwedge_{\Delta''} \left[\left(\bigwedge_{\Delta} T_{i,j}\phi \wedge T_{k,l}\phi \vee \bigwedge_{\Delta} T_{i,j}\psi \wedge T_{k,l}\psi \right) \Rightarrow T_{k,l}(\phi \vee \psi) \right] \\ & \equiv \left[\left(\bigwedge_{\Delta'} T_{i,j}\phi \vee \bigwedge_{\Delta'} T_{i,j}\psi \right) \right] \Rightarrow \bigwedge_{\Delta''} T_{k,l}(\phi \vee \psi) \end{aligned}$$

Par conséquent, par application de la synthèse :

$$(T_{C_{I,J}}\phi \vee T_{C_{I,J}}\psi) \Rightarrow T_{C_{I,J}}(\phi \vee \psi)$$

iii) Supposons $T_{C_{I,J}}\phi$. Donc :

$$\bigwedge_{(i,j) \in I \times J} T_{i,j}\phi$$

Or pour tout $(i, j) \in I \times J$, nous avons $T_{i,j}\phi \Rightarrow \neg T_{i,j}\neg\phi$. En appliquant deux fois (R₇) :

$$\begin{aligned} & \bigwedge_{(i,j) \in I \times J} (T_{i,j}\phi \Rightarrow \neg T_{i,j}\neg\phi) \\ & \bigwedge_{(i,j) \in I \times J} (T_{i,j}\phi \wedge (T_{i,j}\phi \Rightarrow \neg T_{i,j}\neg\phi)) \end{aligned}$$

Par modus ponens, nous déduisons :

$$\bigwedge_{(i,j) \in I \times J} \neg T_{i,j}\neg\phi$$

Or, par le théorème de De Morgan :

$$\bigwedge_{(i,j) \in I \times J} \neg T_{i,j}\neg\phi \equiv \neg \bigvee_{(i,j) \in I \times J} T_{i,j}\neg\phi \equiv \neg T_{C_{I,J}}^*\neg\phi$$

Par synthèse, nous prouvons :

$$T_{C_{I,J}}\phi \Rightarrow \neg T_{C_{I,K}}^*\neg\phi$$

6.4 Liens entre confiances collectives

La transitivité de la confiance au sein d'un groupe $I \cup J$ se retrouve à l'échelle du groupe lorsqu'il y a une confiance commune envers un groupe K .

Proposition 6.2. Pour tout $I, J, K \subseteq \mathcal{N}$,

$$i) \quad T_{A_{I \cup J, K}}\phi \wedge T_{C_{I, J}}T_{C_{J, K}}\phi \Rightarrow T_{C_{I, K}}\phi$$

$$ii) \quad T_{b_{I \cup J, K}}\phi \wedge T_{C_{I, J}}\neg T_{J, K}^*\phi \Rightarrow \neg T_{C_{I, K}}^*\phi$$

Démonstration. i) Supposons $T_{A_{I \cup J, K}}\phi \wedge T_{C_{I, J}}T_{C_{J, K}}\phi$. Par (R₈) :

$$\begin{aligned} & \bigwedge_{(i,j) \in I \times J} T_{i,j} \left(\bigwedge_{(j',k) \in J \times K} T_{j',k}\phi \right) \\ & \equiv \bigwedge_{(i,j) \in I \times J} T_{i,j} \left(\bigwedge_{(j',k) \in \{j\} \times K} T_{j',k}\phi \bigwedge_{k \in K} T_{j,k}\phi \right) \end{aligned}$$

Nous en déduisons⁴ :

$$\begin{aligned} & \bigwedge_{(i,j) \in I \times J} T_{i,j} \left(\bigwedge_{(j',k) \in \{j\} \times K} T_{j',k}\phi \bigwedge_{k \in K} T_{j,k}\phi \right) \\ & \Rightarrow \left(\bigwedge_{(i,j) \in I \times J} T_{i,j} \left(\bigwedge_{k \in K} T_{j,k}\phi \right) \right) \end{aligned}$$

Par modus ponens et par la proposition 5.1.i :

$$\left(\bigwedge_{(i,j) \in I \times J} T_{i,j} \left(\bigwedge_{k \in K} T_{j,k}\phi \right) \right) \equiv \left(\bigwedge_{(i,j) \in I \times J} \bigwedge_{k \in K} T_{i,j}(T_{j,k}\phi) \right)$$

Or, nous avons aussi par (R₈) les propriétés de transitivité pour tous les agents $(i, j) \in I \times J$ selon l'hypothèse $T_{A_{I \cup J, K}}\phi$. Par conséquent, pour tout $(i, j) \in I \times J$ et $k \in K$: $T_{i,j}T_{j,k}\phi \Rightarrow T_{i,k}\phi$. Par (R₇), nous avons :

$$\left(\bigwedge_{(i,j) \in I \times J} \bigwedge_{k \in K} (T_{i,j}T_{j,k}\phi \Rightarrow T_{i,k}\phi) \right)$$

Puis par modus ponens, nous déduisons :

$$\left(\bigwedge_{(i,j) \in I \times J} \bigwedge_{k \in K} T_{i,k}\phi \right) \equiv T_{C_{I, K}}\phi$$

Par synthèse, nous prouvons alors :

$$T_{A_{I \cup J, K}}\phi \wedge T_{C_{I, J}}T_{C_{J, K}}\phi \Rightarrow T_{C_{I, K}}\phi$$

ii) Supposons $T_{b_{I \cup J, K}}\phi \wedge T_{C_{I, J}}\neg T_{J, K}^*\phi$. Par (R₈) :

$$T_{C_{I, J}}\neg T_{J, K}^*\phi \equiv \bigwedge_{(i,j) \in I \times J} \neg \bigvee_{(j',k) \in J \times K} T_{j',k}^*\phi$$

4. Dans le détail, nous appliquons la nécessité pour tous les $T_{i,j}$ puis K , R₇ puis R₈. Pour des soucis de lisibilité, nous nous permettons ce raccourci trivial. □

Donc,

$$\begin{aligned}
& \bigwedge_{(i,j) \in I \times J} \bigwedge_{(j',k) \in J \times K} \neg T_{j',k} \phi \\
& \equiv \bigwedge_{(i,j) \in I \times J} \left(\bigwedge_{(j',k) \in J \setminus \{j\} \times K} \neg T_{j',k} \phi \wedge \neg T_{j,k} \phi \right) \\
& \equiv \bigwedge_{(i,j) \in I \times J} \left(\bigwedge_{(j',k) \in J \setminus \{j\} \times K} \neg T_{j',k} \phi \right) \wedge \bigwedge_{k \in K} \neg T_{j,k} \phi
\end{aligned}$$

Par (R₈) et par la propriété 5.1.i, nous déduisons :

$$\bigwedge_{(i,j) \in I \times J} \bigwedge_{k \in K} \neg T_{j,k} \phi$$

Or, notre hypothèse, nous permet aussi de déduire $Tb_{I \cup J, K} \phi$ par (R₈). Ainsi, pour tout $(i, j) \in I \times I \wedge k \in K, T_{i,j}$:

$$\neg T_{j,k} \phi \Rightarrow \neg T_{i,k} \phi$$

Donc par (R₇) :

$$\bigwedge_{(i,j) \in I \times J} \bigwedge_{k \in K} (T_{i,j} \neg T_{j,k} \phi \wedge (T_{i,j} \neg T_{j,k} \phi \Rightarrow \neg T_{i,k} \phi))$$

Puis par modus ponens, nous déduisons :

$$\bigwedge_{(i,j) \in I \times J} \bigwedge_{k \in K} \neg T_{i,k} \phi$$

Par simplification et par De Morgan, nous avons :

$$\bigwedge_{(i,k) \in I \times K} \neg T_{i,k} \phi \equiv \neg \bigvee_{(i,k) \in I \times K} T_{i,k} \phi$$

Enfin, par synthèse, nous prouvons :

$$Tb_{I \cup J, K} \phi \wedge Tc_{I, J} \neg T_{J, K}^* \phi \Rightarrow \neg Tc_{I, K}^* \phi$$

□

6.5 Analogies avec Singh

Comme pour la confiance individuelle, nous pouvons faire des liens entre notre modèle et des propriétés axiomatisées par Singh. Appliquons par exemple la notion de confiance commune à la la confiance dispositionnelle. Considérons le prédicat :

$$\forall I, J \subseteq N : Td_{I, J}(\phi, \psi) \triangleq \bigwedge_{(i,j) \in I \times J} Ts_{i,j}(\phi, \psi)$$

Nous retrouvons alors l'axiome T_{15} de Singh ($T_{i,j}(\phi, \psi) \Rightarrow T_{i, \{j, k\}}(\phi, \psi)$) qui stipule que si un agent i fait confiance à j pour ψ dans un contexte ϕ , alors i fait confiance au groupe $\{j, k\}$ pour la même chose. Pour déduire une telle proposition dans notre modèle, nous avons simplement besoin de prouver qu'il y a une confiance commune sur $Ts_{i,j}(\phi, \psi)$ et $Ts_{i,k}(\phi, \psi)$. De la même

manière, nous caractérisons avec le prédicat $Td_{I, J}(\phi, \psi)$ l'axiome T_{16} ($T_{i,j}(\phi, \psi) \Rightarrow T_{\{i, k\}, j}(\phi, \psi)$) stipulant que si un agent i fait confiance à j pour ψ dans un contexte ϕ alors le groupe d'agents $\{i, k\}$ fait confiance au groupe j pour la même chose.

En revanche, nous ne pouvons pas déduire l'axiome T_{14} ($T_{i,j}(\phi, \psi) \wedge T_{j,i}(\psi, \phi) \Rightarrow T_{i, \{i, j\}}(\top, \phi \wedge \psi)$) qui dit que si un agent i fait confiance à j pour ψ dans un contexte ϕ et si un agent j fait confiance à i pour ϕ dans le contexte ψ alors i fait confiance au groupe $\{i, j\}$ pour $\phi \wedge \psi$, et ce quelque soit le contexte. En effet, si nous essayons de démontrer une caractérisation de cet axiome (par exemple avec une contraposition), nous obtenons $\neg Td_{\{i, \{i, j\}}(\top, \phi \wedge \psi) \equiv \neg(Ts_{i,j}(\top, \phi \wedge \psi) \wedge Ts_{j,i}(\top, \phi \wedge \psi))$. Le constat est le même pour les axiomes T_{17} ($T_{i,j}(\phi, \psi) \wedge T_{i,k}(\psi, \psi') \Rightarrow T_{i, \{j, k\}}(\phi, \psi \wedge \psi')$) et T_{18} ($T_{i,j}(\phi, \psi), T_{j,k}(\phi', \psi'), \psi' \vdash \psi, \phi \vdash \phi' \Rightarrow T_{i, \{j, k\}}(\phi, \psi')$). Pour caractériser ces axiomes, il nous faudrait considérer des propriétés supplémentaires sur notre notion de confiance. Par exemple pour T_{14} , il s'agirait de considérer $Ts_{i,j}(\phi, \psi) \wedge Ts_{j,i}(\psi, \phi) \Rightarrow Td_{\{i, \{i, j\}}(\top, \phi \wedge \psi)$.

7 Conclusion et ouvertures

Nous avons présenté dans cet article une logique modale pour exprimer la confiance qu'un agent a envers le discours d'un autre agent. Il s'agit essentiellement d'un système KD associé à un formalisme STIT. Ceci nous permet de retrouver les propriétés de la confiance dispositionnelle formalisées par Herzig *et al.*, les propriétés doxastiques de la confiance formalisées par Dundua et Uridia ainsi que de caractériser les principaux axiomes de la confiance postulés par Singh. Ce formalisme présente bien entendu des limites. Tout d'abord, nous n'avons pas présenté la sémantique et les propriétés de cohérence et complétude de notre formalisme. Toutefois, en considérant les travaux de Van Benthem [20] et ceux de Dundua et Uridia [7], notre système est intuitivement complet et cohérent si nous utilisons une sémantique de Kripke. Enfin, nous n'avons pas évoqué les aspects de complexité d'un tel modèle.

En termes de perspectives, il serait intéressant d'introduire des prédicats supplémentaires pour caractériser certains axiomes de Singh concernant la confiance collective. Nous pourrions aussi définir un prédicat de *confiance mutuelle* représenté par $T_{i,j} \phi \wedge T_{j,i} \phi \wedge T_{i,j} T_{j,i} \phi \wedge T_{j,i} T_{i,j} \phi$ ou de *confiance réciproque* par $T_{i,j} \phi \wedge T_{j,i} \phi$. De plus, nous pensons qu'il serait intéressant d'associer notre formalisme avec un système de normes sociales. En effet, la norme sociale pourrait exprimer l'axiomatique choisie par un agent pour faire confiance. Ceci apparaît en filigrane lorsque nous définissons par exemple la confiance par transitivité. Notre objectif est alors de pouvoir raisonner sur ce que certains agents malintentionnés pourraient faire s'ils considéraient ces normes.

Références

- [1] Nuel Belnap and Michael Perloff. Seeing to it that : a canonical form for agentives. *Theoria*, 54(3) :175–199, 1988.
- [2] Christiano Castelfranchi and Rino Falcone. *Trust theory : A socio-cognitive and computational model*. John Wiley & Sons, 2010.
- [3] Brian F Chellas. Time and modality in the logic of agency. *Studia Logica*, 51(3-4) :485–517, 1992.
- [4] Bruce Christianson and William Harbison. Why isn't trust transitive? In *Security protocols*, pages 171–176, 1997.
- [5] Mehdi Dastani, Andreas Herzig, Joris Hulstijn, and Leendert Van Der Torre. Inferring trust. In *5th CLIMA*, pages 144–160. Springer, 2004.
- [6] Robert Demolombe. Reasoning about trust : A formal logical framework. In *2nd iTrust*, pages 291–303, 2004.
- [7] Besik Dundua and Levan Uridia. Trust and belief, interrelation. In *3rd WAT*, 2010.
- [8] Rino Falcone, Giovanni Pezzulo, and Cristiano Castelfranchi. A fuzzy approach to a belief-based trust computation. In *5th Workshop on Deception, Fraud and Trust in Agent Societies*, pages 73–86, 2002.
- [9] Tuan-Fang Fan and Churn-Jung Liau. Reasoning about justified belief based on the fusion of evidence. In *15th JELIA*, pages 240–255. Springer, 2016.
- [10] Andreas Herzig, Emiliano Lorini, Jomi Fred Hübner, and Laurent Vercouter. A logic of trust and reputation. *Logic Journal of the IGPL*, 18(1) :214–244, 2010.
- [11] Audun Josang and Roslan Ismail. The beta reputation system. In *15th Bled Electronic Commerce Conference*, pages 2502–2511, 2002.
- [12] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *12th WWW*, pages 640–651. ACM, 2003.
- [13] Vibhor Kant and Kamal K Bharadwaj. Fuzzy computational models of trust and distrust for enhanced recommendations. *Int. J. of Intelligent Systems*, 28(4) :332–365, 2013.
- [14] Churn-Jung Liau. Belief, information acquisition, and trust in multi-agent systems - a modal logic formulation. *Artificial Intelligence*, 149(1) :31–60, 2003.
- [15] Emiliano Lorini, Guifei Jiang, and Laurent Perrussel. Trust-based belief change. In *21st ECAI*, pages 549–554. IOS Press, 2014.
- [16] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking : bringing order to the web. 1999.
- [17] Yefeng Ruan and Arjan Durrresi. A survey of trust management systems for online social communities – trust modeling, trust inference and attacks. *Knowledge-Based Systems*, 106 :150–163, 2016.
- [18] Munindar P. Singh. Trust as dependence : A logical approach. In *10th AAMAS*, pages 863–870, 2011.
- [19] Clara Smith, Agustín Ambrossio, Leandro Mendoza, and Antonino Rotolo. Combinations of normal and non-normal modal logics for modeling collective trust in normative mas. In *4th AICOL*, pages 189–203, 2011.
- [20] Johan Van Benthem. Correspondence theory. In *Handbook of philosophical logic*, pages 167–247. Springer, 1984.
- [21] Jin-Long Wang and Shih-Ping Huang. Fuzzy logic based reputation system for mobile ad hoc networks. In *11th KES*, pages 1315–1322, 2007.

Vers une représentation des systèmes dynamiques booléens en logique des hypothèses *

Pierre Siegel¹, Andrei Doncescu², Vincent Risch³, Sylvain Sené¹

¹ Aix-Marseille Univ., CNRS, Centrale Marseille, LIF, Marseille, France

² CNRS, LAAS, Toulouse, France

³ Aix-Marseille Univ., Université de Toulon, CNRS, ENSAM, LSIS, Marseille, France

{pierre.siegel,vincent.risch,sylvain.sene}@univ-amu.fr andrei.doncescu@laas.fr

Résumé

Un Système Dynamique Booléen (SDB) représente l'évolution au cours du temps des interactions dans un réseau fini d'entités. La représentation des réseaux de régulation de gènes, c'est à dire des interactions entre les gènes/protéines d'une cellule, en est un exemple. Dans ce domaine des théorèmes fondateurs ont porté sur les cycles d'interactions et l'étude des cycles, circuits et point fixes est fondamentale. Des travaux ont été fait sur la représentation de certains réseaux biologiques en utilisant des formalismes non-monotones, en particulier la logique de défauts ou les ASP. Mais une représentation des SDB par la plupart des formalismes non-monotones n'est pas satisfaisante, car elle ne permet pas de représenter les cycles stables et instables. Ici on représente les cycles en utilisant la logique des hypothèses qui généralise la logique des défauts. Pour cette logique on a toujours des extensions mais certaines d'entre elles, les extensions fantômes, vont jouer un rôle particulier pour la représentation et la gestion des cycles. L'article donne une représentation des SDB en logique des hypothèses. Le but est de permettre de discriminer les états stables, les cycles stables et instables.

Abstract

Boolean dynamical systems (BDSs) represent the evolution of interactions of a finite network of entities taking Boolean states over discrete time. These networks are classically used to model biological networks. Previous seminal results have emphasized some relations between the presence of circuits in such networks and their dynamical behaviors, like the convergence towards stable configurations for instance. Moreover, studies have focused on the modeling of "real" genetic with non-monotonic formalisms, such as default logic and ASP. So far however, BDS representation by most these formalisms is not adequate since it does

not allow to capture cyclic dynamical behaviors. In this paper, our purpose is to show that hypothesis logic is more appropriate to remedy some of these problems. Indeed, this logic always admits extensions among which stable extensions represent BDS stable configurations and ghost extensions play a key role in dealing with BDS cyclic behaviors.

1 Introduction

Du point de vue logique et représentation des connaissances, un système biologique peut être considéré comme un ensemble d'éléments qui interagissent entre eux. Par exemple une cellule est un ensemble de protéines/gènes qui interagissent pour la faire survivre se reproduire et mourir. La cellule pose des problèmes intéressants à l'Intelligence Artificielle. Il faut d'abord formaliser les interactions mais, une formalisation en logique classique donne rapidement des incohérences. Ensuite, ce que l'on sait vient en grande partie d'expériences. On ne connaît donc qu'une petite partie des interactions et cette connaissance peut être révisable, incertaine, contradictoire et même fausse. Enfin la complexité algorithmique est importante et on a besoin d'algorithmes qui donnent des temps de calcul raisonnables. Ces questions sont étudiées en IA depuis la fin des années 1970 en utilisant en particulier des logiques non-monotones. Pour certains réseaux génétiques, sans cycles négatifs, des résultats ont été obtenus en utilisant la logique des défauts [9, 10].

D'un autre côté, les systèmes biologiques peuvent être étudiés dans le contexte des réseaux d'automates et des Systèmes dynamiques discrets (SDD). Des théorèmes fondateurs [8, 19, 20, 25, 26, 27, 28, 29, 35] ont porté sur les cycles d'interactions. Une représentation des SDB par la logique des défauts, par les ASP et par la plupart des formalismes non-monotones, n'est pas adaptée à la représen-

*This work has been partially supported by Turing Centre for living systems (ANR-PIA) and by the project PACA APEX FRI.

tation des cycles. En particulier, pour la logique des défauts l'équivalent d'un circuit négatif n'a pas d'extension. Cette absence d'extension a été étudiée, ce qui a donné la logique des hypothèses [30, 32]. Pour cette logique on a toujours des extensions mais certaines d'entre elles, les extensions fantômes ne sont pas des extensions classiques de la logique des défauts. L'article étudie une représentation des SDB en logique des hypothèses. Le but est de permettre de discriminer les états stables, les cycles stables et les cycles instables en utilisant les extensions fantômes. Les chapitres 2, 3 et 4 présentent les SDB, la logique des défauts et la logique des hypothèses. Le chapitre 5 donne une représentation des graphes d'interactions en logique des hypothèses, des propriétés relatives aux extensions et quelques considérations algorithmiques. Le chapitre 6 utilise cette représentation pour étudier les SDB et en caractériser, via la sémantique de Kripke, les états stables et les cycles stables.

2 Systèmes dynamique discrets

Un système dynamique discret (SDD) décrit l'évolution, au cours d'un temps discret, des interactions dans un réseau fini de n entités notées de 1 à n . Cette évolution est la *dynamique* du système. Un exemple d'un tel système est la représentation des réseaux génétiques, que l'on nommera réseaux dans la suite, qui représentent les interactions entre les gènes ou les protéines d'une cellule [1, 6, 7, 11, 13, 22]. Dans ce contexte, une entité $i \in \{1, \dots, n\}$ représente une protéine et x_i est la concentration de i ; il y a un nombre fini de concentrations possibles. Pour ces réseaux, une interaction donne les conditions pour qu'un ensemble de protéines fasse augmenter ou diminuer la concentration d'une protéine. Dans le contexte des réseaux Booléens, étudiés dans cet article, les concentrations sont dans $\{0, 1\}$. Une protéine i est alors représentée par une variable propositionnelle x_i . Alors, x_i (ou $x_i = 1$, ou $x_i = \text{vrai}$) dénote la présence de i et $\neg x_i$ (ou $x_i = 0$, ou $x_i = \text{faux}$) son absence. On peut aussi considérer qu'est associée à la valeur de x_i une *activation* (resp. *inhibition*), ou encore une *production* (resp. *destruction*), de la protéine.

2.1 Systèmes dynamiques Booléens (SDB)

Soit $V = \{1, \dots, n\}$ un ensemble de n entités. Une *configuration* $x = (x_1, \dots, x_n)$ du système est l'affectation d'une valeur de vérité $x_i \in \{0, 1\}$ à chaque élément i de V . L'ensemble des configurations, également appelé *espace des configurations*, se note $X = \{0, 1\}^n$. La *dynamique* d'un tel système est modélisée par une fonction f , appelée *fonction globale de transition*, et par un *mode de mise à jour* qui dit comment les éléments de V sont mis à jour au cours d'un temps discret. Plus formellement, $f : X \rightarrow X$ est telle que $x = (x_1, \dots, x_n) \mapsto f(x) = (f_1(x), \dots, f_n(x))$, où chaque fonction $f_i : X \rightarrow \{0, 1\}$ est une *fonction locale de transition* qui donne l'évolution de l'état x_i dans le temps.

Il existe une infinité de modes de mise à jour parmi lesquels les modes *parallèle* et *asynchrone* sont les plus utilisés. Le mode de synchronisation *parallèle* (ou *parfaitement synchrone*) est tel que toutes les entités du réseau sont mises à jour à chaque étape temporelle. Pratiquement, une telle mise à jour n'est pas vraiment réaliste d'un point de vue biologique car les changements d'état de l'entité se produisent à des vitesses variables et sont rarement simultanés. À l'opposé, le mode de mise à jour *asynchrone* est une variation non déterministe pour laquelle une seule entité est mise à jour à la fois. Dans la suite, nous limitons notre étude à la dynamique asynchrone [19, 26, 28].

2.2 Graphes de Transition Asynchrones (GTA)

La dynamique étant caractérisée par une fonction f et un mode de mise à jour, une partie importante de l'étude des SDB porte sur l'analyse des configurations stables (c.a.d. des points fixes de f) et des cycles stables et instables de f . Pour le cas asynchrone, cette étude utilise la notion de *graphe de transition asynchrone* (GTA) associé à f .

Dans la suite pour alléger le texte on simplifiera parfois les notations; par exemple $x = (\neg x_1, x_2, \neg x_3)$ pourra se représenter par $(-1, 2, -3)$ ou même par $\bar{1}, 2, \bar{3}$.

Définition 1 Soient $X = \{0, 1\}^n$ l'espace des configurations et la fonction $f : X \rightarrow X$. La dynamique asynchrone de f est donnée par son GTA, noté $\mathcal{G}(f) = (X, T(f))$, le digraphe dont l'ensemble des sommets est l'espace des configurations et dont l'ensemble des arcs est l'ensemble des transitions possibles.

$$T(f) = \{(x, y) \in X^2 \mid x \neq y, x = (x_1, \dots, x_i, \dots, x_n) \text{ et } y = (x_1, \dots, x_{i-1}, f_i(x), x_{i+1}, \dots, x_n)\}.$$

On remarque que si $(x, y) \in T(f)$ alors x et y diffèrent par un unique élément (leur distance de Hamming est 1). On définit une *orbite* de $\mathcal{G}(f)$ comme une suite de configurations (x^0, x^1, x^2, \dots) telle que, soit $(x^t, x^{t+1}) \in T(f)$ soit $x^{t+1} = x^t$ si $x^t = f(x^{t+1})$ (c.a.d., x^t n'a pas de successeur). De plus, on définit un *cycle* de longueur r comme une suite de configurations (x^1, \dots, x^r, x^1) , avec $r \geq 2$, telle que x^1, \dots, x^r sont tous différents.

Définition 2 Soient $X = \{0, 1\}^n$, une fonction $f : X \rightarrow X$ et $\mathcal{G}(f)$ le GTA de f . Un attracteur asynchrone de f est un composant fortement connexe (CFC) terminal de $\mathcal{G}(f)$ (c.a.d. un CFC sans arc sortant).

Dans cet article, nous étudieront plus particulièrement deux attracteurs : les configurations stables (ou les points fixes) et les cycles stables (ou cycles limites) :

- Une *configuration stable* est un CFC trivial, c'est à dire une configuration x telle que $\forall i \in V, x_i = f_i(x)$; ceci implique que $x = f(x)$.
- Un *cycle stable* est un attracteur cyclique de $\mathcal{G}(f)$, tel que $\forall t < r, x^{t+1}$ est l'unique successeur de x^t et tel que x^1 est l'unique successeur de x^r .

Lorsqu'il est possible de sortir d'un cycle (ou d'un CFC non trivial), ce cycle (resp. ce CFC) est considéré comme un *cycle instable* (resp. une oscillation instable).

Exemple 1 Soit $V = \{1, 2, 3\}$, $X = \{0, 1\}^3$ et deux fonctions f et g telles que $f(x_1, x_2, x_3) = (\neg x_2, \neg x_3, x_1)$ et $g(x_1, x_2, x_3) = (\neg x_3, x_1, x_2)$. Les GTA associés $\mathcal{G}(f)$ et $\mathcal{G}(g)$ sont représentés par la Figure 1. Pour chaque arc (x, y) de $\mathcal{G}(f)$ et de $\mathcal{G}(g)$, si $x \neq y$ alors x diffère de y par un seul élément. Il y a donc jusqu'à 3 transitions qui sortent de chaque configuration. Ici, $\mathcal{G}(f)$ a deux configurations stables, $(x_1, \neg x_2, x_3)$ et $(\neg x_1, x_2, \neg x_3)$ représentées en gras dans la figure 1(a) et toutes les autres configurations forment un cycle instable représenté par les flèches en gras dans la figure 1(a). D'autre part $\mathcal{G}(g)$ a un cycle stable représenté par les flèches en gras dans la figure 1(b).

Une orbite qui atteint une configuration stable y reste indéfiniment. De même, si l'orbite atteint un cycle stable, elle adopte indéfiniment un comportement oscillant stable. On note que la principale différence entre un cycle stable et un cycle instable est que le premier représente une orbite unique tandis que le second peut quitter à tout moment l'orbite et représente donc une infinité d'orbites.

En particulier, pour l'exemple 1 le cycle de la figure 1(a) est instable car une orbite qui le parcourt peut en sortir à tout moment pour rejoindre un des états stables. Le cycle de 1(b) est stable car il est impossible d'en sortir.

2.3 Graphes d'interaction et circuits

Un graphe de transition est un excellent outil d'étude du comportement d'une fonction. Cependant, dans la pratique, les données biologiques proviennent d'expériences

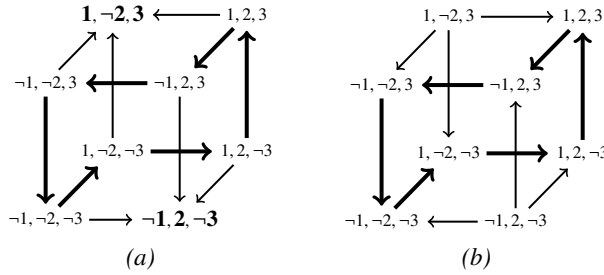


FIGURE 1 – (a) GTA $\mathcal{G}(f)$ du SDB de fonction f ; (b) GTA $\mathcal{G}(g)$ du SDB de fonction g présentés en exemple 1.

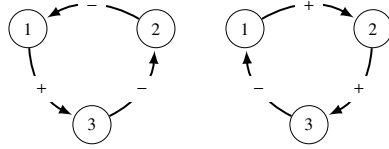


FIGURE 2 – Graphes d'interactions associés aux GTA $\mathcal{G}(f)$ et $\mathcal{G}(g)$ des SDB f et g de l'exemple 1.

qui donnent généralement des corrélations entre l'expression des gènes. Cette information est classiquement modélisée par les digraphes d'interaction. Ces digraphes sont exponentiellement plus compacts et donc plus "lisibles" que les graphes de transition. Mais, contrairement à ces derniers les digraphes d'interactions ne donnent que des informations statiques sur le fait qu'une entité agit sur cette autre entité. Notons que le graphe d'interaction d'un SDB de fonction f est induit par ses fonctions de transition locales f_i . Sur les SDB, un thème de recherche important, traite de ce que l'on peut dire d'un graphe de transition en ne connaissant que sa spécification statique, c'est à dire sa fonction et par induction son graphe d'interaction.

Définition Un *graphe d'interaction* est un digraphe signé $G = (V, I)$, où $V = \{1, \dots, n\}$ est l'ensemble des sommets et $I \subseteq V \times S \times V$, avec $S = \{-, +\}$. Un arc (i, s, j) (resp. $(i, -, j)$) $\in I$ est dit *positif* (resp. *négatif*). Un *circuit* C de taille k est une suite $C = \{(i_1, s_{(1,2)}, i_2), \dots, (i_k, s_{(k,1)}, i_1)\}$. Un circuit est *élémentaire* si tous ses composants sont distincts. Un circuit est *positif* (resp. *négatif*) s'il contient un nombre pair (resp. impair) d'arcs négatifs.

La présence d'un arc (i, s, j) dans un graphe d'interaction G signifie que la valeur du sommet i affecte la valeur du sommet j ; on dit que i régule j . Considérons l'exemple simple où j a un seul arc entrant qui vient de i . Dans ce cas la régulation est très simple : si l'arc est positif (resp. négatif), à la mise à jour l'état de j prendra la valeur (ou la valeur opposée) de i (c.a.d. $f_j(x) = x_i$ (resp. $f_j(x) = \neg x_i$). Les circuits élémentaires sont régulés ainsi.

Comme mentionné ci-dessus, un graphe d'interaction $G = (V, I)$ représente l'existence d'interactions entre les entités de V . La spécification de la nature de ces interactions et des conditions dans lesquelles elles se produisent conduit de manière naturelle à associer G à un SDB de fonction f , tel que G est le graphe d'interaction de f , qui est alors désigné par $G(f) = (V, I(f))$.

Cela se fait en associant à chaque $i \in V$ une fonction de transition locale f_i telle que :

$$\forall j \in V, \exists x \in \{0, 1\}^n \mid f_i(x) \neq f_i(\bar{x}^j) \Leftrightarrow (j, s, i) \in I(f),$$

où, si $x = (x_1, \dots, x_n)$, $\bar{x}^j = (x_1, \dots, x_{j-1}, \neg x_j, x_{j+1}, \dots, x_n)$.

Une telle spécification induit la minimalité de $G(f)$ car chaque arc représente une interaction effective.

Remarque Soient un SDB et son graphe d'interaction associé $G(f)$ tel que $I(f)$ contienne un arc (i, s, i) . Si $s = +$ (resp. $s = -$), cet arc tend à maintenir (resp. à inverser) l'état de i . Cela dépend bien entendu du fait que i admette d'autres voisins que lui-même et de l'influence positive ou négative de ces voisins. Dans le cas où i n'admet aucun autre voisin, si $s = +$ (resp. $s = -$) il est évident que i maintient (resp. inverse) indéfiniment son état.

Exemple 1 (suite) Soient les deux SDB de l'exemple 1 dont les fonctions sont $f(x) = (\neg x_2, \neg x_3, x_1)$ et $g(x) = (\neg x_3, x_1, x_2)$ et dont les GTA respectifs $\mathcal{G}(f)$ et $\mathcal{G}(g)$ sont

en Figure 1. A partir des 2 fonctions, il est facile de déduire leurs graphes d'interaction représentés en figure 2.

2.4 Résultats généraux fondamentaux

En considérant que les SDB sont un bon candidat pour la modélisation qualitative des réseaux génétiques, comme la littérature l'a établi depuis les travaux fondateurs de Kauffman [12] et Thomas [34], la présence de plusieurs attracteurs permet de modéliser, en utilisant les comportements dynamiques, la spécialisation cellulaire. En effet, si un réseau contrôle un phénomène de spécialisation, la cellule se spécialisera (c.a.d. va acquérir un phénotype particulier ou une fonction physiologique spécifique) selon l'attracteur vers lequel évolue son SDB sous-jacent. Parmi les exemples les plus classiques figurent :

- Le contrôle de l'immunité dans le bacteriophage λ [33] qui met l'accent sur le processus de décision entre la lyse et la lysogénéisation de λ . Bien que ce modèle soit basé sur un système dynamique à plusieurs valeurs, il peut être facilement modélisé en SDB sans perte d'informations qualitatives.
- La morphogenèse florale de la plante *Arabidopsis thaliana*, modélisé en SDB dans [21, 22], dont les comportements dynamiques admettent notamment quatre configurations stables qui correspondent aux schémas d'expression génétique des tissus floraux : sépales, pétales, étamines et carpelles. Notons que ce modèle a également permis d'expliquer formellement le rôle de l'hormone gibberelline sur le développement floral [7].

Ces travaux et de nombreux autres qui utilisent les SDB, ou plus généralement les SDD, ont mis en évidence le rôle essentiel de l'étude visant à comprendre les relations formelles entre les graphes d'interaction et de transition et de leurs propriétés respectives. Ils ont également clairement souligné le rôle essentiel des circuits, aujourd'hui appelés moteurs de complexité comportementale dans les systèmes dynamiques. Cela vient en particulier de Robert qui a établi le théorème qui dit que si le graphe d'interaction $G(f)$ d'un SDD f est acyclique, alors f converge vers une configuration stable unique [29]. En outre, dans [35] Thomas, en se basant sur des SDD asynchrones, a conjecturé que le $G(f)$ d'un SDD doit contenir un circuit positif (resp. négatif) pour que $G(f)$ admette plusieurs configurations stables (resp. un attracteur non trivial tel qu'un cycle stable ou plus complexe). Ces deux conjectures ont été prouvées comme étant vraies sous l'hypothèse d'un mode de mise à jour asynchrone [25, 26, 27, 28]. Néanmoins, bien que le premier résultat reliant les circuits positifs à des configurations stables soit vrai pour tout mode de mise à jour [23], le second ne l'est pas (un circuit positif en parallèle admet des cycles stables). De plus, dans [25], les auteurs ont montré qu'un circuit positif asynchrone de taille n admet deux attracteurs, à savoir deux configurations stables x et \bar{x} , et

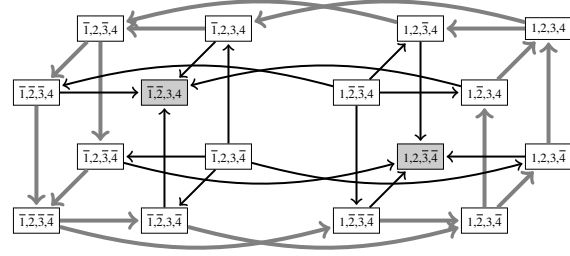


FIGURE 3 – GTA $\mathcal{G}(f)$ du SDB défini dans l'exemple 2.

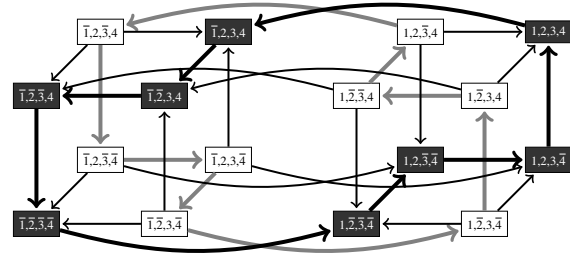


FIGURE 4 – GTA $\mathcal{G}(g)$ du SDB défini dans l'exemple 3.

qu'un circuit négatif asynchrone n'admet qu'un seul attracteur, à savoir un cycle de longueur stable de longueur $2n$.

Les exemples 2 et 3 donnent les comportements asynchrones de circuits booléen positif et négatif de taille 4.

Exemple 2 (Circuit booléen positif de taille 4) Soit le SDB de fonction $f(x_1, x_2, x_3, x_4) = (\neg x_4, x_1, \neg x_2, x_3)$. La figure 3 donne son graphe de transition. Ce SDB a deux configurations stables, $(x_1, x_2, \neg x_3, \neg x_4)$ et $(\neg x_1, \neg x_2, x_3, x_4)$ et une oscillation instable dessinée en gris.

Exemple 3 (Circuit booléen négatif de taille 4) Soit un SDB de fonction $g(x_1, x_2, x_3, x_4) = (\neg x_4, x_1, x_2, x_3)$. La figure 4 donne son graphe d'interaction. Ce SDB a un cycle stable 8 dessiné en gras et un cycle instable de longueur 8 dessiné en gris.

La suite de l'article est une première étape vers une représentation des SDB en logique des hypothèses.

3 Non-monotonie et logique des défauts

L'idée de représenter les graphes d'interaction par un formalisme logique semble naturelle car, la manière dont un arc (i, s, j) est interprété suggère une relation étroite avec ce qu'on appelle l'implication matérielle en logique. Cependant, une représentation en logique classique n'est pas adaptée car elle entraîne des incohérences dans la plupart des cas. Une façon de gérer ces incohérences est fournie par des formalismes non monotones, tels que la logique

par défaut ou les ASP. En effet, il existe clairement un lien entre la notion d'extension stable (ou de modèle stable) et les configurations stables d'un SDB. Mais un inconvénient de ces formalismes est qu'ils ne prennent pas en compte la représentation des cycles stables et instables; le problème fondamental est de capturer, via la logique, la notion de dynamique. Pour ce faire, une généralisation de la logique des défaut, la *logique des hypothèses*, semble appropriée. La suite de ce chapitre donne les bases de la non-monotonie et de la logique des défauts. Le chapitre suivant présente la logique des hypothèses.

3.1 Logiques non-monotones

Les logiques dites «classiques», en particulier le calcul propositionnel et la logique de premier ordre, ont la propriété de monotonie. Cette propriété dit que, si F est un ensemble de formules on garde tous les théorèmes de F (les déductions) en ajoutant à F un nouvel ensemble de formules F' . Cela s'écrit $F \vdash g \Rightarrow (F \cup F') \vdash g$. Dans un cadre de représentation de connaissances la monotonie signifie que si la connaissance augmente (resp. diminue) alors les déductions augmentent (resp. diminuent). La monotonie est une propriété cruciale pour les mathématiques dont la base est l'accumulation de théorèmes qui vont servir à démontrer d'autres théorèmes. À l'inverse, à partir des années 1970, la monotonie semble être discutable en ce qui concerne la formalisation des problèmes pratiques du raisonnement du bon sens du point de vue de l'intelligence artificielle (voir quelques conférences de John McCarthy parmi d'autres). En effet, dans la vie réelle, l'information est incomplète, incertaine, révisable, contradictoire, fautive, venant de sources diverse (sans mentionner les "vérités alternatives" à la mode).

En particulier, deux problèmes théoriques relatifs à ces questions sont apparus : (i) la réponse aux requêtes d'une base de données relationnelle qui a en particulier abouti à la circonscription [17, 18] et aux logiques préférentielles [2, 3, 5, 31]; (ii) le traitement de la négation par échec en Prolog [4] qui a donné la logique des défaut [24], logique qui plus tard a été dérivée en ASP [16].

3.2 Logique des défauts

La logique des défauts de Reiter [24] a été introduite pour formaliser le raisonnement avec exceptions. Par exemple si l'on sait que *Généralement les étudiants sont jeunes* et que *Léa est étudiante* on a tendance à conclure que *Léa est jeune*. Si par la suite on apprend que *Léa est à la retraite* la conclusion risque d'être révisée. Léa est donc une exception à la règle intuitive générale. Le raisonnement sous-jacent est non-monotone car l'ajout d'une information a invalidé la conclusion.

Formellement une *théorie des défauts* est un couple $\Delta = (D, W)$, où W est un ensemble de formules fermées du premier ordre et D un ensemble de défauts qui sont des règles

d'inférences spécifiques. Un *défaut* est une expression de la forme $d = \frac{A(X):B(X)}{C(X)}$ où, $A(X)$, $B(X)$ and $C(X)$ sont des formules du premier ordre, éventuellement ouvertes et, X est l'ensemble de toutes les variables libres qui apparaissent dans d . La formule $A(X)$ est le *prérequis* de d , $B(X)$ est sa *justification* et $C(X)$ est son *consequent*. Si X n'est pas vide d est un *défaut ouvert* et si X est vide c'est un *défaut fermé* qui peut s'écrire $d = \frac{A:B}{C}$. Tout défaut ouvert est identifié à l'ensemble de toutes ses instances terminales sur X donc à un ensemble de défauts fermés. Dans la suite, pour simplifier, on considère que tous les défauts sont fermés, ce qui ne change rien à la généralité du formalisme. Si $B = C$, le défaut est *normal*. Si le prérequis est toujours vrai, le défaut est *sans prérequis* et se note $\frac{B}{C}$.

Un défaut $d = \frac{A:B}{C}$ est donc une règle d'inférence spécifique qui, parmi d'autres interprétations, peut signifier intuitivement "Si A est prouvé, si B est cohérent/consistant avec ce qui est prouvé, alors C est prouvé". Cette intuition se formalise par le concept d'extension. Dans la suite si F est un ensemble de formules on note $\text{Th}(F)$ la *cloture déductive* de F (l'ensemble de tous les théorèmes de F).

Définition 3 Une extension E d'une théorie des défauts $\Delta = (D, W)$ est un ensemble de formules du premier ordre, défini récursivement par :

1. $E = \text{Th}(\bigcup E_i)$
2. $E_0 = \text{Th}(W)$;
3. $E_{i+1} = \text{Th}(E_i \cup \{C \mid \frac{A:B}{C} \in D, A \in E_i, \neg B \notin E\})$.

Une extension est donc un point fixe et une théorie des défauts peut avoir plusieurs extensions. La limitation de la logique des défauts est que, même si W est consistant, Δ peut ne pas avoir d'extension. Ceci vient de la condition $\neg B \notin E$ à la fin du 3. de la définition 3. En effet, pour «construire» une extension il faut ajouter C à E avant la fin de la récursion sans savoir si la condition $\neg B \notin E$ sera vérifiée.

Exemple 4 Soit $\Delta = (D, W)$ une théorie des défauts, avec $W = \emptyset$ et $D = \left\{ \frac{\neg A}{\neg A} \right\}$. Δ n'a pas d'extension car si E était une extension de Δ , deux cas seraient possibles :

(i) $\neg A \notin E$: E n'est pas une extension car, si $\neg A \notin E$ alors, d'après la condition 3., il serait possible d'utiliser le défaut d et d'obtenir $\neg A \in E$.

(ii) $\neg A \in E$: comme W est vide, pour avoir $\neg A \in E$ il faut utiliser le défaut $\left\{ \frac{\neg A}{\neg A} \right\}$. C'est impossible car la condition 3., dit que pour utiliser ce défaut il faut que $\neg A \notin E$.

Cet exemple est intéressant dans le cadre de l'article, car il est équivalent à un SDB de fonction $g(x_1) = (\neg x_1)$ et dont le graphe d'interaction $G(g) = (V = \{1\}, I(f) = \{(1, -)\})$ correspond à un circuit négatif de taille 1. De même, on obtient une traduction d'un circuit positif de taille 2 associé au SDB de fonction $f(x_1, x_2) = (\neg x_1, \neg x_2)$ en une théorie des

défauts $\Delta_2 = (D, W)$, où $W = \emptyset$ et $D = \left\{ \frac{A}{\neg B}, \frac{B}{\neg A} \right\}$. Et, Δ_2 a deux extensions $E_1 = \text{Th}(\{(A, \neg B)\})$ et $E_2 = \text{Th}(\{(\neg A, B)\})$, comme attendu par le formalisme SDB. On a les mêmes comportements avec la traduction des SDB en ASP.

Comme évoqué dans l'introduction, l'absence d'extension est gênante car, si une théorie des défauts Δ contient une forme locale d'inconsistance, par exemple le défaut $\frac{A}{\neg A}$, elle est invalidé. C'est pour cela que l'on se restreint souvent à des défauts normaux ; on sait que si W est consistant et que si tous les défauts de Δ sont normaux, il existe toujours une extension [24]. De plus si les tous les défauts sont normaux, la condition $\neg B \notin E$ de la définition 3 peut être remplacée par $\neg B \notin E_i$. Alors E_i est connu quand E_{i+1} est construit et la condition peut être immédiatement vérifiée. L'algorithme devient constructif. Intuitivement ceci a pour conséquence que pour calculer les extensions, il n'est pas nécessaire de gérer des "boucles". Mais, l'inconvénient de la restriction aux défauts normaux est que l'on perd du pouvoir expressif. La logique des hypothèses a été défini pour pallier à cet inconvénient.

4 Logique des hypothèses, théories des hypothèses

La logique des hypothèses est une logique bi-modale, introduite dans [30, 32]. Avant de la présenter on donne les bases de logique modale. Dans la suite f et g peuvent être n'importe quelle formule, modale ou non.

La logique du premier ordre donne une valeur de vérité dans l'ensemble $\{\text{faux}, \text{vrai}\}$ à toute formule. Les logiques modales nuancent ces valeurs. Par exemple, au lieu d'exprimer uniquement *Il fait beau* ou *Il ne fait pas beau*, on module ces deux états Booléens en exprimant par exemple *Je sais/pense qu'il fait beau*; *Il est possible qu'il fasse beau*, *Je crois qu'il fait beau*; *Demain il fera beau*; *Il fait beau ailleurs...* Pour décrire ces précisions on ajoute un opérateur modal, noté L , à la logique classique. On associe à L son dual M défini par $Mf = \neg L \neg f$. Le sens intuitif donné aux couples $Mf = \neg L \neg f$ dépend du contexte. Par exemple dans un cadre de programmation par contrainte, le couple $\{\text{prouvé}, \text{possible}\}$ peut être utilisé. Dans ce cas f dit que f est *vrai*, $\neg f$ dit que f est *faux*, Lf dit que f est *prouvé* et Mf dit que f est *possible*, car on ne peut pas prouver que f est *faux*. On peut aussi avoir les couples (certain, possible), {savoir, croire}, {obligatoire, permis}... La logique modale est une sorte de couteau Suisse pour la représentation des connaissances. La définition inductive permet de composer les modalités Par exemple $L(p \rightarrow (M(p \rightarrow ML\neg q)))$ est une formule modale.

Comme toute logique, les logiques modales peuvent être définies axiomatiquement par des règles d'inférence et des schémas d'axiomes. La base est l'ensemble des règles et axiomes du calcul propositionnel ou de la logique du premier ordre. Pour les logiques modales dites *normales*, on ajoute toujours à la fois la *règle de nécessité* :

$$(N) : \vdash f \Rightarrow \vdash Lf$$

et le schéma d'axiomes de distribution :

$$(K) : \vdash (L(f \rightarrow g) \wedge Lf) \rightarrow Lg.$$

on obtient alors le système modal K . Si on ajoute à K le schéma d'axiomes de réflexivité :

$$(T) : \vdash Lf \rightarrow f,$$

on obtient le système modal T .

La règle (N) signifie que *si f est une tautologie, alors Lf est également une tautologie*. Le schéma d'axiomes (K) permet de retrouver pour la modalité un analogue axiomatique de Modus Ponens. Le schéma d'axiomes (T) peut signifier *ce qui est prouvé est vrai*; il est logiquement équivalent à $\vdash \neg(Lf \wedge \neg f)$ et aussi à $\vdash \neg f \rightarrow \neg Lf$.

On note que dans le système T deux schémas d'axiomes essentiels peuvent être dérivés

$$(i) \vdash L(f \wedge g) \leftrightarrow (Lf \wedge Lg)$$

et surtout

$$(ii) \vdash \neg(Lf \wedge L\neg f).$$

Ces deux schémas d'axiomes ne sont pas valides dans K .

4.1 Logique des hypothèses

La logique des hypothèses, notée \mathcal{H} , est une logique bi-modale avec deux opérateurs modaux L et H . Avant d'étudier le formalisme, on en donne l'intuition. On a dit qu'un défaut $d = \frac{A:B}{C}$ peut signifier : *Si A est prouvé, si B est cohérent/consistant, alors C est prouvé*.

Si on traduit *A est prouvé* par LA , *B est cohérent/consistant* par HB et *C est prouvé* par LC , alors le défaut d se traduit par la formule modale de \mathcal{H} :

$$LA \wedge HB \rightarrow LC$$

Et en ASP, la règle $p \leftarrow q, \text{not } r$ pourrait se traduire par :

$$Lq \wedge H\neg r \rightarrow Lp$$

Ensuite, on se rapproche de l'idée de preuve mathématique en posant que Hf signifie que *f est une hypothèse*. Et si $[H]$ est l'opérateur dual de H , défini par $[H]f = \neg H \neg f$, la formule $[H]f$ dit que $\neg f$ n'est pas une hypothèse. On peut alors traduire dans \mathcal{H} une théorie des défauts $\Delta = (D, W)$ en traduisant chaque défaut $d \in D$ comme ci dessus et en traduisant chaque formule $f \in W$ par Lf [30, 32].

On donne le formalisme. Le langage de \mathcal{H} , noté $\mathcal{L}(\mathcal{H})$, est une extension de la logique du premier ordre définie par les règles suivantes :

1. Les formules de la logique du premier ordre sont dans $\mathcal{L}(\mathcal{H})$;
2. Si f et g sont dans $\mathcal{L}(\mathcal{H})$ alors les formules $\neg f$, $(f \wedge g)$, $(f \vee g)$, $(f \rightarrow g)$, Lf , Mf , $[H]f$, Hf le sont également.

L'opérateur L a les propriétés du système modal T et l'opérateur $[H]$ celles du système modal K . Donc les règles d'inférences et schémas d'axiomes de \mathcal{H} sont :

- Toutes les règles d'inférence et schémas d'axiome de la logique du premier ordre.

- (N[H]) : $\vdash f \Rightarrow \vdash [H]f$, la règle de nécessité pour [H];
- (NL) : $\vdash f \Rightarrow \vdash Lf$, la règle de nécessité pour L;
- (K[H]) : $\vdash [H](f \rightarrow g) \rightarrow ([H]f \rightarrow [H]g)$, le schéma d'axiomes de distribution pour [H];
- (K[L]) : $\vdash L(f \rightarrow g) \rightarrow (Lf \rightarrow Lg)$, le schéma d'axiomes de distribution pour L;
- (TL) : $\vdash Lf \rightarrow f$, le schéma d'axiomes de réflexivité pour T.

Contrairement à L, [H] n'a pas de schéma d'axiome de distribution et ceci entraîne qu'il n'y a pas de lien entre H et L (entre l'hypothèse et la preuve). On donne un tel lien en ajoutant le schéma d'axiome de liaison suivant :

$$(LI) : \vdash \neg(Lf \wedge H\neg f)$$

Cet axiome, très faible, est fondamental en logique des hypothèses. Il dit qu'il est impossible de prouver f et de faire l'hypothèse $\neg f$ en même temps. On a les équivalences :

$$\neg(Lf \wedge H\neg f) \iff Lf \rightarrow \neg H\neg f \iff Hf \rightarrow \neg L\neg f.$$

La première signifie que *si on fait l'hypothèse f , on ne peut pas prouver $\neg f$* , la seconde que *si on prouve f , on ne peut pas faire l'hypothèse $\neg f$* .

4.2 Théories des hypothèses et extensions

La logique des hypothèses \mathcal{H} est une logique monotone. Pour capturer la non-monotonie on définit, comme pour la logique des défauts, une notion d'extension. Cependant, contrairement à la logique des défauts, on a deux types d'extensions, les *extensions stables* et les *extensions fantômes*.

Formellement une *théorie des hypothèses* est une paire $\mathcal{T} = \{HY, F\}$, où F est un ensemble de formules de \mathcal{H} et HY est un ensemble d'hypothèses. Une *extension* E de \mathcal{T} est un ensemble maximal consistant $E = F \cup HY'$ avec $HY' \subseteq HY$; elle est donc obtenue en ajoutant à F le plus d'hypothèses possible, tout en testant consistant.

(a) E est une *extension stable* si elle satisfait la *propriété de cohérence* :

$$\forall Hf, \neg Hf \in E \Rightarrow L\neg f \in E$$

(b) E est une *extension fantôme* sinon, c'est à dire que :

$$\exists Hf \in E \text{ tel que } \neg Hf \in E \text{ et } L\neg f \notin E.$$

Les théorèmes 1 et 2 ci dessous donnent les propriétés fondamentales de la logique des hypothèses.

Théorème 1 Si F est consistant, alors $\mathcal{T} = \{HY, F\}$ a au moins une extension.

Théorème 2 Soit $\Delta = \{D, W\}$ une théorie des défauts quelconque. Δ peut être traduite en une théorie des hypothèses $\mathcal{T}(\Delta)$ telle que :

- si Δ est consistant, $\mathcal{T}(\Delta)$ a au moins une extension;
- l'ensemble des extensions de Δ est isomorphe à l'ensemble des extensions stables de $\mathcal{T}(\Delta)$.

La traduction utilisée pour le théorème 2 a été présenté au début du chapitre 4.1. On obtient donc une extension, en ajoutant à F un grand ensemble d'hypothèses possible tout en restant consistant. Et si F est consistant il existe toujours une extension, contrairement à la logique des défauts. Une extension E de \mathcal{H} est stable si chaque fois qu'il est interdit de faire l'hypothèse f alors $\neg f$ est prouvé. Elle est fantôme sinon. Les extensions stables de \mathcal{H} correspondent aux extensions stables de la logique des défauts. Dans le chapitre 6 on montrera, en utilisant une traduction des SDB vers \mathcal{H} , que les extensions stables de \mathcal{H} correspondent également aux configurations stables des SDB.

Les extensions fantômes de \mathcal{H} n'ont pas leur correspondance en logique des défauts. D'un autre côté, on a remarqué que les théories des défauts qui n'ont pas d'extension semblent contenir ce qui ressemble beaucoup à des cycles stables ou instables. Dans le chapitre 6 on montre qu'un cycle stable est caractérisé par un ensemble d'extensions fantômes symétriques Certains cycles instables sont également caractérisés par des ensembles d'extensions fantômes, mais la généralisation formelle reste à faire.

Remarque La définition des extensions est en fait basée sur une approche non-monotone préférentielle. Donc la logique des hypothèses a la propriété de cumulativité. La cumulativité permet d'obtenir des algorithmes de calcul d'extensions qui, bien que non déterministes, sont constructifs.

5 Représentation des graphes d'interactions en logique des hypothèses

En chapitre 2 on a vu qu'un réseau génétique peut être représenté par un graphe de transitions asynchrone (GTA) et qu'il peut aussi être représenté par un graphe d'interactions. Un sujet d'étude important est de faire des liens formels entre ces deux représentations. Dans ce chapitre on ne s'intéresse qu'aux graphes d'interactions. Un tel graphe G sera traduit en une théorie des hypothèses $\mathcal{T}(G)$ et on démontrera des propriétés relatives aux extensions de $\mathcal{T}(G)$. Dans le chapitre 6, G sera le graphe d'interaction associé à un GTA et on étudiera les états stables et les cycles stables du GTA en utilisant $\mathcal{T}(G)$. Le formalisme SDB associe à une entité $i \in \{1, \dots, n\}$ une variable booléenne x_i mais, pour alléger les notations, on fera quand c'est possible un abus de notation en considérant que $i = x_i$.

On donne l'intuition de l'utilisation de la logique des hypothèses pour la représentation des réseaux génétiques. Considérons que i est une protéine :

- i (resp. $\neg i$) signifie que i est activée (resp. n'est pas activée);
- Li (resp. $\neg Li$) signifie que le réseau active i (resp. n'active pas); Li est une action;
- $L\neg i$ (resp. $\neg L\neg i$) signifie que le réseau inhibe i (n'inhibe pas); $L\neg i$ est également une action;
- Hi (resp. $\neg Hi$) signifie que le réseau donne (ne donne

- pas) l'autorisation d'essayer d'activer i . Ou encore, qu'une cellule a (n'a pas) la capacité d'activer i ;
- $H\rightarrow i$ (resp. $\neg H\rightarrow i$) signifie que le réseau donne (ne donne pas) l'autorisation d'essayer d'inhiber i . Ou qu'une cellule a (n'a pas) la capacité d'inhiber i ;

Le rôle d'une extension est d'admettre un maximum d'hypothèses tout en restant consistant. Pour cet article les hypothèses sont des autorisations. On remarque que, même si H_i donne au réseau l'autorisation d'activer i , cette activation n'est pas obligatoire. Elle peut être effectuée ou non, suivant le contexte (c.a.d. la théorie des hypothèses qui représente le réseau) et en accord avec l'axiomatique de \mathcal{H} .

Remarque 3. On s'intéresse aux propriétés importantes et suffisantes de \mathcal{H} et à leur sens pour les réseaux génétiques :

$$Li \rightarrow i \quad \text{et} \quad L\neg i \rightarrow \neg i \quad (1)$$

Si le réseau active i (resp. inhibe), alors i est activée (resp. inhibée).

$$\neg(Li \wedge H\rightarrow i) \quad \text{et} \quad \neg(L\neg i \wedge Hi) \quad (2)$$

Il est interdit d'activer (resp. d'inhiber) i et de donner l'autorisation d'inhiber (resp. d'activer) i en même temps.

$$\neg(Li \wedge L\neg i) \quad (3)$$

Il est impossible d'activer et inhiber i en même temps.

L'équation 1 est une instance de l'axiome (T); 2 est une instance de l'axiome de liaison (LI); 3 se dérive immédiatement de 1. On note que ces formules sont équivalentes à des ensembles de clauses binaires qui contiennent au moins un littéral négatif (c.a.d. des clauses de Horn binaires).

5.1 Représentation des graphes d'interaction

Un graphe d'interaction $G = (V, I)$ est traduit en une théorie des hypothèses $\mathcal{T}(G) = \{\text{HY}(G), F(G)\}$ de la manière suivante :

(i) Chaque arc $(i, s, j) \in I(f)$ est traduit par une paire d'implications de \mathcal{H} . Plus précisément, un arc positif $(i, +, j)$ est traduit par $\{Hi \rightarrow Lj, H\rightarrow i \rightarrow L\neg j\}$ et un arc négatif $(i, -, j)$ est traduit par $\{Hi \rightarrow L\neg j, H\rightarrow i \rightarrow Lj\}$;

(ii) $F(G)$ est l'union des traductions des arcs de G .

(iii) $\text{HY}(G)$ est l'ensemble des Hi et $H\rightarrow i$ qui apparaissent dans $F(G)$.

5.2 Propriétés et considérations algorithmiques

En logique des hypothèses il est possible d'avoir Hi et $H\rightarrow i$ en même temps. Ici, c'est interdit à cause de la double implication obtenue par la traduction d'un arc (i, s, j) :

Propriété 1 .

Si $\mathcal{T}(G) = \{\text{HY}(G), F(G)\}$ est la traduction d'un graphe d'interaction G alors $\forall Hi \mid Hi \in \text{HY}(G) \Rightarrow \neg(Hi \wedge H\rightarrow i)$.

Démonstration. Par construction, Hi appartient à $\text{HY}(G)$ s.s.i. G contient un arc (i, s, j) . Avec la traduction de cet arc, une des paires de clauses $\{Hi \rightarrow Lj, H\rightarrow i \rightarrow L\neg j\}$ ou $\{Hi \rightarrow L\neg j, H\rightarrow i \rightarrow Lj\}$ est dans $F(G)$. Dans les deux cas, si $(Hi \wedge H\rightarrow i)$ est vrai on obtient $(Lj \wedge L\neg j)$, d'où l'on dérive $(i \wedge \neg i)$, en utilisant l'axiome (T). On a alors une inconsistance. Donc on a bien $\neg(Hi \wedge H\rightarrow i)$. \square

Remarque. Pour traduire un graphe d'interaction, seul un fragment de \mathcal{H} est utilisé. Ce fragment ne contient que des clauses de Horn binaires. En effet :

1) $F(G)$ est un ensemble fini d'implications du type $Hx \rightarrow Ly$, D'autre part il n'y a pas de variable ni d'imbrication de modalités. On peut donc considérer que Hx et Ly sont des littéraux du calcul propositionnel. Avec ces considération $\{Hx \rightarrow Ly \Leftrightarrow \neg Hx \vee Ly\}$ est une clause de Horn binaire et donc $F(G)$ est un ensemble de clauses de Horn binaires.

2) Les propriétés de la remarque 3, peuvent être également considérées comme des ensembles de clauses de Horn binaires. Il en est de même pour les formules du type $\neg(Hi \wedge H\rightarrow i)$ données par la propriété 1.

Donc un algorithme de calcul d'extension va travailler sur 2-SAT ce qui améliore beaucoup la complexité algorithmique.

Propriété 2 Soit G un graphe d'interactions dont la traduction est $\mathcal{T}(G) = \{\text{HY}(G), F(G)\}$, on a :

1) $F(G)$ est consistant.

2) $\mathcal{T}(G)$ a au moins une extension.

Démonstration. On vient de voir en remarque que $F(G)$ est équivalent à un ensemble de clauses de Horn binaires. Chaque clause contient donc au moins un littéral négatif et donc l'affectation de tous les littéraux à *faux* est un modèle de $F(G)$. Donc $F(G)$ est consistant et $\mathcal{T}(G)$ a au moins une extension d'après le théorème 1. \square

On est en logique des hypothèses, donc à tout couple $\{i, j\}$ sont associés 10 formules modales : les 5 formules $\{i, Hi, H\rightarrow i, Li, L\neg i\}$ associées à i et les 5 formules $\{j, Hj, H\rightarrow j, Lj, L\neg j\}$ associées à j . C'est le prix à payer pour capturer les dynamiques via les extension fantômes. Mais, on montre que l'affectation à *vrai* de Hi (resp. $H\rightarrow i$) fixe 7 des 10 formules modales associées à $\{i, j\}$:

Propriété 3 Si $\mathcal{T}(G)$ est la traduction de G , alors :

- Pour tout $(i, +, j) \in G$, tel que $i \neq j$:

(1) $Hi \rightarrow \{\neg H\rightarrow i, \neg L\neg i, Lj, \neg L\neg j, j, \neg H\rightarrow j\}$

(2) $H\rightarrow i \rightarrow \{\neg Hi, \neg Li, \neg Lj, L\neg j, \neg j, \neg Hj\}$

- Pour tout $(i, -, j) \in G$, tel que $i \neq j$:

(3) $Hi \rightarrow \{\neg H\rightarrow i, \neg L\neg i, L\neg j, \neg Lj, \neg j, \neg Hj\}$

(4) $H\rightarrow i \rightarrow \{\neg Hi, \neg Li, Lj, \neg L\neg j, j, \neg H\rightarrow j\}$

Démonstration. Pour la ligne (1) on affecte Hi à vrai. La propriété 1 donne $\neg H\neg i$. A partir de Hi , l'axiome de liaison (LI) sur Hi donne $\neg L\neg i$. Toujours à partir de Hi , $Hi \rightarrow Li$ donne Li . A partir de Li l'axiome (T) donne i . A partir de Li , l'axiome (T) donne aussi $\neg L\neg i$ avec la remarque 3. Enfin à partir de Li , l'axiome de liaison (LI) donne $\neg H\neg j$. Les lignes (2), (3) et (4) se traitent de la même façon. \square

Donc en affectant Hi ou $H\neg i$, 7 des 10 propositions associées i et j sont affectées ; seules les propositions i, Hj et Li ne sont pas affectées. Ceci simplifie beaucoup le calcul des extensions. D'autre part, comme une extension est obtenue en ajoutant un nombre maximal consistant d'hypothèses et on a des algorithmes constructifs non déterministes.

5.3 Extension complète, degré de liberté

Soient G un graphe d'interaction et $\mathcal{T}(G) = \{HY(G), F(G)\}$ sa traduction en \mathcal{H} . Soit $E = \{Th(F(G) \cup Hx_i)\}$ une extension de $\mathcal{T}(G)$ obtenue en ajoutant à $F(G)$ un ensemble maximal consistant $\{Hx_i\}$ d'hypothèses.

- Le miroir de E , est l'ensemble $E' = Th(F(G) \cup \{H\neg x_i\})$
- E est complète si pour tout i , $Hi \in E$ ou $H\neg i \in E$.
- Un sommet i de G est libre dans E si $Li \notin E$ et $L\neg i \notin E$. Il est lié sinon.
- Le degré de liberté de E , noté $deg(E)$ est le nombre de ses sommets libres.

Propriété 4 Le miroir d'une extension E de $\mathcal{T}(G)$ est une extension de $\mathcal{T}(G)$.

Démonstration. Par construction de $\mathcal{T}(G)$ si $Hx \rightarrow Ly$ est dans $F(G)$ alors $H\neg x \rightarrow L\neg y$ est aussi dans $F(G)$. De même si Hx est dans $HY(G)$, alors $H\neg x$ est aussi dans $\mathcal{T}(G)$. E et son miroir sont alors symétriques. \square

Propriété 5 Si tout sommet d'un graphe d'interaction G a au moins un arc entrant, alors toute extension complète de $\mathcal{T}(G)$ est de degré 0.

Démonstration. Soit E extension complète de $\mathcal{T}(G)$. Pour prouver que E est de degré 0 il faut prouver que tous ses sommets sont liés, c'est dire que pour tout sommet j , $Hj \in E$ ou $H\neg j \in E$. Par hypothèse si j est un sommet il a un arc entrant, $(i, +, j)$ ou $(i, -, j)$. Si $(i, +, j)$ est cet arc entrant, il est traduit par la paire de formules $\{Hi \rightarrow Lj, H\neg i \rightarrow \neg Lj\}$. Comme E est complète, par définition $Hi \in E$ ou $H\neg i \in E$. Si $Hi \in E$ on dérive Li avec $Hi \rightarrow Lj$. De même si $H\neg i \in E$. on dérive $L\neg i$ avec $H\neg i \rightarrow L\neg j$. Le cas où $(i, -, j)$ est l'arc entrant se traite de la même manière. On a prouvé que i est lié. \square

Propriété 6 Si tout sommet de G a au moins un arc entrant, alors toute extension complète est stable.

Démonstration. Soit E une extension complète de G . E est stable si, pour tout sommet i :

$$(1) \neg H\neg i \in E \Rightarrow Li \in E \text{ et } (2) \neg Hi \in E \Rightarrow L\neg i \in E$$

Soit donc un sommet i . Comme E est complète elle est de degré 0 d'après la propriété 5. Donc soit Li soit $L\neg i$ est dans E . On considère le cas $Li \in E$. (1) est alors trivialement vérifié. D'autre part comme $Li \in E$, avec l'axiome (LI) on obtient $\neg H\neg i \in E$. Comme par hypothèse E est complète et $\neg H\neg i \in E$ alors $Hi \in E$. Donc (2) est trivialement vérifié.

Le cas où $L\neg i \in E$ se démontre de la même manière. \square

Cette propriété est importante pour la suite. On considère maintenant que tout sommet a un arc entrant. Si un sommet i n'en a pas on peut ajouter un arc $(i, +, i)$.

5.4 Ensemble générateur, graphe d'une extension

Définition 4 Soit $E = Th(F(G) \cup \{Hx_k\})$ une extension de $\mathcal{T}(G)$, telle que $x_k \in \{1, \dots, n, \neg 1, \dots, \neg n\}$. L'ensemble générateur de E , noté $Gen(E)$ est l'ensemble des $(Hx \rightarrow Ly) \in F(G)$ tels que $x \in \{Hx_k\}$

La notion d'ensemble générateur vient de la logique des défauts. On montre que $Gen(E)$ suffit à caractériser E et que $Gen(E)$ ne peut pas contenir simultanément les deux formules $Hx \rightarrow Ly$ les $H\neg x \rightarrow L\neg y$. Une extension peut alors être représentée par un graphe d'interaction non signé :

Propriété 7 Soit $E = Th(F(G) \cup \{Hx_i\})$ une extension de $\mathcal{T}(G)$ et $Gen(G)$ son ensemble générateur, alors :

- 1) $E = Th(Gen(E) \cup \{Hx_i\})$
- 2) Si $Hx \in \{Hx_i\}$ alors $Gen(E)$ ne peut pas contenir à la fois $Hx \rightarrow Ly$ et $H\neg x \rightarrow L\neg y$.
- 3) Si $deg(E) = 1$ existe un ordre sur les x_i avec une permutation circulaire telle que $Gen(E) = \{Hi \rightarrow L(i+1), H(i+1) \rightarrow L(i+2), \dots, H(i-2) \rightarrow L(i-1)\}$

Démonstration. .

1) E contient un ensemble de littéraux $\{Hx_i\}$, donc d'après la propriété 1, E contient aussi l'ensemble de littéraux $\{\neg H\neg x_i\}$. Les littéraux de cet ensemble subsument toutes les clauses de $F(G)$ du type $H\neg x_i \rightarrow Ly$. Donc $E = Th(Gen(E) \cup \{Hx_i\})$

2) Si $Gen(E)$ contenait $Hx \rightarrow Ly$ et $H\neg x \rightarrow L\neg y$. alors Ly et $L\neg y$ seraient dans E , et d'après l'axiome T, y et $\neg y$ seraient dans E . E serait alors inconsistant.

3) Si $deg(E) = 1$, il existe un unique élément $i \in \{1, \dots, n\}$ qui n'est pas lié. Donc $Li \notin E$ et $L\neg i \notin E$. Comme $deg(E) = 1$, tous les autres éléments de $\{1, \dots, n\} \setminus \{i\}$ sont liés et si $E = \{Lk\}, k \in \{1, \dots, n\} \setminus \{i\}$ on a le résultat. \square

Définition 5 Si E est une extension, le graphe de E noté $G(E)$, est le graphe non signé de sommets $\{1, \dots, n\}$ tel que :

$$(i, j) \in G(E) \Leftrightarrow (Hi \rightarrow Lj) \in Gen(E)$$

Propriété 8 Si $G = (V = \{1, \dots, n\}, I)$ est un circuit négatif de taille n alors :

- (1) $\text{Th}(G)$ n'a pas d'extension de degré 0.
- (2) $\text{Th}(G)$ a $2n$ extensions de degré 1.

Démonstration.

(1) Si $\text{Th}(G)$ était de degré 0, alors tous les i seraient liés et son ensemble générateur serait :

$$\{Hx_1 \rightarrow Lx_2, \dots, Hx_{n-1} \rightarrow Lx_n, Hx_n \rightarrow Lx_1\}$$

Ceci voudrait dire que G est un circuit positif.

(2) On peut montrer que si :

$$\{Hx_1 \rightarrow Lx_2, Hx_2 \rightarrow Lx_3, \dots, Hx_n \rightarrow Lx_1\}$$

est le système générateur d'une extension E_1 , alors :

$$\{Hx_2 \rightarrow Lx_3, \dots, Hx_n \rightarrow Lx_1, Hx_1 \rightarrow Lx_2\}$$

est le système générateur d'une extension E_2 .

En répétant cette opération sur toutes les permutations circulaires, qui laissent toutes un élément libre, on obtient n extensions différentes de degré 1. De plus, d'après la propriété 4 les miroirs de ces extensions sont aussi des extensions toutes différentes. Ces n extensions miroirs sont également différentes des n premières extensions. On a donc $2n$ extensions qui forment une orbite d'extensions symétriques. \square

Dans la suite, ces $2n$ extensions sont dites *équivalentes*. Il semble que la propriété 8 pourrait s'étendre aux extensions de degré supérieur à 1. Pour ces extensions il y a plus d'un élément libre et des contraintes sur la position de ces éléments libres jouent un rôle.

6 Liens entre les SDB et la logique des hypothèses

Soit un SDB asynchrone dont le graphe d'interaction est $G = (V, I)$ et le graphe de transition est \mathcal{G} . Soit $\mathcal{T}(G) = \{\text{HY}(G), \text{F}(G)\}$ la théorie des hypothèses associée à G . On montre (théorème 3) qu'il existe un isomorphisme entre les configurations stables et les extensions stables. De plus on montre (théorème 4) que tout circuit négatif de taille n correspond à un unique ensemble de $2n$ extensions fantômes de degré 1 équivalentes.

Théorème 3 Soit \mathcal{S} un BDS asynchrone, G et \mathcal{G} ses graphes d'interaction et de transition et $\mathcal{T}(G)$ la théorie des hypothèses associée à G . On a alors :

1. Si E est une extension stable de $\mathcal{T}(G)$ et si $\{Lx_1, \dots, Lx_n\}$ est l'ensemble de tous les Li et $L-i$ qui sont vrais dans E , alors $\{x_1, \dots, x_n\}$ est une configuration stable de \mathcal{S} .
2. If $\{x_1, \dots, x_n\}$ est une configuration stable de \mathcal{S} , alors $\{Lx_1, \dots, Lx_n\}$ est une extension stable de $\mathcal{T}(G)$.

Théorème 4 Soit \mathcal{S} un SDB asynchrone \mathcal{G} dont le graphe d'interaction G est un circuit négatif. Soit E_1 une extension fantôme de $\mathcal{T}(G)$. L'ensemble de toutes les extensions équivalentes à E_1 correspond à un cycle stable de \mathcal{G} .

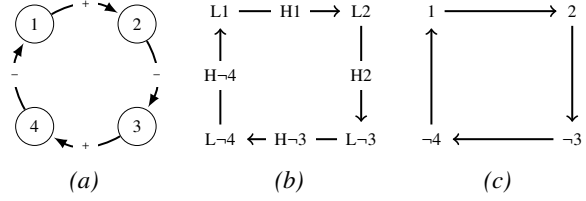


FIGURE 5 - (a) Graphe d'interaction de $G(f)$, (b) Construction de l'extension E , (c) Circuit positif obtenu.

Pour démontrer ces deux théorèmes on a besoin de la sémantique des logiques modale de Kripke [14, 15]. Les démonstrations des théorèmes sont données en 6.2 après la définition de cette sémantique en 6.1. Pour mieux comprendre on commence par regarder des exemples.

Exemple 2 (suite) Soit le SDB asynchrone de fonction f défini dans l'exemple 2. On a vu qu'il admet deux attracteurs qui sont des configurations stable, $(x_1, x_2, \neg x_3, \neg x_4)$ et $(\neg x_1, \neg x_2, x_3, x_4)$. Sa traduction en \mathcal{H} est $\mathcal{T}(G(f)) = \{\text{HY}(G(f)), \text{F}(G(f))\}$, avec :

$$\begin{aligned} - \text{HY}(G(f)) &= \{H1, H2, H3, H4, H-1, H-2, H-3, H-4\} \\ - \text{F}(G(f)) &= \{H1 \rightarrow L2, H2 \rightarrow L-3, H3 \rightarrow L4, H4 \rightarrow L-1, \\ &\quad H-1 \rightarrow L-2, H-2 \rightarrow L3, H-3 \rightarrow L-4, L-4 \rightarrow L1\} \end{aligned}$$

$\mathcal{T}(G(f))$ a deux extensions stables, E et son miroir E' , qui correspondent aux deux configurations stables données plus haut :

$$\begin{aligned} E &= \text{Th}(\text{F}(G(f)) \cup H1, H2, H-3, H-4) \\ E' &= \text{Th}(\text{F}(G(f)) \cup H-1, H-2, H3, H4). \end{aligned}$$

L'ensemble générateur de E est :

$$\{H1 \rightarrow L2, H2 \rightarrow L-3, H-3 \rightarrow L-4, L-4 \rightarrow L1\}.$$

Le processus de traduction est décrit en figure 5. En appliquant la propriété 3 on obtient E par construction :

$$\begin{aligned} E &= \text{Th}(\{H1, H2, H-3, H-4, \neg H-1, \neg H-2, \neg H3, \neg H4, \\ &\quad L1, L2, L-3, L-4, \neg L-1, \neg L-2, \neg L3, \neg L4, \\ &\quad 1, 2, \neg 3, \neg 4\} = \{L1, L2, L-3, L-4 \end{aligned}$$

E est de degré 0 car chaque $i \in V$ est lié (c.a.d. que E contient Li ou $L-i$).

$\mathcal{T}(G(f))$ n'a pas d'extension fantôme car, pour tout x , si $\neg Hx \in E$ on a $L-x \in E$. C'est donc une extension stable dont l'ensemble des Lx vrais dans E est $\{L1, L2, L-3$ et $L-4\}$. En utilisant l'axiome (T), cet ensemble donne le circuit positif $\{1, 2, \neg 3$ et $\neg 4\}$ qui représente les motifs d'expression de chaque gène ; c'est la première configuration stable du SDB.

Le même raisonnement permet de construire l'extension miroir de E , $E' = \{L-1, L-2, L3, L4\}$ qui correspond à la deuxième configuration stable du SDB.

Exemple 3 (suite) Soit le SDB défini dans l'exemple 3, de fonction $g(x_4, x_1, x_2, x_3) = (\neg x_4, x_1, x_2, x_3)$. Ce SDB a un attracteur, un cycle stable de longueur 8. Il a aussi un cycle instable de longueur 8. Sa traduction en \mathcal{H} est $\mathcal{T}(G(g)) = \{HY(G(g)), F(G(g))\}$, avec :

$$F(G(g)) = \{H1 \rightarrow L2, H2 \rightarrow L3, H3 \rightarrow L4, \\ H4 \rightarrow L\bar{1}, H\bar{1} \rightarrow L\bar{2}, H\bar{2} \rightarrow L\bar{3}, \\ H\bar{3} \rightarrow L\bar{4}, H\bar{4} \rightarrow L1\}$$

On obtient une orbite de huit extensions fantômes symétriques, équivalentes. La première est :

$$E1 = Th(F(GI)) \cup \{H1, H2, H3\} \\ = \{H1, H2, H3, \neg H\bar{1}, \neg H\bar{2}, \neg H\bar{3}, \neg H\bar{4} \\ L2, L3, L4, \neg L\bar{1}, \neg L\bar{2}, \neg L\bar{3}, \neg L\bar{4}, 2, 3, 4\}.$$

L'ensemble générateur de $E1$ est :

$$\{H1 \rightarrow L2, H2 \rightarrow L3, H3 \rightarrow L4\}$$

Le littéral $\neg H\bar{1}$ caractérise $E1$ comme extension fantôme car $E1$ ne contient pas $L1$. Le sommet 1 est libre car $E1$ ne contient ni $L1$ ni $L\bar{1}$. C'est le seul sommet libre donc, $E1$ est une extension fantôme de degré 1. Pour cette extension, 2, 3 et 4 sont liés et on connaît rien sur 1.

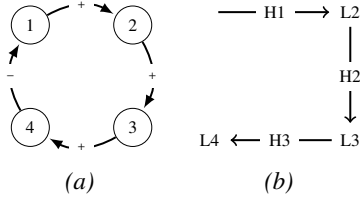


FIGURE 6 – (a) graphe d'interaction $G(g)$, (b) Construction de $E1$

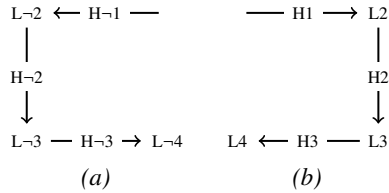


FIGURE 7 – (a) et (b) sont 2 extensions fantômes miroir.

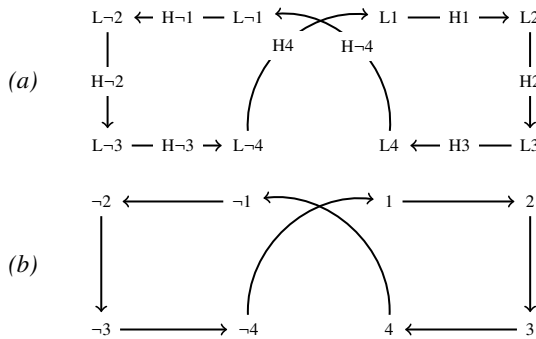


FIGURE 8 – Construction du circuit négatif.

Les figures 6, 7 et 8 montrent intuitivement ce qui se passe. $E1$ est construite à partir de $\{H1, H2, H3\}$. On obtient donc $\{L2, L3, L4\}$ qui, avec l'axiome (T) donnent $\{2, 3, 4\}$. (figure 6(b)). La place de 1 est vide car, l'extension ne contient ni $L1$ ni $L\bar{1}$ (c.a.d que 1 est libre). En effet pour obtenir $L1$ on devrait utiliser $H\bar{4} \rightarrow L1$, ce qui est impossible car $L4$ est vrai et l'axiome de cohérence $L4 \rightarrow \neg H\bar{4}$ donne $\neg H\bar{4}$. De même on ne peut pas avoir $L\bar{1}$ car $H1$ est vrai et l'axiome de liaison $H1 \rightarrow \neg L\bar{1}$ donne $\neg L\bar{1}$

Mais, $E1$ est une extension donc on obtient une deuxième extension miroir $E2$ (figure 7). Et comme 1 et 2 sont libres dans les deux extensions, on peut raccorder $E1$ et $E2$ en allant de $L4$ à $L\bar{1}$ en utilisant $H\bar{4}$ et, en allant de $L\bar{4}$ à $L1$ en utilisant $H4$ (figure 8(a)). On trouve alors un cycle $(1, 2, 3, 4, \bar{1}, \bar{2}, \bar{3}, \bar{4})$ (figure 8(b)).

Pour cet exemple on a une orbite de 8 extensions symétriques : $E1 = \{L2, L3, L4\}$, son miroir $E2 = \{L\bar{2}, L\bar{3}, L\bar{4}\}$ et 6 autres extensions symétriques à $E1$: $\{L3, L4, L\bar{1}\}, \{L4, L\bar{1}, L\bar{2}\}, \{L\bar{1}, L\bar{2}, L\bar{3}\}, \{L\bar{2}, L\bar{3}, L\bar{4}\}, \{L\bar{3}, L\bar{4}, L1\}, \{L\bar{4}, L1, L2\}$

On remarque que chacune de ces extensions correspond à 3 entités successives du cycle $(1, 2, 3, 4, \bar{1}, \bar{2}, \bar{3}, \bar{4})$ ci-dessus. Dans un certain sens, ce cycle contient implicitement les 8 extensions. Mais on a triché, car on a en fait utilisé une mise à jour synchrone qui n'est pas autorisée. Cette tricherie semble vertueuse car, en utilisant la sémantique de Kripke, on va voir que les 8 extensions fantômes correspondent aux configurations du cycle stable de $\mathcal{G}(g)$.

6.1 Sémantique de Kripke

La sémantique de Kripke [14, 15] est une sémantique pour les logiques modales normales c'est-à-dire celles qui vérifient au moins le système K. On en donne les bases utiles à cet article.

Un cadre de Kripke est un couple (W, R) où W , l'univers est un ensemble $\{w_k\}$ de mondes et $R \in W^2$ est une relation binaire : la relation d'accessibilité. Si wRw' on dit que w' est accessible à partir de w .

Un modèle de Kripke est obtenu en affectant, dans chaque monde w , à chaque proposition i une valeur de vérité dans $\{0, 1\}$. Tout monde est alors associé à une interprétation de la logique propositionnelle et donc implicitement, à un état d'un SDB.

Les formules modales autres que les i sont affectées avec la condition suivante : Pour tout f , Lf est vrai dans un monde w ssi f est vrai dans tous les mondes accessibles à partir de w .

Il est alors démontré que :

- 1) Pour le système K, R peut être n'importe quelle relation.
- 2) L'axiome (T) est valide ssi R est réflexive.

Exemple 5 Cet exemple est représenté par la figure 9.

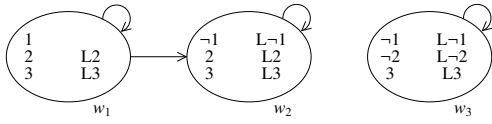


FIGURE 9 – Modèle de Kripke de l'exemple 5

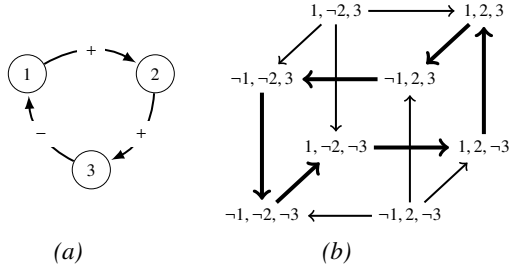


FIGURE 10 – (a) Graphe d'interaction et (b) de transition du SDB de l'exemple 1.

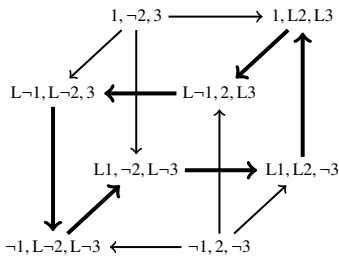


FIGURE 11 – Graphe de transition du SDB vu comme un modèle de Kripke. On a simplement pris la figure 1(b) et mis des L devant les i quand la sémantique de Kripke le demande.

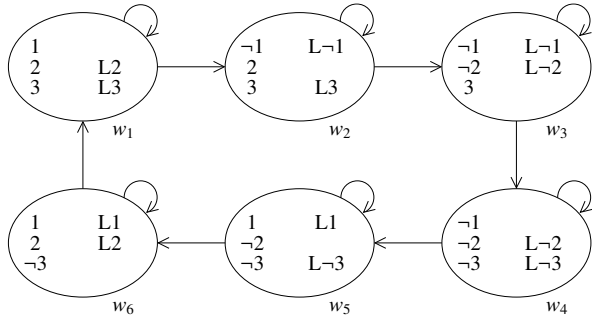


FIGURE 12 – Autre représentation des états du cycle stable par un modèle de Kripke. Il manque les états extérieurs au cycle ; ces états sont inaccessibles car ils sont de liberté supérieurs à 1.

Soit un univers (W, R) avec $W = \{w_1, w_2, w_3\}$ et $R = \{(w_1, w_2), (w_2, w_3), (w_1, w_1), (w_2, w_2), (w_3, w_3)\}$

Dans chaque monde, on affecte des valeurs de vérité aux i de la manière suivante :

$$w_1 = \{1, 2, 3\}, w_2 = \{\neg 1, 2, 3\}, w_3 = \{\neg 1, \neg 2, 3\}.$$

La relation R permet d'associer à tout L_i une valeur de vérité. Par exemple dans w_1 , L_2 est vrai car 2 est vrai dans w_1 et w_2 (c.a.d. les mondes accessibles de w_1). On a aussi $\neg L_1$ dans w_1 car 1 est faux dans w_2 qui est accessible à partir de w_1 .

On remarque que si w_1 était une extension, ce serait une extension fantôme de degré 1 car L_2 and L_3 sont vrais dans w_1 et ni L_1 ni $L\neg 1$ ne sont vrais. De manière similaire si w_3 était une extension, ce serait une extension stable de degré 0 car pour tout i , soit L_i soit $L\neg i$ est vrai ; w_3 devrait alors correspondre à une configuration stable car aucun arc n'en sort.

Exemple 6 Soit le circuit négatif de l'exemple 1, $g(x_1, x_2, x_3) = (\neg x_3, x_1, x_2)$ De la même manière que pour l'exemple 3-suite, on obtient :

$$F(G(g)) = \{H1 \rightarrow L2, H2 \rightarrow L3, H3 \rightarrow L\neg 1, H\neg 1 \rightarrow L\neg 2, H\neg 2 \rightarrow L\neg 3, H\neg 3 \rightarrow L1\}$$

Il y a 6 extensions symétriques, équivalentes. En ne s'intéressant qu'aux Lx vrais dans les extensions, on obtient :

$$E1 = (L2, L3), E2 = (L\neg 1, L3), E3 = (L\neg 1, L\neg 2), \\ E4 = (L2, L\neg 3), E5 = (L\neg 2, L\neg 3), E6 = (L1, L2)$$

On a 6 extensions symétriques qui correspondent aux 6 sommets du cycle du cycle stable $GG(G)$. La figure 11 montre comment le graphe d'interaction peut être considéré comme un modèle de Kripke.

La figure 12 donne une autre représentation des 6 extensions en sémantique de Kripke. Le monde w_1 contient L_2 et L_3 et correspond à $E1$. Le w_2 contient $L\neg 1$ et L_3 et correspond à $E2$. Dans w_1 , L_2 est vrai car 2 est vrai dans tous les mondes (w_1 and w_2) accessibles de w_1 . Il en est de même pour L_3 . Dans w_1 , L_1 est faux car 1 est faux dans w_2 et $L\neg 1$ est faux car 1 est vrai dans w_1 . Donc, 1 est libre. Les autres mondes sont traités de la même manière.

Cet exemple suggère les démonstrations des théorèmes 3 et 4.

6.2 Démonstration des théorèmes 3 et 4

On utilise des modèles de Kripke (W, R) pour lesquels l'ensemble fini des mondes W est l'ensemble des états/interprétations (c'est l'univers canonique) et tels que wRw' ssi w et w' diffèrent au plus d'une proposition. Avec ces conditions tous les GTA asynchrone sont des modèles de Kripke. Comme T est réflexif on a des boucles sur tous les états. Etant donné un tel modèle, pour tout monde w et toute entité x , Lx est obtenu de manière classique : $Lx = 1$ ssi $x = 1$ dans tout w' accessible à partir de w . Pour simplifier les démonstrations, on étend la notion de degré de liberté d'une extension aux états : le degré de liberté d'un état w est le nombre de ses i tels que ni L_i ni $L\neg i$ n'est vrai.

On montre alors qu'il est impossible d'aller d'un état de degré k à un état de degré $k' < k$:

Propriété 9 .

- 1) Si w est un monde de (W, R) de degré k , alors exactement k mondes différents de w sont accessibles à partir de w .
 2) Si wRw' alors $\text{deg}(w) \leq \text{deg}(w')$.

Démonstration.

1) Soit i (resp. $\neg i$) vrai dans w . Si i est lié alors, soit Li soit $L\neg i$ est vrai dans w . On a $\neg L\neg i$ dans w car R est réflexive et i est vrai. Donc Li doit être vrai et il est nécessaire que i soit vrai dans tous les mondes accessibles à partir de w .

Par définition, si un i est libre, on a $L\neg i$ dans un monde w' accessible de w . Par construction, de w il ne peut exister qu'un seul monde w' qui vérifie ceci. Cela est le cas pour tous les i libres et, si le degré de liberté est k on a exactement k mondes accessibles.

2) Si on avait $\text{deg}(w) > \text{deg}(w')$, on aurait plus de k mondes accessibles à partir de w . \square

Théorème 3

Soit \mathcal{S} un BDS asynchrone, G et \mathcal{G} ses graphes d'interaction et de transition et $\mathcal{T}(G)$ la théorie des hypothèses associée à G . On a alors :

1. Si E est une extension stable de $\mathcal{T}(G)$ et si $\{Lx_1, \dots, Lx_n\}$ est l'ensemble de tous les Li et $L\neg i$ qui sont vrais dans E , alors $\{x_1, \dots, x_n\}$ est une configuration stable de \mathcal{S} .
2. If $\{x_1, \dots, x_n\}$ est une configuration stable de \mathcal{S} , alors $\{Lx_1, \dots, Lx_n\}$ est une extension stable de $\mathcal{T}(G)$.

Démonstration. Soit E représenté par le monde w . Comme E est stable son degré de liberté est 0. Donc d'après la propriété 10 le seul monde accessible de w est lui même. C'est donc un état stable du GTA associé à GI .

Réciproquement si w est un état stable du GTA associé, aucun arc ne sort de w . Et, comme R est réflexive, le seul monde accessible à partir de w est lui même. On obtient donc que pour tout i (resp $\neg i$) de w , Li (resp. $L\neg i$) est vrai. Donc, tous les i sont liés et E est une extension stable. \square

Théorème 4 Soit \mathcal{S} un SDB asynchrone \mathcal{G} dont le graphe d'interaction G est un circuit négatif. Soit $E1$ une extension fantôme de $\mathcal{T}(G)$. L'ensemble de toutes les extensions équivalentes à $E1$ correspond à un cycle stable de \mathcal{G} .

Démonstration. La preuve est analogue à celle du théorème 3. Il faut de plus utiliser le fait que les extensions sont obtenues en ajoutant un ensemble maximum consistant d'extensions. Ceci implique que dans la structure de Kripke tous les arcs vont d'une extension de degré 1 vers une autre extension de degré 1. On retrouve alors le cycle stable. \square

Remarque Avec les mêmes arguments que ceux utilisés pour la démonstration de la propriété 9, si $\text{deg}(E) > 1$ le modèle de Kripke donne au moins une possibilité de sortir d'un cycle. Ce cycle devrait être instable.

7 Conclusion

L'article est un premier pas vers la représentation des SDB par la logique des hypothèses. La difficulté est de trouver comment représenter les cycles instables. Les premiers résultats semblent intéressants.

Il y a encore beaucoup à étudier par exemple une généralisation du théorème 4 aux cycles instables. On a une conjecture qui dit qu'un cycle instable est un ensemble d'extensions fantômes de degré de liberté supérieur à 1. Pour étudier cette conjecture on peut partir de la remarque de la fin du chapitre 6.

Une autre perspective est la validation de la représentation des SDB en logique des hypothèses par l'obtention des théorèmes fondamentaux. Le fait que la représentation en logique des hypothèses d'un circuit positif ait deux extensions miroirs stables et, qu'un circuit négatif est équivalent à un unique ensemble de $2n$ d'extensions fantômes équivalentes est un petit pas vers cette validation car, cela ressemble aux résultats établis dans [25].

Il faut aussi étudier la prise en compte par la logique des hypothèses d'autres modes de mise que le mode asynchrone, en commençant par exemple par le mode parallèle, puis en généralisant aux blocs-séquentiels, qui sont des modes de mise à jour déterministes périodiques.

Egalement, l'utilisation d'un algorithme d'extensions pour dénombrer les cycles peut être pertinente.

Pour cette première approche seul un petit fragment de la logique des hypothèses HH a été utilisé. Ce fragment est suffisant pour décrire les SDB étudiés dans cet article, mais on peut utiliser le formalisme pour décrire d'autres propriétés des réseaux biologiques, par exemple pour représenter la liaison de protéines. Pour tester les réseaux, il est aussi possible d'affirmer une proposition i , ou Li , ou encore Hi seule. On peut aussi ne pas utiliser la double implication $\{Hx \rightarrow Ly, H\neg x \rightarrow L\neg y\}$ donnée par la traduction des SDB; pour certaines applications une seule implication suffit. Tout cela augmente la puissance d'expression. La représentation peut alors sortir du cadre de 2-SAT et la complexité algorithmique est plus importante.

Références

- [1] Aracena, J., M. González, A. Zuñiga, M. A. Mendez et V. Cambiazo: *Regulatory network for cell shape changes during Drosophila ventral furrow formation*. Journal of Theoretical Biology, 239 :49–62, 2006.
- [2] Bossu, G. et P. Siegel: *Nonmonotonic Reasoning and Databases*. Dans *Advances in Data Base Theory, Vol. 2, Based on the Proceedings of the Workshop on Logical Data Bases, December 14-17, 1982, Centre d'études et de recherches de Toulouse, France*, pages 239–284, 1982.
- [3] Bossu, G. et P. Siegel: *Nonmonotonic Reasoning and the Closed-World Assumption*. Artif. Intell., 25(1) :13–63, 1985.

- [4] Colmerauer, A. et P. Roussel: *History of Programming Languages – II*, chapitre The Birth of Prolog, pages 331–367. Addison Wesley, 1996.
- [5] Delgrande, J. P., T. Schaub, H. Tompits et K. Wang: *A classification and survey of preference handling approaches in nonmonotonic reasoning*. Computational Intelligence, 20 :308–334, 2004.
- [6] Demongeot, J., A. Elena, M. Noual, S. Sené et F. Thuderoz: *"Immunetworks", intersecting circuits and dynamics*. Journal of Theoretical Biology, 280 :19–33, 2011.
- [7] Demongeot, J., E. Goles, M. Morvan, M. Noual et S. Sené: *Attraction basins as gauges of the robustness against boundary conditions in biological complex systems*. PLoS One, 5 :e11793 (18 pages), 2010.
- [8] Demongeot, J., M. Noual et S. Sené: *Combinatorics of Boolean automata circuits dynamics*. Discrete Applied Mathematics, 160 :398–415, 2012.
- [9] Doncescu, Andrei et Pierre Siegel: *DNA Double-Strand Break-Based Nonmonotonic Logic*. Dans *Computational Biology, Bioinformatics and Systems Biology*, pages 409–427. Elsevier, 2015.
- [10] Doncescu, Andrei, Pierre Siegel et Tan Lee: *Representation an efficient algorithms for the study of cell signaling pathways*. Dans *ICAI*, pages 504–510. IEEE Computer Society, 2014.
- [11] Fauré, A., A. Naldi, C. Chaouyia et D. Thieffry: *Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle*. Bioinformatics, 22 :e124–e131, 2006.
- [12] Kauffman, S. A.: *Metabolic stability and epigenesis in randomly constructed nets*. Journal of Theoretical Biology, 22 :437–467, 1969.
- [13] Kauffman, S. A., C. Peterson, B. Samuelsson et C. Troein: *Random Boolean network models and the yeast transcriptional network*. PNAS, 100 :14796–14799, 2003.
- [14] Kripke, S. A.: *Semantical analysis of modal logic I Normal modal propositional calculi*. Mathematical Logic Quarterly, 9 :67–96, 1963.
- [15] Kripke, S. A.: *Semantical Analysis of Intuitionistic Logic I*. Studies in Logic and the Foundations of Mathematics, 40 :92–130, 1965.
- [16] Lifschitz, V.: *The Logic Programming Paradigm : A 25-Year Perspective*, chapitre Action languages, answer sets, and planning, pages 357–373. Springer, 1999.
- [17] McCarthy, J.: *Circumscription – A form of non-monotonic reasoning*. Artificial Intelligence, 13 :27–39, 1980.
- [18] McCarthy, J.: *Applications of circumscription to formalizing common-sense knowledge*. Artificial Intelligence, 28 :89–116, 1986.
- [19] Melliti, T., M. Noual, D. Regnault, S. Sené et J. Sobieraj: *Asynchronous dynamics of Boolean automata double-cycles*. Dans *Proceedings of UCNC'2015*, tome 9252 de LNCS, pages 250–262. Springer, 2015.
- [20] Melliti, T., D. Regnault, A. Richard et S. Sené: *On the convergence of Boolean automata networks without negative cycles*. Dans *Proceedings of AUTOMATA'2013*, tome 8155 de LNCS, pages 124–138. Springer, 2013.
- [21] Mendoza, L. et E. R. Alvarez-Buylla: *Dynamics of the genetic regulatory network for Arabidopsis thaliana flower morphogenesis*. Journal of Theoretical Biology, 193 :307–319, 1998.
- [22] Mendoza, L., D. Thieffry et E. R. Alvarez-Buylla: *Genetic control of flower morphogenesis in Arabidopsis thaliana*. Bioinformatics, 15 :593–606, 1999.
- [23] Noual, M.: *Updating Automata Networks*. Thèse de doctorat, École normale supérieure de Lyon, 2012.
- [24] Reiter, R.: *A logic for default reasoning*. Artificial Intelligence, 13 :81–132, 1980.
- [25] Remy, É., B. Mossé, C. Chaouyia et D. Thieffry: *A description of dynamical graphs associated to elementary regulatory circuits*. Bioinformatics, 19 :ii172–ii178, 2003.
- [26] Remy, É., P. Ruet et D. Thieffry: *Graphic requirement for multistability and attractive cycles in a Boolean dynamical framework*. Advances in Applied Mathematics, 41 :335–350, 2008.
- [27] Richard, A.: *Negative circuits and sustained oscillations in asynchronous automata networks*. Advances in Applied Mathematics, 44 :378–392, 2010.
- [28] Richard, A. et J. P. Comet: *Necessary conditions for multistationarity in discrete dynamical systems*. Discrete Applied Mathematics, 155 :2403–2413, 2007.
- [29] Robert, F.: *Discrete Iterations : A Metric Study*. Springer, 1986.
- [30] Schwind, C. et Siegel P.: *A modal logic for hypothesis theory*. Fundamenta Informaticae, 21 :89–101, 1994.
- [31] Shoham, Y.: *Readings in Nonmonotonic Reasoning*, chapitre A semantical approach to nonmonotonic logics, pages 227–250. Morgan Kaufmann Publishers Inc., 1987.
- [32] Siegel, P. et C. Schwind: *Modal logic based theory for non-monotonic reasoning*. Journal of Applied Non-classical Logic, 3 :73–92, 1993.
- [33] Thieffry, D. et R. Thomas: *Dynamical behaviour of biological regulatory networks – II. Immunity control in bacteriophage Lambda*. Bulletin of Mathematical Biology, 57 :277–297, 1995.
- [34] Thomas, R.: *Boolean formalization of genetic control circuits*. Journal of Theoretical Biology, 42 :563–585, 1973.
- [35] Thomas, R.: *Numerical Methods in the Study of Critical Phenomena*, chapitre On the relation between the logical structure of systems and their ability to generate multiple steady states or sustained oscillations, pages 180–193. Springer, 1981.

Allocation d'objets par des échanges le long d'un réseau social

Laurent Gourvès¹Julien Lesca¹Anaëlle Wilczynski¹

¹ Univ. Paris-Dauphine, PSL Research University, CNRS, LAMSADE, Paris, France
 {laurent.gourves, julien.lesca, anaëlle.wilczynski}@dauphine.fr

Résumé

Cet article s'intéresse à un problème particulier d'allocation de ressources où chaque agent reçoit un unique objet. Initialement dotés d'un objet, les agents peuvent effectuer des échanges afin d'obtenir un objet qui leur plaît mieux. Seulement, tous les échanges ne sont pas plausibles en raison de l'incapacité de certains agents à communiquer entre eux. En considérant un réseau social sur les agents, on propose d'étudier les allocations possibles obtenues par des séquences d'échanges entre paires de voisins dans le réseau. Ce modèle soulève des questions naturelles concernant (i) la possibilité pour une allocation complète d'être atteignable par une séquence d'échanges dans le réseau, (ii) la possibilité pour un agent d'obtenir un objet donné, et (iii) la recherche d'allocations Pareto-efficaces. On se concentre sur la complexité de ces questions en déterminant, selon la structure du réseau social, des cas polynomiaux ou NP-complets.

Abstract

This article deals with the allocation of objects where each agent receives a single item. Starting from an initial endowment, the agents can be better off by exchanging their objects. However, not all trades are likely because some participants are unable to communicate. By considering that the agents are embedded in a social network, we propose to study the possible allocations emerging from a sequence of simple swaps between pairs of neighbors in the network. This model raises natural questions regarding (i) the reachability of a given full allocation, (ii) the ability of an agent to obtain a given object, and (iii) the search of Pareto-efficient allocations. We investigate the complexity of these problems by providing, according to the structure of the social network, polynomial and NP-complete cases.

1 Introduction

L'allocation de ressources indivisibles apparaît comme une problématique fondamentale en IA [9], se situant à l'interface entre l'informatique et l'économie. Lorsque l'ensemble des solutions est restreint à celles où chaque

agent reçoit une ressource unique, le problème correspond à un cas spécifique de couplage [22, 21], que l'on peut appeler unilatéral (*one-sided matching*), car seulement un des deux groupes a des préférences sur le second. Les couplages unilatéraux ont été largement étudiés selon différentes approches, notamment par le biais du problème d'affectation [18], ou du *house allocation problem* [1]. Quand, de plus, chaque agent est initialement doté d'une ressource, le problème est connu dans la littérature sous le nom de *housing market problem* [24]. Une approche standard afin de réallouer les ressources consiste à effectuer des séquences d'échanges entre agents à partir de l'allocation initiale. L'algorithme *top-trading cycle* et ses variantes [24, 2, 7] exploitent cette idée dans le but d'obtenir une meilleure allocation du point de vue des agents. Une autre façon d'appréhender le problème réside dans le fait de laisser les agents effectuer eux-mêmes des échanges. Une riche littérature s'est développée autour de cette perspective distribuée, formalisant des conditions de réalisme pour un échange et analysant la qualité des issues possibles [23, 15, 11, 17, 12, 14, 10, 6]. La plus grande part de cette littérature est dédiée au problème général d'allocation de ressources où les agents peuvent recevoir plusieurs ressources, mais seulement très peu s'intéressent spécifiquement au problème du marché du logement [13].

Dans le contexte du problème de *housing market*, il est implicitement supposé que tous les agents peuvent directement effectuer des échanges entre eux. Or, cette hypothèse paraît vraiment improbable, en particulier dans des instances de grande taille où les agents ne sont pas forcément en mesure de communiquer entre eux. Restreindre l'ensemble des échanges directs à ceux qui sont effectivement possibles apparaît donc plus réaliste et pertinent dans des problèmes de grande taille. La possibilité pour deux agents d'échanger leur ressource peut alors être modélisée par le biais d'un réseau social [20, 16]. De récents travaux sur les couplages ont introduit un réseau social dans leur modélisation afin de capturer certains types de comportement social comme l'altruisme [4], l'influence du jugement

des paires [8] ou la collaboration [19]. La plupart de ces travaux se concentrent sur des couplages généraux bilatéraux, et seulement peu d'entre eux se sont intéressés à l'allocation de ressource en particulier. Dans cette perspective, la notion de *graphe de topologie de négociation* [12], où les échanges sont restreints à ceux entre agents d'une même clique du graphe, est à noter.

Cet article s'intéresse à une variante du problème de *housing market*, dans laquelle les agents sont intégrés dans un réseau social qui détermine leur possibilité d'échange d'objets. Chaque participant possède initialement un objet et a des préférences ordinales strictes sur les objets. Les agents peuvent échanger leurs objets sous deux conditions : l'échange est mutuellement profitable, et les agents sont liés socialement. Bien que des échanges plus sophistiqués impliquant plusieurs agents peuvent être effectués [23, 14, 13], on s'intéresse à des échanges simples au sein de paires de voisins. Les échanges sont faits sans paiement ni compensation monétaire. La question principale que l'on se propose d'étudier est de savoir, en partant de l'allocation initiale, quelles allocations d'objets peuvent émerger. En effet, certaines solutions sont éliminées en raison des préférences des agents sur les objets (personne n'est intéressé pour échanger son objet contre un autre qu'il aime moins), et le réseau limite l'accès de certains participants à d'autres. Il apparaît donc particulièrement intéressant de comprendre comment la combinaison de ces deux ingrédients influence l'issue de séquences d'échanges dans lesquelles une paire d'agents connectés entre eux s'entendent sur un échange d'objets mutuellement profitable. De plus, la question de la qualité d'une allocation résultant d'une séquence d'échanges profitables s'avère également digne d'intérêt. Dans le contexte de préférences ordinales, la Pareto-optimalité semble être une exigence minimale pour qu'une allocation soit socialement acceptable. La Pareto-optimalité a été largement étudiée dans le cadre du *house allocation problem* [3] et du *housing market* [6]. Pour autant que nous le sachions, la construction d'allocations Pareto-efficaces n'a pas été étudiée lorsque les allocations possibles sont contraintes par un réseau social.

L'article est organisé de la façon suivante. Le modèle d'allocation d'objets le long d'un réseau social est défini en Section 2, ainsi que les problèmes que nous considérons. Puis, en Section 3, le problème d'accessibilité d'un objet est étudié, et en Section 4 l'atteignabilité d'une allocation complète donnée. Enfin, avant de conclure, la recherche d'une allocation Pareto-efficace parmi celles atteignables est examinée en Section 5.

2 Modèle

2.1 Notations

Soit $N = \{1, \dots, n\}$ un ensemble d'agents et $X = \{x_1, \dots, x_n\}$ un ensemble d'objets tels que $|N| = |X| = n$. Une allocation est une bijection $\sigma : N \rightarrow X$, où $\sigma(i)$ représente l'objet affecté à l'agent i dans σ . On peut écrire σ comme un vecteur de taille n : $\sigma = (\sigma(1), \dots, \sigma(n))$. Chaque agent est initialement doté d'un objet ; l'allocation initiale est notée σ_0 . On suppose sans perte de généralité que $\sigma_0(i) = x_i$ pour tout agent i . Chaque agent i a des préférences strictes sur l'ensemble des objets, ces préférences étant exprimées par un ordre total $>_i$ sur X . Le profil global de préférences est noté $>$.

Soit $G = (N, E)$ un graphe non orienté représentant un réseau social sur les agents. Les arêtes symbolisent une possibilité de communication et d'échange entre deux agents. Une instance est un vecteur $(N, X, >, G, \sigma_0)$.

2.2 Dynamique d'échanges

Dans cet article, on se concentre sur des *échanges rationnels* au sein de paires d'agents. Un échange entre deux agents est dit *rationnel* si les deux agents gagnent en échangeant leurs objets. Une allocation σ' résulte d'un échange rationnel à partir de l'allocation σ , s'il existe deux agents i et j tels que $\sigma'(i) = \sigma(j)$, $\sigma'(j) = \sigma(i)$, $\sigma'(i) >_i \sigma(i)$ et $\sigma'(j) >_j \sigma(j)$, et pour tout agent $k \notin \{i, j\}$, $\sigma'(k) = \sigma(k)$.

On suppose qu'un échange ne peut être effectué qu'entre deux voisins dans le réseau social. On définit donc la réalisabilité d'un échange en fonction de G . Une allocation σ' résulte d'un échange *réalisable* à partir de σ selon $G = (N, E)$ s'il existe deux agents i et j tels que $(i, j) \in E$ et $\sigma'(i) = \sigma(j)$, $\sigma'(j) = \sigma(i)$ et $\sigma'(k) = \sigma(k)$ pour tout agent $k \notin \{i, j\}$. A partir de maintenant, dès que l'on parlera d'échange, on supposera qu'il est rationnel et réalisable. De même, les termes allocation et affectation seront employés indifféremment.

Une séquence d'échanges est une séquence d'affectations $(\sigma_0, \sigma_1, \sigma_2, \dots, \sigma_t)$ où pour tout $k \in \{1, \dots, t\}$, σ_k résulte d'un échange à partir de σ_{k-1} . Une allocation σ est *stable* si aucun échange ne peut être effectué à partir de σ . On peut remarquer que toute séquence d'échanges converge vers un état stable en $O(n^2)$ échanges.

Exemple 1 On considère une instance où $n = 4$. Le réseau social, les préférences des agents et une séquence d'échanges sont décrits ci-dessous.

$$\begin{array}{l}
 G : \quad \textcircled{1} \text{---} \textcircled{2} \text{---} \textcircled{3} \text{---} \textcircled{4} \quad 1 : \quad x_4 > x_2 > \boxed{x_1} > x_3 \\
 \sigma_0 : \quad x_1 \leftrightarrow x_2 \leftrightarrow x_3 \quad x_4 \quad 2 : \quad x_3 > x_1 > \boxed{x_2} > x_4 \\
 \sigma_1 : \quad x_2 \quad x_1 \leftrightarrow x_3 \quad x_4 \quad 3 : \quad x_4 > x_2 > x_1 > \boxed{x_3} \\
 \sigma_2 : \quad x_2 \quad x_3 \quad x_1 \quad x_4 \quad 4 : \quad x_2 > \boxed{x_4} > x_1 > x_3
 \end{array}$$

À partir de l'allocation σ_0 , représentée dans les préférences des agents par des encadrés, les agents 1 et 2

peuvent effectuer un échange : ils sont liés dans le réseau et ils préfèrent tous deux l'objet possédé par l'autre agent. Il en va de même pour les agents 2 et 3. Si les agents 1 et 2 échangent leurs objets (flèche en gras dans la figure), on obtient l'allocation σ_1 . À partir de σ_1 , seuls les agents 2 et 3 peuvent échanger leurs objets, aboutissant à l'allocation σ_2 . À partir de σ_2 , les agents 1 et 4 préfèrent chacun l'objet de l'autre mais ils ne sont pas reliés dans G , donc cet échange n'est pas réalisable. Il s'ensuit qu'aucun échange n'est possible à partir de σ_2 , ainsi σ_2 est stable.

Une affectation σ' est atteignable s'il existe une séquence d'échanges $(\sigma_0, \dots, \sigma_t)$ où $\sigma_t = \sigma'$. On note RA (reachable assignments) l'ensemble de toutes les affectations atteignables. De même, un objet $x \in X$ est dit accessible pour un agent $i \in N$ s'il existe une séquence d'échanges $(\sigma_0, \dots, \sigma_t)$ où $\sigma_t(i) = x$.

La dynamique d'échanges est un processus distribué où les agents échangent leurs objets sans intervention extérieure à partir de l'allocation initiale jusqu'à l'obtention d'une allocation stable.

2.3 Enjeux

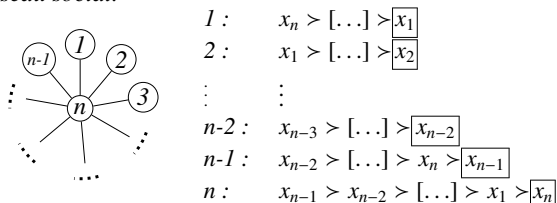
Afin d'analyser le processus distribué des dynamiques d'échanges, on s'intéresse en particulier à toutes les allocations atteignables à partir de σ_0 par des séquences d'échanges. Les deux problèmes de décision suivants émergent naturellement de la définition de notre modèle.

OBJET ACCESSIBLE
Instance : $(N, X, >, G, \sigma_0)$, agent i , objet x
Question : L'objet x est-il accessible pour l'agent i ?

AFFECTATION ATTEIGNABLE
Instance : $(N, X, >, G, \sigma_0)$, affectation σ
Question : L'affectation σ est-elle atteignable ?

Les dynamiques d'échanges convergent toujours vers une allocation stable. Seulement, si les agents troquent leurs objets de manière non coordonnée, il est possible d'aboutir à une allocation assez mauvaise, comme le montre l'exemple suivant.

Exemple 2 On considère une instance avec n agents dont les préférences sont décrites ci-dessous, ainsi que le réseau social.



L'allocation σ_1 résulte du troc entre l'agent $n-1$ et l'agent n . L'agent central n obtient alors l'objet qu'il préfère et

n'a plus intérêt à effectuer d'autres échanges, rendant σ_1 stable. Cependant, si l'on considère la séquence d'échanges entre les agents suivants $(n,1), (n,2), \dots, (n,n-1)$, alors on aboutit à l'allocation σ' dans laquelle tous les agents obtiennent leur objet préféré.

Une allocation σ est Pareto-efficace s'il n'existe pas d'affectation σ' la dominant au sens de Pareto, c'est-à-dire telle que pour tout agent $i \in N$, $\sigma'(i) \geq_i \sigma(i)$ et il existe $j \in N$ tel que $\sigma'(j) >_j \sigma(j)$. Dans tout l'article, on restreint la notion de Pareto-efficacité à l'ensemble RA des affectations atteignables. Autrement, une allocation Pareto-efficace dans le sens standard peut ne pas être atteignable par une séquence d'échanges.

L'exemple 2 montre que les séquences d'échanges peuvent aboutir à une allocation Pareto-dominée. Cette observation nous amène à considérer les dynamiques d'échanges d'un point de vue plus centralisé où les échanges, toujours régis par les contraintes du réseau et du profit mutuel immédiat pour les deux agents, sont guidés par un coordinateur, dans le but d'atteindre une allocation Pareto-efficace parmi toutes les allocations atteignables. On étudie donc également le problème d'optimisation suivant.

PARETO
Instance : $(N, X, >, G, \sigma_0)$
Problème : Trouver une allocation Pareto-efficace parmi RA

3 Objet accessible

Cette section est consacrée au problème de décision OBJET ACCESSIBLE. On prouve qu'il est NP-complet, même dans le cas où le réseau est un arbre. Cependant, avec encore plus de restrictions sur le graphe, le problème devient polynomial.

On peut commencer par observer que dans tout réseau, un objet ne peut passer deux fois par le même agent. Ceci découle de la définition d'un échange rationnel, dans lequel les deux agents doivent être gagnants.

Observation 1 Si pour une affectation atteignable σ et un agent i , $\sigma(i) \neq \sigma_0(i)$ alors pour toute affectation σ' atteignable à partir de σ , $\sigma'(i) \neq \sigma_0(i)$.

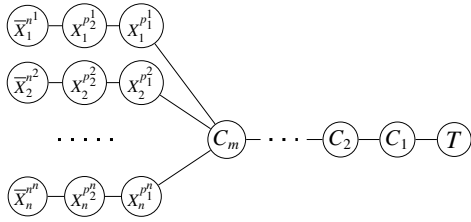
Théorème 1 OBJET ACCESSIBLE est NP-complet même lorsque le réseau est un arbre.

Preuve : On peut vérifier aisément que le problème est dans NP. La réduction utilisée se fait à partir du problème NP-complet 2P1N-SAT [25]. Dans 2P1N-SAT, étant donné un ensemble $V = \{v_1, \dots, v_n\}$ de variables et une collection $C = \{C_1, \dots, C_m\}$ de clauses sur V telle que chaque littéral positif (resp., négatif) apparaît exactement 2 (resp., 1) fois dans C , est-ce que C est satisfiable ?

Soit v_k^j (resp., \bar{v}_k^j) le littéral positif (resp., négatif) de la variable v_k , s'il est présent dans la clause C_i . L'indice p_k^j (resp., n^k) renvoie quant à lui à la clause contenant la j^{eme} occurrence ($j \in \{1, 2\}$) du littéral v_k (resp., la seule occurrence de \bar{v}_k).

L'instance $(N, X, >, G, \sigma_0)$ du problème OBJET ACCESSIBLE est construite comme suit. Chaque littéral v_k^j (resp., \bar{v}_k^j) est associé à un agent X_k^i (resp., \bar{X}_k^i), qui possède initialement l'objet x_k^j (resp., \bar{x}_k^j). A chaque clause C_i , correspond un agent C_i initialement doté de l'objet c_i . Un agent T possédant initialement l'objet t est ajouté, nous amenant à $|N| = |X| = m + 3n + 1$.

Le graphe $G = (N, E)$ est défini de la manière suivante.



Le profil de préférences $>$ est quant à lui présenté ci-après. Ne sont représentés, pour chaque agent, que les objets préférés à l'objet initialement détenu, et l'ensemble des objets associés aux littéraux de la clause C_i est noté $\{\ell_i\}$, où les littéraux sont ordonnés de manière arbitraire.

$$\begin{aligned}
T &: \{\ell_1\} > \boxed{t} \\
C_i &: \{\ell_{i+1}\} > t > \{\ell_i\} > c_1 > \{\ell_{i-1}\} > \dots > c_{i-1} > \{\ell_1\} > \boxed{c_i} \\
C_m &: t > \{\ell_m\} > c_1 > \{\ell_{m-1}\} > \dots > c_{m-1} > \{\ell_1\} > \boxed{c_m} \\
X_k^{p_1^k} &: c_{m-n^k+1} > c_{m-p_2^k+1} > x_k^{p_2^k} > c_{m-p_1^k+1} > \bar{x}_k^{n^k} > \boxed{x_k^{p_1^k}} \\
X_k^{p_2^k} &: c_{m-p_1^k+1} > x_k^{p_1^k} > \bar{x}_k^{n^k} > \boxed{x_k^{p_2^k}} \\
\bar{X}_k^{n^k} &: x_k^{p_2^k} > \boxed{\bar{x}_k^{n^k}}
\end{aligned}$$

A présent qu'une instance de OBJET ACCESSIBLE a été construite, on affirme que C est satisfiable si et seulement si t est atteignable pour C_m .

On peut observer dans un premier temps qu'au sein du chemin τ de T à C_m , tous les agents préfèrent t à leur objet initial et n'acceptent d'échanger t , avec leur successeur dans τ , que contre un objet correspondant à un littéral satisfaisant la clause associée au successeur. Il s'ensuit que l'unique façon d'amener t de T à C_m est de fournir à chaque agent C_i un objet associé à un littéral satisfaisant la clause C_i . Donc, pour $i = 1$ jusqu'à m , un objet de $\{\ell_i\}$ doit être amené jusqu'à C_i .

D'un autre côté, il est garanti pour chaque branche correspondant à une variable v_k que si un objet x_k^j (resp., \bar{x}_k^j) sort de la branche, alors un objet \bar{x}_k^j (resp., x_k^j) ne peut plus

en sortir par la suite. Plus précisément, si $\bar{x}_k^{n^k}$ est le premier objet à sortir de la branche, alors l'agent $X_k^{p_1^k}$ obtient son objet préféré et plus aucun autre objet ne peut sortir de la branche. Sinon, si $x_k^{p_1^k}$ ou $x_k^{p_2^k}$ bougent en premier, alors l'agent $X_k^{p_1^k}$ reçoit un objet qu'il préfère à $\bar{x}_k^{n^k}$, donc $\bar{x}_k^{n^k}$ est bloqué et ne peut plus sortir.

Supposons qu'il existe une affectation ϕ des variables satisfaisant toutes les clauses de C . Il suffit alors de choisir, pour chaque clause C_i , un littéral ℓ_i qui rend C_i vraie dans l'affectation ϕ , et d'amener l'objet correspondant jusqu'à l'agent C_i par des séquences d'échanges en procédant itérativement à partir de $i = 1$ jusqu'à m . Ces séquences d'échanges sont possibles. En effet, on peut remarquer que chaque agent C_j préfère les objets de $\{\ell_i\}$, puis c_i , puis les objets de $\{\ell_{i+1}\}$, puis c_{i+1} pour tout $i < j$, donc C_j permet bien le passage d'un objet de $\{\ell_i\}$ vers C_i pour tout $i < j$. De plus, si l'on choisit d'amener ℓ_i jusqu'à C_i , dans la branche correspondant à la variable associée à ℓ_i , si d'autres objets sont déjà sortis, ils sont forcément de la même polarité que ℓ_i puisqu'on les a choisis dans ϕ . Donc, rien ne bloque la sortie de l'objet ℓ_i de sa branche. Ainsi, les séquences d'échanges amenant jusqu'à C_i un objet associé à un littéral ℓ_i mis à vrai dans ϕ sont réalisables et permettent le passage ensuite de t jusqu'à C_m .

Supposons maintenant que t est atteignable pour C_m , et notons σ une allocation atteignable où $\sigma(C_m) = t$. On considère alors une affectation des variables notée ϕ , dans laquelle chaque littéral correspondant à un objet possédé par les agents T, \dots, C_{m-1} dans σ est mis à vrai. Comme précédemment mentionné, avant de pouvoir amener t jusqu'à C_m , chaque C_i doit obtenir un objet ℓ_i correspondant à l'un des littéraux de la clause qui lui est associée. Donc, une fois que C_m obtient t , ℓ_i est possédé par l'agent C_{i-1} ou T si $i = 1$. Par construction, les agents T, \dots, C_{m-1} ne peuvent obtenir un objet qu'ils préfèrent à leur objet courant après cela. Il suffit donc juste de vérifier qu'il n'existe pas dans ϕ deux littéraux ℓ_1 et ℓ_2 tels que $\ell_1 = \bar{\ell}_2$. Ceci est garanti par construction de chaque branche associée à une variable v_k , comme précédemment décrit. Ainsi, l'affectation ϕ construite satisfait C . ■

Le réseau de la précédente preuve est un arbre dans lequel tous les sous-arbres découlant d'une même racine sont des chemins (on l'appelle *étoile généralisée*). Néanmoins, OBJET ACCESSIBLE est résoluble en temps polynomial quand le réseau est une étoile.

Proposition 1 Lorsque G est une étoile, il existe un algorithme polynomial pour OBJET ACCESSIBLE.

Preuve : Le problème consiste à savoir si un agent donné i peut obtenir un objet donné x . Quant au réseau social, il est constitué d'un agent central noté n et de $n-1$ feuilles notées $1, \dots, n-1$. Tout troc implique donc forcément le centre et

une feuille. Une fois qu'une feuille a échangé son objet initial, elle ne peut plus intervenir dans un prochain échange (Observation 1). Ainsi, toute séquence d'échanges peut être réduite à une liste ordonnée sans répétition de feuilles, indiquant avec quel agent le centre doit échanger son objet.

Commençons par le cas où l'agent i est le centre. Le problème revient à la recherche d'un chemin dans un graphe orienté composé de l'ensemble de sommets N et de l'ensemble d'arcs décrit de la manière suivante : il existe un arc entre $a \in N$ et $b \in N \setminus \{n\}$ si et seulement si le centre et la feuille b peuvent effectuer un échange lorsque le centre possède l'objet initial de a et b a son objet initial. On déduit qu'il existe un chemin de n à b dans ce graphe orienté si et seulement si le centre peut obtenir l'objet initial de l'agent b . Un algorithme linéaire résout ce problème de chemin et la construction de ce graphe orienté est clairement polynomial.

Dans le cas où i est une feuille, le problème se réduit au précédent : le centre n doit obtenir l'objet x , et ensuite i et n échantent leurs objets. Si ces deux étapes sont réalisables, la réponse à OBJET ACCESSIBLE est positive. ■

On s'intéresse à présent au cas où le réseau est une chaîne. On suppose sans perte de généralité que $E = \{(j, j+1) : 1 \leq j < n\}$. L'Observation 1 implique, lorsque le réseau est un chemin, qu'une fois qu'un objet se déplace dans une certaine direction, alors il ne peut plus se déplacer dans la direction opposée.

On définit comme séquence d'échanges canonique $\kappa(j, k)$ une séquence d'échanges qui permet de faire glisser directement le long du chemin de j à k l'objet x_j jusqu'à l'agent k . Il s'agit donc de la séquence d'échanges entre les paires d'agents suivantes $(j, j+1), (j+1, j+2), \dots, (k-1, k)$, qui transforme σ_0 en une affectation σ où $\sigma(\ell) = x_\ell$ si $\ell < j$, $\sigma(\ell) = x_{\ell+1}$ si $j \leq \ell < k$ et $\sigma(k) = x_j$. Cette séquence est dite rationnelle si tous les échanges qui la composent le sont.

La séquence d'échanges canonique s'avère être la seule séquence d'échanges potentiellement réalisable lorsqu'il s'agit de tester l'accessibilité d'un objet pour un agent situé sur une feuille de la chaîne.

Proposition 2 *Lorsque G est une chaîne, si l'objet x_j est accessible pour n , alors la séquence d'échanges canonique $\kappa(j, n)$ est rationnelle.*

Preuve : Supposons par contradiction qu'il existe $k \in \{j+1, \dots, n\}$ tel que l'échange dans la séquence canonique entre $k-1$ et k n'est pas rationnel, c'est-à-dire $x_k >_k x_j$ ou $x_j >_{k-1} x_k$.

Dans un premier temps, supposons que $x_k >_k x_j$. S'il existe une séquence d'échanges aboutissant à affecter x_j à n , alors x_j doit être affecté une fois à l'agent k puisque le chemin entre j et n est unique. Comme $x_k >_k x_j$, l'agent k n'acceptera jamais x_j , contradiction.

Supposons à présent que $x_j >_{k-1} x_k$. S'il existe une séquence d'échanges aboutissant à l'affectation de x_j à n , alors x_j doit être échangé avec x_k à un moment donné, puisque x_j doit atteindre n et qu'il n'y a pas d'autre agent après n pour recevoir x_k . Cet échange ne peut pas se situer entre $k-1$ et k puisque $x_j >_{k-1} x_k$, et se situe donc entre des agents situés avant k . Il faut donc au préalable que les agents $k-1$ et k aient effectué un échange afin de déplacer x_k dans la direction des agents indicés inférieurement à k . À la suite de cet échange, l'objet alors détenu par $k-1$ se déplace donc dans la direction de n avant x_j , et x_j ne peut dépasser cet objet, contradiction. ■

Pour tester l'accessibilité d'un objet pour un agent situé à une feuille de la chaîne, il suffit alors de vérifier que la séquence d'échanges canonique associée est rationnelle.

Corollaire 1 *Lorsque G est une chaîne et l'agent i une feuille de G , OBJET ACCESSIBLE est résoluble en temps polynomial.*

D'autres cas polynomiaux du problème OBJET ACCESSIBLE dans une chaîne peuvent être listés. Le principal concerne le cas où la distance entre l'agent et l'objet est une constante.

Proposition 3 *Lorsque G est une chaîne et k une constante représentant la distance entre l'agent i et le possesseur de x , OBJET ACCESSIBLE est résoluble en temps polynomial.*

Preuve : Observons tout d'abord que si l'agent i peut obtenir l'objet x_j avec $i < j$, alors on peut ignorer l'agent ℓ et l'objet x_ℓ pour tout $\ell > j$. On suppose donc sans perte de généralité que le possesseur de x est une feuille du graphe. Par ailleurs, pour toute paire d'agents $j, \ell \in N$ tels que $j < \ell$, l'objet affecté à j doit avoir un indice plus petit que celui affecté à ℓ dans toute affectation atteignable à partir de σ_0 . En effet, dans le cas contraire, il s'ensuivrait qu'un objet passerait deux fois par le même agent, en se déplaçant dans un sens de la chaîne, puis dans l'autre, ce qui contredirait l'Observation 1.

L'objet x doit obligatoirement passer par tous les agents du chemin τ menant du possesseur de x dans σ_0 à l'agent i . Il faut donc que tous les agents situés sur τ préfèrent x à leur objet actuel. Mais pour faire passer l'objet x d'un agent j à son voisin ℓ sur τ , il faut que j préfère l'objet détenu par ℓ à x . Si x est accessible pour i , l'allocation finale consiste donc en une affectation σ' où $\sigma'(i) = x$ et $\sigma'(j) = y_j$ pour tout agent $j \in \tau \setminus \{i\}$ avec y_j un objet que l'agent j préfère à x . Ainsi, cela revient à choisir un ensemble de k objets parmi $X \setminus \{x\}$ afin de les affecter aux agents dans $\tau \setminus \{i\}$ pour créer une affectation finale potentiellement atteignable. Le reste des objets est affecté aux agents restants de manière à respecter la seconde observation, en suivant l'ordre croissant des indices. Il suffit alors de vérifier que l'affectation

ainsi construite est bien atteignable en fonction des préférences des agents, ce qui peut être effectué en temps polynomial (voir Proposition 4 de la section suivante). Il y a donc $\binom{n-1}{k} = O(n^k)$ telles affectations possibles, ce qui nous donne un algorithme polynomial pour OBJET ACCESSIBLE lorsque k est une constante.

Vérifions à présent la validité de l'algorithme. Si l'algorithme renvoie vrai, alors il a trouvé une affectation σ atteignable où $\sigma(i) = x$. Donc, l'accessibilité de l'objet est forcément établie. Supposons à présent que l'algorithme renvoie faux mais qu'il existe une affectation atteignable σ dans laquelle l'agent i reçoit l'objet x , c'est-à-dire $\sigma(i) = x$. Soit $X' \subseteq X \setminus \{x\}$ l'ensemble des objets affectés aux agents appartenant à $\tau \setminus \{i\}$. Puisqu'un même objet ne peut être affecté à plusieurs agents, on a bien $|X'| = k$ et chaque objet est affecté aux agents au sein de $\tau \setminus \{i\}$ par indice croissant, faute de quoi d'après la seconde observation, il y aurait un objet qui passerait deux fois par le même agent. Pour le reste de l'affectation, concernant les agents n'appartenant pas à τ , il en va de même : les objets affectés dans σ le sont forcément par ordre croissant d'indice. Ainsi, l'atteignabilité de l'affectation σ a bien été testée par l'algorithme, contradiction. ■

Malgré son apparente simplicité, OBJET ACCESSIBLE dans une chaîne est un problème ouvert quand il n'y a pas de restriction sur la position des agents. Il nous semble que ce cas se situe à la frontière entre problèmes polynomiaux et NP-difficiles.

4 Affectation atteignable

On s'intéresse dans cette section au problème de décision AFFECTATION ATTEIGNABLE : est-ce que l'affectation σ appartient à RA ? Ce problème est montré NP-complet en général mais polynomial dans le cas d'un arbre.

Théorème 2 AFFECTATION ATTEIGNABLE est NP-complet.

Preuve : On peut aisément vérifier que le problème est dans NP. On propose une réduction à partir du problème OBJET ACCESSIBLE, dont on considère une instance $I = (N, X, G, >, \sigma_0)$. Ce problème consiste à décider si un agent, appelé sans perte de généralité 1, peut obtenir un objet donné, noté x_ℓ . On construit une instance $I' = (N \cup N', X \cup X', G', >', \sigma'_0)$ de AFFECTATION ATTEIGNABLE où tout élément de N (resp., X) a une copie dans N' (resp., X'). Les copies de j et x_j sont notées respectivement j' et x'_j . L'ensemble E' des arêtes de G' contient E , mais également chaque $j' \in N'$ est relié à chaque $k \in N' \cup \{j\}$ dans E' . L'affectation initiale σ'_0 est telle que $\sigma'_0(j) = \sigma_0(j)$ si $j \in N$, et $\sigma'_0(j) = x'_j$ si $j \in N'$. Pour tout $j \in N$, $>'_j$ est constitué de x'_j en première position, suivi de $>_j$, puis des objets restants. En ce qui concerne les agents de N' , les préférences $>'$ sont définies comme suit.

$$\begin{aligned} 1' : x_\ell >' \boxed{x'_1} \\ \ell' : x_1 >' \dots >' x_n >' \boxed{x'_\ell} >' x_\ell \\ j' \notin \{1', \ell'\} : x_j >' x_{j+1} >' \dots >' x_n >' x_{j-1} >' \dots >' x_1 >' \boxed{x'_j} >' x_\ell \end{aligned}$$

On affirme que x_ℓ est atteignable pour l'agent 1 dans I si et seulement si chaque agent obtient son objet préféré dans I' .

Supposons que x_ℓ est atteignable pour l'agent 1 dans I . Par construction, c'est également le cas dans I' . Une fois que x_ℓ a été obtenu par l'agent 1, chaque agent $j \in N$ échange son objet avec sa copie $j' \in N'$. Ainsi, chaque $j \in N$ possède son objet préféré x'_j , de même que pour l'agent 1'. Puis, en considérant l'agent ℓ' et après cela tout agent j' par ordre croissant d'indices, on procède à l'échange avec l'agent parmi N' possédant son objet préféré. Ces échanges sont réalisables par construction et permettent à tous les agents de N' d'obtenir leur objet préféré.

Supposons que tous les agents obtiennent leur objet préféré dans I' . L'agent 1' doit recevoir x_ℓ après un échange avec l'agent 1 car aucun agent de $N' \setminus \{1'\}$ ne préfère x_ℓ à son objet initial. Pour la même raison, x_ℓ ne peut être obtenu par 1' par le biais des agents de $N' \setminus \{1'\}$. De plus, les préférences et la topologie du réseau imposent qu'aucun des objets de X' n'est impliqué dans le mouvement de x_ℓ jusqu'à l'agent 1. Donc, x_ℓ est bien atteignable par l'agent 1 dans I . ■

Le graphe construit dans la précédente réduction contient des cycles. Cependant, lorsque G est un arbre, un algorithme polynomial (Algorithme 1) résout AFFECTATION ATTEIGNABLE. Dans cet algorithme, la commande $pop(P)$ renvoie le premier arc du chemin P et supprime cet arc de P .

L'idée globale de l'algorithme est la suivante. Tout objet doit se déplacer le long d'un unique chemin afin d'atteindre son possesseur dans σ . Ainsi, pour que σ soit atteignable, il suffit de vérifier que les arcs des chemins que doivent suivre les objets s'intersectent bien à chaque fois en des échanges rationnels pour les agents concernés.

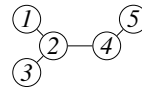
On illustre l'Algorithme 1 avec l'exemple suivant.

Exemple 3 On considère une instance avec 5 agents essayant d'atteindre l'allocation $(x_4, x_5, x_1, x_3, x_2)$. Le réseau social et les préférences des agents sont décrites comme suit.

$$\begin{array}{l} 1 : x_4 > \boxed{x_1} \\ 2 : x_5 > x_3 > x_1 > x_4 > \boxed{x_2} \\ 3 : x_1 > \boxed{x_3} \\ 4 : x_3 > x_5 > x_2 > \boxed{x_4} \\ 5 : x_2 > \boxed{x_5} \end{array}$$

Les chemins calculés à la ligne 3 de l'Algorithme 1 sont :

$$\begin{aligned} P_{x_1} &= \{(1, 2), (2, 3)\}, & P_{x_3} &= \{(3, 2), (2, 4)\}, & P_{x_5} &= \{(5, 4), (4, 2)\}, \\ P_{x_2} &= \{(2, 4), (4, 5)\}, & P_{x_4} &= \{(4, 2), (2, 1)\}. \end{aligned}$$



Algorithm 1:

Input: $(N, X, >, G, \sigma_0)$, affectation σ
Output: si σ est atteignable à partir de σ_0

- 1 $L \leftarrow \emptyset$; $\sigma' \leftarrow \sigma_0$;
- 2 **for each** $x \in X$ **do**
- 3 $P_x \leftarrow$ l'unique chemin entre le possesseur de x
 dans σ_0 et le possesseur de x dans σ ;
- 4 $L \leftarrow L \cup \{pop(P_x)\}$;
- 5 **while** $L \neq \emptyset$ **do**
- 6 **if** $\exists_{i,j \in N, i \neq j}: (i, j), (j, i) \in L$ **then**
- 7 **if** $(\sigma'(i) >_i \sigma'(j)) \vee (\sigma'(j) >_j \sigma'(i))$ **then**
- 8 **return false**;
- 9 Mettre à jour σ' avec l'échange entre i et j ;
- 10 $L \leftarrow L \setminus \{(i, j), (j, i)\}$;
- 11 $L \leftarrow L \cup \{pop(P_{\sigma'(i)})\} \cup \{pop(P_{\sigma'(j)})\}$;
- 12 **else return false**;
- 13 **return true**;

Le tableau suivant résume les différentes étapes de la boucle while (lignes 5-12) :

L	Echange	σ'
$\{(1, 2), (2, 4), (3, 2), (4, 2), (5, 4)\}$	$2 \leftrightarrow 4$	$(x_1, x_4, x_3, x_2, x_5)$
$\{(1, 2), (4, 5), (3, 2), (2, 1), (5, 4)\}$	$1 \leftrightarrow 2$	$(x_4, x_1, x_3, x_2, x_5)$
$\{(2, 3), (4, 5), (3, 2), \emptyset, (5, 4)\}$	$2 \leftrightarrow 3$	$(x_4, x_3, x_1, x_2, x_5)$
$\{\emptyset, (4, 5), (2, 4), \emptyset, (5, 4)\}$	$4 \leftrightarrow 5$	$(x_4, x_3, x_1, x_5, x_2)$
$\{\emptyset, \emptyset, (2, 4), \emptyset, (4, 2)\}$	$2 \leftrightarrow 4$	$(x_4, x_5, x_1, x_3, x_2)$
\emptyset	-	-

A chaque étape, L garde en mémoire le premier arc de chaque chemin P_{x_i} . A l'étape 1, un seul échange est possible, celui entre les agents 2 et 4. Cet échange étant rationnel, σ' est mis à jour en effectuant cet échange. Les arcs $(2, 4)$ et $(4, 2)$ sont alors supprimés de L . Les arcs $(4, 5)$ et $(2, 1)$, respectivement les nouveaux arcs des chemins P_{x_2} et P_{x_4} , sont insérés dans L . L'algorithme s'arrête quand L est vide, ce qui implique que σ est atteint.

Proposition 4 L'algorithme 1 résout AFFECTATION ATTEIGNABLE en temps polynomial lorsque G est un arbre.

Preuve : On considère une instance avec un nombre minimal d'agents pour laquelle σ est atteignable mais Algorithme 1 renvoie faux. Soit s une séquence d'échanges réalisable menant à σ à partir de σ_0 . L'allocation atteinte par l'algorithme avant de conclure à la non-atteignabilité est notée σ' . Puisque $\sigma' \neq \sigma$, certains objets n'ont pas atteint leur destination prévue au sein de l'algorithme afin d'obtenir σ . La liste L n'est donc pas vide à la fin de l'exécution. *Cas 1.* La liste L ne contient aucune paire d'arcs opposés. Commençons par un agent j tel que $\sigma(j) \neq \sigma'(j)$ et suivons les arcs de L . Puisque G est acyclique, le chemin est fini et se termine par un arc, disons (a, b) . Cela signifie que

l'objet $\sigma'(a)$ doit passer par l'agent b afin d'atteindre son propriétaire dans σ . Or, l'objet $\sigma'(b)$ a atteint sa destination, c'est-à-dire aucun arc ne sort de b car $\sigma'(b) = \sigma(b)$.

Puisqu'une unique chaîne relie deux sommets dans un graphe, on sait que dans s , l'agent a échange $\sigma'(a)$ avec un objet détenu par l'agent b . Supposons que cet objet est affecté dans σ' à un agent c . Il s'ensuit que $\sigma'(c) \neq \sigma'(b)$ et $\sigma'(c) >_a \sigma'(a)$.

Considérons la chaîne (unique) τ entre a et c dans G et supposons que $b \in \tau$. En considérant b comme la racine de G , on observe que a et c appartiennent à deux sous-arbres distincts. De plus, $\sigma'(c) \neq \sigma(c)$ et $\sigma'(a) \neq \sigma(a)$. Cela veut dire qu'en enlevant $\sigma'(a)$ ou $\sigma'(c)$, on obtient un plus petit contre-exemple à l'Algorithme 1, contradiction. Maintenant, supposons que $b \notin \tau$. Puisque $\sigma'(c)$ passe par b puis a dans s , a a reçu l'objet $\sigma'(c)$ avant d'obtenir $\sigma'(a)$, ce qui implique que $\sigma'(a) >_a \sigma'(c)$, contradiction.

Cas 2. Deux arcs opposés (a, b) et (b, a) sont présents dans L mais $\sigma'(a) >_a \sigma'(b)$. Dans s , l'agent a échange $\sigma'(a)$ avec un objet détenu par l'agent b . Supposons que cet objet est affecté dans σ' à un agent c . Il s'ensuit que $\sigma'(c) \neq \sigma'(b)$ et $\sigma'(c) >_a \sigma'(a)$. Puisque $\sigma'(c)$ passe par l'arc (b, a) dans s , si a avait reçu $\sigma'(c)$ avant d'obtenir $\sigma'(a)$, alors $\sigma'(a) >_a \sigma'(c)$, contradiction. Mais si a n'a pas reçu $\sigma'(c)$ avant d'obtenir $\sigma'(a)$, alors $\sigma'(c)$, ainsi que $\sigma'(a)$ et $\sigma'(b)$, n'ont pas atteint leur destination. Un contre-exemple plus petit existe donc, contradiction. ■

5 Allocations Pareto-efficaces

La question soulevée dans cette section est celle de la manière dont un coordinateur externe pourrait coordonner les échanges entre les agents afin d'obtenir une allocation Pareto-efficace sur l'ensemble des allocations atteignables RA. Notons tout d'abord qu'une telle allocation est forcément stable, sinon au moins deux agents pourraient obtenir un meilleur objet en effectuant un échange. Cependant, la réciproque est fautive, comme on peut le remarquer à travers l'Exemple 2 avec l'allocation σ_1 qui est stable mais Pareto-dominée par une autre allocation atteignable.

Proposition 5 PARETO est NP-difficile.

Preuve : La réduction est la même que celle exhibée dans la preuve du Théorème 2. L'instance \mathcal{I} est une instance positive si et seulement si tous les agents obtiennent leur objet préféré dans \mathcal{I}' , qui est donc l'unique allocation Pareto-efficace. Il s'ensuit qu'un algorithme calculant une allocation Pareto-efficace pourrait être utilisé pour reconnaître une instance positive du problème OBJET ACCESSIBLE. ■

Ce résultat négatif n'empêche cependant pas l'existence d'un algorithme polynomial pour construire une allocation Pareto-efficace dans des classes spécifiques d'instances. Le

reste de la section est consacré à la résolution de PARETO dans une chaîne et dans une étoile.

Un algorithme classique pour obtenir la Pareto-optimalité de manière centralisée est celui du *Serial Dictatorship* ([1]). Il s'agit de ranger les agents dans un certain ordre et de leur assigner tour à tour leur objet préféré parmi les objets encore disponibles, jusqu'à ce que tous les objets soient affectés. Lorsque le réseau est une chaîne, il est possible de réexploiter cette idée afin de guider les échanges des agents le long du réseau pour obtenir une allocation Pareto-efficace au sein de RA.

Algorithm 2:

Input: $(N, X, >, G, \sigma_0)$, indice k

Output: affectation σ

```

1 if  $k = 1$  then
2   | return  $\sigma_0$ ;
3 else
4   |  $x_i \leftarrow$  objet préféré accessible pour  $k$ ;
5   |  $\sigma \leftarrow$  effectuer  $\kappa(i, k)$  sur  $\sigma_0$ ;
6   | return Algorithme 2  $((N, X, >, G, \sigma), k - 1)$ ;

```

Le paramètre k de l'Algorithme 2 renvoie au dictateur qui choisit son objet préféré accessible, et $\kappa(i, k)$ est la séquence d'échanges canonique (voir Proposition 2).

Proposition 6 *Lorsque G est une chaîne, l'Algorithme 2 avec $k = n$ renvoie une allocation Pareto-efficace parmi RA en temps polynomial.*

Preuve : L'algorithme commence avec une feuille de la chaîne, et modifie l'allocation courante de façon à ce que cet agent feuille obtienne son objet préféré (le Corollaire 1 est utilisé), puis continue sur la sous-chaîne entre $k - 1$ et 1. La preuve est par induction sur k : quand la décision pour l'agent k est effectuée, l'allocation partielle pour les agents $\{k + 1, \dots, n\}$ est Pareto-efficace. ■

On étudie à présent le cas de l'étoile. Le réseau est constitué d'un agent central n et de $n - 1$ feuilles $1, \dots, n - 1$ (voir Exemple 2). On suppose sans perte de généralité que $\forall i < n, x_i >_n x_{i+1}$. En effet, si $x_n >_n x_j$ pour un certain j , alors le centre ne troquera jamais son objet avec j , donc j gardera son objet dans toute séquence d'échanges. L'algorithme est assez simple : pour $i = n - 1$ jusqu'à 1, on pratique l'échange entre n et i si celui-ci est rationnel.

Proposition 7 *Lorsque G est une étoile, il existe un algorithme linéaire pour PARETO.*

Preuve : Comme déjà mentionné, une feuille qui a déjà échangé son objet avec le centre, ne peut plus intervenir dans un autre échange. L'algorithme considère les objets par ordre croissant des préférences de l'agent central (du

moins préféré au préféré). Supposons par contradiction que l'allocation σ renvoyée par l'algorithme est dominée par une autre allocation atteignable σ' . Notons s et s' les séquences d'échanges aboutissant respectivement à σ et σ' . L'argument principal repose sur le fait que toute séquence d'échanges réalisable est effectuée par ordre décroissant des indices des feuilles. Au premier échange où s et s' diffèrent, le centre échange son objet respectivement avec ℓ et ℓ' . Puisque $\ell < \ell'$, ℓ et n ne peuvent plus échanger leurs objets dans s' . Ainsi, $\sigma(\ell) >_\ell \sigma'(\ell)$, contradiction. ■

Il apparaît intéressant de regarder si PARETO est résoluble en temps polynomial dans une étoile généralisée par une combinaison de techniques utilisées dans la résolution des cas de la chaîne et de l'étoile.

6 Conclusion

Des problèmes naturels soulevés lorsque des agents échantent leurs objets le long d'un réseau social ont été étudiés. Nos résultats montrent qu'au-delà des préférences des agents, le réseau peut grandement influencer et contraindre les allocations possibles. En particulier, nous avons prouvé que décider si un agent peut obtenir un objet donné (OBJET ACCESSIBLE) est difficile d'un point de vue computationnel, même lorsque le réseau social est un arbre. Néanmoins, un algorithme efficace peut déterminer si une allocation complète (AFFECTATION ATTEIGNABLE) est atteignable dans un arbre. En ce qui concerne des structures simples de graphe comme des chaînes, nous pouvons décider si un agent situé sur une feuille du réseau peut acquérir un objet donné. Ce résultat peut être étendu au cas où la distance entre un agent et la position initiale d'un objet est borné par une constante. Nous avons laissé ouverte la question de savoir si OBJET ACCESSIBLE peut être résolu efficacement dans une chaîne, sans restriction sur la position de l'agent.

Dans cet article, la qualité d'un point de vue social des allocations atteignables a été examinée par le biais de la recherche d'allocations Pareto-efficaces. Ce problème est montré difficile en général. Cependant, des algorithmes polynomiaux ont été exhibés pour des chaînes et des étoiles. La complexité de construction d'une allocation Pareto-efficace dans un arbre apparaît comme un problème intéressant pour la suite.

Comme perspective, de nombreux aspects de ce modèle méritent d'être explorés. Par exemple, l'impact d'un comportement stratégique de la part des agents n'a pas été étudié. Raisonner de façon stratégique peut conduire un agent à refuser un échange profitable dans l'immédiat pour avoir une vision à plus long terme. Il apparaît également important de se pencher sur le bien-être social des allocations atteignables. Au-delà de la Pareto-optimalité, est-il difficile de maximiser le bien-être social égalitaire ou utilitaire ?

Dans l'idée des prix d'anarchie et de stabilité, à quel point une allocation stable peut-elle être mauvaise, en comparaison d'une allocation obtenue suite à des échanges non contraints par un réseau social ? Une autre piste de travail est d'autoriser plus de deux agents à effectuer des échanges le long d'un réseau social et d'analyser quelles allocations peuvent émerger. En outre, notre travail ne suppose aucune restriction sur le domaine des préférences des agents (par exemple *single peakedness*), ce qui pourrait également constituer une extension intéressante.

Références

- [1] Abdulkadiroğlu, A. et T. Sönmez: *Random Serial Dictatorship and the Core from Random Endowments in House Allocation Problems*. *Econometrica*, 66(3) :689–701, 1998.
- [2] Abdulkadiroğlu, A. et T. Sönmez: *House Allocation with Existing Tenants*. *Journal of Economic Theory*, 88(2) :233–260, 1999.
- [3] Abraham, D. J., K. Cechlárová, D. Manlove et K. Mehlhorn: *Pareto Optimality in House Allocation Problems*. Dans *Proceedings of the 16th International Symposium on Algorithms and Computation (ISAAC)*, pages 1163–1175, 2005.
- [4] Anshelevich, E., O. Bhardwaj et M. Hoefer: *Friendship and Stable Matching*. Dans *Proceedings of the 21st European Symposium on Algorithms (ESA)*, pages 49–60, 2013.
- [5] Arcaute, E. et S. Vassilvitskii: *Social Networks and Stable Matchings in the Job Market*. Dans *Proceedings of the 5th International Workshop on Internet and Network Economics (WINE)*, pages 220–231, 2009.
- [6] Aziz, H., P. Biró, J. Lang, J. Lesca et J. Monnot: *Optimal Reallocation under Additive and Ordinal Preferences*. Dans *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 402–410, 2016.
- [7] Aziz, H. et B. De Keijzer: *Housing Markets with Indifferences : A Tale of Two Mechanisms*. Dans *Proceedings of the 26th Conference on Artificial Intelligence (AAAI)*, pages 1249–1255, 2012.
- [8] Bodine-Baron, E., C. Lee, A. Chong, B. Hassibi et A. Wierman: *Peer Effects and Stability in Matching Markets*. Dans *Proceedings of the 4th International Symposium on Algorithmic Game Theory (SAGT)*, pages 117–129, 2011.
- [9] Chevaleyre, Y., P. E. Dunne, U. Endriss, J. Lang, M. Lemaître, N. Maudet, J. Padget, S. Phelps, J. A. Rodriguez-Aguilar et P. Sousa: *Issues in Multiagent Resource Allocation*. *Informatica*, 30(1) :3–31, 2006.
- [10] Chevaleyre, Y., U. Endriss, S. Estivie et N. Maudet: *Multiagent Resource Allocation in k -additive Domains : Preference Representation and Complexity*. *Annals of Operations Research*, 163(1) :49–62, 2008.
- [11] Chevaleyre, Y., U. Endriss, J. Lang et N. Maudet: *Negotiating over Small Bundles of Resources*. Dans *Proceedings of the 4th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 296–302, 2005.
- [12] Chevaleyre, Y., U. Endriss et N. Maudet: *Allocating Goods on a Graph to Eliminate Envy*. Dans *Proceedings of the 22nd Conference on Artificial Intelligence (AAAI)*, pages 700–705, 2007.
- [13] Damamme, A., A. Beynier, Y. Chevaleyre et N. Maudet: *The Power of Swap Deals in Distributed Resource Allocation*. Dans *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 625–633, 2015.
- [14] Dunne, P. E. et Y. Chevaleyre: *The Complexity of Deciding Reachability Properties of Distributed Negotiation Schemes*. *Theoretical Computer Science*, 396(1-3) :113–144, 2008.
- [15] Dunne, P. E., M. Wooldridge et M. Laurence: *The Complexity of Contract Negotiation*. *Artificial Intelligence*, 164(1-2) :23–46, 2005.
- [16] Easley, D. et J. Kleinberg: *Networks, Crowds, and Markets : Reasoning about a Highly Connected World*. Cambridge University Press, 2010.
- [17] Endriss, U., N. Maudet, F. Sadri et F. Toni: *Negotiating Socially Optimal Allocations of Resources*. *Journal of Artificial Intelligence Research*, 25 :315–348, 2006.
- [18] Gardenfors, P.: *Assignment Problem Based on Ordinal Preferences*. *Management Science*, 20(3) :331–340, 1973.
- [19] Hoefer, M.: *Local Matching Dynamics in Social Networks*. *Information and Computation*, 222 :20–35, 2013.
- [20] Jackson, M. O.: *Social and Economic Networks*. Princeton University Press, 2008.
- [21] Klaus, B., D. F. Manlove et Rossi F.: *Matching under Preferences*. Dans *Handbook of Computational Social Choice*, pages 333–355. Cambridge University Press, 2016.
- [22] Manlove, D. F.: *Algorithmics of Matching under Preferences*, tome 2. World Scientific, 2013.
- [23] Sandholm, T.: *Contract Types for Satisficing Task Allocation*. Dans *Proceedings of the AAAI Spring Symposium*, pages 23–25, 1998.
- [24] Shapley, L. et H. Scarf: *On Cores and Indivisibility*. *Journal of Mathematical Economics*, 1(1) :23–37, 1974.

- [25] Yoshinaka, R.: *Higher-order Matching in the Linear Lambda Calculus in the Absence of Constants is NP-complete*. Dans *Proceedings of the 16th International Conference on Rewriting Techniques and Applications (RTA)*, pages 235–249, 2005.

TouIST again. . .

(formalisez et résolvez facilement des problèmes avec des solveurs SAT, SMT et QBF)

Olivier Gasquet[†], Andreas Herzig[‡], Dominique Longin[‡], Frédéric Maris[†], Maël Valais[†]

IRIT (Institut de Recherche en Informatique de Toulouse)

[‡]CNRS, [†]Université Paul Sabatier, Toulouse, France

{Olivier.Gasquet,Andreas.Herzig,Dominique.Longin,Frederic.Maris,Mael.Valais}@irit.fr

1 Introduction

Depuis 2010, nous développons TouIST¹, un logiciel dédié à la logique propositionnelle dont les principales fonctionnalités sont (1) d’offrir un langage logique de haut niveau pour exprimer succinctement des formules complexes et (2) de trouver des modèles à ces formules en utilisant un solveur SAT performant.

Dans ce qui suit, nous présentons une extension de TouIST à QBF (*Quantified Boolean Formulas*) au travers d’un exemple : le jeu de Nim.

Tout d’abord, nous survolons succinctement les principales caractéristiques de TouIST (Section 2) et montrons comment modéliser le jeu de Nim dans le langage d’entrée de TouIST (Section 3). Enfin, après une brève présentation de QBF nous montrons comment modéliser la recherche d’une stratégie gagnante dans TouIST pour ce jeu (Section 4).

2 Présentation générale de TouIST

Dès le départ, nos exigences étaient que l’outil devait : être très facile à installer et à utiliser, sans syntaxe complexe ; pouvoir être utilisé comme une boîte noire sans connaître son fonctionnement interne (éditeur intégré de formules traduites automatiquement et de manière transparente dans le langage d’entrée du solveur considéré) ; ne requérir aucune mise en forme normale, ordonnancement de clauses, *etc.* ; ne nécessiter qu’un minimum de connaissances en logique.

Avec TouIST, les étudiants peuvent expérimenter par eux-même qu’un langage logique n’est pas seule-

ment descriptif mais peut aussi conduire à des calculs qui résolvent des problèmes concrets. En particulier, TouIST leur permet de résoudre des Sudokus assez facilement, ainsi que beaucoup d’autres problèmes combinatoires (emplois du temps, coloration de carte, circuits électroniques, *etc.*).

Voici les principales facilités qu’offre TouIST :

- les formules entrées n’ont pas besoin d’être sous forme clausale et des connecteurs arbitraires peuvent être utilisés, la mise sous forme normale est faite dynamiquement pendant la saisie au clavier de l’utilisateur ;
- facilités d’utilisation de conjonctions ou disjonctions indicées comme :

$$\bigwedge_{i \in \{1..9\}} \bigvee_{j \in \{1..9\}} \bigwedge_{n \in \{1..9\}} \bigwedge_{m \in \{1..9\}, m \neq n} (p_{i,j,n} \rightarrow \neg p_{i,j,m})$$

exprimant qu’une case de coordonnées (i, j) contient au plus un nombre compris entre 1 et 9.

- plusieurs solveurs sont disponibles (SAT, QF_LRA, QF_LIA, QF_RDL, QF_IDL et QBF) et le langage admissible par TouIST s’adapte facilement ;
- définition d’ensembles de domaines : $\bigwedge_{i \in A}$ vs. $\bigwedge_{i \in \{Paris, London, Roma, Madrid\}}$
- liaisons multiples sur les indices : $\bigwedge_{i \in A, j \in B}$ vs. $\bigwedge_{i \in \dots} \bigwedge_{j \in \dots}$
- calculs riches sur les indices ainsi que sur les ensembles de domaines : $\bigwedge_{i \in (A \cup (B \cap C))}$
- primitives de contraintes de cardinalité : *au moins, au plus, exactement, n valeurs sont vraies parmi un ensemble de valeurs données, etc.*
- prédicats pouvant être des variables définies sur des ensembles de domaines : $\bigwedge_{X \in \{A, B\}, i \in \{1, 2\}} X(i)$ vs. $\bigwedge_{i \in \{1, 2\}} (A(i) \wedge B(i))$

1. Historiquement, TouIST est le successeur de SATTOULOUSE, présenté pour la première fois lors de la conférence ICT-TL’2011 [2].

- littéraux spéciaux définissant des contraintes entre nombres entiers ou réels : $(x + y \leq z)$
- parcours facile des modèles successivement calculés par les solveurs
- expressions régulières permettant un filtrage des littéraux pertinents
- possibilité d'utiliser le logiciel en ligne de commande et/ou batch
- nombreuses fonctionnalités d'édition et améliorations

Ainsi, il est possible de montrer la puissance de la logique propositionnelle à des étudiants qui ont été formés quelques heures à la formalisation de phrases en logique et qui ont acquis les notions de bases de validité et satisfiabilité pour résoudre automatiquement des Sudokus par exemple.

Une présentation plus complète de TOUIST peut être trouvée dans [4]. TOUIST est téléchargeable à l'adresse <https://www.irit.fr/touist>. Cette page donne aussi accès au manuel complet du langage et de l'utilisation du logiciel.

3 Description du jeu de Nim

Le principe du jeu de Nim est le suivant : on dispose au départ d'un nombre NA non nul d'allumettes et un nombre NJ de joueurs peuvent prendre 1 ou plusieurs allumette(s). Le joueur qui perd est celui qui, le premier, ne peut plus prendre d'allumette.² Le nombre de tours de jeu possibles est au plus égal à celui des allumettes (à minima, chaque joueur ne prend à chaque tour qu'une seule allumette). Ainsi, l'ensemble des indices des tours possibles est $T = \{0, \dots, NA\}$ où 0 est l'indice de l'état initial. De même, l'ensemble des nombres possibles d'allumettes encore disponibles est $A = \{0, 1, \dots, NA\}$.

Afin de simplifier au maximum le langage utilisé, nous modélisons ici une variante où $NA = 4$ et $NJ = 2$. Les joueurs sont notés 0 et 1 et c'est au tour de 0 de jouer au tour t ssi $tour_de_0(t)$ est vrai (considérant que si ce n'est pas le tour de 0 alors c'est celui de 1). De plus, $reste(t, n)$ est vrai ssi au tour t il reste n allumettes.

Ainsi, l'état initial du jeu est le suivant :

$$reste(0, NA) \wedge tour_de_0(0) \quad (1)$$

indiquant qu'au tour 0 il y a encore NA de disponible et que c'est au tour de 0 de jouer.

2. Il existe différentes variantes de ce jeu, notamment en faisant varier les nombres d'allumettes et de joueurs, mais également en faisant varier les actions possibles ou en introduisant des contraintes (par exemple, on ne peut pas prendre le même nombre d'allumettes que le joueur précédent).

Dans la version présentée ici du jeu de Nim, on limite également le nombre des actions possibles à deux : un joueur peut prendre soit 1 allumette, soit 2 allumettes. Ainsi, $prend(t, 2)$ est vrai ssi un agent prend 2 allumettes au tour t (considérant ainsi que $prend(t, 2)$ est faux ssi il n'en prend qu'une).

Ainsi :

$$\bigwedge_{\substack{t \in T \\ n \in A \\ n \geq 2}} \left(\left(reste(t, n) \wedge prend(t, 2) \rightarrow \right. \right. \\ \left. \left. reste(t+1, n-2) \right) \wedge \right. \quad (2) \\ \left. \left(reste(t, n) \wedge \neg prend(t, 2) \rightarrow \right. \right. \\ \left. \left. reste(t+1, n-1) \right) \right)$$

capture le fait que si au tour t il reste au moins 2 allumettes et qu'un joueur en prend 2 alors au tour suivant il en reste 2 de moins, et que si il n'en prend qu'une alors au tour suivant il en reste 1 de moins.

En revanche, si au tour t il reste exactement 1 allumette, alors nécessairement le joueur en prendra 1 et il en restera 0 au tour suivant :

$$\bigwedge_{t \in T} \left(reste(t, 1) \rightarrow \neg prend(t, 2) \wedge reste(t+1, 0) \right) \quad (3)$$

Notre modèle spécifie ensuite que :

$$\bigwedge_{t \in T} \bigvee_{n \in A} reste(t, n) \quad (4)$$

$$\bigwedge_{\substack{t \in T \\ n1, n2 \in A \\ n1 \neq n2}} \left(reste(t, n1) \rightarrow \neg reste(t, n2) \right) \quad (5)$$

La première formule stipule qu'à chaque tour t il existe au moins un nombre n d'allumettes restant, et la seconde que ce nombre est unique.

Il faut maintenant définir quand un joueur a perdu :

$$0_perd \leftrightarrow \bigvee_{\substack{t \in T \\ t > 0}} \left(tour_de_0(t) \wedge reste(t, 0) \right) \quad (6)$$

signifie que le jouer 0 a perdu ssi il existe un tour t où il reste 0 allumettes alors qu'à l'instant d'avant il y en avait au moins une.

Finalement, à chaque tour t , ce n'est pas au joueur 0 de jouer ssi c'est à lui de jouer au tour suivant :

$$\bigwedge_{t \in T \setminus \{NA\}} \left(\neg tour_de_0(t) \leftrightarrow tour_de_0(t+1) \right) \quad (7)$$

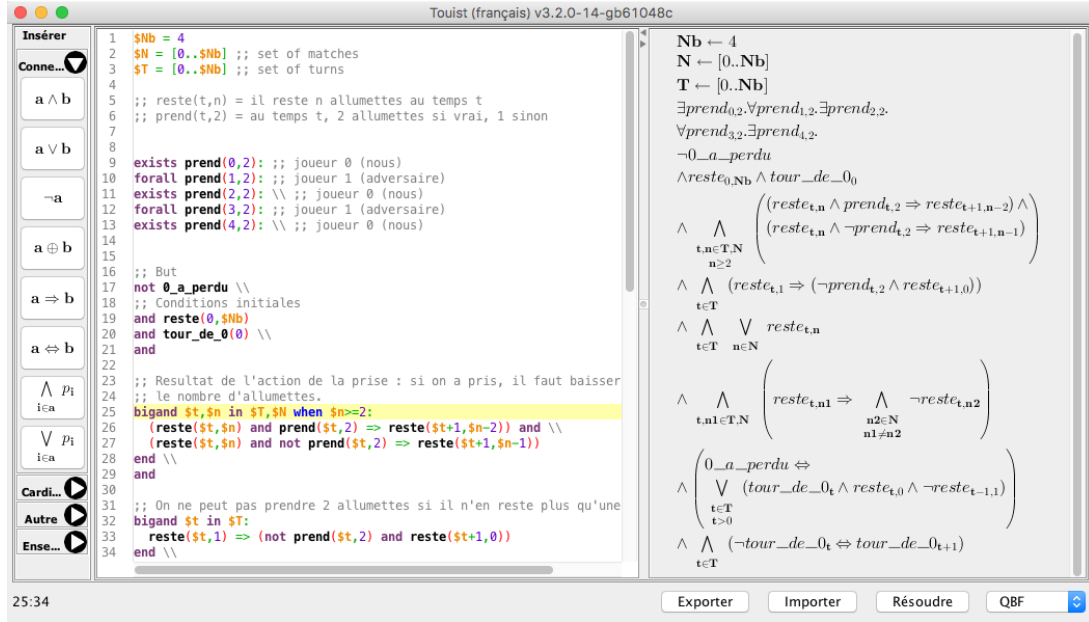


FIGURE 1 – Capture d’écran de TOUIST avec le jeu de Nim. Le fichier est disponible à l’adresse <https://github.com/maelvalais/allumettes>

4 Formalisation d’une stratégie gagnante à l’aide de QBF

Dans cette section, nous allons présenter à l’aide de notre exemple du jeu de Nim l’extension de TOUIST à Quantified Boolean Formulas (QBF) connu comme étant le problème de référence pour la classe de complexité PSPACE ([5]). C’est une extension de la logique propositionnelle permettant de quantifier sur les variables propositionnelles.

Par exemple, $\forall p \exists q. p \leftrightarrow q$ se lit : pour toute valeur de vérité de p , il existe une valeur de vérité de q tel que $p \leftrightarrow q$ est vrai. Cette formule est vraie (il suffit de choisir la même valeur pour q que pour p). Alors que $\exists p \forall q. p \vee q$ ne l’est pas. Ainsi, une formule booléenne quantifiée est toujours SOIT vraie SOIT fausse.

De fait, à toute formule QBF peut être associée une formule propositionnelle sans variables car par définition : $\forall p. \Phi$ est vraie ssi $\Phi_{[p:=\top]} \wedge \Phi_{[p:=\perp]}$ l’est, et $\exists p. \Phi$ est vraie ssi $\Phi_{[p:=\top]} \vee \Phi_{[p:=\perp]}$.

La formule QBF peut être exponentiellement plus compacte que la formule propositionnelle correspondante.

Par exemple à la formule $\forall p \exists q. p \leftrightarrow q$ correspond la formule propositionnelle

$$((\top \leftrightarrow \top) \vee (\top \leftrightarrow \perp)) \wedge ((\perp \leftrightarrow \top) \vee (\perp \leftrightarrow \perp))$$

Le langage de QBF permet d’exprimer naturellement et de manière concise l’existence de stratégies ga-

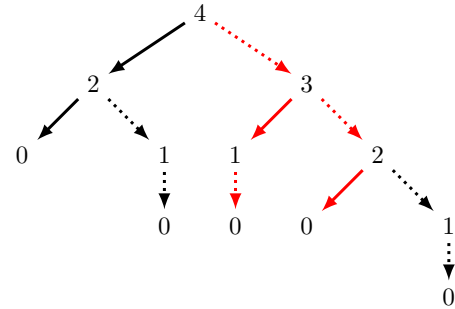


FIGURE 2 – Solutions pour le jeu de Nim (4 all./2 joueurs), en rouge : stratégie gagnante du joueur 0

gnantes ainsi que décrit dans [3]. Les coups du joueur 0 (pour lequel on cherche une stratégie gagnante) seront existentiellement quantifiés alors que ceux de son adversaire seront universellement quantifiés. (On cherche les coups du joueur 0 qui le mèneront à la victoire quels que soient les coups joués par le joueur 1.)

TOUIST a été étendu pour être compatible avec le solveur QBF *Quantor 3.2* [1]. La sélection de ce prouveur dans TOUIST autorise *de facto* l’utilisation des quantificateurs \forall et \exists (respectivement définis par **exists** et **forall** dans TOUIST).

FIGURE 2 présente l’ensemble exhaustif des solutions de notre exemple. La racine de l’arbre représente le nombre initial d’allumettes, et chaque flèche l’action de retirer 1 (.....►) ou 2 (————►) allumette(s). Au

bout de la flèche, le nombre d'allumettes après exécution de l'action concernée. D'après cette figure, on voit que si le joueur 0 commence (ce qui est imposé par (1)) et qu'il retire une seule allumette (il en reste donc 3) on voit qu'il a une stratégie gagnante :

- si le joueur 1 retire ensuite 2 allumettes il en restera 1 seule que le joueur 0 peut retirer pour gagner (puisque le joueur 1 ne pourra ensuite plus retirer d'allumette);
- si le joueur 1 retire une seule allumette, il en restera 2 et le joueur 0 pourra au coup suivant les retirer en un seul coup et le joueur 1 perd.

Nous tirons parti de QBF pour écrire cette stratégie dans TOUIST. Si on note Φ la conjonction des formules (1) à (7) alors la recherche d'une stratégie gagnante pour le joueur 0 s'écrit simplement :

$$\begin{aligned} & \exists \text{prend}(0, 2) \forall \text{prend}(1, 2) \\ & \quad \exists \text{prend}(2, 2) \forall \text{prend}(3, 2) \\ & \quad \exists \text{prend}(4, 2) . \neg 0_perd \wedge \Phi \end{aligned} \quad (8)$$

Autrement dit, on cherche à satisfaire le fait qu'il existe une action du joueur 0 au tour 0 telle que quelle que soit l'action du joueur 1 au tour 1, il existe une action du joueur 0 au tour 2, telle que pour toute action du joueur 1 au tour 3 il existe une action du joueur 0 (qui sera donc le dernier à jouer) telle que le joueur 0 ne perd pas et que les contraintes inhérentes au jeu de Nim soient satisfaites.

L'exécution du programme dans TOUIST indique que cette formule est vraie, ce qui signifie l'existence d'une stratégie gagnante pour le joueur 0. Le solveur retourne la valeur des (ici une seule) variables existentielles correspondant au prochain coup du joueur 0. À ce stade, le joueur adverse doit fournir son coup qui fixe la valeur des variables universelles correspondant à ses possibles prochains coups. On exécute alors de nouveau le programme modifié de la façon suivante (de manière à prendre en compte le calcul de la valuation de $\text{prend}(0, 2)$) :

$$\begin{aligned} & \exists \text{prend}(0, 2) \exists \text{prend}(1, 2) \\ & \quad \exists \text{prend}(2, 2) \forall \text{prend}(3, 2) \\ & \quad \exists \text{prend}(4, 2) . \neg 0_perd \wedge c_0 \wedge c_1 \wedge \Phi \end{aligned}$$

où c_0 est soit $\text{prend}(0, 2)$ soit $\neg \text{prend}(0, 2)$ en fonction du coup du joueur 0, et similairement pour c_1 en fonction du coup choisi par l'adversaire. La situation après ces deux coups est la nouvelle situation initiale pour le solveur et la recherche du coup suivant du joueur 0... jusqu'à sa victoire! On réitère ce processus jusqu'à ce que toutes les variables aient reçu une valeur.

5 Conclusion

TOUIST peut être vu comme un compilateur de langages logiques étendus et de haut niveau vers des prouveurs efficaces indépendants. Ces deux facettes lui confèrent une grande facilité d'utilisation, un large spectre d'application et de bonnes performances calculatoires. À ce titre, il constitue un outil complètement original et unique en son genre.

Nous l'utilisons dans le cadre du cours d'initiation à la logique en licence d'informatique, mais aussi en master, dans le cadre de travaux pratiques et de projets. Les étudiants sont ainsi appelés à parcourir tout le processus allant de la formalisation à la résolution de problèmes qui vont au-delà des problèmes-jouets faisables sur papier.

Mais plus encore, TOUIST est d'ores et déjà utilisé par des chercheurs dans le cadre de travaux menés au sein de notre laboratoire et impliquant une modélisation logique (planification, raisonnement épistémique via traduction en QBF, ...), il comble un manque existant au sein des logiciels de calculs formels comme Maple, SageMath, Mathematica ou Maxima qui n'intègrent qu'anecdotiquement des outils logiques.

Références

- [1] Armin Biere. Resolve and expand. In *Proceedings of the 7th International Conference on Theory and Applications of Satisfiability Testing, SAT'04*, pages 59–70, Berlin, Heidelberg, 2005. Springer-Verlag.
- [2] Olivier Gasquet, François Schwarzentruher, and Martin Strecker. Satoulouse : The computational power of propositional logic shown to beginners. In Patrick Blackburn, Hans van Ditmarsch, María Manzano, and Fernando Soler-Toscano, editors, *Tools for Teaching Logic : Third International Congress, TICTTL 2011, Salamanca, Spain, June 1-4, 2011. Proceedings*, pages 77–84. Springer, Berlin, Heidelberg, 2011.
- [3] Daniel Kroening and Ofer Strichman. *Decision Procedures - An Algorithmic Point of View, Second Edition*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2016.
- [4] Khaled Skander Ben Slimane, Alexis Comte, Olivier Gasquet, Abdelwahab Heba, Olivier Lezaud, Frederic Maris, and Mael Valais. Twist your logic with touist. *CoRR*, abs/1507.03663, 2015.
- [5] L. J. Stockmeyer and A. R. Meyer. Word problems requiring exponential time (preliminary report). In *Proceedings of the Fifth Annual ACM Symposium on Theory of Computing, STOC '73*, pages 1–9, New York, NY, USA, 1973. ACM.