

Table des matières

Introduction	9
1 Préliminaires	13
1.1 Ensembles et relations d'ordre	13
1.1.1 Ensembles	13
1.1.2 Relations, ordres et pré-ordres	13
1.2 Logique propositionnelle	14
1.2.1 Aspect sémantique	14
1.2.2 Aspect axiomatique	16
1.2.3 Principales propriétés de la logique propositionnelle	17
1.2.4 Les méthodes énumératives et le problème SAT	18
1.2.5 La résolution	22
1.3 Logique des prédicats	23
1.3.1 Variables liées, variables libres et formules closes	23
1.3.2 Aspects sémantiques et syntaxiques	23
1.4 La logique possibiliste	26
1.4.1 Base de croyances et distribution de possibilité	27
I Etat de l'art	31
2 La programmation logique	33
2.1 PROLOG	33
2.1.1 Les faits	34
2.1.2 Les règles	36
2.1.3 Les questions	37
2.1.4 Négation par échec	37
2.1.5 Les sémantiques de PROLOG	37
2.2 Programmes logiques	39
2.2.1 Programme logique basique	40
2.2.2 Dénomination des programmes logiques	40
2.3 Sémantique bien-fondée	40
2.3.1 Ensemble non-fondé	40

2.3.2	Modèle partiel bien-fondé	41
2.3.3	Modèle bien-fondé	42
2.4	Sémantique des modèles stables	42
2.4.1	Définition des modèles stables	42
2.4.2	Une définition alternative des modèles stables	43
2.4.3	Différentes utilisations de programme ASP	45
2.5	Langage ASP et solveurs	45
2.5.1	Langage ASP et instructions supplémentaires	46
2.6	Les préférences dans un programme logique	48
2.6.1	Théorie et définition	48
2.7	Solveurs ASP	51
2.7.1	Lparse	51
2.7.2	Smodels	51
2.7.3	Kernel Normal Form	55
2.7.4	CLASP	59
2.8	Genérer et tester	60
2.9	Programmation logique possibiliste	61
3	Changement de croyances	63
3.1	Révision	64
3.1.1	Changement de théories	64
3.1.2	Approche sémantique	66
3.1.3	Opérateurs de révision syntaxique	68
3.1.4	Révision itérée	70
3.2	Fusion de croyances	72
3.2.1	Flock et profil de croyances	72
3.2.2	La question des contraintes d'intégrité	72
3.2.3	Postulats	73
3.2.4	Arbitrage et majorité	74
3.2.5	Opérateurs	75
3.2.6	Fusion réversible	81
3.2.7	Fusion possibiliste	84
3.2.8	Fusion de bases de croyances exprimées en logique possibiliste . . .	85
3.2.9	Résultat d'équivalence	86
II	Fusion par R-ensembles	89
4	Fusion par R-ensembles	93
4.1	Principes et définitions	93
4.1.1	Fusion pure par R-ensembles	94
4.1.2	Fusion avec contraintes	96
4.1.3	Réécriture des postulats de Konieczny-Pino Perez	98
4.2	Différents opérateurs de fusion syntaxique	98

4.2.1	<i>Somme</i>	99
4.2.2	<i>Card</i>	104
4.2.3	<i>Max</i>	106
4.2.4	<i>GMax</i>	109
4.3	Contrepartie sémantique	110
4.3.1	Contrepartie sémantique de l'opérateur <i>Somme</i>	112
4.3.2	Contrepartie sémantique de l'opérateur <i>Card</i>	113
4.3.3	Contrepartie sémantique de l'opérateur <i>Max</i>	114
4.3.4	Contrepartie sémantique de l'opérateur <i>GMax</i>	115
4.4	Généralisation de la Révision par R-ensembles	116
5	Mise en œuvre	119
5.1	Traduction en programme logique avec sémantique des modèles stables	119
5.1.1	Première étape : génération des interprétations	120
5.1.2	Deuxième étape : détermination des R-ensembles potentiels correspondants	120
5.1.3	Troisième étape : génération des R-ensembles restants	122
5.1.4	Quatrième étape : Représentation des contraintes <i>IC</i>	123
5.2	Équivalence entre modèle stable et R-ensemble potentiel	123
5.3	Définition des modèles stables préférés	126
5.4	Équivalence entre modèles stables préférés et R-ensembles	127
6	Le calcul des R-ensembles	131
6.1	Adaptation de l'algorithme Smodels	131
6.2	L'heuristique de choix	132
6.3	Le nouvel algorithme	132
6.4	Sélection Off The Shelf	133
6.4.1	Stratégie <i>Somme</i>	133
6.4.2	Stratégie <i>Card</i>	134
6.4.3	Stratégie <i>Max</i>	135
6.4.4	Stratégie <i>GMax</i>	137
7	Expérimentations	141
7.1	Expérimentations sur des bases aléatoires	141
7.1.1	Stratégie <i>Somme</i>	142
7.1.2	Stratégie <i>Max</i>	142
7.1.3	Comparaison avec la méthode de N. Gorogiannis et A. Hunter	144
7.2	Gestion de l'incohérence sur le site de Pianosa	146
7.2.1	Expérimentations et résultats	156
8	Fusion avec préférences	159
8.1	Préférences externes uniquement	159
8.1.1	Fusion classique avec opérateur lexicographique	160
8.1.2	Fusion comme révision itérée	161

8.1.3	Mise en œuvre de la stratégie <i>lexipref</i>	164
8.2	Préférences externes et préférences internes	166
9	Fusion de programmes logiques	171
9.1	Première approche : opération forte de fusion par R-ensembles	172
9.1.1	Relation d'inférence entre programmes logiques	174
9.2	Deuxième approche : fusion faible par R-ensembles	181
9.2.1	Operation de fusion faible et postulats IC	184
	Conclusion	188
	Bibliographie	195

Introduction

John McCarthy, un des pères fondateurs de l'Intelligence Artificielle, a défini celle-ci comme étant :

"La science et l'ingénierie de la création de machines intelligentes."

L'acquisition et la représentation des connaissances est un aspect central dans le domaine de l'Intelligence Artificielle car une machine intelligente qui souhaite effectuer l'action appropriée à une situation ou tenir un raisonnement correct doit avant tout s'appuyer sur des informations représentant le monde de façon suffisamment précise.

Cette difficulté à disposer d'une représentation correcte du monde est particulièrement importante lorsque l'on a affaire à un monde changeant ou à des informations provenant de sources multiples. La question est donc de savoir ce qu'un agent choisit de croire lorsque ses seules sources sont d'autres agents ou experts? Lorsque les informations apportées par les agents sont cohérentes entre elles, il suffit de considérer l'union de ces informations comme étant la représentation du monde. Mais dans le cas où les agents possèdent des informations conflictuelles, cette solution n'est pas souhaitable car elle conduirait à une représentation incohérente du monde. Il sera donc nécessaire pour notre agent de transiger, de choisir, parmi les informations qui lui sont fournies celles qui lui permettent d'obtenir une représentation du monde qui soit cohérente.

C'est ce problème, appelé fusion de croyances, que nous allons développer dans cette thèse.

La logique propositionnelle avec son solide socle formel pour la représentation, l'inférence ainsi que sa propriété de décidabilité est un cadre privilégié pour la représentation de croyances. C'est pour cela que les travaux sur le changement de croyances, qui provenaient à l'origine du domaine de l'épistémologie et des bases de données [FUV83, RGJM86], ont été développés dans le cadre de la logique classique. D'autres cadres ont été utilisés, notamment quantitatifs comme les cadres probabilistes et possibilistes [Spo88a, Spo90, SK94, Pea96]. Des postulats proposés par Alchourron, Gärdenfors et Makinson [AGM85, Men94] ont permis de caractériser le bon comportement d'opérateurs de révision de croyances.

Dans le domaine de la fusion de croyances, les approches basées sur la logique propositionnelle ont été développées [BKMS91, Rev93, Lin96, Rev97, Cho98]. Puis, inspirés par les postulats AGM, Konieczny et Pino-Perez les ont étendus à la fusion de croyances dans [KP98, KP99]. Ainsi, de nombreux opérateurs ont été proposés dans ce cadre [DDL06, Kon00, KLM02]. Cependant, peu d'implantations de ces opérations ont

été données. Parmi ces quelques implantations, des opérateurs de fusion basés sur des notions de morphologie mathématique présentés par Bloch et Lang [BL02] ont été mis en œuvre à l’aide des Diagrammes de Décision Binaires (ou BDD) par Gorogiannis et Hunter [GH08].

Les approches de changement de croyances basées sur la logique propositionnelle se divisent en deux grandes familles :

- les opérations sémantiques auxquelles appartiennent les travaux cités jusque-là, qui considèrent que la solution à une opération de fusion est une interprétation ou un ensemble d’interprétations représentant le mieux le monde.
- les opérations syntaxiques qui considèrent que le résultat de l’opération de fusion est un ensemble de formules issues des croyances initiales. Les approches syntaxiques ont été globalement moins étudiées que les approches sémantiques. Les principaux travaux sont issus de [MGC01, YLH07] pour la logique propositionnelle et [DLP94, BDKP02b] pour la logique possibiliste.

Nous proposons dans cette thèse une méthode de fusion syntaxique de croyances dans le cas de la logique propositionnelle. Nous nous situons dans un premier temps dans le cas où il n’existe pas de priorités explicites ni entre les agents, ni entre les croyances exprimées par les agents. Cette méthode, appelée fusion par R-ensembles, repose sur la recherche des sous-ensembles de formules à retirer afin de restaurer la cohérence. Nous avons réalisé une mise en œuvre de cette méthode basée sur la traduction du problème de fusion en un programme logique avec sémantique des modèles stables. Nous avons d’abord proposé une adaptation de l’algorithme *smodels* mais nous souhaitons réaliser une mise en œuvre indépendante du solveur utilisé et nous avons proposé une autre implantation basée sur les instructions fournies par les solveurs *Lparse/Smodels* et *Gringo/CLASP*. Nous avons testé cette dernière implantation avec des expérimentations portant sur des profils de croyances générés aléatoirement ainsi que sur les données issues d’un projet européen portant sur le relevé en archéologie sous-marine. Dans un deuxième temps, nous proposons d’étendre la fusion par R-ensembles dans deux directions. Nous nous sommes ainsi intéressés au cas où des préférences sont exprimées entre les agents ou entre les croyances exprimées par chaque agent. Nous nous sommes également penchés sur le cas où les croyances sont exprimées sous forme de programmes logiques.

Le plan de cette thèse est le suivant.

Après quelques rappels, le chapitre 1 introduit les notions essentielles de la théorie des ensembles, la logique propositionnelle, la logique des prédicats et la logique possibiliste.

Le chapitre 2 propose un rappel sur la programmation logique notamment sur la programmation logique avec sémantique des modèles stables. Nous présentons ici sa définition formelle telle qu’elle a été donnée par Gelfond et Lifschitz [GL88]. Nous présentons également une autre définition basée sur la logique du *Here-and-There* de Heyting [Hey30]. Nous présentons ensuite les principaux solveurs de modèles stables existants avant de parler des préférences en programmation logique avec sémantique des modèles stables. Finalement, nous rappelons la programmation logique possibiliste présentée par Nicolas et al. [NGS05].

Le chapitre 3 présente le changement de croyances en rappelant les principaux travaux sur la révision de croyances, en fonction des différentes définitions d'état épistémique. Ainsi la révision de théories, d'ensembles d'interprétations ou de formules est développée et nous montrons quelques exemples d'opérateurs. Puis, nous présentons les principaux travaux sur la fusion de croyances ; notamment les postulats proposés par Konieczny et Pino-Perez [KP99] ainsi que certains opérateurs présentés dans ce cadre. Nous présentons également les approches de Benferhat et al. sur la fusion de bases de croyances possibilistes [BK03].

Le chapitre 4 présente la fusion par R-ensembles que nous appelons RSF. Nous définissons plusieurs opérateurs de fusion syntaxique reposant sur différentes stratégies (*Somme*, *Card*, *Max* et *GMax*) et nous montrons les propriétés de ces différents opérateurs. Nous proposons également une contrepartie sémantique pour ces opérateurs. Nous montrons que les opérateurs définis syntaxiquement sont équivalents à leur contrepartie sémantique. Nous montrons également que la fusion par R-ensembles est la généralisation de la Révision par R-ensembles.

Le chapitre 5 présente la mise en œuvre de la fusion par R-ensembles basée sur la traduction du problème de fusion en un programme logique avec sémantique des modèles stables. La première partie de la traduction du problème de fusion en un programme logique est donnée. Cette partie fournit l'ensemble des ensembles de formules qui permettent de restaurer la cohérence, que nous appelons R-ensembles potentiels. L'équivalence entre les R-ensembles potentiels du profil et les modèles stables du programme logique est montrée. Parmi l'ensemble des R-ensembles potentiels, il reste à choisir ceux qui sauvegardent le maximum de croyances en fonction de la stratégie choisie. Pour réaliser ceci, nous définissons la notion de modèles stables préférés et nous montrons que les modèles stables préférés selon une stratégie donnée sont équivalents aux R-ensembles du profil selon cette même stratégie.

Le chapitre 6 a pour objet le calcul effectif des modèles stables préférés. Pour cela, nous proposons deux méthodes : la première repose sur l'adaptation de l'algorithme Smodels et la deuxième repose sur l'utilisation des instructions supplémentaires fournies par les solveurs Lparse/Smodels et Gringo/CLASP et permet une implantation indépendante du solveur ASP.

Le chapitre 7 présente une expérimentation de notre mise en œuvre de la fusion par R-ensembles dans le cas propositionnel. Une première partie présente une expérimentation utilisant des profils de croyances aléatoires. Nous comparons notre mise en œuvre avec celle présentée dans [GH08]. Une seconde partie présente l'adaptation de la fusion par R-ensembles à la gestion de l'incohérence dans le cadre du projet européen VENUS qui porte sur le relevé en archéologie sous-marine. Cette expérimentation permet d'illustrer le comportement de la fusion par R-ensembles pour des problèmes de taille réelle.

Le chapitre 8 se place dans le cas où il existe des préférences. Dans un premier temps, nous proposons une extension de la fusion par R-ensembles dans le cas où il existe une préférence entre les agents. Dans un second temps, nous nous plaçons dans le cas où il existe une préférence entre les agents mais également à l'intérieur des formules fournies par chaque agent.

Le chapitre 9 propose une extension de la fusion par R-ensembles dans le cas où les croyances sont exprimées sous forme de programmes logiques. Nous proposons une adaptation directe de la fusion par R-ensembles, appelée opération forte de fusion, ainsi qu'une analyse des propriétés des opérateurs ainsi définis. Nous présentons une autre définition de l'opération de fusion par R-ensembles, appelée opération faible de fusion, qui repose sur la logique du Here-and-There.

Après une conclusion, nous donnons des perspectives pour des travaux futurs.

Chapitre 1

Préliminaires

1.1 Ensembles et relations d'ordre

1.1.1 Ensembles

De manière générale, on nomme ensemble une collection d'objets quelconques. Soient A et B deux ensembles, leur union, notée $A \cup B$, est la collection des objets appartenant à A ou à B . Leur intersection, notée $A \cap B$ est la collection des objets appartenant à la fois à A et à B . Un multi-ensemble est un ensemble qui peut contenir plusieurs fois le même objet. On note $\sqcup$ l'union de multi-ensembles et $\sqcap$ leur intersection.

1.1.2 Relations, ordres et pré-ordres

Soit E un ensemble. Une relation binaire sur E est un sous-ensemble de $E \times E$. Autrement dit, une relation binaire est un ensemble de couples (x, y) tels que $x \in E$ et $y \in E$. On note $x \mathcal{R} y$ le fait qu'un couple (x, y) appartient à une relation $\mathcal{R}$. On note $x \not\mathcal{R} y$ si (x, y) n'appartient pas à la relation $\mathcal{R}$. $\mathcal{P}(E)$ représente l'ensemble des sous-ensembles de E .

Une relation $\mathcal{R}$ définie sur $E \times E$ peut vérifier les propriétés suivantes :

réflexive si et seulement si $\forall x \in E, x \mathcal{R} x$;

transitive si et seulement si $\forall x \in E, \forall y \in E, \forall z \in E$, si $x \mathcal{R} y$ et $y \mathcal{R} z$, alors $x \mathcal{R} z$;

symétrique si et seulement si $\forall x \in E, \forall y \in E$, si $x \mathcal{R} y$ alors $y \mathcal{R} x$;

antisymétrique si et seulement si $\forall x \in E, \forall y \in E$, si $x \mathcal{R} y$ et $y \mathcal{R} x$ alors $x = y$;

totale si et seulement si $\forall x \in E, \forall y \in E$, $x \mathcal{R} y$ ou $y \mathcal{R} x$;

Une relation qui n'est pas totale est dite partielle et une relation qui n'est pas réflexive est dite irréflexive. Un ordre est une relation réflexive, transitive et symétrique. Un pré-ordre est une relation réflexive et transitive. Une relation d'équivalence est une relation réflexive, transitive et antisymétrique. Un ordre strict est une relation irréflexive et transitive.

Soit un pré-ordre $\leq$ défini sur $E \times E$, on définit l'ordre strict $<$ associé par $x < y$ si et seulement si $x \leq y$ et $y \not\leq x$. On définit également une relation d'équivalence $\sim$ par $x \sim y$ si et seulement si $x \leq y$ et $y \leq x$.

1.2 Logique propositionnelle

La langage de la logique propositionnelle est construit à partir d'un ensemble infini dénombrable de variables propositionnelles $\mathcal{V}$ (que nous noterons à l'aide de lettres minuscules), des constantes $\top$ (représentant le Vrai) et $\perp$ (représentant le Faux), des connecteurs $\neg, \vee, \wedge, \rightarrow, \leftrightarrow$ ainsi que des symboles de ponctuation (et). L'ensemble des formules finies formées à partir de ces symboles est noté $\mathcal{L}$.

Définition 1.2.1. *L'ensemble des formules bien formées de $\mathcal{L}$ est le plus petit ensemble tel que :*

- $\perp$ et $\top$ sont des formules ;
- une variable propositionnelle est une formule ;
- si A et B sont des formules, alors $\neg A, A \vee B, A \wedge B, A \rightarrow B, A \leftrightarrow B$ sont des formules.

Exemple 1.2.1. *Par exemple, les formules suivantes sont des formules bien-formées de la logique propositionnelle :*

$$\mathcal{F} = \{a, \neg a \vee b, (a \wedge b) \rightarrow c\}$$

1.2.1 Aspect sémantique

La sémantique permet de définir des règles d'interprétations pour les formules à partir de la valeur de vérité de chacune des propositions qui les composent.

Définition 1.2.2. *On appelle interprétation toute application ω de $\mathcal{V}$ dans $\{0, 1\}$ telle que $\omega(\perp) = 0$ et $\omega(\top) = 1$. L'application ω est étendue aux formules de la façon suivante : $\forall A, B \in \mathcal{L}$:*

- $\omega(\neg A) = 1 - \omega(A)$
- $\omega(A \vee B) = \max(\omega(A), \omega(B))$
- $\omega(A \wedge B) = \min(\omega(A), \omega(B))$
- $\omega(A \rightarrow B) = \max(1 - \omega(A), \omega(B))$
- $\omega(A \leftrightarrow B) = \min(\max(1 - \omega(A), \omega(B)), \max(\omega(A), 1 - \omega(B)))$

L'ensemble des interprétations sur un langage $\mathcal{L}$ est noté $\mathcal{W}$. Une interprétation est souvent représentée sous la forme d'une table de vérité.

Une interprétation ω satisfait une formule A , noté $\omega \models A$, si et seulement si $\omega(A) = 1$. Une interprétation ω satisfaisant une formule A est appelée modèle de A . L'ensemble des modèles de A est notée $Mod(A)$. Par extension, on notera $A \models B$ lorsque l'ensemble des modèles de A satisfont B , i.e. $Mod(A) \subseteq Mod(B)$.

w	a	b	$\neg a$	$a \vee b$	$a \wedge b$	$a \rightarrow b$	$a \leftrightarrow b$
w_0	0	0	1	0	0	1	1
w_1	0	1	1	1	0	1	0
w_2	1	0	0	1	0	0	0
w_3	1	1	0	1	1	1	1

FIGURE 1.1 – Table de vérité des opérateurs logiques

On dira d'une formule qu'elle est valide (ou qu'elle est une tautologie) si elle est vraie pour toute interprétation. Une formule fausse pour toute interprétation est dite insatisfaisable ou incohérente. On dit qu'une formule est satisfaisable ou cohérente s'il existe une interprétation qui la satisfait. Une interprétation qui satisfait une formule A (ou toutes les formules d'un ensemble de formules $\mathcal{A}$) est appelée modèle de A (ou de $\mathcal{A}$). À l'inverse, une interprétation qui ne satisfait pas une formule A (ou toutes les formules d'un ensemble de formules $\mathcal{A}$) est appelée contre-modèle de A (ou de $\mathcal{A}$).

Définition 1.2.3. La formule B est une conséquence logique ou conséquence valide de la formule A si $\omega(A) = 1$ implique $\omega(B) = 1$, on écrit alors $A \models B$. La formule B est une conséquence logique de l'ensemble de formules $\mathcal{F}$, si $\forall A \in \mathcal{F}, \omega(A) = 1$ alors $\omega(B) = 1$, on écrit $\mathcal{F} \models B$.

L'ensemble des conséquences logiques déductibles d'une formule A est noté $\text{Cons}(A)$.

Définition 1.2.4. Soit $\mathcal{F}$ un ensemble de formules et G une formule, $\text{Cons}(\mathcal{F})$ représente l'ensemble des conséquences de $\mathcal{F}$. $G \in \text{Cons}(\mathcal{F})$ si et seulement si $\mathcal{F} \models G$.

Si deux formules ont un même ensemble de conséquences, on dit alors qu'elles sont sémantiquement équivalentes.

Définition 1.2.5. Les formules A et B sont équivalentes si et seulement si $A \models B$ et $B \models A$, on écrit alors $A \equiv B$.

Une interprétation est représentée soit comme un ensemble de variables soit comme un vecteur dont les composantes correspondent aux valeurs de vérité des variables propositionnelles rangées selon un ordre lexicographique.

Exemple 1.2.2. Soient a , b et c des variables propositionnelles. Reprenons l'exemple 1.2.1. L'ensemble des interprétations des formules de $\mathcal{F}$ est :

w	a	b	c	a	$\neg a \vee b$	$(a \wedge b) \rightarrow c$
w_0	0	0	0	0	1	1
w_1	0	0	1	0	1	1
w_2	0	1	0	0	1	1
w_3	0	1	1	0	1	1
w_4	1	0	0	1	0	1
w_5	1	0	1	1	0	1
w_6	1	1	0	1	1	0
w_7	1	1	1	1	1	1

On voit ici que la seule interprétation satisfaisant l'ensemble des formules de $\mathcal{F}$ est $\{a, b, c\}$. Ainsi $\{a, b, c\} \models \mathcal{F}$.

1.2.2 Aspect axiomatique

La logique propositionnelle peut être considérée comme un système formel. Un système formel est un ensemble de règles permettant en un nombre fini d'étapes et selon des règles explicites de déterminer si une proposition est un théorème. Un tel procédé s'appelle une démonstration. En logique propositionnelle, on définit un langage grâce à des règles d'écriture permettant d'effectuer de telles démonstrations. Le système formel de la logique propositionnelle est basé sur des axiomes - des propositions évidentes par elles-mêmes ne nécessitant pas de démonstrations - ainsi que sur deux règles de déductions : la substitution et le modus ponens.

Parmi les ensembles d'axiomes proposés, le système de Lukasiewicz [Luk64] est le plus succinct. Il est basé sur les trois axiomes suivants :

- A1 : $A \rightarrow (B \rightarrow A)$
- A2 : $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$
- A3 : $(\neg A \rightarrow \neg B) \rightarrow (B \rightarrow A)$

Grâce à ces axiomes et aux règles suivantes, il est possible de démontrer si de nouvelles formules sont des théorèmes pour le système, on appelle cela une déduction. Pour pouvoir réaliser des déductions, deux règles sont définies : la règle de la substitution et le modus ponens. La règle de substitution autorise le remplacement d'une variable propositionnelle, partout où elle apparaît, par n'importe quelle formule bien-formée. Le modus ponens stipule que si de l'ensemble de formules $\mathcal{G}$ on peut déduire A et de l'ensemble de formules $\mathcal{H}$ on peut déduire $A \rightarrow B$, alors de $\mathcal{G}$ et $\mathcal{H}$ on peut déduire B . On utilise le symbole $\vdash$ pour noter la relation de déduction ou d'inférence. Le symbole $\vdash$ n'est pas un nouveau symbole du langage mais appartient au métalangage.

$$\frac{\mathcal{G} \vdash A, \mathcal{H} \vdash A \rightarrow B}{\mathcal{G}, \mathcal{H} \vdash B}$$

Exemple 1.2.3. Reprenons l'exemple traité dans l'exemple 1.2.1. Soit l'ensemble de formules $\mathcal{F} = \{a, \neg a \vee b, (a \wedge b) \rightarrow c\}$.

La formule $\neg a \vee b$ est équivalente à $a \rightarrow b$. Ainsi, le modus ponens avec $\mathcal{G} = a$ et $\mathcal{H} = a \rightarrow b$ permet de déduire b . De façon similaire, cette information associée à la formule $(a \wedge b) \rightarrow c$ grâce au modus ponens avec $\mathcal{G} = \{a, b\}$ et $\mathcal{H} = \{(a \wedge b) \rightarrow c\}$ permet de déduire c .

On peut donc constater que, parallèlement aux calculs des interprétations, $\{a, b, c\} \vdash \mathcal{F}$.

Définition 1.2.6. Soit $H_1, H_2, \dots, H_m \in \mathcal{L}$ et $F_1, F_2, \dots, F_p \in \mathcal{L}$. Une déduction à partir d'hypothèses $H_1, H_2, \dots, H_m$ est une suite de formules bien-formées $F_1, F_2, \dots, F_p$ où chaque F_i est soit :

- une hypothèse ;
- un axiome ;

- une formule obtenue à partir des règles d'inférence (substitution ou *modus ponens*) appliquées aux formules placées avant F_i dans la déduction.

Un théorème est une déduction pour laquelle l'ensemble des hypothèses est vide .

Proposition 1.2.1. $\forall P_1, \dots, P_{n-1}, Q \in \mathcal{L}$, si $P_1, \dots, P_{n-1} \vdash (P_n \rightarrow Q)$ alors $P_1, \dots, P_{n-1}, P_n \vdash Q$.

Théorème 1.2.1 (Théorème de déduction). $\forall P_1, \dots, P_{n-1}, Q \in \mathcal{L}$, si $P_1, \dots, P_{n-1}, P_n \vdash Q$ alors $P_1, \dots, P_{n-1} \vdash (P_n \rightarrow Q)$.

1.2.3 Principales propriétés de la logique propositionnelle

Les aspects axiomatiques et sémantiques de la logique propositionnelle sont équivalents. Les théorèmes suivants sont issus de la thèse de doctorat de K.Gödel [FJK⁺86].

Le théorème d'adéquation stipule que tout théorème est une tautologie. Autrement dit, toute déduction effectuée axiomatiquement peut également être effectuée sémantiquement.

Théorème 1.2.2 (Théorème d'adéquation). $\forall A \in \mathcal{L}$, si $\vdash A$ alors $\models A$. Autrement dit, tous les théorèmes sont des tautologies.

Le théorème de complétude stipule que toute proposition démontrable est un théorème. Autrement dit, toute déduction effectuée sémantiquement peut également être effectuée axiomatiquement.

Théorème 1.2.3 (Théorème de complétude). $\forall A \in \mathcal{L}$, si $\models A$ alors $\vdash A$. Autrement dit, toutes les tautologies sont des théorèmes.

Parmi les résultats importants, il est impossible pour un système déductif de contenir une proposition ainsi que sa négation.

Théorème 1.2.4 (Théorème de cohérence). $\forall A \in \mathcal{L}$, il est impossible d'avoir à la fois $\vdash A$ et $\vdash \neg A$.

Le théorème suivant est central dans les questions que nous allons traiter. In extenso, toute formule démontrable en logique propositionnelle peut être démontrée par un algorithme et cette démonstration peut être effectuée en un temps fini. Formellement :

Théorème 1.2.5 (Théorème de décidabilité). $\forall A \in \mathcal{L}$, il existe un programme qui pour toute formule A , calcule en temps fini si $\vdash A$.

En effet, la plupart des logiques ne vérifient pas ce résultat et sont donc inappropriées pour représenter des croyances sur lesquelles il serait demandé d'effectuer des raisonnements. Ainsi, on cherche un sous-ensemble décidable de ces logiques pour résoudre ce problème.

D'un point de vue algorithmique, il est souvent plus facile de résoudre des problèmes lorsque les croyances sont représentées dans une syntaxe particulière appelée CNF.

Définition 1.2.7. *On appelle littéral une proposition ou la négation d'une proposition.*

On donne ici la définition de Forme Normale Conjonctive.

Définition 1.2.8. *On appelle clause une disjonction de littéraux. Un ensemble de formules qui est une conjonction de clauses est dite en forme normale conjonctive ou CNF.*

Il est toujours possible de transformer n'importe quel ensemble de formules en un ensemble de formules sémantiquement équivalent qui est en Forme Normale Conjonctive.

Théorème 1.2.6. *Tout ensemble de formules propositionnelles admet une forme normale conjonctive équivalente.*

Exemple 1.2.4. *Soit l'ensemble de formules $\mathcal{F} = \{a, \neg a \vee b, (a \wedge b) \rightarrow c\}$ de l'exemple 1.2.1. La base $\mathcal{F}' = \{a, \neg a \vee b, \neg a \vee \neg b \vee c\}$ est la forme normale conjonctive équivalente à $\mathcal{F}$.*

Il se pose naturellement la question de la cohérence d'un ensemble de formules. Ce problème, central en algorithmique, est appelé problème de satisfaisabilité (ou SAT) [GJ79]. Nous rappelons brièvement quelques méthodes classiques pour la mise en œuvre du problème de satisfaisabilité d'un ensemble de formules propositionnelles.

1.2.4 Les méthodes énumératives et le problème SAT

1.2.4.1 Présentation du problème SAT

On nomme problème SAT le problème de décision consistant à savoir s'il existe une interprétation satisfaisant un ensemble de formules de la logique propositionnelle. Les formules sont généralement présentées en Forme Normale Conjonctive.

Théorème de Cook et NP-complétude

Un problème est NP-complet s'il est dans NP et si tout problème de NP peut se réduire à celui-ci par une réduction polynomiale. Le problème SAT est le premier problème à avoir été montré NP-complet par S.Cook [Coo71].

1.2.4.2 Méthode énumérative

Les méthodes énumératives sont des méthodes permettant de résoudre une instance du problème SAT au niveau sémantique. Cette résolution se fait par énumération de toutes les interprétations jusqu'à ce qu'une solution soit trouvée ou que toutes les interprétations se soient révélées être des contre-modèles de l'ensemble de formules.

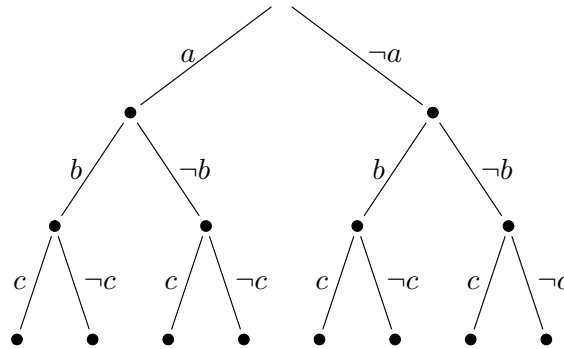


FIGURE 1.2 – Exemple d'arbre sémantique

Arbre sémantique

L'idée des méthodes énumératives est la construction d'un arbre binaire où chaque noeud représente une interprétation partielle sur les variables propositionnelles et chaque feuille représente une interprétation totale. Soit $\mathcal{A} = \{c_1, \dots, c_m\}$ le problème à résoudre où chaque c_i représente une clause du problème. Soit $Atome(\mathcal{A}) = \{p_1, \dots, p_n\}$ l'ensemble des variables propositionnelles présentes dans $\mathcal{A}$. On construit l'arbre de la manière suivante :

- Chaque arc est étiqueté par un littéral p_i ou $\neg p_i$. La branche p_i correspond à l'interprétation de p_i à $\top$ et la branche $\neg p_i$ correspond à l'interprétation de p_i à $\perp$.
- Les littéraux étiquetant les arcs d'un même noeud sont opposés.
- Aucune branche ne comporte plus d'une occurrence de chaque variable propositionnelle.

Cette construction est illustrée par la figure 1.2.

Définition 1.2.9 (branche fermée). *Une branche d'un arbre sémantique correspondant à un ensemble de clauses $\mathcal{A}$ est dite fermée si et seulement si il existe un nœud n tel que l'interprétation partielle correspondante à celui-ci est un contre-modèle des clauses de $\mathcal{A}$ et tel que les interprétations partielles de tout nœud ancêtre de n ne sont pas des contre-modèles d'une des clauses de $\mathcal{A}$.*

Définition 1.2.10 (arbre fermé). *Un arbre sémantique correspondant à un ensemble de clauses $\mathcal{A}$ est dit fermé si et seulement si toutes les branches de cet arbre sont fermées.*

Les deux théorèmes suivants nous autorisent à travailler sur des arbres sémantiques :

Théorème 1.2.7. *Un ensemble de clauses $\mathcal{A}$ est incohérent si et seulement si pour tout arbre sémantique complet correspondant à $\mathcal{A}$ il existe un arbre fini fermé correspondant à $\mathcal{A}$.*

Théorème 1.2.8. *Un ensemble de clauses $\mathcal{A}$ est incohérent si et seulement si il existe un sous-ensemble de clauses $\mathcal{A}' \subseteq \mathcal{A}$ incohérent.*

Ainsi, pour prouver qu'un ensemble de clauses $\mathcal{A}$ est cohérent, il suffit de trouver une feuille qui produit un modèle de $\mathcal{A}$. Pour prouver qu'un ensemble de clauses est incohérent, il est nécessaire de prouver que toutes les feuilles de l'arbre sémantique correspondant produisent des contre-modèles.

Méthode de Quine

La méthode de Quine est une amélioration de la méthode des arbres sémantiques qui permet de ne pas construire l'intégralité de l'arbre sémantique. Elle a été présentée dans [Qui52].

Définition 1.2.11. Soit p une variable apparaissant dans l'ensemble de clauses $\mathcal{A}$. $\mathcal{A}$ se partitionne vis-à-vis de p en deux sous-ensembles de clauses :

- L'ensemble des clauses où p apparaît, noté $\mathcal{A}_p$.
- L'ensemble des clauses où $\neg p$ apparaît, noté $\mathcal{A}_{\neg p}$.
- L'ensemble des clauses où ni p , ni $\neg p$ n'apparaît, noté $\mathcal{A}'' = \mathcal{A} \setminus (\mathcal{A}_p \cup \mathcal{A}_{\neg p})$.

On note $\mathcal{A}'_p$ l'ensemble des clauses de $\mathcal{A}_p$ privées de p et $\mathcal{A}'_{\neg p}$ l'ensemble des clauses de $\mathcal{A}_{\neg p}$ privées de $\neg p$.

On note $\mathcal{A}|_p = \mathcal{A}'_p \cup \mathcal{A}''$ (respectivement $\mathcal{A}|\neg p = \mathcal{A}'_{\neg p} \cup \mathcal{A}''$).

Proposition 1.2.2. Soient $\mathcal{A}$ un ensemble de clauses et $\{l_1, \dots, l_n\}$ un ensemble de littéraux. Un modèle de $\mathcal{A} \cup \{l_1\} \cup \dots \cup \{l_n\}$ est un modèle de $\mathcal{A}|_{l_1, \dots, l_n}$.

Proposition 1.2.3. Soient $\mathcal{A}$ un ensemble de clauses et p une variable. $\mathcal{A}$ est incohérent si et seulement si $\mathcal{A}'_p \cup \mathcal{A}''$ et $\mathcal{A}'_{\neg p} \cup \mathcal{A}''$ sont incohérents.

Algorithme 1 QUINE($\mathcal{A}$: un ensemble de clauses ; d : profondeur dans l'arbre)

```

si  $\mathcal{A} = \emptyset$  alors
  renvoyer  $\{\}$  { $\mathcal{A}$  est cohérent}
sinon si  $\mathcal{A}$  alors
  renvoyer ECHEC { $\mathcal{A}$  est incohérent}
sinon
   $\mathbf{L} = \text{QUINE}(\mathcal{A}|_{p_{d+1}}, d + 1)$ 
  si  $\mathbf{L} \neq \text{ECHEC}$  alors
    renvoyer  $\mathbf{L} \cup \{p\}$ 
  fini
   $\mathbf{L} = \text{QUINE}(\mathcal{A}|\neg p_{d+1}, d + 1)$ 
  si  $\mathbf{L} \neq \text{ECHEC}$  alors
    renvoyer  $\mathbf{L} \cup \{\neg p\}$ 
  fini
  renvoyer ECHEC
fini

```

1.2.4.3 Méthode de Davis-Putnam-Logemann-Loveland

Certaines assignations dans la procédure conduisent à une exploration inutile dans l'arbre. Par exemple, nous considérons l'ensemble de clauses $\mathcal{A} = \{\{\neg a, b\}, \{\neg b, \neg c\}, \{c, \neg d\}\}$ pour lequel, au niveau 1 de l'arbre, la variable a est assignée à $\top$. L'ensemble obtenu est $\mathcal{A}|_a = \{\{b\}, \{\neg b, \neg c\}, \{c, \neg d\}\}$. Il est clair qu'à ce stade de l'arbre, l'assignation de b à $\perp$ est inutile car elle produit une contradiction. On peut donc, avant de faire toute autre opération, assigner b à $\top$. Une telle assignation est appelée résolution unitaire. Cette amélioration a été présentée dans [DP60].

Nous définissons la fonction $ResolutionUnitaire(\mathcal{A})$ qui renvoie deux objets distincts :

- Un ensemble $\mathbf{I}$ de littéraux qui étaient soit déjà présents comme clause unitaire, soit ont été déduits par résolution unitaire.
- Une nouvelle version Γ de l'ensemble A résultant de la simplification de A par les littéraux de $\mathbf{I}$ ($\Gamma = \mathcal{A}|_{\mathbf{I}}$)

Ainsi l'algorithme *DPLL* est une amélioration de l'algorithme de *QUINE* sur plusieurs points. Le premier point est l'utilisation de la fonction de résolution unitaire pour réduire l'arbre de recherche. Le deuxième point est que le choix de la variable ne se fait plus dans le même ordre en fonction du niveau dans l'arbre mais s'adapte à la forme du programme ; de plus, les variables ne sont plus systématiquement assignées à $\top$ avant d'être assignées $\perp$ ce qui peut permettre de résoudre les ensembles de clauses cohérents plus rapidement.

L'algorithme de Davis-Putnam-Logemann-Loveland est présenté dans la figure 2

Algorithme 2 $DPLL(\mathcal{A} : \text{un ensemble de clauses})$

```

( $\mathbf{I}, \Gamma$ )  $\leftarrow ResolutionUnitaire(\mathcal{A})$ 
si  $\Gamma = \emptyset$  alors
    renvoyer  $\mathbf{I}$ 
sinon si  $\mathcal{A}$  alors
    renvoyer ECHEC
sinon
    Choisir un littéral  $l$  apparaissant dans  $\Gamma$ 
     $\mathbf{L} = DPLL(\mathcal{A} \cup l)$ 
    si  $\mathbf{L} \neq ECHEC$  alors
        renvoyer  $\mathbf{L} \cup \mathbf{I}$ 
    finsi
     $\mathbf{L} = DPLL(\mathcal{A} \cup \neg l)$ 
    si  $\mathbf{L} \neq ECHEC$  alors
        renvoyer  $\mathbf{L} \cup \mathbf{I}$ 
    finsi
    renvoyer ECHEC
fin
```

1.2.5 La résolution

La règle de résolution a été introduite par J.-A. Robinson [Rob65]. La résolution est une règle d'inférence logique que l'on peut voir comme une généralisation du modus ponens. Elle est utilisée dans les systèmes de preuve automatiques, notamment PROLOG.

Théorème 1.2.9. *Soient A, B, X des formules propositionnelles telles que $A \vee X$ et $B \vee \neg X$ sont cohérents.*

- *Si X est cohérent, alors B est cohérent ;*
- *Si $\neg X$ est cohérent, alors A est cohérent.*

Ainsi,

$$\{A \vee X, B \vee \neg X\} \models A \vee B$$

La déduction réalisée grâce à cette règle d'inférence est appelée résolvente.

Le principe de résolution est à la base de nombreux algorithmes de déduction dont le principe est illustré par l'algorithme qui se trouve dans la figure 3 :

Algorithme 3 Résolution (S : un ensemble de clauses)

tantque $\square \notin S$ **faire**

 Choisir l, c_1, c_2 tel que $l \in c_1$ et $\neg l \in c_2$

 Calculer la résolvente r

 Remplacer S par $S \cup \{r\}$

fin tantque

Si on peut déduire la clause vide, notée $\square$, de l'ensemble de formules, alors celui-ci est incohérent.

Proposition 1.2.4. $\mathcal{F} \models a$ si et seulement si $\mathcal{F} \cup \{\neg a\}$ est incohérent.

Si il ne reste plus de formules susceptibles de mener à de nouvelles déductions, alors le programme est cohérent.

Exemple 1.2.5. *Soit l'ensemble $\mathcal{A}$ de formules suivant :*

$$\mathcal{A} = \left\{ \begin{array}{lll} a \vee b \vee c & \neg c \vee \neg d & d \\ e \vee \neg b & \neg e & \neg d \vee \neg a \end{array} \right\}$$

On peut pratiquer une première résolution sur $a \vee b \vee c$ et $\neg c \vee \neg d$ pour déduire $a \vee b \vee \neg d$.

De la même manière, on obtient :

- $\{a \vee b \vee \neg d, d\} \models a \vee b$
- $\{e \vee \neg b, \neg e\} \models \neg b$
- $\{a \vee b, \neg b\} \models a$
- $\{d, \neg d \vee \neg a\} \models \neg a$
- $\{a, \neg a\} \models \square$

L'ensemble $\mathcal{A}$ est donc incohérent.

1.3 Logique des prédicats

La logique des prédicats, appelée également logique du premier ordre, est un formalisme logique plus expressif que la logique propositionnelle. Cependant, cette amélioration de l'expressivité a un coût : la logique des prédicats n'est que semi-décidable [Chu36] (un système est semi-décidable s'il n'existe pas de méthodes pour prouver en un temps fini qu'une formule n'est pas un théorème).

Le langage $\mathcal{L}_{Pr}$ de la logique des prédicats est construit à partir d'un ensemble infini dénombrable de variables, d'un ensemble éventuellement infini de fonctions, d'un ensemble éventuellement infini de prédicats, des connecteurs $\neg$, $\vee$, $\wedge$, $\rightarrow$, $\leftrightarrow$ et des quantificateurs $\forall$ et $\exists$, des constantes $\top$ et $\perp$ ainsi que des symboles de ponctuation (et). Une fonction d'arité 0 est appelée constante. Un prédicat d'arité 0 est appelé proposition.

Une expression pouvant servir d'argument à une fonction ou à un prédicat est appelée terme.

Définition 1.3.1. *Toute variable est un terme. Toute constante est un terme. Soit f une fonction n -aire et soient $t_1, t_2, \dots, t_n$ des termes, alors $f(t_1, t_2, \dots, t_n)$ est un terme.*

La brique élémentaire de la logique des prédicats est l'atome.

Définition 1.3.2. *Soient $t_1, t_2, \dots, t_n$ des termes (n peut être nul) et P un symbole de prédicat, alors $P(t_1, t_2, \dots, t_n)$ est un atome.*

Une formule qui respecte la grammaire du langage est appelée formule bien formée. Formellement :

Définition 1.3.3. *Soient A et B des formules bien formées, alors $\neg A$, $A \vee B$, $A \wedge B$, $A \rightarrow B$ et $A \leftrightarrow B$ sont des formules bien formées. Soit x une variable alors $\forall x A$ et $\exists x A$ sont des formules bien formées.*

1.3.1 Variables liées, variables libres et formules closes

Lorsqu'une variable x appartient à une sous-formule précédée d'un quantificateur universel noté $\forall$, $\forall x$ ou d'un quantificateur existentiel noté $\exists$, $\exists x$, elle est dite liée par ce quantificateur. Une variable est dite libre si elle n'est liée par aucun quantificateur.

Une formule close (ou formule fermée) est une formule dont toutes les variables sont liées.

1.3.2 Aspects sémantiques et syntaxiques

D'un point de vue sémantique, la logique des prédicats a pour but de déterminer quelles formules sont vraies ou fausses. Mais d'un point de vue axiomatique, on cherche à démontrer des théorèmes indépendamment de leur interprétation (signification). C'est pourquoi la logique des prédicats est considérée comme un système formel pour la démonstration des théorèmes.

1.3.2.1 Aspect sémantique

Définition 1.3.4. On appelle interprétation le triplet $\omega = (D, I_c, I_v)$ où D est un ensemble non-vidé appelé domaine d'interprétation, I_c est la fonction qui associe à tout symbole fonctionnel une valeur du domaine D et à tout prédicat une valeur dans $\{0, 1\}$, et I_v est la fonction qui associe à toute variable une valeur de D .

Définition 1.3.5. L'interprétation d'une formule de la logique des prédicats A associe une interprétation $\omega(A)$ à A de la manière suivante :

- Si x est une variable libre alors $\omega(x) = I_v(x)$;
- $\omega(f(t_1, t_2, \dots, t_n)) = (I_c(f))(\omega(t_1), \dots, \omega(t_n))$;
- $\omega(P(t_1, t_2, \dots, t_m)) = (I_c(P))(\omega(t_1), \dots, \omega(t_m))$;
- Si A et B sont des formules alors $\neg A$, $A \vee B$, $A \wedge B$, $A \rightarrow B$, $A \leftrightarrow B$ s'interprètent comme dans la logique propositionnelle ;
- Si A est une formule et x une variable alors $\omega(\forall x A) = 1$ si $\omega_{x/d}(A) = 1$ ¹ pour tout élément $d \in D$;
- Si A est une formule et x une variable alors $\omega(\exists x A) = 1$ si $\omega_{x/d}(A) = 1$ pour au moins un élément $d \in D$.

Comme pour la logique propositionnelle, la relation de conséquences entre deux formules peut se comprendre en termes d'inclusion des interprétations qui satisfont les formules.

Définition 1.3.6. Soient A et B deux formules bien formées. B est une conséquence de A si pour toute interprétation ω , $\omega(A) = 1$ alors $\omega(B) = 1$, on écrit $A \models B$.

Le fait qu'une formule est la conséquence d'un ensemble de formules est défini par :

Définition 1.3.7. Soient A et B deux formules bien formées. B est une conséquence de $\mathcal{F}$ si pour toute interprétation ω , telle que $\forall A \in \mathcal{F}, \omega(A) = 1$ alors $\omega(B) = 1$, on écrit $\mathcal{F} \models B$.

Comme pour la logique propositionnelle, on dira d'une formule vraie pour toute interprétation qu'elle est valide (ou qu'elle est une tautologie). Une formule fausse pour toute interprétation est dite insatisfaisable ou incohérente. S'il existe une interprétation qui satisfait une formule, on dit qu'elle est satisfaisable ou cohérente. Une interprétation qui satisfait une formule A (ou toutes les formules d'un ensemble de formules $\mathcal{A}$) est appelée modèle de A (ou de $\mathcal{A}$). À l'inverse, une interprétation qui ne satisfait pas une formule A (ou toutes les formules d'un ensemble de formules $\mathcal{A}$) est appelée contre-modèle de A (ou de $\mathcal{A}$).

1.3.2.2 Aspect axiomatique

Le système axiomatique de Lukasiewicz pour la logique propositionnelle peut être étendu à la logique des prédicats. Ce système est basé sur les axiomes suivants.

1. $\omega_{x/d}(A)$ désigne $\omega(A)$ où chaque occurrence de x est remplacée par d .

Soient A, B, C des formules de $\mathcal{L}_{Pr}$, x une variable, t un terme et E une formule n'ayant pas x pour variable libre :

1. $(A \rightarrow (B \rightarrow A))$;
2. $((A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C)))$;
3. $((\neg A \rightarrow \neg B) \rightarrow (B \rightarrow A))$;
4. $(\forall x A(x) \rightarrow A(t))$;
5. $((E \rightarrow B) \rightarrow (E \rightarrow \forall x B))$.

Les règles de substitution ainsi que le modus ponens sont également applicables en logique des prédicats. Soit $\mathcal{G}$ et $\mathcal{H}$ des ensembles de formules.

$$\frac{\mathcal{G} \vdash A, \mathcal{H} \vdash A \rightarrow B}{\mathcal{G}, \mathcal{H} \vdash B}$$

Mais le système formel de la logique des prédicats utilise également la règle de généralisation.

$$\frac{\mathcal{G} \vdash A}{\mathcal{G} \vdash \forall x A}$$

1.3.2.3 Propriétés de la logique des prédicats

La preuve de Gödel [FJK⁺86] est également valable pour la logique des prédicats.

Théorème 1.3.1 (Théorème d'adéquation). *Soit $A \in \mathcal{L}_{Pr}$ si $\vdash A$ alors $\models A$. Autrement dit, tous les théorèmes sont des tautologies.*

Théorème 1.3.2 (Théorème de complétude). *Soit $A \in \mathcal{L}_{Pr}$ si $\models A$ alors $\vdash A$. Autrement dit, toutes les tautologies sont des théorèmes.*

Contrairement à la logique propositionnelle, la logique des prédicats n'est que semi-décidable.

Théorème 1.3.3 (Semi-décidabilité). *La logique des prédicats est semi-décidable : il existe une procédure effective telle que pour toute formule A en entrée,*

- *si A est valide alors la procédure s'arrête et retourne 'oui' ;*
- *sinon ou bien la procédure s'arrête et retourne 'non', ou bien elle ne s'arrête pas.*

Alors qu'il est possible de déterminer si un ensemble de formules est satisfaisable en logique propositionnelle, cette question est plus délicate en logique des prédicats. Le théorème de Herbrand permet de donner un début de réponse à cette question au travers de l'instanciation des prédicats par l'ensemble des termes apparaissant dans l'ensemble de formules.

1.3.2.4 Univers de Herbrand

L'univers de Herbrand contient l'ensemble des termes qui peuvent être définis à partir des constantes et des fonctions apparaissant dans l'ensemble de formules.

Définition 1.3.8 (Univers de Herbrand). *Soit $\mathcal{F}$ un ensemble de formules :*

- *L'ensemble des constantes apparaissant dans $\mathcal{F}$ est dans $H(\mathcal{F})$;*
- *Pour toute fonction f apparaissant dans $\mathcal{F}$, si $t_1, \dots, t_n$ sont dans $H(\mathcal{F})$ alors $f(t_1, \dots, t_n)$ l'est aussi.*

La plupart du temps $H(\mathcal{F})$ est infini dénombrable comme dans l'exemple suivant :

Exemple 1.3.1. *Soient x, y des variables et a, b des constantes. Soit l'ensemble $\mathcal{F}$ de formules suivants :*

$$\mathcal{F} = \{P(x) \vee Q(a) \vee \neg P(f(a)), \neg Q(b) \vee P(g(x, y))\}$$

L'ensemble de Herbrand $H(\mathcal{F})$ est le suivant :

$$H(\mathcal{F}) = \{a, b, f(a), f(b), g(a, a), g(a, b), g(b, a), g(b, b), f(f(a)), f(f(b)), f(g(a, a)), \dots\}$$

À partir de l'univers de Herbrand, on définit la base de Herbrand comme suit :

Définition 1.3.9 (Base de Herbrand). *La base de Herbrand est l'ensemble des termes que l'on peut construire à l'aide de l'univers de Herbrand et des prédicats du langage.*

Exemple 1.3.2. *La base de Herbrand de l'ensemble $\mathcal{F}$ de formules décrit dans l'exemple 1.3.1 est le suivant :*

$$\{P(a), Q(a), P(b), Q(b), P(f(a)), Q(f(a)), P(f(b)), Q(f(b)), P(g(a, a)), \dots\}$$

Le lien entre la logique des prédicats et la logique propositionnelle s'effectue à travers le théorème de Herbrand. Nous appelons système de Herbrand $SH_{\mathcal{F}}$ associé à un ensemble de formules $\mathcal{F}$, l'ensemble des formules obtenues à partir de $\mathcal{F}$ en remplaçant les variables par des éléments de l'univers de Herbrand, on obtient le résultat suivant :

Théorème 1.3.4 (Théorème de Herbrand). *Soit $\mathcal{F}$ un ensemble fini de formules, et $\mathcal{F}$ et $SH_{\mathcal{F}}$ le système de Herbrand associé à $\mathcal{F}$, $\mathcal{F}$ est satisfaisable si et seulement si $SH_{\mathcal{F}}$ est satisfaisable.*

1.4 La logique possibiliste

La logique possibiliste est une logique de l'incertain permettant de traiter des informations incomplètes et partiellement incohérentes. Elle a été proposée par D. Dubois et H. Prade [DP94] à partir de la logique floue de Zadeh [Zad65]. Les informations sont représentées à l'aide de formules de la logique du premier ordre auxquelles sont associées deux valeurs numériques : le degré de possibilité et le degré de nécessité. Le degré de possibilité d'une formule représente la compatibilité de la formule avec le reste des informations. Le degré de nécessité représente la certitude que l'on a sur la vérité de la formule.

1.4.1 Base de croyances et distribution de possibilité

La clé de voute de la logique possibiliste est la distribution de possibilité π qui est une fonction de l'ensemble des interprétations $\omega \in \mathcal{W}$ vers $[0, 1]$. La valeur $\pi(\omega)$ représente le degré de satisfaction de l'interprétation ω vis-à-vis d'une base de croyances possibiliste K . De manière ordinaire, on choisit que $\pi(\omega) = 0$ représente l'impossibilité de l'interprétation ω et que $\pi(\omega) = 1$ représente la compatibilité complète de ω avec la base de croyances. Dans le cas où la base de croyances possibiliste est cohérente, on a : $\sup\{\pi(\omega) \mid \omega \in \mathcal{W}\} = 1$ qui signifie qu'il existe au moins une interprétation ω telle qu'elle satisfait toutes les formules de la base.

Définition 1.4.1. *Une formule de la logique possibiliste est un triplet de la forme $F = (A, \alpha, \beta)$ avec :*

- $N(F) = \alpha$ est un réel compris entre 0 et 1 (inclus), représentant la nécessité de F .
- $\Pi(F) = \beta$ est un réel compris entre 0 et 1 (inclus), représentant la possibilité de F .
- $F^* = A$ est la formule la logique classique associée à F .

Une base de croyances possibiliste est représentée par un ensemble de formules de la logique possibiliste. On la notera $\mathcal{K} = \{(A_i, \alpha_i, \beta_i)\}_{i \in I}$. Les formules ayant une possibilité de 0 ne sont pas représentées explicitement.

Soit $F = (A, \alpha, \beta)$ une formule de la logique possibiliste. La possibilité de F représente le degré de compatibilité de la croyance avec le reste de la base. Ainsi :

- $\Pi(F) = 0$ signifie qu'il est impossible que A soit vraie.
- $\Pi(F) = 1$ signifie que rien ne contredit la possibilité de A . Cela ne signifie en rien que A est vraie.

Par exemple, lors d'événements équiprobables (comme un tir à pile ou face, ou le tirage d'une carte dans un jeu dans des conditions normales), tous les événements auront une possibilité de 1.

La mesure de possibilité d'une formule peut être obtenue par l'intermédiaire de la distribution de possibilité. La possibilité d'une formule correspond à la possibilité la plus élevée parmi celles des interprétations qui satisfont cette formule. Ainsi :

Définition 1.4.2 (Mesure de possibilité d'une formule). *Soit $\mathcal{K} = \{F_i : (A_i, \alpha_i, \beta_i)\}_{i \in I}$ une base de croyances possibiliste et $\pi(\omega)$ une distribution de possibilité sur $\mathcal{K}$. La mesure de possibilité d'une formule F_i est donnée par :*

$$\Pi(F_i) = \max\{\pi(\omega) \mid \omega \models A_i\}$$

Soit $F = (A, \alpha, \beta)$ une formule de la logique possibiliste. La mesure de nécessité représente les preuves, les raisons de croire que A est vraie.

- $N(F) = 0$ signifie que A n'a aucune raison d'être vraie. Cela n'implique pas que A soit fausse.
- $N(F) = 1$ signifie que A est absolument vraie.

La mesure de nécessité d'une formule est duale de sa possibilité. La possibilité de F est liée à la nécessité de $\neg F$ et vice versa.

Définition 1.4.3 (Mesure de nécessité). *Soit F une formule de la logique possibiliste, on a :*

$$\Pi(F) = 1 - N(\neg F)$$

Grâce à cette dualité, il est également possible d'obtenir la mesure de nécessité en passant par la distribution de possibilité.

Définition 1.4.4. *Soit $\mathcal{K} = \{F_i : (A_i, \alpha_i, \beta_i)\}_{i \in I}$ une base de croyances possibiliste et $\pi(\omega)$ une distribution de possibilité sur $\mathcal{K}$. La mesure de nécessité d'une formule F_i est donnée par :*

$$N(F_i) = 1 - \Pi(\neg F_i) = 1 - \max\{\pi(\omega) \mid \omega \not\models A_i\}$$

De plus, un ordonnancement des croyances de la base découle des mesures de possibilité. Soit A et B deux formules possibilistes, si $\Pi(A) > \Pi(B)$ ou si $N(A) > N(B)$ alors A est préférée à B .

Il est également possible d'obtenir une distribution de possibilité à partir d'une base de croyances possibiliste. Dans un premier temps, on identifie les distributions de possibilité compatibles avec la base de croyances. Seule une distribution telle que $\forall F = (A, \alpha, \beta) \in \mathcal{K}, N(F) \geq \alpha$ est dite compatible. Parmi ces distributions là, on choisira la distribution dite la plus spécifique. Une distribution $\pi(\omega)$ est dite la plus spécifique lorsque il n'existe pas de distribution $\pi'(\omega)$ telle que $\forall \omega, \pi'(\omega) \geq \pi(\omega)$. Il existe toujours une distribution plus spécifique [DP94].

La logique possibiliste propose un cadre complet de traitement de l'information incertaine. Elle propose notamment une mesure de l'incohérence. Dans le cadre de la logique classique, l'incohérence se caractérise par la présence dans l'ensemble de conséquences de $\mathcal{K}$ d'un atome et de sa négation. Ici, en cas d'incohérence l'égalité $\sup\{\pi(\omega) \mid \omega \in \mathcal{W}\} = 1$ est perdue et on a $\sup\{\pi(\omega) \mid \omega \in \mathcal{W}\} < 1$. En d'autres termes, il n'existe pas de distributions de possibilité satisfaisant pleinement l'ensemble des formules de la base de croyances.

Exemple 1.4.1. *Soit la base de croyances possibiliste $\mathcal{K}$ suivante, tirée de [DP94] :*

$$\mathcal{K} = \left\{ \begin{array}{cc} (\neg p \vee r, 0, 6) & (p, 0, 8) \\ (\neg q \vee \neg r, 0, 9) & (q, 0, 9) \end{array} \right\}$$

Il n'existe pas d'interprétation satisfaisante pour $\mathcal{K}$ ainsi :

$$\begin{array}{lll} \pi(\{p, q, r\}) = 0, 1 & \pi(\{p, q, r\}) = 0, 4 & \pi(\{p, q, r\}) = 0, 7 \\ \pi(\{p, q, r\}) = 0, 4 & \pi(\{p, q, r\}) = 0, 1 & \pi(\{p, q, r\}) = 0, 2 \\ \pi(\{p, q, r\}) = 0, 2 & \pi(\{p, q, r\}) = 0, 2 & \end{array}$$

et la condition $\sup\{\pi(\omega) \mid \omega \in \mathcal{W}\} = 1$ n'est pas respectée.

Grâce à la distribution de possibilité, nous disposons d'une méthode pour évaluer la cohérence d'une base de croyances possibiliste et par conséquent son incohérence. Ainsi, soit K une base de croyances possibiliste :

$$Coh(\mathcal{K}) = \sup_{\pi \models \mathcal{K}} \sup_{\omega \in \mathcal{W}} \pi(\omega)$$

$Coh(\mathcal{K})$ représente le degré de cohérence de $\mathcal{K}$ qui est lié à son degré d'incohérence $Incoh(\mathcal{K})$ par la formule :

$$Incoh(\mathcal{K}) = 1 - Coh(\mathcal{K})$$

Exemple 1.4.2. Reprenons l'exemple 1.4.1. Le degré de cohérence de $\mathcal{K}$ est égal à la mesure de possibilité la plus élevée parmi les $\omega \in \mathcal{W}$, soit $Coh(\mathcal{K}) = 0,7$. Et ainsi, son degré d'incohérence est égal à $Incoh(\mathcal{K}) = 1 - Coh(\mathcal{K}) = 0,3$.

De nombreux travaux proposent une modélisation du raisonnement non-monotone en logique possibiliste comme dans [BK03, DDC09].

Première partie

Etat de l'art

Chapitre 2

La programmation logique

Introduction

La programmation logique diffère de la programmation impérative qui oblige l'indication pas à pas à l'ordinateur de ce qu'il doit faire, au profit d'une programmation plus déclarative qui consiste à représenter dans un formalisme donné les données du problème et à abandonner le traitement à un programme nommé solveur. Le premier langage de programmation logique est PROLOG. Nous commencerons donc par PROLOG avant de parler des programmes logiques dans un sens plus général. Puis, nous parlerons des deux sémantiques généralement associées aux programmes logiques : la sémantique bien-fondée (ou *well-founded semantics*) et la sémantique des modèles stables. Finalement, nous parlerons des différents solveurs permettant de traiter les programmes logiques au regard de la sémantique des modèles stables.

2.1 PROLOG

La programmation logique fait partie de la famille de la programmation déclarative. Contrairement à la programmation impérative où l'on décrit *comment* sera résolu le problème ; dans le cadre de la programmation déclarative, c'est le *quoi* qui est donné, la description du problème lui-même.

Développé dans un premier temps par R. Kowalski et A. Colmerauer (une liste globale d'articles est compilée dans [Kow88]) pour le traitement des langages naturels, PROLOG a été développé comme un outil pour représenter le monde et traiter les questions sur le monde. Quelques années plus tard, D.H. Warren a implanté une version efficace de PROLOG [Dav80].

Le problème est décrit à l'aide d'un ensemble de faits ou de règles définissant les relations entre les objets où les liens qui unissent ces relations entre elles. Les questions fournissent également un mécanisme permettant d'interroger le programme sur l'existence d'une relation entre plusieurs objets. Dans le cadre de cette thèse, les programmes logiques sont utilisés pour représenter les croyances et raisonner sur elles.

Les objets qui permettent la description du monde sont : les faits, les règles et les questions. Ils seront détaillés dans les sections suivantes.

2.1.1 Les faits

Dans la programmation logique, on décrit l'état des croyances à l'aide de faits et de règles. Le fait est le type d'énoncé le plus simple, il affirme l'existence d'une propriété pour un objet, ou de relations entre plusieurs objets.

Exemple 2.1.1. *On peut, par exemple, exprimer les faits suivants :*

$$\text{felin}(\text{lion}) \quad \text{nation}(\text{renault}, \text{france})$$

Le premier fait possède un argument et stipule que le lion est un félin. Le second fait possède deux arguments et représente le fait que Renault est une marque française.

On peut remarquer que l'écriture de faits peut se faire de plusieurs manières car les faits précédents auraient pu s'écrire :

$$\text{famille}(\text{felin}, \text{lion}) \quad \text{francais}(\text{renault})$$

Ainsi, la compréhension et la lecture des faits nécessitent la mise en place de convention. Sans convention, le fait $\text{mere}(\text{sophie}, \text{amelie})$ pourrait aussi bien se comprendre comme “*sophie est la mère d'amelie*” que comme “*amelie est la mère de sophie*”

Un programme qui serait composé uniquement de faits peut être considéré comme une base de données.

Remarques 2.1.1. *Le nom des constantes est toujours représenté en minuscules.*

2.1.1.1 Les variables

Une variable est la représentation d'un objet non spécifié. Dans le cadre de la programmation logique, elle s'apparente plus à une variable mathématique qu'à une variable de programmation impérative.

Dans le cas où elles sont utilisées dans des faits ou dans des règles, les variables se comportent comme s'elles étaient quantifiées universellement.

Exemple 2.1.2. *Avec des variables, on peut exprimer des faits ayant des portées universelles. Par exemple,*

$$\text{Addition}(0, X, X)$$

permet de représenter que l'addition de 0 à n'importe quel nombre X vaudra X . Mais encore, dans le cadre d'une base de Herbrand composée uniquement de personnages historiques, la règle

$$\text{Mort}(X)$$

représente que tous les personnages présents dans la base sont décédés.

Les variables trouveront une autre lecture lorsqu'elles seront utilisées pour des énoncés de types questions.

Remarques 2.1.2. *Le nom des variables commence toujours par une majuscule.*

2.1.1.2 Les termes

La seule structure de données au sein des programmes logiques est le *terme*. Constantes et variables sont des termes, on les appelle des *termes simples*.

Il est également possible de définir des *termes composés* qui représentent des objets complexes de l'univers. Un terme composé est décrit par un *foncteur* et un ensemble d'arguments qui sont des termes. Le foncteur est le descriptif du terme composé : il comprend son nom ainsi que le nombre d'arguments, également appelé arité et est noté nom/n où n est l'arité.

Exemple 2.1.3. Reprenons les exemples précédents. Le terme $addition(0, X, X)$ a pour foncteur $addition/3$.

De la même manière, le terme $felin(lion)$ a pour foncteur $felin/1$

Les termes permettent également de définir des types abstraits de données à l'aide de structures récursives comme dans les exemples suivants :

Exemple 2.1.4. On peut définir des listes :

$$liste(a, liste(b, liste(c, d)))$$

ou des arbres :

$$arbre(arbre(nil, 3, nil), 5, R)$$

2.1.1.3 Le mécanisme de substitution

Définition 2.1.1 (Substitution). Une substitution est une fonction qui associe des termes à des variables.

Une substitution Θ appliquée à un terme A , noté $A\Theta$, est l'ensemble des termes obtenus en remplaçant, pour tout couple (X, t) de Θ , toutes les occurrences de X par t dans A .

Exemple 2.1.5. Le résultat de la substitution de $\{X = charlemagne\}$ au terme $Mort(X)$ est le terme $Mort(charlemagne)$.

Ainsi, le résultat d'une substitution appliquée à un ensemble de termes est appelé une instantiation. Formellement,

Définition 2.1.2 (Instance). A est une instance de B s'il existe une substitution Θ telle que $A = B\Theta$.

Exemple 2.1.6. Le terme $mere(sophie, amelie)$ est une instance de $mere(X, Y)$ avec la substitution $\{X = sophie, Y = amelie\}$ ou pour reprendre l'exemple précédent le terme $mort(charlemagne)$ est une instance de $mort(X)$ avec la substitution $\{X = charlemagne\}$

2.1.1.4 Les faits universels

Afin de faciliter l'écriture des faits portant sur tous les objets de l'univers décrits, on peut utiliser des variables. Par exemple, si l'univers est composé de l'ensemble des rois de France, pour pouvoir écrire que tous sont dorénavant morts, nous devrions écrire :

$$\begin{aligned} & mort(clodionlechevelu) \\ & mort(merovee) \\ & \dots \\ & mort(louis - philippe) \end{aligned}$$

Grâce aux variables, le seul fait universel $mort(X)$ porte la propriété “est mort” sur tous les éléments de l'univers.

Exemple 2.1.7. *Les termes $mort(X)$ et $addition(0, X, X)$ sont des faits universels.*

2.1.2 Les règles

Les règles peuvent être comprises comme une manière de définir des relations complexes à l'aide d'un ensemble de relations plus simples. Par exemple, une règle peut permettre de décrire la relation “être mère” comme étant l'intersection entre les concepts : “être parent” et “être une femme”.

La plupart de ces règles font intervenir la présence d'une ou plusieurs variables. Autrement dit, l'exemple précédent sera traduit par : quelque soit X , si X est une femme et que X est parent, alors X est une mère. La variable porte seulement sur la règle. Formellement, une règle s'écrit avec la syntaxe suivante :

$$A \leftarrow B_1, \dots, B_n$$

où $A, B_1, \dots, B_n$ sont des termes et $n \geq 0$. A est la tête de la règle et les B_i forment le corps de la règle.

Exemple 2.1.8. *Une règle qui exprime la relation mere :*

$$mere(X, Y) \leftarrow parent(X, Y), femme(X)$$

Une autre règle qui exprime la relation grandparent :

$$grandparent(X, Z) \leftarrow parent(X, Y), parent(Y, Z)$$

Remarques 2.1.3. *Un fait est une règle pour laquelle le corps ne contient pas de termes.*

On peut donc définir un problème à partir de règles et ainsi :

Définition 2.1.3. *Un programme logique est un ensemble fini de règles.*

Les règles permettent notamment d'effectuer des déductions à travers la règle du *modus ponens*.

Définition 2.1.4 (Modus ponens). *Soient une règle $A \leftarrow B_1, \dots, B_n$ et un ensemble de faits $B'_1, \dots, B'_n$. Alors on peut déduire A' si $A' \leftarrow B'_1, \dots, B'_n$ est une instance du programme.*

2.1.3 Les questions

Les dernières formes possibles d'énoncés en PROLOG sont les questions. La question est une recherche de l'existence d'une relation parmi les croyances ou les conséquences de celles-ci. Dans le cas où la relation demandée existe, la réponse est *yes*, sinon la réponse est *no*.

Les questions possèdent la même syntaxe que les faits.

Pour pouvoir répondre à la question, il faut déterminer si la question est une conséquence logique du programme interrogé. Les règles de déduction sont au nombre de quatre :

- L'*identité* : d'un fait A , on peut déduire A . Un fait est sa propre conséquence logique.
- La *généralisation* : une question contenant des variables A est la conséquence logique, quelle que soit la substitution Θ , d'une de ses instances $A\Theta$.
- L'*instanciation* : à partir d'un énoncé contenant des variables A , on peut déduire, pour toute substitution Θ , une instance de celui-ci, $A\Theta$.
- Le *modus ponens* (voir la définition 2.1.4)

2.1.4 Négation par échec

La négation classique n'existe pas en PROLOG. Il n'existe que la négation par échec représentée la plupart du temps par le symbole *not*. La négation par échec d'un atome signifie qu'il ne peut pas être déduit ou prouvé. Lorsqu'un atome est soumis à une négation par échec, on considère qu'il est faux s'il n'est pas possible de le déduire en un temps fini. En d'autres termes, un terme est faux s'il n'est pas montré comme étant vrai.

Exemple 2.1.9. La règle :

$$\text{roule}(X) \leftarrow \text{voiture}(X), \text{not panne}(X).$$

signifie que si X est un voiture, a priori elle roule sauf s'il est prouvé qu'elle est en panne.

2.1.5 Les sémantiques de PROLOG

Les résultats obtenus par PROLOG peuvent être interprétés de différentes façons, appelées sémantiques. Naïvement, la signification d'un programme logique Π est l'ensemble des instances de bases déductibles à partir de Π par un nombre fini d'applications de la règle du modus ponens.

On en dénombre trois : la sémantique opérationnelle, la sémantique déclarative et la sémantique dénotationnelle.

2.1.5.1 La sémantique opérationnelle

La sémantique opérationnelle est l'ensemble des buts de base auxquels le programme Π donne une solution en utilisant la SLD-résolution.

SLD-résolution

Pour résoudre les questions qui lui sont posées, PROLOG utilise l'algorithme de résolution SLD basé sur la résolution (voir la section 1.2.5). Nous donnerons d'abord un exemple informel de son déroulement avant d'en donner une écriture plus formelle.

Prenons pour exemple le programme suivant :

$$\begin{array}{l} p \leftarrow q. \quad q. \\ r \leftarrow s. \quad r \leftarrow p, q. \end{array}$$

auquel on pose la question : $r?$. D'abord, on regarde les règles permettant de déduire r , il y en a deux : $r \leftarrow s$. et $r \leftarrow p, q$.. Pour la première règle, on peut déduire r si s fait partie du programme, or il n'existe aucune règle permettant de déduire s , on est donc dans une impasse avec cette règle-là. Pour la seconde règle, on peut déduire r si p et q font partie du programme. La déduction de q est immédiate puisqu'il s'agit d'un fait. La déduction de p se fait grâce aux règles $p \leftarrow q$ et q . On peut donc conclure que r est une conséquence du programme.

Ce raisonnement est généralement représenté par le schéma ci-dessous qui représente le chaînage arrière qui part de la question et utilise les règles dont la tête est la question et la remplace par le corps de cette règle.

$$\begin{array}{c} \overline{\quad} \\ \quad \quad \quad \overline{\quad} \\ \quad \quad \quad \quad \quad \overline{\quad} \\ \overline{\{s\}} \quad \overline{\{q\}} \\ \quad \quad \quad \overline{\{p, q\}} \\ \quad \quad \quad \quad \quad \overline{\{r\}} \end{array}$$

Le déroulement formalisé d'un tel algorithme est donné dans la figure 4, avec Q une question de base.

Algorithme 4 Calcul SLD (Q : une question)

```

Initialiser la résolvante à  $Q$ 
tantque La résolvante  $a_1, \dots, a_n$  n'est pas vide faire
  Choisir un but  $a_i$ 
  Choisir une instance de base d'une clause  $a \leftarrow b_1, \dots, b_k$ 
  tels que  $a = a_i$  (si une telle instance n'existe pas, sortir de la boucle)
  Déterminer la nouvelle résolvante :
     $a_1, \dots, a_{i-1}, b_1, \dots, b_k, a_{i+1}, \dots, a_n$ 
fin tantque
si La résolvante est vide alors
  renvoyer yes
sinon
  renvoyer no
finsi

```

2.1.5.2 La sémantique déclarative

Définition 2.1.5 (Interprétation). *Une interprétation est un sous-ensemble de la base de Herbrand. Un but dans la base de Herbrand est vrai par rapport à une interprétation si ce but en est un élément et faux sinon.*

Une interprétation I est un modèle pour un programme logique Π si pour chaque instance de base d'une règle du programme $A \leftarrow B_1, \dots, B_n$, A est dans I à chaque fois que $B_1, \dots, B_n$ sont dans I . Un modèle est une interprétation qui respecte les clauses du programme.

Ainsi, l'intersection de tous les modèles de Π est toujours un modèle et est appelé *modèle minimal* de Π , et est noté $M(\Pi)$.

Le modèle minimal d'un programme est sa sémantique déclarative.

2.1.5.3 La sémantique dénotationnelle

Pour la sémantique dénotationnelle, on construit le plus petit point fixe du programme Π .

Soit $T_P(\omega)$ une fonction qui associe une interprétation à une interprétation tel que $T_P(\omega) = \{A \mid A \leftarrow B_1, \dots, B_n \in \text{inst}(\Pi) \text{ et } B_1, \dots, B_n \in \omega\}$

Une interprétation supportée du programme Π est une interprétation ω telle que pour tout $A \in \omega$ il existe une règle $A \leftarrow B_1, \dots, B_n \in \text{inst}(\Pi)$ et $B_1, \dots, B_n \subseteq \omega$.

Soit Π un programme logique et ω une interprétation :

- ω est un modèle de Π , noté $\omega \models \Pi$ si et seulement si $T_P(\omega) \subseteq \omega$;
- ω est un modèle supporté de Π si et seulement si $T_P(\omega) = \omega$.

Le plus petit point fixe de $T_P(\emptyset)$ constitue donc la sémantique dénotationnelle de PROLOG.

2.2 Programmes logiques

Un programme logique est un ensemble de règles de la forme :

$$r : a \leftarrow b_1, \dots, b_m, \text{not } c_1, \dots, \text{not } c_n.$$

où a, b_i et c_j sont des termes. Ici, le symbole *not* représente la négation par echec. a est appelé tête de la règle et est noté $tete(r)$. b_i et c_j constituent le corps de la règle, noté $corps(r)$. Le corps d'une règle peut être divisé en corps positif et corps négatif. Les atomes b_i constituent le corps positif de la règle, noté $corps^+(r)$, et les atomes c_i constituent le corps négatif de la règle, noté $corps^-(r)$. Ainsi, $corps(r) = corps^+(r) \cup corps^-(r)$.

De manière intuitive, une règle peut être comprise de la façon suivante : si tous les atomes b_i du corps positif de la règle sont vrais et si aucun des atomes c_i du corps négatif n'est vrai, alors la tête de la règle peut être déduite.

2.2.1 Programme logique basique

Un programme basique est un programme logique sans négation par échec, c'est à dire où toutes les règles sont de la forme $a \leftarrow b_1, \dots, b_m$.

Définition 2.2.1. *Un ensemble d'atomes X est clos sous un programme basique Π si et seulement si pour n'importe quelle règle $r \in \Pi$, $tete(r) \in X$ à chaque fois que $corps(r) \subseteq X$. Le plus petit ensemble clos sous un programme basique Π est noté $CN(\Pi)$.*

Exemple 2.2.1. *Soit le programme Π suivant :*

$$\Pi = \left\{ \begin{array}{lll} a. & b \leftarrow a. & f \leftarrow e. \\ c \leftarrow a. & d \leftarrow c. & g \leftarrow e \end{array} \right\}$$

Pour ce programme Π , les ensembles $\{a, b, c, d, e, f, g\}$, $\{a, b, c, d, f\}$, $\{a, b, c, d, g\}$ sont des ensembles clos sous Π . Mais le plus petit ensemble clos sous Π est $\{a, b, c, d\}$.

Un programme basique possède un et un seul plus petit ensemble clos que l'on considère comme étant l'ensemble de ses conséquences.

2.2.2 Dénomination des programmes logiques

Lorsque qu'un programme logique utilise la négation par echec mais pas la négation classique, on parle de *programme logique général*.

Lorsque qu'un programme logique utilise la négation par echec et la négation classique, on parle de *programme logique étendu*.

2.3 Sémantique bien-fondée

En PROLOG, les programmes logiques généraux n'ont pas toujours de modèle total satisfaisant. La sémantique bien-fondée de Van Gelder et al. [vGRS91] propose de définir des modèles partiels appelés modèles bien-fondés.

Les modèles bien-fondés sont divisés en deux ensembles : les conséquences positives d'un coté, appelées modèles partiels bien-fondés, et les conséquences négatives de l'autre, appelées ensemble non-fondé.

2.3.1 Ensemble non-fondé

L'ensemble non-fondé d'un programme logique Π peut être considéré comme l'ensemble des atomes qui ne feront partie d'aucun ensemble de conséquences quelle que soit la sémantique, raisonnable, choisie.

Définition 2.3.1. *Soient un programme logique Π , sa base de Herbrand associée H et une interprétation I . On dit que $A \subseteq H$ est un modèle non-fondé Π par rapport à I si chaque atome de $p \in A$ satisfait la condition suivante :*

Pour que chaque règle $r \in \Pi$ telle que $p \in tete(r)$, l'une (au moins) des conditions suivantes est vérifiée :

- $\exists a \in \text{corps}(r), a \notin I$;
- $\exists a \in \text{corps}^+(r), a \in A$

Exemple 2.3.1. Soit Π le programme logique suivant :

$$\begin{array}{ll} q. & p \leftarrow q. \\ r \leftarrow s. & r \leftarrow p, q. \\ a \leftarrow r. & a \leftarrow b. \end{array}$$

L'atome s ne peut pas être déduit à partir de ce programme, l'ensemble $\{s\}$ est donc un modèle non-fondé de Π .

Définition 2.3.2. Le plus grand modèle non-fondé de Π par rapport à I , noté $\mathbf{U}_\Pi(I)$, est l'union de tous les modèles non-fondés de Π par rapport à I .

Exemple 2.3.2. Dans le programme Π de l'exemple 2.3.1, l'ensemble $\{b\}$ constitue également un modèle non-fondé. Le plus grand modèle non-fondé de Π est $\{s, b\}$ et représente l'ensemble des atomes qui ne pourront jamais être déduits de ce programme.

2.3.2 Modèle partiel bien-fondé

Le modèle partiel bien-fondé d'un programme logique Π peut être considéré comme l'ensemble des atomes qui feront partie de tous les ensembles de conséquences quelle que soit la sémantique, raisonnable, choisie.

Dans la définition suivante, $\mathbf{T}$ représente les conséquences positives, $\mathbf{U}$ représente les conséquences négatives et $\mathbf{W}$ l'union des deux (ou modèle partiel bien-fondé)

Définition 2.3.3. Soient Π un programme logique et I une interprétation sur $\text{Atome}(\Pi)$. Les transformations $\mathbf{T}_\Pi$, $\mathbf{U}_\Pi$ et $\mathbf{W}_\Pi$ sont définies de la manière suivante :

- $p \in \mathbf{T}_\Pi(I)$ si et seulement si il existe une règle $r \in \Pi$ telle que $p \in \text{tete}(r)$ et que $\forall a \in \text{corps}(r), a \in I$.
- $\mathbf{U}_\Pi(I)$ est le plus grand modèle non-fondé de Π par rapport à I .
- $\mathbf{W}_\Pi(I) = \mathbf{T}_\Pi(I) \cup \neg \mathbf{U}_\Pi(I)$

Exemple 2.3.3. Pour l'exemple 2.3.1 et l'interprétation $I = \emptyset$, les transformations sont les suivantes :

- $\mathbf{T}_\Pi(\emptyset) = \{q\}$
- $\mathbf{U}_\Pi(\emptyset) = \{s, b\}$
- $\mathbf{W}_\Pi(\emptyset) = \{q, \neg s, \neg b\}$

La déduction de q est immédiate puisqu'il s'agit d'un fait. Quant à celle de $\neg s$ et $\neg b$, elles peuvent être réalisées car il n'existe pas de règles dont la tête les contiennent.

Définition 2.3.4. Soit α un ordinal. Les ensembles I_α et I^∞ , dont les éléments sont des littéraux dans la base de Herbrand du programme logique Π , sont définis récursivement par :

- Pour la limite ordinal α , $I_\alpha = \cup_{\beta < \alpha} I_\beta$. On remarque qu'au point 0, $I_0 = \emptyset$.
- Pour les successeurs de $\alpha = \gamma + 1$, $I_{\gamma+1} = \mathbf{W}_\Pi(I_\gamma)$

- Finalement, $I^\infty = \cup_\alpha I_\alpha$

Exemple 2.3.4. *L'ensemble des atomes du plus grand modèle non-fondé est déjà maximal pour $\mathbf{U}_\Pi(\emptyset)$. Par contre, pour $\mathbf{T}_\Pi()$, on a la progression suivante :*

- $\mathbf{T}_\Pi(\emptyset) = \{q\}$
- $\mathbf{T}_\Pi(\{q\}) = \{q, p\}$
- $\mathbf{T}_\Pi(\{p, q\}) = \{p, q, r\}$
- $\mathbf{T}_\Pi(\{p, q, r\}) = \{p, q, r, a\}$
- $\mathbf{T}_\Pi(\{p, q, r, a\}) = \{p, q, r, a\}$

2.3.3 Modèle bien-fondé

Le modèle bien-fondé d'un programme logique Π est l'union du modèle partiel bien-fondé et de l'ensemble non-fondé. On le note $\mathbf{WFS}(\Pi)$.

Définition 2.3.5. *La sémantique bien-fondée d'un programme Π est la signification représentée par le point fixe de $\mathbf{W}_\Pi$, ou la limite I^∞ décrite dans 2.3.4 ; tous les littéraux positifs expriment qu'un atome est vrai et tous les atomes négatifs expriment qu'un atome est faux, et les atomes absents n'ont pas de valeurs de vérité attribuée par la sémantique.*

Exemple 2.3.5. *Ainsi, pour l'exemple 2.3.1, on obtient le modèle bien-fondé suivant : $\mathbf{WFS}(\Pi) = \{a, p, q, r, \neg b, \neg s\}$. On peut noter que ce modèle bien-fondé est total.*

2.4 Sémantique des modèles stables

Les premières définitions d'un programme logique avec sémantique des modèles stables ont été données par V.Lifschitz [Lif99], [GL88]. La programmation logique avec sémantique des modèles stables, Answer Set Programming en anglais, est souvent abrégée par l'acronyme ASP.

Contrairement à un programme PROLOG [SS94], un programme logique avec sémantique des modèles stables peut avoir plusieurs ensembles de conséquences et ces ensembles de conséquences peuvent être contradictoires. Ces ensembles de conséquences, appelés modèles stables, sont ce que l'on pourrait considérer comme les conséquences qu'un agent raisonnable pourrait tirer de ce programme.

2.4.1 Définition des modèles stables

Un modèle stable est un ensemble d'atomes qui est un ensemble de conséquences satisfaisant pour un programme. Par satisfaisant, on entend ici qu'il soit à la fois cohérent avec les croyances exprimées dans le programme mais également tel que chacun de ses atomes soit justifié. Autrement dit, un modèle stable est forcément minimal au sens de l'inclusion - un sous-ensemble d'atomes d'un modèle stable ne peut pas être lui-même un modèle stable.

Définition 2.4.1. La réduction, ou transformation de Gelfond-Lifschitz [GL88], Π^X d'un programme Π et d'un ensemble d'atomes X est définie par :

$$\Pi^X = \{r^+ \mid r \in \Pi \text{ et } \text{corps}^-(r) \cap X = \emptyset\}$$

La réduction est illustrée par l'exemple suivant :

Exemple 2.4.1. Soit le programme Π suivant :

$$\Pi^X = \left\{ \begin{array}{ll} a. & b \leftarrow \text{not } c \\ c \leftarrow \text{not } b & d \leftarrow a, \text{not } b \end{array} \right\}$$

La réduction du programme Π par l'interprétation $X = \{a, c\}$ est :

$$\Pi^X = \left\{ \begin{array}{l} a. \\ c. \\ d \leftarrow a. \end{array} \right\}$$

La réduction du programme Π par l'interprétation $X' = \{a, b\}$ est :

$$\Pi^{X'} = \{ a. \quad b. \}$$

La réduction d'un programme par un ensemble d'atomes permet de donner une définition simple des modèles stables d'un programme logique.

Définition 2.4.2. Soient X un ensemble d'atomes et Π un programme logique. X est un modèle stable du programme Π si et seulement si $CN(\Pi^X) = X$.

Cette définition assure deux propriétés clés. La première propriété est qu'un modèle stable est minimal. Le sous-ensemble d'un modèle stable ne peut être un modèle stable. La seconde propriété est qu'un modèle stable est justifié. Tous les atomes appartenant à un modèle stable sont justifiés par le programme. Si un atome est vrai, il a une bonne raison de l'être.

Exemple 2.4.2. Reprenons l'exemple précédent :

$$CN(\Pi^X) = \{a, c, d\} \neq X$$

X n'est donc pas un modèle stable de Π .

$$CN(\Pi^{X'}) = \{a, b\} = X'$$

X' est donc un modèle stable de Π .

2.4.2 Une définition alternative des modèles stables

La logique du Here-and-There proposée par A. Heyting [Hey30], au départ comme une définition alternative de la logique propositionnelle, permet de donner une autre définition des modèles stables d'un programme logique.

2.4.2.1 La logique du Here-and-There

Dans [Tur03], H.Turner donne une définition alternative des modèles stables d'un programme logique qui est basée sur la logique du Here-and-There. Cette logique est une logique monotone à 3 valeurs où un atome peut avoir trois valeurs de vérité :

- 0 s'il est vrai ;
- 1 s'il est faux ;
- 2 s'il n'est pas faux mais ne peut pas être prouvé comme étant vrai.

Cette logique représente les interprétations d'un programme logique sous la forme d'une paire (X, Y) d'ensembles d'atomes appelée HT-interprétation. Les valeurs de vérité sont matérialisées par une paire d'ensembles d'atomes (X, Y) telle que $X \subseteq Y$. Soit a un atome, si $a \in X$ alors sa valeur de vérité est 0. Si $a \notin X$ et $a \notin Y$ alors sa valeur de vérité est 1. Si $a \notin X$ et $a \in Y$ alors sa valeur de vérité est 2. L'ensemble d'atomes Y représente l'ensemble des conséquences plausibles du programme. L'ensemble d'atomes X représente un ensemble de conséquences du programme Π sachant Y .

Exemple 2.4.3. Soit Π le programme suivant :

$$\Pi = \{a \leftarrow \text{not } b. \quad b \leftarrow \text{not } a.\}$$

Plusieurs exemples de HT-interprétations sont : $(\{a\}, \{a, b\})$, $(\{b\}, \{b\})$ ou $(\{\emptyset\}, \{\emptyset\})$

On donne la définition d'un modèle pour la HT-logique, appelé HT-modèle.

Définition 2.4.3. Soient Π un programme logique et (X, Y) une HT-interprétation de Π . (X, Y) est un HT-modèle de Π si et seulement si $Y \models \Pi$ et $X \models \Pi^Y$. L'ensemble des HT-modèles de Π est noté $HT(\Pi)$.

Exemple 2.4.4. Continuons l'exemple 2.4.3.

L'ensemble des atomes cohérents pour toutes les règles de Π sont : $\{a\}$, $\{b\}$ et $\{a, b\}$. Il n'est pas possible d'avoir un HT-modèle ayant $\emptyset$ pour deuxième élément parce que l'absence de a entraîne la déduction de b (par la règle $a \leftarrow \text{not } b.$) et vice versa.

Pour $Y = \{a\}$, $\Pi^Y = \{a \leftarrow\}$ alors $X = \{a\}$ est le seul ensemble d'atomes qui soit cohérent avec Π^Y .

De la même manière, $Y = \{b\}$ alors $X = \{b\}$ est le seul ensemble d'atomes qui soit cohérent avec Π^Y .

Pour $Y = \{a, b\}$, $\Pi^Y = \emptyset$ alors tous les ensembles $X \subseteq Y$ sont cohérents pour Π^Y .

Enfin $HT(\Pi) = \{(\{a\}, \{a\}), (\{b\}, \{b\}), (\emptyset, \{a, b\}), (\{a\}, \{a, b\}), (\{b\}, \{a, b\}), (\{a, b\}, \{a, b\})\}$

Une propriété notable des HT-modèles est donnée dans [Tur03].

Proposition 2.4.1. Soient Π et Π' deux programmes logiques, on a

$$HT(\Pi \cup \Pi') = HT(\Pi) \cap HT(\Pi')$$

2.4.2.2 Une définition des modèles stables en termes de HT-modèles

Comme nous l'avons vu précédemment, un modèle stable est un ensemble d'atomes qui doit pouvoir être considéré comme étant cohérent avec les règles d'un programme logique et, en même temps, justifié vis-à-vis de lui. Si (X, Y) est un HT-modèle du programme logique Π , la définition de HT-modèle assure la cohérence de l'ensemble Y mais n'assure pas le fait que les atomes soient justifiés vis-à-vis du programme Π . On peut imaginer que si (Y, Y) est le seul HT-modèle dont le second élément est Y , les atomes appartenant à Y possèdent une justification. C'est exactement le sens de la définition ci-dessous :

Définition 2.4.4. *Soit Π un programme logique et Y un ensemble d'atomes de $\text{Atome}(\Pi)$, un ensemble Y d'atomes est un modèle stable de Π si et seulement si (Y, Y) est le seul HT-modèle de Π dont le second élément est Y .*

Exemple 2.4.5. *Continuons l'exemple 2.4.3.*

L'ensemble des HT-modèles de Π est : $HT(\Pi) = \{(\{a\}, \{a\}), (\{b\}, \{b\}), (\emptyset, \{a, b\}), (\{a\}, \{a, b\}), (\{b\}, \{a, b\}), (\{a, b\}, \{a, b\})\}$.

On note que $\emptyset$ n'apparaît pas en tant que second élément d'un HT-modèle. $\{a, b\}$ apparaît plusieurs fois. Mais $\{a\}$ et $\{b\}$ n'apparaissent chacun qu'une seule fois en tant que second élément d'un HT-modèle respectivement dans $(\{a\}, \{a\})$ et $(\{b\}, \{b\})$ et sont donc des modèles stables.

Ce qui correspond bien à la définition classique des modèles stables.

2.4.3 Différentes utilisations de programme ASP

La programmation logique avec sémantique des modèles stables a été utilisé pour la résolution nombreux problèmes fortement combinatoires. Quelques exemples : reconstitution des échanges linguistiques entre civilisations [ELR05], système de prévision des crues [BnBPW04], planification [Lif99], configuration [SN98] avec, par exemple, le calcul de dépendances de paquetages dans la distribution Debian de Linux [Syr99].

2.5 Langage ASP et solveurs

La résolution des programmes logiques avec sémantique des modèles stables s'effectue au travers de programmes appelés solveurs ASP.

Ceux-ci proposent plusieurs raccourcis et instructions pour faciliter l'écriture. Par exemple, des variables peuvent être utilisées pour réduire l'écriture d'une collection de règles similaires. Un programme logique n'est donc pas utilisable dans sa forme première.

C'est pour cela que la résolution est effectuée par deux tâches distinctes. Dans un premier temps, le programme logique est instancié par un préprocesseur. Ensuite, le solveur résout ce programme instancié.

Nous présentons d'abord les instructions et raccourcis présentés par les solveurs. Nous ferons une présentation de la proposition faite par J. Delgrande et al. pour gérer des

préférences entre règles. Puis, nous rappelons quelques uns des principaux solveurs de modèles stables.

2.5.1 Langage ASP et instructions supplémentaires

Nous introduisons tout d'abord quelques instructions offertes par Lparse [Syr02] et reprise par Gringo [GKO⁺09] permettant d'écrire de manière compacte certains ensembles de règles.

2.5.1.1 Gestion des cardinalités

Prenons le problème qui consiste à demander qu'au plus deux atomes parmi $\{a, b, c, d\}$ soient présents dans les modèles stables du programme. Une solution intuitive consiste en l'interdiction, par le biais de contraintes, de toutes solutions contenant trois de ces atomes. Ainsi,

$$\begin{array}{ll} \leftarrow a, b, c. & \leftarrow a, b, d. \\ \leftarrow a, c, d. & \leftarrow b, c, d. \end{array}$$

Cependant, l'écriture de telles règles risque de devenir fastidieuse si le nombre d'atomes dans l'ensemble à restreindre augmente.

La programmation ASP autorise une écriture compacte de cette contrainte :

$$\leftarrow \{a, b, c, d\} 2.$$

On peut également demander à un minimum d'atomes de l'ensemble d'être vrais. Par exemple, si l'on veut qu'au moins deux des atomes de l'ensemble $\{a, b, c, d\}$ soient vrais, il est possible d'écrire :

$$\leftarrow 1 \{a, b, c, d\}.$$

De manière générale, une règle de cardinalité est décrite de la manière suivante :

$$k \{a_1, \dots, a_m, \text{not } b_1, \dots, \text{not } b_n\} l$$

Cet atome sera vrai si au moins k et au plus l atomes parmi l'ensemble sont vrais. Il est possible d'omettre une des deux restrictions k ou l .

Il est également possible de donner un poids à chaque variable. Dans l'instruction suivante, les X_i sont des poids :

$$k [a_1 = X_1, \dots, a_m = X_m, \text{not } b_1 = X_{m+1}, \dots, \text{not } b_n = X_{m+n}] l$$

Dans ce cas, le poids cumulé des atomes vrais dans l'ensemble $\{a_1, \dots, a_m, \text{not } b_1, \dots, \text{not } b_n\}$ doit être supérieur à k et inférieur à l pour que l'atome soit vrai.

2.5.1.2 Optimiser les modèles stables

Une question proche de celle de la cardinalité est celle de la minimisation, ou de la maximisation, des modèles stables. Par exemple, pour un problème de configuration, on préférera probablement garder, parmi toutes les solutions, uniquement celles possédant le moins de pièces.

Formellement, une instruction de minimisation est écrite :

$$\text{minimize}\{a_1, \dots, a_m, \text{not } b_1, \dots, \text{not } b_n\}$$

Cette instruction a pour effet de ne conserver, parmi l'ensemble des modèles stables du reste du programme, que ceux ayant le minimum d'atomes vrais parmi l'ensemble $\{a_1, \dots, a_m, \text{not } b_1, \dots, \text{not } b_n\}$. L'instruction duale de celle-ci est :

$$\text{maximize}\{a_1, \dots, a_m, \text{not } b_1, \dots, \text{not } b_n\}$$

qui conserve les modèles stables possédant le plus d'atomes dans ce même ensemble.

L'optimisation peut également se faire en fonction d'une certaine pondération des atomes. Dans l'instruction suivante, les X_i sont des poids, ainsi :

$$\text{minimize}[a_1 = X_1, \dots, a_m = X_m, \text{not } b_1 = X_{m+1}, \dots, \text{not } b_n = X_{m+n}]$$

minimise le poids cumulé des atomes de l'ensemble.

2.5.1.3 Variables et domaines de définition

La définition de modèles stables telle qu'elle est présentée dans la section 2.4.2 pour les programmes logiques sous-entend que ces programmes soient instanciés. Il est possible d'étendre cette définition aux programmes avec variables à l'aide de la notion de modèle de Herbrand [Nie99]

Il est nécessaire, afin de conserver la décidabilité et la finitude, de ne pas avoir de symboles de fonction.

Exemple 2.5.1. Soit le programme logique Π_1 suivant :

$$\Pi_1 = \left\{ \begin{array}{ll} p(1). & p(2). \\ p(3). & q(X) \leftarrow p(X). \end{array} \right\}$$

L'univers de Herbrand de Π_1 est $\{1, 2, 3\}$. Sa base de Herbrand est $\{p(1), p(2), p(3), q(1), q(2), q(3)\}$. L'instantiation de la règle $q(X) \leftarrow p(X)$ se fera pour toutes les variables susceptibles de rendre $p(X)$ vraie. Ainsi, l'instantiation du programme Π sera :

$$\begin{array}{ll} p(1). & q(1) \leftarrow p(1). \\ p(2). & q(2) \leftarrow p(2). \\ p(3). & q(3) \leftarrow p(3). \end{array}$$

Il est possible d'associer à une variable donnée un ensemble d'atomes. Ceci aura pour effet de forcer l'instantiation de chaque règle où une telle variable apparaît par l'ensemble des atomes qui lui sont associés. Cette déclaration de domaine prend la forme de l'instruction :

$$\#domain\ a(X).$$

où la variable X est instanciée pour toutes les valeurs telles que $a(X)$ est vraie.

Exemple 2.5.2. Soit le programme logique Π_2 suivant :

$$\Pi_2 = \left\{ \begin{array}{l} p(1). \qquad \qquad p(2). \quad p(3). \\ \#domain\ p(X). \\ a(X). \end{array} \right\}$$

dont l'instantiation est :

$$\begin{array}{l} p(1). \quad p(2). \quad p(3). \\ a(1). \quad a(2). \quad a(3). \end{array}$$

2.6 Les préférences dans un programme logique

2.6.1 Théorie et définition

Les préférences jouent un rôle crucial dans le cadre de la représentation de croyances. En effet, les croyances ne possèdent pas toujours le même degré d'importance ou de crédibilité. Dans le cadre légal par exemple, la loi française punit le téléchargement par la coupure de l'accès internet alors que la loi européenne stipule qu'aucun citoyen ne peut se voir priver de son accès internet. Pour résoudre un tel conflit, il est nécessaire d'établir une préférence entre ces deux lois. Dans le cas présent, la loi européenne a plus de valeur que la loi française.

L'expression de préférences permet de choisir entre plusieurs solutions possibles celles qui correspondent le mieux ou qui sont les plus sûres. Ainsi, l'expression de préférences dans un programme basique n'a pas de sens, car un programme basique ne possède, de toutes façons, qu'un seul ensemble de conséquences. Les préférences sont donc exprimées dans le cas de programmes généraux ou de programmes étendus.

Afin de traiter ce genre de situation, les programmes logiques incorporent des préférences entre leurs règles. Soient r_1 et r_2 deux règles d'un programme logique, on notera $r_1 < r_2$ le fait que la règle r_1 est préférée à la règle r_2 . Ainsi un programme logique ordonné est une paire $(\Pi, <)$ où Π est un programme logique et $<$ un ensemble de préférences exprimés sur les règles de Π .

Exemple 2.6.1. Considérons le programme logique ordonné $(\Pi, <)$ avec :

$$\Pi = \{ \ r_1 : a \quad r_2 : b \leftarrow a, not \neg c \quad r_3 : \neg c \leftarrow not\ b. \ }$$

et la relation de préférence $<$ définie par $\{r_2 < r_3\}$.

La déduction de a ne pose pas de problème. Sans préférences, ce programme Π aurait deux modèles stables : $X_1 = \{a, b\}$ et $X_2 = \{a, \neg c\}$ correspondant respectivement au choix

de préférer r_2 à r_3 et r_3 à r_2 . Ainsi lorsque la préférence $r_2 < r_3$ est prise en compte, il semble normal que $(\Pi, <)$ ne possède qu'un seul modèle stable, à savoir $\{a, b\}$.

Pour formaliser cette notion J.Delgrande [Del97] a proposé la définition d'un modèle stable préservant les préférences exprimées. Cette définition est basée sur la notion d'ordre d'application des règles. Par exemple, dans l'exemple précédent, on a appliqué r_1 puis r_2 et r_3 . L'ordre d'application a respecté la préférence : r_2 a été traitée avec r_3 . Cette définition garantit que la question de l'applicabilité des règles doit être réglée dans un ordre respectant les préférences exprimées.

Exemple 2.6.2. Pour le programme logique ordonné $(\Pi, <)$ de l'exemple 2.6.1, le modèle stable X_1 ne peut se calculer qu'en résolvant la question de l'applicabilité des règles dans l'ordre $r_1 < r_2 < r_3$. Pour X_2 , les seuls ordres possibles sont $r_1 < r_3 < r_2$, $r_3 < r_1 < r_2$ et $r_3 < r_2 < r_1$ et aucun ne respecte la préférence $\{r_2 < r_3\}$.

Il est clair que seul X_1 respecte la préférence exprimée par $\{r_2 < r_3\}$.

Ainsi, pour formaliser ces notions, il faut définir la notion de séquences fondées.

Un ordre d'application des règles et appelé séquence est noté $< r_i >_{i \in I}$. Il paraît normal qu'une règle ne puisse pas être appliquée ou défaite tant que les atomes qui permettent l'application ou le rejet de la règle n'ont pas été déduits. Une séquence qui respecte cette règle de bon sens est appelée séquence fondée.

Définition 2.6.1 (Séquence fondée). Soit r_i un ensemble de règles. Une séquence $< r_i >_{i \in I}$ est fondée si et seulement si, pour tout $i \in I$, $\{tete(r_j) \mid j < i\}$ est incohérent ou si $corps^+(r_i) \subseteq \{tete(r_j) \mid j < i\}$

Exemple 2.6.3. Pour le programme logique Π :

$$\Pi = \{ r_1 : a. \quad r_2 : b \leftarrow a, not \neg c. \quad r_3 : \neg c \leftarrow not b. \}$$

L'ensemble des séquences fondées sont : $< r_1, r_2, r_3 >$ (qui génère $X_1 = \{a, b\}$), $< r_1, r_3, r_2 >$ (qui génère $X_2 = \{a, \neg c\}$), $< r_3, r_1, r_2 >$, $< r_3, r_2, r_1 >$.

r_2 ne peut pas être appliqué avant r_1 car il faut pouvoir déduire a d'abord.

2.6.1.1 Modèle stable $<$ -préservant

On étend ainsi la définition de modèle stable au cas où des préférences sont exprimées entre les règles.

L'ensemble des règles génératrices est l'ensemble des règles $r \in \Pi$ qui ne sont pas défaites par X et dont le corps de r est vrai pour X .

Définition 2.6.2 (Règles génératrices). Soient Π un programme logique et X un modèle stable de Π . L'ensemble Γ_{Π}^X de toutes les règles génératrices de X est donné par :

$$\Gamma_{\Pi}^X = \{r \in \Pi \mid corps^+(r) \subseteq X \text{ et } corps^-(r) \cap X = \emptyset\}$$

Exemple 2.6.4. Dans le programme logique Π donné dans l'exemple 2.6.3, l'ensemble des règles génératrices de X_1 et $\Gamma_{\Pi}^{X_1} = \{r_1, r_2\}$ et l'ensemble des règles génératrices de X_2 et $\Gamma_{\Pi}^{X_2} = \{r_1, r_3\}$.

Il est maintenant possible de donner les différentes définitions de modèles stables qui respectent les préférences exprimées. De telles modèles stables sont appelés modèles stables $<$ -préservants. Pour simplifier, un modèle stable $<$ -préservant est un modèle stable qui possède une séquence fondée dont l'ordre respecte les préférences exprimées. Plusieurs définitions de la $<$ -préservation ont été données suivant différentes visions du respect de l'ordre. La première de ces définitions a été donnée par Delgrande dans [DST02].

Définition 2.6.3 ($<_D$ -préservation). Soient $(\Pi, <)$ un programme logique ordonné et X un modèle stable cohérent de Π .

Alors X est appelé $<_D$ -préservant si il existe une séquence fondée $\langle r_i \rangle_{i \in I}$ de Γ_{Π}^X telle que, pour tout $i, j \in I$, on a :

1. $\text{corps}^+(r_i) \subseteq \{\text{tete}(r_k) \mid k < i\}$
2. Si $r_i < r_j$ alors $j < i$
3. Si $r_i < r'$ et $r' \in \Pi \setminus \Gamma_{\Pi}^X$, alors
 - $\text{corps}^+(r') \subseteq X$ ou
 - $\text{corps}^-(r') \cap \{\text{tete}(r_k) \mid k < i\} \neq \emptyset$

Wang et al. [WZL00] ont proposé une autre définition d'un modèle stable préservant l'ordre d'application.

Définition 2.6.4 ($<_W$ -préservation). Soient $(\Pi, <)$ un programme logique ordonné et X un modèle stable cohérent de Π .

Alors X est appelé $<_W$ -préservant si il existe une séquence fondée $\langle r_i \rangle_{i \in I}$ de Γ_{Π}^X telle que, pour tout $i, j \in I$, on a :

1. – $\text{corps}^+(r_i) \subseteq \{\text{tete}(r_j) \mid j < i\}$ ou
– $\text{tete}(r_i) \in \{\text{tete}(r_j) \mid j < i\}$
2. Si $r_i < r_j$ alors $j < i$
3. Si $r_i < r'$ et $r' \in \Pi \setminus \Gamma_{\Pi}^X$, alors
 - $\text{corps}^+(r') \not\subseteq X$ ou
 - $\text{corps}^-(r') \cap \{\text{tete}(r_j) \mid j < i\} \neq \emptyset$ ou
 - $\text{tete}(r') \in \{\text{tete}(r_j) \mid j < i\}$

De la même manière, Brewka et Eiter [BE99] ont également proposé une définition de modèle stable préservant l'ordre d'application.

Définition 2.6.5 ($<_B$ -préservant). Soient $(\Pi, <)$ un programme logique ordonné et X un modèle stable cohérent de Π .

Alors X est appelé $<_B$ -préservant si il existe une séquence fondée $\langle r_i \rangle_{i \in I}$ de Γ_{Π}^X telle que, pour tout $i, j \in I$, on a :

1. Si $r_i < r_j$ alors $j < i$
2. Si $r_i < r'$ et $r' \in \Pi \setminus \Gamma_{\Pi}^X$, alors
 - $\text{corps}^+(r') \not\subseteq X$ ou
 - $\text{corps}^-(r') \cap \{\text{tete}(r_j) \mid j < i\} \neq \emptyset$ ou
 - $\text{tete}(r') \in X$

Il a été montré dans [DST02] que tout modèle stable $<_D$ -préservant est un modèle stable $<_W$ -préservant. Il a également été montré que tout modèle stable $<_W$ -préservant est un modèle stable $<_B$ -préservant.

2.7 Solveurs ASP

Les programmes ainsi instanciés sont ensuite résolus par des programmes appelés solveurs de modèles stables : CLASP [GKNS07a], NoMore [AKL02], ASSAT [LZ04], CMODELS [GLM04], Smodels [Sim02].

2.7.1 Lparse

Lparse [Syr02] n'est pas un solveur ASP à proprement parler. C'est un programme qui permet de traduire les programmes logiques dans un format utilisable par les solveurs. Il instancie les programmes avec variables et effectue les déductions triviales du programme.

2.7.2 Smodels

Smodels [Sim02] est la première et la plus simple des méthodes de calcul de modèles stables. C'est un algorithme de Branch and Bound qui construit, au fur et à mesure, un ensemble d'atomes A représentant un modèle stable potentiel. Pour ce faire, il utilise les fonctions suivantes :

- $\text{expand}(A)$ qui renvoie les conséquences immédiates de A ;
- $\text{conflict}(A)$ qui détecte les incohérences ayant pu survenir après une expansion. La fonction renvoie *Vrai* en cas de conflit et *Faux* dans le cas contraire ;
- $\text{heuristic}(A)$ qui choisit l'atome qui doit être ajouté à l'ensemble A . Le choix, heuristique, de l'atome permettant le plus de déductions réduit considérablement la taille de l'espace de recherche en découvrant les conflits inhérents à une branche.

Définition 2.7.1. Soient L un ensemble de littéraux et A un ensemble d'atomes. L couvre A si et seulement si $A \subseteq \text{Atome}(L)$.

L'algorithme Smodels est présenté dans la figure 5.

2.7.2.1 noMoRe

noMoRe est un solveur ASP basé sur un principe de coloration de graphes non-standards présenté dans [AKL02]. Le programme logique est traduit sous la forme d'un graphe représentant les relations de dépendance entre les règles. La coloration du graphe

Algorithme 5 $Smodels(A)$

```

 $A \leftarrow expand(A)$ 
si  $conflict(A)$  alors
  renvoyer  $Faux$ 
sinon si  $A$  couvre  $Atome(E)$  alors
  renvoyer  $Vrai$ 
sinon
   $x \leftarrow heuristic(A)$ 
  si  $Smodels(A \cup \{x\})$  alors
    renvoyer  $Vrai$ 
  sinon
    renvoyer  $Smodels(A \cup \{not\ x\})$ 
finsi
finsi

```

permet de répartir les règles en deux ensembles : règles appliquées et règles bloquées. L'ensemble des règles appliquées permet ensuite de déduire l'ensemble des atomes qui constituera le modèle stable.

On construit pour cela le graphe suivant :

Définition 2.7.2 (Graphe de dépendance de règles). *Soit Π un programme logique. Le graphe de dépendance de règles (ou RDG, pour Rule Dependency Graph) $\Gamma_\Pi = (\Pi, E_0, E_1)$ du programme Π est un graphe orienté étiqueté avec :*

- $E_0 = \{(r, r') \mid r, r' \in \Pi, tete(r) \in corps^+(r')\}$
- $E_1 = \{(r, r') \mid r, r' \in \Pi, tete(r) \in corps^-(r')\}$

Un i -sous graphe (V, E) de Γ est un sous-graphe de Γ avec $E \subseteq E_i$ pour $i \in \{0, 1\}$.

Exemple 2.7.1. Soit $\Pi_1 = \{r_1, \dots, r_6\}$ un programme logique où :

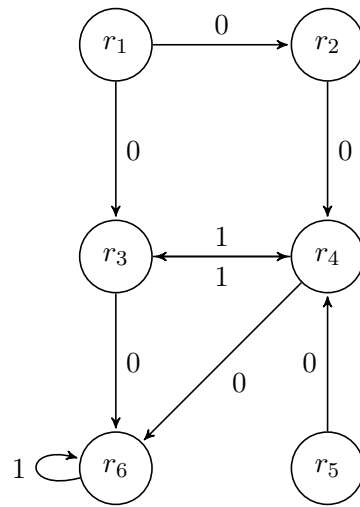
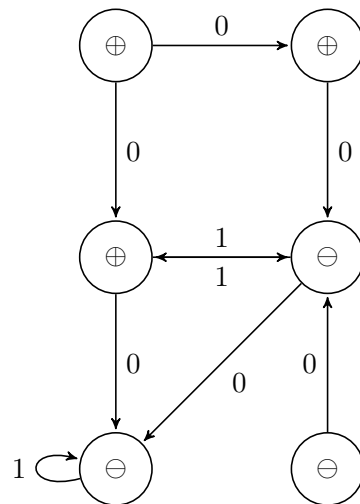
$$\begin{array}{lll}
 r_1 & : & p \leftarrow \\
 r_2 & : & b \leftarrow p. \\
 r_3 & : & f \leftarrow b, not\ f'. \\
 r_4 & : & f' \leftarrow p, not\ f. \\
 r_5 & : & b \leftarrow m \\
 r_6 & : & x \leftarrow f, f', not\ x.
 \end{array}$$

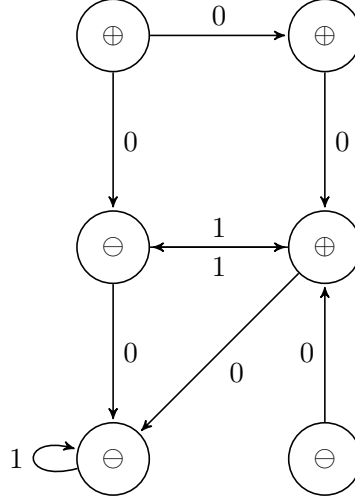
Le programme Π_1 est transformé en graphe de dépendance de règles Γ_{Π_1} présenté dans la figure 2.1 :

À partir de ce graphe là, on va réaliser un coloriage de Γ_Π . Un coloriage C est une fonction $C : \Pi \rightarrow \{\oplus, \ominus\}$. Intuitivement, les couleurs $\oplus$ et $\ominus$ indiquent si une règle est supposée appliquée ou bloquée. On définit $C_\oplus = \{r \mid C(r) = \oplus\}$ et $C_\ominus = \{r \mid C(r) = \ominus\}$.

Dans un premier temps, pour tout modèle stable X , on définit un coloriage admissible C de Γ comme étant $C = (R_\Pi(X), \Pi \setminus R_\Pi(X))$. L'ensemble des coloriages admissibles de Γ_Π sont notés $AC(\Pi)$. On note également $AC_\Pi(C)$ l'ensemble des coloriages admissibles compatibles avec C . En d'autres termes, $AC_\Pi(C) = \{C' \in AC(\Pi) \mid C \subseteq C'\}$.

Exemple 2.7.2. Reprenons l'exemple 2.7.1. Soit C_2 le coloriage partiel $C_2 = (\{r_1, r_2\}, \{r_6\})$. On peut voir que $AC_{\Pi_1}(C_2) = AC(\Pi) = \{(\{r_1, r_2, r_3\}, \{r_4, r_5, r_6\}), (\{r_1, r_2, r_4\}, \{r_3, r_5, r_6\})\}$. Ce coloriage est présenté dans les figures 2.2 et 2.3.

FIGURE 2.1 – Graphe de dépendance de Π_1 FIGURE 2.2 – Coloriage correspondant au modèle b, p, f

FIGURE 2.3 – Coloriage correspondant au modèle b, p, f'

On définit l'ensemble des modèles stables compatibles avec un coloriage comme étant $AS_{\Pi}(C) = \{X \in AS(\Pi) \mid C_{\oplus} \subseteq R_{\Pi}(X) \text{ et } C_{\ominus} \cap R_{\Pi}(X) = \emptyset\}$.

Exemple 2.7.3. On continue l'exemple 2.7.2, et on constate que $AS_{\Pi_1}(C_2) = AS(\Pi_1) = \{\{b, p, f\}, \{b, p, f'\}\}$.

À partir de ces définitions, on peut préciser les liens entre les règles déjà appliquées ou bloquées et les autres.

Définition 2.7.3. Soit $\Gamma = (\Pi, E_0, E_1)$ le graphe de dépendance de règles du programme logique Π , C un coloriage partiel de Γ et r une règle de Π , on dit que :

1. r est supportée dans (Γ, C) , si $\text{corps}^+(r) \subseteq \{\text{tete}(r') \mid (r', r) \in E_0, r' \in C_{\oplus}\}$;
2. r n'est pas supportée dans (Γ, C) , si $\{r' \mid (r', r) \in E_0, \text{tete}(r') = q\} \subseteq C_{\ominus}$ pour un $q \in \text{corps}^+(r)$;
3. r est bloquée dans (Γ, C) , si $r' \in C_{\oplus}$ pour un $(r', r) \in E_1$;
4. r est bloquée dans (Γ, C) , si $r' \in C_{\ominus}$ pour tout $(r', r) \in E_1$.

On notera $S(\Gamma, C), \bar{S}(\Gamma, C), B(\Gamma, C)$ et $\bar{B}(\Gamma, C)$ pour l'ensemble des règles supportées, non-supportées, bloquées et non-bloquées.

Théorème 2.7.1. Soit Γ le RDG du programme logique Π et C un coloriage partiel de Γ . Alors, on a pour tout $X \in AS_{\Pi}(C)$ que :

1. $S(\Gamma, C) \cap \bar{B}(\Gamma, C) \subseteq R_{\Pi}(X)$
2. $\bar{S}(\Gamma, C) \cap B(\Gamma, C) \subseteq \Pi \subseteq R_{\Pi}(X)$

Si C est admissible, on a pour $\{X\} = AS_{\Pi}(C)$ que :

3. $S(\Gamma, C) \cap \bar{B}(\Gamma, C) = R_{\Pi}(X)$

$$4. \bar{S}(\Gamma, C) \cap B(\Gamma, C) = \Pi \subseteq R_\Pi(X)$$

Exemple 2.7.4. Continuons l'exemple 2.7.3. Pour tout $X \in AS_{\Pi_1}(C) = \{\{b, p, f\}, \{b, p, f'\}\}$:

- $S(\Gamma_{\Pi_1}, C_2) \cap \bar{B}(\Gamma_{\Pi_1}, C_2) = \{r_1, r_2\} \subseteq R_{\Pi_1}(X)$
- $\bar{S}(\Gamma_{\Pi_1}, C_2) \cup B(\Gamma_{\Pi_1}, C_2) = \{r_5\} \subseteq \Pi_1 \setminus R_{\Pi_1}(X)$

2.7.3 Kernel Normal Form

Dans [GM05], Grossi et Marchi proposent un autre algorithme basé sur la représentation des programmes logiques sous forme de graphes. À la différence de noMoRe, cet algorithme est orienté sur les atomes et non sur les règles. Il est à noter qu'une première approche sur ce thème a été formulée par Viet Phan Luong dans [Luo97].

Définition 2.7.4. Un programme logique Π est en Kernel Normal Form (ou Π est un Kernel-Programme) si et seulement si les conditions suivantes sont vérifiées :

- Π est WFS-irréductible, i.e., $WFS(\Pi) = (\emptyset, \emptyset)$;
- Toutes les règles ont un corps composé uniquement de littéraux négatifs ;
- Tous les atomes de Π apparaissent dans le corps d'au moins une règle.

Une conséquence immédiate de cette définition est que dans un Kernel-Programme, tous les atomes apparaissent au moins une fois en tant que tête de règle. De la manière, un Kernel-Programme ne possède pas de faits.

De manière générale, tous les programmes logiques peuvent être représentés sous la forme d'un graphe orienté, fini et étiqueté appelé Graphe de Dépendance Étendu (ou EDG pour Extended Dependency Graph). Nous ne présenterons ici qu'une partie de sa définition correspondant au cas des Kernel-programmes.

Définition 2.7.5 (Kernel-EDG). Pour un Kernel-Programme Π_K , le graphe de dépendance étendu $EDG(\Pi_K)$ est le graphe (V, E) défini comme suit :

1. Pour chaque règle dans Π_K , $a_i^{(k)} \in V$, où a_i est le nom de la tête et k est l'index correspondant aux règles définissant a_i ;
2. Pour tout $a_j \in V$, $(a_j, a_i^{(k)}) \in E$ si et seulement si a_j apparaît dans le corps de la k -ème règle définissant a_i .

Nous donnons maintenant la définition d'un coloriage admissible pour un EDG et sa signification vis-à-vis des modèles stables de Π .

Définition 2.7.6 (Coloriage Rouge-Vert). Un coloriage R - V pour le graphe EDG est une fonction $\rho : V \rightarrow \{R, V\}$. De plus, un coloriage est admissible si les conditions suivantes sont vérifiées :

1. Si $\exists i$ tel que $\rho(i) = V$, alors $\forall j \in \mathcal{N}(i) \rho(j) = R$;
2. Si $\exists i$ tel que $\rho(i) = R$, alors $\exists j \in \mathcal{N}^+(i)$ tel que $\rho(j) = V$.

En d'autres termes : il ne peut pas y avoir deux noeuds verts adjacents (V-V violation) et un noeud rouge ne peut pas avoir tous ces voisins entrants rouges (R-R violation).

Théorème 2.7.2. *Une interprétation S est un modèle stable de Π si et seulement si il existe un coloriage associé col_S qui est un EDG admissible.*

Exemple 2.7.5. *Considérons le programme logique Π_2 :*

$$\Pi_2 = \left\{ \begin{array}{ll} r_1 : p \leftarrow \text{not } a. & r_2 : p \leftarrow \text{not } b. \\ r_3 : a \leftarrow \text{not } b. & r_4 : b \leftarrow \text{not } a. \end{array} \right\}$$

Ce programme possède deux modèles stables : $S_1 = \{p, a\}$ et $S_2 = \{p, b\}$. Le graphe de dépendance étendu correspondant à Π_2 est présenté dans la figure 2.1 :

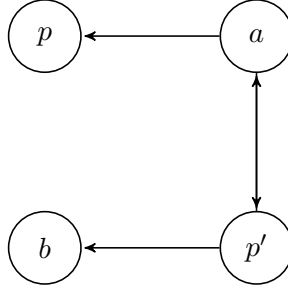


TABLE 2.1 – Graphe de dépendance étendu du programme Π_2

Les deux seuls coloriages admissibles de ce graphe sont $\rho(p) = \rho(b) = V$ et $\rho(p') = \rho(a) = V$.

2.7.3.1 L'algorithme de coloriage

L'algorithme prend en entrée un Graphe de dépendance étendu $G = (V, E)$ et retourne les coloriages admissibles. À chaque étape $V = C \cup \bar{C}$, où C et $\bar{C} = V \setminus C$ représente respectivement l'ensemble des noeuds déjà coloriés et les noeuds restants à colorier. i_V et i_R signifie respectivement que i a été colorié en Vert ou en Rouge.

L'algorithme de coloriage est présenté dans la figure 6.

2.7.3.2 ASSAT

Dans [LZ02], Lin et Zhao proposent un algorithme de calcul des modèles stables basé sur la traduction du problème en problème SAT. Cette traduction est basée sur la complétion de Clark [Fag94] et la transcription des formules de boucle [LL03].

2.7.3.3 La complétion de Clark

Définition 2.7.7. *Pour tout atome $p \in \text{Atome}(\Pi)$, soient $p \leftarrow G_1, \dots, p \leftarrow G_n$ l'ensemble des règles ayant p pour tête dans Π , alors $p \equiv G_1 \vee \dots \vee G_n$ fait partie de la complétion de Clark de Π . En particulier, si $n = 0$, $p \equiv \text{false}$.*

Algorithme 6 *coloriage_EDG*($G = (C \cup \bar{C}, E)$)

```

si  $\exists i \in \bar{C}$  avec  $\deg(i, \bar{C}) \geq 1$  alors
     $i \leftarrow \text{choisir\_un\_noeud\_a\_colorier}(\bar{C})$ 
sinon
     $\text{final\_check}(G)$ ;
fin
si colorier en R tous les  $j \in \mathcal{N}(i)$  n'induit pas de R-R violation alors
     $\text{coloriage\_EDG}( (C \cup \{i_V\} \cup \{j_R : j \in \mathcal{N}(i)\} \cup \bar{C}, E) )$ 
fin
si colorier en rouge  $i$  induit une R-R violation alors
    renvoyer ;
sinon
     $\text{coloriage\_EDG}( (C \cup \{i_R\} \cup \bar{C}, E) )$ 
fin

```

Exemple 2.7.6. Soit le programme logique Π suivant : $\Pi = \{a \leftarrow b. \quad a \leftarrow c, \text{not } b. \quad b \leftarrow \text{not } c, \text{not } d.\}$

La complétion de Clark de Π est : $\{a \equiv (b) \vee (c \wedge \neg b), b \equiv (\neg c \wedge \neg d)\}$.

Si S est un modèle stable de Π , alors S est un modèle de sa complétion de Clark alors que l'inverse n'est pas forcément vrai.

2.7.3.4 Les formules de boucles

Lorsque l'on représente les dépendances positives d'un programme sous la forme d'un graphe, les atomes peuvent se montrer dépendants entre eux.

Pour expliquer le principe, si les règles suivantes font partie d'un même programme : $\{a \leftarrow b. \quad b \leftarrow a.\}$ alors la présence de a dans un modèle stable implique la présence de b et vice versa.

Définition 2.7.8. Étant donné un programme logique Π qui peut contenir des contraintes, un sous-ensemble non-vide L de $\text{Atome}(\Pi)$ est appelé boucle de Π si pour tout p et q dans L , il existe un chemin > 0 allant de p à q dans le graphe de dépendance positive de Π , tel que tous les noeuds dans le chemin sont dans L .

On sépare les atomes appartenant à une boucle de ceux qui n'appartiennent à aucune boucle en deux ensembles qui forment une partition d' $\text{Atome}(\Pi)$. Soient Π un programme logique et une boucle L , on définit les ensembles de règles suivants :

$$R^+(L, \Pi) = \{p \leftarrow G \mid (p \leftarrow G) \in \Pi, p \in L, (\exists q). q \in G \wedge q \in L\}$$

$$R^-(L, \Pi) = \{p \leftarrow G \mid (p \leftarrow G) \in \Pi, p \in L, \neg(\exists q). q \in G \wedge q \in L\}$$

Aucun atome appartenant à $R^+(L, \Pi)$ ne peut appartenir à un modèle stable de Π que s'il y a une justification à sa présence et cette justification appartient forcément à $R^-(L, \Pi)$.

Définition 2.7.9. Soit Π un programme logique et L une boucle de Π . Soit $R^-(L)$ l'ensemble de règles suivant :

$$\begin{aligned} p_1 &\leftarrow G_{11}, \dots, p_1 \leftarrow G_{1k_1}, \\ &\vdots \\ p_n &\leftarrow G_{n1}, \dots, p_1 \leftarrow G_{nk_n}. \end{aligned}$$

Alors la formule de boucle associée à L dans Π , notée $LF(L, \Pi)$, est l'implication suivante :

$$\neg[G_{11} \vee \dots \vee G_{1k_1} \vee \dots \vee G_{n1} \vee \dots \vee G_{nk_n}] \supset \bigwedge_{p \in L} \neg p.$$

Exemple 2.7.7. Reprenons l'exemple tiré de [LZ02]. Soit Π le programme logique suivant :

$$\Pi = \left\{ \begin{array}{ll} a \leftarrow b & b \leftarrow a \\ a \leftarrow \text{not } c. & c \leftarrow d \\ d \leftarrow c & c \leftarrow \text{not } a. \end{array} \right\}$$

Dans cet exemple, on a deux boucles $L_1 = \{a, b\}$ et $L_2 = \{c, d\}$. Et pour ces deux boucles, on a les ensembles suivants :

$$R^+(L_1, \Pi) = \{a \leftarrow b. \quad b \leftarrow a.\}, R^-(L_1, \Pi) = \{a \leftarrow \text{not } c.\}$$

$$R^+(L_2, \Pi) = \{c \leftarrow d. \quad d \leftarrow c.\}, R^-(L_2, \Pi) = \{c \leftarrow \text{not } a.\}$$

Ainsi, $LF(L_1, \Pi)$ est $c \supset (\neg a \wedge \neg b)$ et $LF(L_2, \Pi)$ est $a \supset (\neg c \wedge \neg d)$.

On peut également noter que la complétion de Π est :

$$\Pi = \left\{ \begin{array}{ll} a \equiv \neg c \vee b & b \equiv a \\ c \equiv \neg a \vee d & d \equiv c \end{array} \right\}$$

2.7.3.5 L'algorithme ASSAT

L'algorithme est basé sur le résultat suivant.

Théorème 2.7.3. Soit Π un programme logique, $\text{comp}(\Pi)$ sa complétion et $LF(\Pi)$ l'ensemble des formules de boucles associées aux boucles de Π . Pour tout sous-ensembles S de $\text{Atome}(\Pi)$, S est un modèle stable de Π si et seulement si c'est un modèle $\text{comp}(\Pi) \cup LF(\Pi)$.

Les auteurs proposent la mise en œuvre présentée dans la figure 7 :

Algorithme 7 L'algorithme ASSAT

Instancier le programme logique à l'aide de lparse ;
 Effectuer les simplifications simples (par exemple les règles $p \leftarrow G$ tel que $p \in G$) ;
 Calculer la complétion du programme et la convertir en clauses ;
tantque **faire**
 Trouver un modèle M en utilisant un solveur SAT ;
 si $\neg M$ **alors**
 renvoyer *FALSE*
 fin
 Calculer $M^- = M - \text{cons}(P_M)$;
 Si $M^- = \emptyset$, M est un modèle stable ;
 Calculer les boucles maximales de M ;
 Pour chacune de ses boucles, calculer les formules de boucles et les convertir en clauses et les ajouter à l'ensemble de clauses.
fin tantque

2.7.4 CLASP

Proposé par Gebser et al. [GKNS07b], CLASP est un solveur ASP qui développe les techniques venant des domaines de CSP et de SAT : les nogoods.

Soit Π un programme logique. Une assignation est une séquence $(a_1, \dots, a_n)$ de littéraux a_i de la forme $\mathbf{T}_p$ ou $\mathbf{F}_p$ tel que $p \in \text{Atome}(\Pi)$ et $1 \leq i \leq n$. $\mathbf{T}_p$ signifie que p est vraie et $\mathbf{F}_p$ signifie que p est fausse. Les ensembles d'atomes vrais ou faux d'une assignation A sont définis par : $A^{\mathbf{T}} = \{p \in \text{Atome}(\Pi) \mid \mathbf{T}_p \in A\}$ et $A^{\mathbf{F}} = \{p \in \text{Atome}(\Pi) \mid \mathbf{F}_p \in A\}$.

Un nogood est un ensemble $\{a_1, \dots, a_n\}$ de littéraux, exprimant une contrainte violée par n'importe quelle assignation comprenant $a_1, \dots, a_n$. Une assignation A tel que $A^{\mathbf{F}} \cup A^{\mathbf{T}} = \text{Atome}(\Pi)$ et $A^{\mathbf{F}} \cap A^{\mathbf{T}} = \emptyset$ est une solution pour un ensemble Δ de nogoods si $\delta \not\subseteq A$ pour tout $\delta \in \Delta$.

L'idée de l'approche de Gebser et al. consiste à représenter le programme logique Π sous la forme de nogoods d'après l'analyse de Lin et Zhao [LZ04]. Cette analyse (décrite dans 2.7.3.2) décompose un programme logique en deux parties : les inférences issues de la complétion de Clark et celles obtenues à l'aide des formules de boucles.

2.7.4.1 Complétion de Clark

L'atome p_β représente un corps de règle.

$$\begin{aligned}
 (1) \quad p_\beta &\equiv p_1 \wedge \dots \wedge p_m \wedge p_{m+1} \wedge \dots \wedge p_n \mid \beta \in \text{corps}(\Pi), \\
 &\quad \beta = \{p_1, \dots, p_m, \text{not } p_{m+1}, \dots, \text{not } p_n\} \\
 &\quad \cup \\
 (2) \quad \{p &\equiv p_{\beta_1} \vee \dots \vee p_{\beta_k} \mid p \in \text{Atome}(\Pi), \text{corps}(p) = \{\beta_1, \dots, \beta_k\}\}
 \end{aligned}$$

L'équivalence (1) peut être décomposée en deux parties :

- La première est l'implication $p_\beta \rightarrow p_1 \wedge \dots \wedge p_m \wedge \neg p_{m+1} \wedge \dots \wedge \neg p_n$ qui peut être décomposée en un ensemble de clauses : $\neg p_\beta \vee p_1, \dots, \neg p_\beta \vee p_m, \neg p_\beta \vee \neg p_{m+1}, \dots, \neg p_\beta \vee \neg p_n$;
- Cette implication est également vraie dans l'autre sens : $p_\beta \leftarrow p_1 \wedge \dots \wedge p_m \wedge \neg p_{m+1} \wedge \dots \wedge \neg p_n$.

Ce qui donne lieu aux nogoods suivants :

$$\Delta(\beta) = \{\{\mathbf{T}_\beta, \mathbf{F}_{p_1}\}, \dots, \{\mathbf{T}_\beta, \mathbf{F}_{p_m}\}, \{\mathbf{T}_\beta, \mathbf{T}_{p_{m+1}}\}, \dots, \{\mathbf{T}_\beta, \mathbf{T}_{p_n}\}\}$$

$$\delta(\beta) = \{\mathbf{F}_\beta, \mathbf{T}_{p_1}, \dots, \mathbf{T}_{p_m}, \mathbf{F}_{p_{m+1}}, \dots, \mathbf{F}_{p_n}\}$$

L'équivalence (2) donne les nogoods suivants :

$$\Delta(p) = \{\{\mathbf{F}_p, \mathbf{T}_{\beta_1}\}, \dots, \{\mathbf{F}_p, \mathbf{T}_{\beta_k}\}\}$$

$$\delta(p) = \{\mathbf{T}_p, \mathbf{F}_{\beta_1}, \mathbf{F}_{\beta_k}\}$$

$$\Delta(\Pi) = \Delta(\beta) \cup \delta(\beta) \cup \Delta(p) \cup \delta(p)$$

2.7.4.2 Formules de boucles

Pour capturer les effets d'une formule de boucles induite par un ensemble U d'atomes de Π tel que l'ensemble des formules de boucles comprenant les atomes d'un ensemble U est $LF(U, \Pi) = \{\beta_1, \dots, \beta_k\}$, on définit le nogood de boucle d'un atome p comme étant :

$$\lambda(p, U) = \{\mathbf{F}_{\beta_1}, \dots, \mathbf{F}_{\beta_k}, \mathbf{T}_p\}$$

Ainsi l'ensemble des nogoods de boucles pour un programme logique Π est :

$$\Lambda_\Pi = \cup_{U \subseteq \text{Atome}(\Pi), U \neq \emptyset} \{\lambda(p, U) \mid p \in U\}$$

Grâce à cette traduction, il est possible d'adapter les méthodes propres aux méthodes CSP et SAT au calcul de modèles stables : les nogoods bénéficient de la propagation unitaire et du retour arrière guidé par conflit.

2.8 Générer et tester

La programmation logique avec sémantique des modèles stables est avant tout utilisée pour la résolution de problèmes fortement combinatoires. Pour représenter ces problèmes, une méthode commune employée dans la programmation logique est celle du générer-et-tester [Lif08].

Cette méthode découpe le programme logique en deux parties : la génération de l'ensemble des configurations, des interprétations ou des plans (selon le domaine) possibles,

puis la mise à l'écart des solutions impropres. Bien entendu, les solveurs récents ne génèrent pas l'ensemble des solutions avant de les tester, mais effectuent des coupes efficaces dans l'arbre de recherche.

2.9 Programmation logique possibiliste

P. Nicolas et al. [NGS05] ont proposé une manière d'améliorer le pouvoir d'expression de la programmation logique en étendant la sémantique des modèles stables (voir la section 2) au programme logique dont les règles sont munis d'une mesure de nécessité.

L'ensemble fini des atomes classiques est noté $\mathcal{X}$. On note $\mathcal{N}$ l'ensemble totalement ordonné des valeurs de nécessité $\mathcal{N} \subseteq]0, 1]$. Un atome possibiliste est donc une paire $a = (p, \alpha)$ où p est un atome de la logique classique et $\alpha \in \mathcal{N}$.

On appelle règle possibiliste une règle de programme logique à laquelle est associée une mesure de nécessité de la forme :

$$r = (c \leftarrow a_1, \dots, a_n, \text{not } b_1, \dots, \text{not } b_m, \alpha)$$

avec $n \geq 0, m \geq 0$ et $(a_i)_{1 \leq i \leq n}, (b_j)_{1 \leq j \leq m}, c$ sont des atomes de la logique classique. La projection classique d'une règle possibiliste est définie par $r^* = c \leftarrow a_1, \dots, a_n, \text{not } b_1, \dots, \text{not } b_m$ et $n(r) = \alpha$ est le degré de nécessité de la règle r . Un programme logique possibiliste est défini comme étant un ensemble de règles possibilistes. Ainsi si R est un programme possibiliste, alors $R^* = \{r^* \mid r \in R\}$ est le programme logique obtenu à partir de R en oubliant les valeurs de nécessité.

Comme pour les programmes logiques standards, on définit la réduction du programme logique possibiliste de la manière suivante.

Définition 2.9.1. Soient P un programme logique possibiliste et X un ensemble d'atomes. La réduction possibiliste de P par rapport à A est le programme logique possibiliste défini par $P^X = \{((r^*)^+, n(r)) \mid r \in P, \text{corps}^-(r) \cap X = \emptyset\}$

Le programme logique possibiliste P^X ainsi obtenu est un programme logique ne contenant pas de négation par echec. Il est possible de définir un ensemble de conséquences uniques et une distribution de possibilité associée.

Définition 2.9.2. Considérons $\mathcal{A} = 2^{\mathcal{X} \times \mathcal{N}}$ l'ensemble fini de tous les atomes possibilistes induit par $\mathcal{X}$ et $\mathcal{N}$. $\forall A, B \in \mathcal{A}$, nous définissons :

$$\begin{aligned} A \sqcap B &= \{(x, \min\{\alpha, \beta\}), (x, \alpha) \in A, (x, \beta) \in B\} \\ A \sqcup B &= \begin{aligned} &\{(x, \alpha) \mid (x, \alpha) \in A, x \notin B^*\} \\ \cup &\{(x, \beta) \mid x \notin A^*, (x, \beta) \in B\} \\ \cup &\{(x, \max\{\alpha, \beta\}) \mid (x, \alpha) \in A, (x, \beta) \in B\} \end{aligned} \\ A \sqsubseteq B &= \left\{ \begin{array}{l} A^* \subseteq B^*, \text{ et } \forall a, \alpha, \beta, \\ (a, \alpha) \in A \wedge (a, \beta) \in B \Rightarrow \alpha \leq \beta \end{array} \right\} \end{aligned}$$

Pour la définition suivante, si A et B sont deux ensembles d'atomes possibilistes, on définit $A \subseteq B$ comme étant l'inclusion ensembliste classique.

Définition 2.9.3. Soient $r = (c \leftarrow a_1, \dots, a_n, \alpha)$ une règle possibiliste et A un ensemble d'atomes possibilistes :

- r est α -applicable dans A si $\text{corps}(r^*) = \emptyset$
- r est β -applicable dans A si $\beta = \min\{\alpha, \alpha_1, \dots, \alpha_n\}, \{(a_1, \alpha_1), \dots, (a_n, \alpha_n)\} \subseteq A$
- r est 0-applicable dans tous les autres cas.

Et, pour un programme logique possibiliste sans négation par échec P et un atome x donné, $\text{App}(P, A, x) = \{r \in P \mid \text{tete}(r^*) = x, r \text{ est } \nu\text{-applicable dans } A, \nu > 0\}$

Définition 2.9.4. Soient P un programme logique possibiliste sans négation par échec et A un ensemble d'atomes possibilistes. L'opérateur possibiliste de conséquence immédiate ΠT_P est une fonction qui associe un ensemble d'atomes possibilistes à un autre.

$$\Pi T_P(A) = \left\{ \begin{array}{l} (x, \delta) \mid x \in \text{tete}(P^*), \text{App}(P, A, x) \neq \emptyset, \\ \delta = \max_{r \in \text{App}(P, A, x)} \{\nu \mid \nu\text{-applicable dans } A\} \end{array} \right\}$$

alors, l'opérateur itéré ΠT_P^k est défini par $\Pi T_P^0 = \emptyset$ et $\Pi T_P^{n+1} = \Pi T_P(\Pi T_P^n), \forall n \geq 0$.

La proposition suivante est donnée dans [NGS05].

Proposition 2.9.1. Soit P un programme logique possibiliste sans négation par échec, alors ΠT_P possède un point fixe $\sqcup_{n \geq 0} \Pi T_P^n(\emptyset)$ que nous appelons ensemble de conséquences possibilistes de P et que nous notons $\Pi Cn(P)$.

On peut maintenant donner la définition d'un modèle stable possibiliste.

Définition 2.9.5. Soient P un programme logique possibiliste et X un ensemble d'atomes possibilistes. X est un modèle stable possibiliste si et seulement si $X = \Pi Cn(P^{X^*})$

Conclusion

Dans ce chapitre, nous avons présenté la programmation logique en commençant par PROLOG. Puis, nous avons introduit la sémantique des modèles bien-fondés et la sémantique des modèles stables qui nous servira à mettre en œuvre la fusion par R-ensembles. Ensuite, nous avons présenté différents solveurs de modèles stables. En particulier, les solveurs Smodels et CLASP qui nous serviront pour la mise en œuvre. Nous avons également présenté la programmation logique possibiliste.

Chapitre 3

Changement de croyances

Introduction

La logique propositionnelle est un outil pour la représentation des croyances qu'un agent ou que plusieurs agents peuvent avoir sur le monde. Ainsi, il semble nécessaire de disposer d'outils pour la gestion de ces croyances (vérification de la cohérence, recherche de modèles, ...). Les années 80 ont marqué le début de l'étude des opérations de changement de croyances, qu'il s'agisse de révision, de mise à jour ou de fusion de croyances.

La révision concerne la modification dans les croyances, dans un monde inchangé, d'informations considérées comme plus sûres ou plus fiables que les précédentes. Par exemple, en voyant l'affiche de *Taken* avec Liam Neeson, le titre ainsi que les acteurs principaux pourraient permettre de déduire qu'il s'agit d'un film américain. Mais en regardant la bande-annonce, on constate qu'il a été réalisé par Pierre Morel et produit par EuropaCorp, *Taken* est donc un film français. Ici, une information peu sûre (le design de l'affiche) a été contredite par une information plus précise (la bande-annonce) et les premières conclusions ont du être révisées pour restaurer la cohérence.

La mise à jour, quant à elle, concerne le changement du monde lui même. Les nouvelles informations sont des modifications du monde qui doivent être rapportées à la base de croyances. Par exemple, imaginons que j'ai un couloir avec deux portes ainsi qu'une croyance stipulant que l'une de ces portes seulement est ouverte. Si j'apprends que la porte de droite est ouverte dans le contexte de la révision, cela veut dire que j'ai amélioré, précisé, ma croyance précédente et je sais que seule la porte de droite est ouverte et que la gauche est fermée. Dans le cas de la mise à jour, j'ai appris que le monde a changé et cela mène à deux conclusions possibles : la porte de gauche était celle qui était ouverte et maintenant elles le sont toutes les deux et le monde a changé ou la porte de droite était celle qui était ouverte et le monde n'a pas changé.

La fusion est l'opération consistant à construire à partir des bases de croyances représentant les croyances d'un groupe d'agents une nouvelle base de croyances ou un ensemble d'interprétations qui représente les croyances du groupe tout entier. Cette opération nécessite de résoudre les conflits existants entre les agents mais également de supprimer les

croyances redondantes.

Toutes ces opérations sont guidées par le principe de minimalité (les croyances, précieuses, ne doivent pas être supprimées à la légère). Néanmoins, ces cadres sont soumis à des contraintes spécifiques. Par exemple, pour la révision, la primauté va à la nouvelle information par rapport aux anciennes ; alors que, pour la fusion, toutes les bases de croyances doivent être également considérées. Ces exigences particulières ont conduit à la définition d'ensembles de postulats qu'une opération de révision dite raisonnable devrait respecter. Ces postulats ont été définis en fonction du cadre dans lequel sont représentées les croyances. Dans ce chapitre, nous allons présenter différentes manières de représenter les croyances : comme un ensemble déductivement clos, une formule, une interprétation ou un ensemble de formules. Pour chacun de ces cadres, nous allons montrer comment les postulats AGM y ont été adaptés ainsi que quelques exemples d'opérations qui y ont été définies.

3.1 Révision

La révision de croyances désigne l'ensemble des opérations à effectuer dans la représentation des croyances d'un agent lorsque de nouvelles croyances, considérées comme plus fiables que les précédentes, doivent être intégrées. Si les nouvelles croyances sont compatibles avec les anciennes, alors leur conjonction est suffisante ; mais que faire si celle-ci est incohérente ?

Les travaux sur la révision seront présentés ici en fonction du choix de l'état épistémique sous-jacent. Les premiers travaux ont porté sur les théories, autrement dit sur des ensembles de croyances déductivement clos ; puis lorsque les croyances sont représentées par une formule représentant cette théorie. Nous verrons ensuite les travaux portant sur la révision définie lorsque l'état épistémique est décrit par une interprétation ou un ensemble d'interprétations. Ensuite, nous examinons la révision syntaxique, où un ensemble de formules représente l'état épistémique. Finalement, nous montrons les principaux travaux portant sur l'itération du processus de révision.

3.1.1 Changement de théories

Les premiers travaux sur la révision de croyances ont porté sur la révision de théories. Une première opération, appelée expansion, a été définie. Cette opération correspond à l'introduction d'une nouvelle formule sans modification de la théorie initiale. L'opération d'expansion d'une théorie T par une formule A , notée $T + a$, est suffisante dans le cas où la formule ajoutée est cohérente avec la théorie initiale. Dans le cas contraire, les opérations de contraction et de révision sont définies.

lorsqu'une formule est retirée d'une théorie, on définit l'opération de contraction, notée $T - A$.

L'opération de révision, notée $T \circ A$, correspond à la prise en compte d'une formule au sein d'une théorie. Le résultat de cette opération doit être cohérent, et doit changer le moins possible la théorie initiale.

3.1.1.1 Identité de Levi et identité de Harper

L'opération de révision a été originalement définie à partir de l'expansion et de la contraction :

- Réduction de T pour s'adapter à la nouvelle information $(T - \neg a)$;
- Expansion de cette nouvelle théorie par la nouvelle information $((T - \neg a) + a)$.

On appelle cette définition de l'opérateur de révision l'identité de Levi [Lev80].

$$T \circ a = \text{Cons}((T - \neg a) \cup \{a\})$$

Il est également possible de définir l'opération de contraction à partir de l'opération de révision, c'est l'identité de Harper [Har75].

$$T - a = \text{Cons}(T \cap (T \circ \neg a))$$

3.1.1.2 Postulats AGM

Dans [AGM85], Alchourron, Gärdenfors et Makinson, ont défini la première série de postulats pour les opérateurs de révision. Les postulats AGM sont exprimés dans le cas où l'objet à réviser est une théorie, c'est à dire un ensemble de formules déductivement clos.

On note $\mathcal{K}_{\mathcal{L}}$ l'ensemble des théories de croyances que l'on peut définir sur le langage $\mathcal{L}$. Une opération de révision $\star$ est une fonction de $\mathcal{K}_{\mathcal{L}} \times \mathcal{L}$ vers $\mathcal{K}_{\mathcal{L}}$, qui, à une théorie K et une formule A , associe une nouvelle théorie $K \star A$ qui vérifie les postulats suivants :

AGM1 $K \star A$ est une théorie

AGM2 $A \in K \star A$

AGM3 $K \star A \subseteq K + A$

AGM4 Si $\neg A \notin K$, alors $K + A \subseteq K \star A$

AGM5 $K \star A = \perp$ si et seulement si $\vdash \neg A$

AGM6 Si $A \leftrightarrow B$, alors $K \star A = K \star B$

AGM7 $K \star (A \wedge B) \subseteq (K \star A) + B$

AGM8 Si $\neg B \notin K \star A$, alors $(K \star A) + B \subseteq K \star (A \wedge B)$

Ainsi un opérateur de révision au sens de ces postulats AGM doit avoir les propriétés suivantes. Sa solution doit être une théorie **AGM1**. La nouvelle croyance doit faire partie de la solution **AGM2**. **AGM3** et **AGM4** donnent le résultat de la révision lorsque A est cohérent avec T . La solution ne peut être incohérent que lorsque l'on révisé par une information incohérente **AGM5**. Le résultat doit être indépendant de la syntaxe **AGM6**. Les deux derniers postulats - **AGM7** et **AGM8** - appelés postulats supplémentaires par Gärdenfors caractérisent le bon comportement de l'opérateur vis-à-vis de la minimalité.

Soient T une théorie et A une formule. Les opérateurs dans ce cadre sont basés sur l'identité de Levi, elles consistent à définir des opérations de contraction, $T - \neg A$ puis de définir l'opération de révision par $T \circ A = (T \downarrow \neg A) + A$.

L'ensemble des sous-ensembles maximaux cohérents est défini par :

$$T \downarrow \neg A = \{T' \mid T' \subseteq T, T \not\models \neg A \text{ et si } T' \subseteq M \subseteq T \text{ alors } M \vdash \neg A\}$$

3.1.1.3 Full meet contraction

Une première approche intuitive consiste à considérer que le résultat de la contraction est l'intersection des sous-ensembles maximaux cohérents.

Définition 3.1.1.

$$T - \neg A = \begin{cases} \cap(T \downarrow \neg A) & \text{si } T \downarrow \neg A \text{ est non-vide,} \\ T & \text{sinon.} \end{cases}$$

Mais cette opération se révèle trop prudente pour être un opérateur de révision raisonnable. Pour plus de détails, voir [Som94].

3.1.1.4 Maxi choice contraction

Une manière de résoudre la prudence de l'opération de Full meet contraction est de définir un opérateur de contraction basé sur la sélection d'un seul sous-ensemble maximal cohérent. Ce choix est réalisé par une fonction de sélection s . Cette opération de révision se trouve être trop hardie pour être satisfaisante.

Définition 3.1.2.

$$T - \neg A = \begin{cases} s(T \downarrow \neg A) & \text{si } T \downarrow \neg A \text{ est non-vide,} \\ T & \text{sinon.} \end{cases}$$

3.1.1.5 Partial meet contraction

La définition d'une opération de contraction qui serait un compromis entre les deux opérateurs précédents est réalisé au travers de l'intersection d'un ensemble de sous-ensembles maximaux cohérents choisis par une fonction de sélection.

Définition 3.1.3.

$$T - \neg A = \begin{cases} \cap s(T \downarrow \neg A) & \text{si } T \downarrow \neg A \text{ est non-vide,} \\ T & \text{sinon.} \end{cases}$$

La critique la plus importante adressée aux postulats AGM est l'absence de prise en considération de la répétition des opérations de révision. C'est à ce problème que Darwiche et Pearl ont tenté d'apporter une solution.

3.1.2 Approche sémantique

Katsuno et Mendelzon [KM91] ont proposé une reformulation des postulats AGM dans le cas où une théorie est représentée par une seule formule φ et le lien entre φ et K est défini de la manière suivante : $K = \{\phi \mid \varphi \models \phi\}$. Les postulats sont reformulés comme suit :

KM1 $\varphi \circ \mu \models \mu$

KM2 Si $\varphi \wedge \mu$ est cohérent, alors $\varphi \circ \mu \leftrightarrow \varphi \wedge \mu$

KM3 Si μ est cohérent, alors $\varphi \circ \mu$ est cohérent

KM4 Si $\varphi_1 \leftrightarrow \varphi_2$ et $\mu_1 \leftrightarrow \mu_2$, alors $\varphi_1 \circ \mu_1 \leftrightarrow \varphi_2 \circ \mu_2$

KM5 $(\varphi \circ \mu) \wedge \phi \vdash \varphi \circ (\mu \wedge \phi)$

KM6 Si $(\varphi \circ \mu) \wedge \phi$ est cohérent, alors $\varphi \circ (\mu \wedge \phi) \vdash (\varphi \circ \mu) \wedge \phi$

On peut établir une correspondance entre les opérateurs de révision $\star$ sur des théories et sur les opérateurs de révision $\circ$ sur des bases de croyances propositionnelles.

Théorème 3.1.1. *Soit un opérateur de révision $\star$ et son opérateur $\circ$ correspondant, alors $\star$ satisfait les postulats **AGM1-AGM8** si et seulement si $\circ$ vérifie les postulats **KM1-KM6**.*

3.1.2.1 Théorème de représentation

Le théorème de représentation montre comment les opérateurs peuvent être caractérisés à l'aide de pré-ordre sur les interprétations. Ainsi, Katsuno et Mendelzon ont proposé dans [KM91] un théorème de représentation basé sur la notion d'assignement fidèle.

Un assignement fidèle est une fonction qui associe une formule propositionnelle φ à un pré-ordre total sur les interprétations de $\mathcal{W}$, noté $\leq_\varphi$, tel que pour toutes formules φ, φ_1 et φ_2 et pour toutes interprétations ω et ω' :

Définition 3.1.4 (Assignement fidèle). *Une fonction qui associe à chaque état épistémique φ un pré-ordre total $\leq_\varphi$ sur les interprétations est un assignement fidèle si et seulement si :*

- Si $\omega \models \varphi$ et $\omega' \models \varphi$, alors $\omega =_\varphi \omega'$
- Si $\omega \models \varphi$ et $\omega' \not\models \varphi$, alors $\omega <_\varphi \omega'$
- Si $\varphi_1 = \varphi_2$, alors $\leq_{\varphi_1} = \leq_{\varphi_2}$

Un assignement fidèle représente la préférence entre les interprétations par rapport à l'opération de révision. C'est à dire $\omega \leq_\varphi \omega'$ signifie que lors d'une opération de révision d'une source représentée par la formule φ , l'interprétation ω sera préféré à l'interprétation ω' . Le théorème de représentation permet de lier les pré-ordre totaux aux postulats KM au travers de l'assignation fidèle.

Théorème 3.1.2. *Un opérateur de révision $\circ$ satisfait les postulats **KM1-KM6** si et seulement si il existe un assignement fidèle $\leq_\varphi$ tel que :*

$$Mod(\varphi \circ \mu) = \min(Mod(\mu), \leq_\varphi)$$

Les opérateurs de révision sémantique considèrent le résultat d'une opération de révision $\varphi \circ \mu$ comme les modèles de μ les plus proches des modèles de φ . Parmi les opérateurs de révision sémantique, on peut citer les opérateurs de Bordiga [Bor85] ou encore les Fonctions Ordinales Conditionnelles (OCF) [Spo88b]. Nous ne présentons ici que l'opération de révision de Dalal.

3.1.2.2 Révision selon Dalal

L'opérateur dit de Dalal [Dal88] est basé sur la distance de Hamming entre les interprétations [Dal88]. La distance de Hamming entre deux interprétations est le nombre de variables propositionnelles sur lesquelles elles diffèrent.

Grâce à la distance de Hamming, nous pouvons définir la révision selon Dalal.

Définition 3.1.5 (Révision selon Dalal). *Soient φ une formule représentant les croyances et μ une formule. Les pré-ordre $\leq_{d_H}$ est définie par :*

$$\omega_i \leq_{d_H} \omega_j \text{ si et seulement } d_H(\text{Mod}(\mu), \omega_i) \leq d_H(\text{Mod}(\mu), \omega_j)$$

Ainsi, la révision de φ par μ selon Dalal est définie de la manière suivante :

$$\varphi \circ \mu = \min(\text{Mod}(\mu), \leq_{d_H})$$

Exemple 3.1.1. *Soit $\varphi = a \wedge b \wedge \neg c$ et $\mu = \neg b$. D'après la définition, la solution de $\varphi \circ \mu$ se trouve parmi les modèles de μ , en d'autres termes $\{\{\neg a, \neg b, \neg c\}, \{\neg a, \neg b, c\}, \{a, \neg b, \neg c\}, \{a, \neg b, c\}\}$. Et la solution est, parmi ces modèles, celui qui est le plus proche de $\{a, b, \neg c\}$.*

$$\begin{aligned} d_H(\{\neg a, \neg b, \neg c\}, \{a, b, \neg c\}) &= 2 & d_H(\{\neg a, \neg b, c\}, \{a, b, \neg c\}) &= 1 \\ d_H(\{a, \neg b, \neg c\}, \{a, b, \neg c\}) &= 1 & d_H(\{a, \neg b, c\}, \{a, b, \neg c\}) &= 2 \end{aligned}$$

Ainsi,

$$\varphi \circ \mu = \{a, \neg b, \neg c\}$$

3.1.3 Opérateurs de révision syntaxique

On appelle base de croyances tout ensemble fini de formules bien formées de la logique propositionnelle. L'ensemble des bases de croyances pouvant être définies sur un langage $\mathcal{L}$ est noté $\mathcal{K}_{\mathcal{L}}$, ou lorsqu'il n'y a pas d'ambiguïtés $\mathcal{K}$.

Ainsi, une autre approche pour la révision de croyances est de s'attacher à la syntaxe en représentant les croyances d'un agent à l'aide de bases de croyances. Les opérateurs de révision syntaxique donneront le résultat de l'opération $K \circ a$ comme un ensemble de formules, le plus souvent un sous-ensemble de $K \cup a$. Les opérateurs de révision syntaxique vont ainsi s'attacher à produire des sous-ensembles maximaux cohérent de K .

L'ensemble des conséquences possibles d'un ensemble de formules $\mathcal{K}$ est appelé clôture déductive, notée $\text{Cons}(\mathcal{K})$. Il est défini par $\text{Cons}(\mathcal{K}) = \{A \mid \mathcal{K} \models A\}$.

3.1.3.1 Révision selon Nebel

B. Nebel [Neb91] a proposé un opérateur de révision pour les bases de croyances finies. Soit K une base de croyances et a une formule. $K \downarrow a$ représente l'ensemble des sous-ensembles maximaux cohérents qui n'impliquent pas a :

$$K \downarrow a = \{K', K' \subseteq K, K' \not\models a \text{ et si } K' \subset M \subseteq K \text{ alors } M \models a\}$$

L'opération de contraction est définie par :

$$K - a = \begin{cases} (\bigvee K')_{K' \in K \downarrow a} & \text{si } \not\models a; \\ K & \text{dans le cas contraire.} \end{cases}$$

L'opération de révision est alors défini par :

$$K \circ a = (K - \neg a) \wedge a$$

Défini de la sorte, l'opérateur de Nebel ne vérifiait pas le postulat **(AGM5)**. Pour corriger ce défaut, l'opérateur de contraction a été redéfini de la manière suivante :

$$K - a = \begin{cases} (\bigvee K')_{K' \in K \downarrow a} \wedge (K \vee \neg a) & \text{si } \not\models a; \\ K & \text{dans le cas contraire.} \end{cases}$$

3.1.3.2 Système de sphères de Grove

A. Grove [Gro88] a proposé une opération de révision utilisant la sémantique des sphères définie par D. Lewis [Lew73]. Le système de sphères de Grove s'intéresse à l'ensemble des sous-ensembles maximaux cohérents d'un langage $\mathcal{L}$, notés $\mathcal{M}$. Les éléments de $\mathcal{M}$ sont également appelés mondes possibles.

Une base de croyances est représentée par un ensemble de formules de $\mathcal{L}$, noté $[K]$, qui est un sous-ensemble de $\mathcal{M}$ et qui est constitué des sous-ensembles maximaux cohérents contenant les formules de K . Formellement :

$$[K] = \{M \in \mathcal{M} \mid K \subseteq M\}$$

Un système de sphères est défini de la manière suivante :

Définition 3.1.6. *Un système de sphère centré sur $[K]$ est une collection de sous-ensembles de $\mathcal{M}$, noté $\mathcal{S}$, qui satisfait les conditions suivantes :*

- $\mathcal{S}$ est totalement ordonné par $\subseteq$;
- $[K]$ est un $\subseteq$ -minimum de $\mathcal{S}$;
- $\mathcal{M}$ est dans $\mathcal{S}$;
- Si A est une formule de $\mathcal{L}$ et si il existe une sphère qui intersecte $[A]$, alors il existe une plus petite sphère de $\mathcal{S}$ qui intersecte $[A]$.

Dans ce cadre, l'opération de révision peut être défini de la manière suivante. Soit S_A la plus petite sphère de $\mathcal{S}$ qui intersecte $[A]$. La révision de la base de croyances K par une formule A peut être représentée par $[A] \cap S_A$. En fait, $[A] \cap S_A$ est l'ensemble des éléments de $\mathcal{M}$ les plus proche de $[K]$ qui contiennent A . Cette opération de révision, noté $K \star A = K_{c(A)}$, satisfait les huit postulats AGM.

3.1.3.3 Révision par R-ensembles

Un exemple est la Révision par R-ensembles qui est définie dans [BnBPW04]. Cette méthode se place dans le contexte de bases de croyances constituées de formules sous forme CNF. Si on considère l'opération de révision de la base de croyances K par un ensemble de formules A . L'opération de révision par R-ensembles est basée sur le retrait de sous-ensembles minimaux de clauses K permettant de restaurer la cohérence dans $K \cup A$, appelés R-ensembles ou *removed sets*. De façon formelle :

Définition 3.1.7 (R-ensembles). *Soient K une base de croyances et A une base de croyances telles que $K \cup A$ soit incohérent. Soit R un sous-ensemble de clauses de K , R est un R-ensemble de $K \circ A$ si et seulement si :*

1. $(K \cup A) \setminus R$ est cohérent ;
2. $\forall R' \subseteq K, R' \neq R$, si $(K \cup A) \setminus R'$ est cohérent, alors $|R| < |R'|$.

L'ensemble des R-ensembles pour $K \circ A$ est noté $\mathcal{R}(K \cup A)$.

Ainsi, on obtient un nouvel opérateur de révision syntaxique :

Définition 3.1.8 (Révision par R-ensembles). *Soient K une base de croyances et A une formule. La révision de K par A est définie comme :*

$$K \circ A = \bigvee_{R \in \mathcal{R}(K \cup A)} \text{Cons}(\{(K \cup A) \setminus R\})$$

Exemple 3.1.2. *Soit la base de croyances $K = \{d, \neg a \vee \neg b, \neg c \vee \neg b, a \vee \neg d, c \vee \neg d\}$. K est cohérent. Si K est révisé par $A = \{b\}$, $K \cup A$ est incohérent, il faut donc supprimer des formules de K pour restaurer cette cohérence.*

Le plus petit sous-ensemble de formules de K dont la suppression restaure la cohérence dans $K \cup A$, et donc seul R-ensemble, est $R = \{d\}$. Ainsi,

$$K \circ A = \{b, \neg a \vee \neg b, \neg c \vee \neg b, a \vee \neg d, c \vee \neg d\}$$

Dans cette thèse, nous allons généraliser cette approche à la fusion de bases de croyances.

3.1.4 Révision itérée

Il peut arriver cependant que les nouvelles informations ne soient pas disponibles immédiatement et que le processus de révision doive être répété plusieurs fois. Ainsi, la notion de révision itérée a été formalisée par Daniel Lehmann dans [Leh95]. Dans ce cadre, au lieu de réviser par une nouvelle information μ , on réviser par une séquence d'informations $(\mu_1, \dots, \mu_n)$. Cette approche est décrite dans un cadre suffisamment large pour que l'état épistémique Φ puisse être représenté de plusieurs manières : un pré-ordre total sur les interprétations, une distribution de possibilité, une fonction conditionnelle ordinale, etc. À chaque état épistémique correspond un ensemble de croyances $Bel(\Phi)$ qui est une formule propositionnelle.

Définition 3.1.9. Soient Φ et Φ' deux états épistémiques. On dit que Φ et Φ' sont équivalents, noté $\Phi \equiv \Phi'$, si et seulement si leur base de croyances contiennent des formules équivalentes. Autrement dit, $Bel(\Phi) \equiv Bel(\Phi')$. Φ et Φ' sont égaux, noté $\Phi = \Phi'$, si et seulement si ils sont identiques.

3.1.4.1 Reformulation des postulats AGM

Les postulats KM sont donc reformulés et réécrits dans ce cadre. Ainsi, soit Φ un état épistémique et φ une formule. L'opérateur $\circ$ est un opérateur de révision s'il satisfait les propriétés suivantes :

R1 $Bel(\Phi \circ \varphi) \models \varphi$

R2 Si $Bel(\Phi) \wedge \varphi$ est cohérent, alors $Bel(\Phi \circ \varphi) \equiv Bel(\Phi) \wedge \varphi$.

R3 Si φ est cohérent, alors $Bel(\Phi \circ \varphi)$ est cohérent.

R4 Si $\Phi_1 = \Phi_2$ et $\varphi_1 \equiv \varphi_2$, alors $\Phi_1 \circ \varphi_1 \equiv \Phi_2 \circ \varphi_2$

R5 $Bel(\Phi \circ \varphi) \wedge \mu \models Bel(\Phi \circ (\varphi \wedge \mu))$

R6 Si $Bel(\Phi \circ \varphi) \wedge \mu$ est cohérent, alors $Bel(\Phi \circ (\varphi \wedge \mu)) \models Bel(\Phi \circ \varphi) \wedge \mu$

Dans ce cadre-là, Darwiche et Pearl [DP97a] ont ajouté quatre postulats pour prendre en compte l'itération du processus de révision dans l'état épistémique. Soit Φ un état épistémique, soient α, μ deux formules, un opérateur de révision doit satisfaire les postulats suivants :

DP1 Si $\varphi \models \mu$, alors $Bel((\Phi \circ \mu) \circ \varphi) \equiv Bel(\Phi \circ \varphi)$

DP2 Si $\varphi \models \neg\mu$, alors $Bel((\Phi \circ \mu) \circ \varphi) \equiv Bel(\Phi \circ \varphi)$

DP3 Si $Bel(\Phi \circ \varphi) \models \mu$, alors $Bel((\Phi \circ \mu) \circ \varphi) \models \mu$

DP4 Si $Bel(\Phi \circ \varphi) \not\models \neg\mu$, alors $Bel((\Phi \circ \mu) \circ \varphi) \not\models \neg\mu$

Ces postulats expriment des conditions sur la succession des opérations de révision. Par exemple, le postulat **(DP2)** stipule que si on révisé une base un état épistémique par une information μ , puis par son contraire φ , le résultat soit être identique à celui obtenu en révisant directement par φ . Le postulat **(DP2)** assure qu'une information doit être gardée après une opération de révision si la nouvelle information n'est pas contradictoire avec elle. La caractérisation des postulats KP a été discutée, en particulier, le très controversé postulat (C_2) , car à la fois trop forte et trop faible [FH99].

Les postulats **(DP1)**-**(DP4)** peuvent être réécrits en fonction des pré-ordres totaux de la manière suivante :

CR1 Si $\omega_1 \models \mu$ et $\omega_2 \models \mu$ alors $\omega_1 \leq_\Psi \omega_2$ si et seulement si $\omega_1 \leq_{\Psi \circ \mu} \omega_2$

CR2 Si $\omega_1 \models \neg\mu$ et $\omega_2 \models \neg\mu$ alors $\omega_1 \leq_\Psi \omega_2$ si et seulement si $\omega_1 \leq_{\Psi \circ \mu} \omega_2$

CR3 Si $\omega_1 \models \mu$ et $\omega_2 \models \neg\mu$ alors $\omega_1 <_\Psi \omega_2$ si et seulement si $\omega_1 <_{\Psi \circ \mu} \omega_2$

CR4 Si $\omega_1 \models \mu$ et $\omega_2 \models \neg\mu$ alors $\omega_1 \leq_\Psi \omega_2$ si et seulement si $\omega_1 \leq_{\Psi \circ \mu} \omega_2$

Parmi les opérateurs de révision itérée, on peut citer : la révision naturelle de Boutilier[Bou93], Lehmann [Leh95], les enracinements épistémiques [NFPS94, NFPS96], les transmutations

[Wil94], [Wil95], la révision possibiliste [DP92, DP97b] ou encore la révision avec mémoire [Pap01, BDLP02].

Dans la section suivante, nous allons étudier la fusion de croyances.

3.2 Fusion de croyances

Les croyances sont de plus en plus distribuées, que ce soit à travers des bases de données réparties ou des systèmes multi-agents. Dans ce cadre, les agents distribués ne sont pas toujours, et même rarement, en accord et la question de la combinaison de ces croyances se pose naturellement. Il s'agit ici de déterminer, à un moment donné, quel est l'ensemble des croyances associées à un groupe d'agents.

3.2.1 Flock et profil de croyances

Un multi-ensemble de bases de croyances est appelé profil de croyances. Dans le cas des opérations de révision ou de fusion syntaxique de croyances, le résultat sera exprimé sous la forme d'ensemble de bases de croyances. Ces ensembles de bases de croyances, qui ne sont pas des multi-ensembles et ont une sémantique différente des profils de croyances, sont appelés flock [FUV83]. Les flocks peuvent être considérés comme un ensemble de solutions alternatives, à la manière des extensions de la logique des défauts [Rei80].

On définit l'équivalence entre profil de croyances de la manière suivante :

Définition 3.2.1. Soit $E = \{K_1, \dots, K_n\}$ et $E' = \{K'_1, \dots, K'_n\}$ deux profils de croyances. On dit que E et E' , noté $E \equiv E'$ sont équivalents si et seulement si il existe une bijection f entre les bases de croyances appartenant au profil, i.e. pour tout i , il existe un unique j tel que $K_i = K_{f(i)=j}$.

Soit $E = \{K_1, \dots, K_n\}$ un multi-ensemble de n bases de croyances cohérentes à fusionner, E est appelé profil de croyances. On note $\Delta(E)$ le résultat de l'opération de fusion des bases $K_1, \dots, K_n$. L'opération de fusion peut se faire en présence de contraintes que le résultat devra respecter. Nous préciserons d'abord la question des contraintes d'intégrité. Puis nous détaillons les principaux postulats pour la fusion de croyances. Nous donnons finalement quelques exemples d'opérateurs de fusion, tant sémantiques que syntaxiques.

3.2.2 La question des contraintes d'intégrité

Les contraintes sont généralement exprimées sous la forme de formules. Les postulats de Konieczny et Pino-Perez (cf. *infra*) demandent à ce que les contraintes soient explicites dans le résultat et pas seulement implicitement cohérent avec le résultat comme dans les travaux antérieurs [Rev97, LM98]. L'opération de fusion du profil E avec IC pour contraintes est notée $\Delta_{IC}(E)$. Dans le cas où il n'y aurait pas de contraintes portant sur le résultat, on parle de fusion pure et on pourra considérer que l'opération se déroule avec la contrainte $IC = \top$ ou alors simplement noté $\Delta(E)$.

Pour reprendre la terminologie de S.Konieczny, on parlera de fusion contrainte en présence de contraintes d'intégrité et de fusion pure sinon.

3.2.3 Postulats

Dans la continuité des travaux d'Alchourron, Gardenfors et Makinson (cf. 3.1.1.2), S. Konieczny et R. Pino-Perez a dans [KP99] défini un ensemble de postulats qu'un opérateur de fusion contrainte raisonnable devrait respecter.

Nous donnons maintenant la définition de l'équivalence entre profils.

Définition 3.2.2. Soient E_1 et E_2 deux profils de croyances. On dit que E_1 et E_2 sont équivalents, noté $E_1 \leftrightarrow E_2$ si et seulement si il existe une bijection f de $E_1 = \{K_1, \dots, K_m\}$ vers $E_2 = \{K_{m+1}, \dots, K_n\}$ telle que $f(K) \leftrightarrow K$.

L'opération de fusion Δ_{IC} est définie comme étant une fonction qui, à chaque ensemble de croyances, associe une base de croyances cohérente, notée $\Delta_{IC}(E)$. Différents postulats permettant de caractériser le comportement rationnel des opérateurs de fusion ont été proposés [KP99].

(KP0) $\Delta_{IC}(E) \models IC$.

(KP1) Si IC est cohérent, alors $\Delta_{IC}(E)$ est cohérent.

(KP2) Si E est cohérent avec IC , alors $\Delta_{IC}(E) = \bigwedge E \wedge IC$.

(KP3) Si $E_1 \equiv E_2$ et $IC_1 \equiv IC_2$, alors $\Delta_{IC_1}(E_1) \equiv \Delta_{IC_2}(E_2)$.

(KP4) Si $K_1 \models IC$ et $K_2 \models IC$, alors $\Delta_{IC}(K_1 \sqcup K_2) \wedge K_1 \not\models \perp$ implique $\Delta_{IC}(K_1 \sqcup K_2) \wedge K_2 \not\models \perp$.

(KP5) $\Delta_{IC}(E_1) \wedge \Delta_{IC}(E_2) \models \Delta_{IC}(E_1 \sqcup E_2)$.

(KP6) Si $\Delta_{IC}(E_1) \wedge \Delta_{IC}(E_2)$ est cohérent, alors $\Delta_{IC}(E_1 \sqcup E_2) \models \Delta_{IC}(E_1) \wedge \Delta_{IC}(E_2)$.

(KP7) $\Delta_{IC_1}(E) \wedge IC_2 \models \Delta_{IC_1 \wedge IC_2}(E)$.

(KP8) Si $\Delta_{IC_1}(E) \wedge IC_2$ est cohérent, alors $\Delta_{IC_1 \wedge IC_2}(E) \models \Delta_{IC_1}(E) \wedge IC_2$.

(KP0) stipule que le résultat de la fusion doit satisfaire les contraintes d'intégrité. D'après **(KP1)**, le résultat de la fusion doit être cohérent si les contraintes d'intégrité sont cohérentes. La conjonction des bases de croyances est cohérente, **(KP2)** indique qu'il doit s'agir du résultat de la fusion. Selon **(KP3)**, la fusion est indépendante de la syntaxe. **(KP4)** stipule qu'il ne peut pas être donné de priorité aux bases de croyances. Les postulats **(KP5)** et **(KP6)** fonctionnent ensemble; ils indiquent que si deux sous-groupes sont d'accord sur une alternative alors cette alternative doit faire partie du résultat de la fusion. **(KP7)** et **(KP8)** expriment des conditions sur les conjonctions de contraintes d'intégrité.

Certains de ces postulats peuvent être, et ont d'ailleurs été, remis en question. Pour le postulat **(KP0)**, par exemple, est-il vraiment nécessaire que les contraintes soient explicites dans le résultat de l'opération de fusion? Quel est l'intérêt de l'équivalence syntaxique, postulat **(KP3)**, lorsque l'on sait que dans la plupart des bases, les variables propositionnelles ne sont pas indépendantes? Pour un récapitulatif de ces discussions, voir [Kon99].

3.2.3.1 Postulats KP pour la fusion pure

Dans le même article, un ensemble de postulats pour la fusion pure a été proposé.

(FP1) $\Delta(E)$ est cohérent.

(FP2) Si E est cohérent, alors $\Delta(E) = \bigwedge E$.

(FP3) Si $E_1 \equiv E_2$, alors $\Delta(E_1) \equiv \Delta(E_2)$.

(FP4) Si $K_1 \wedge K_2$ est incohérent, alors $\Delta(K_1 \sqcup K_2) \not\models K_1$.

(FP5) $\Delta(E_1) \wedge \Delta(E_2) \models \Delta(E_1 \sqcup E_2)$.

(FP6) Si $\Delta(E_1) \wedge \Delta(E_2)$ est cohérent, alors $\Delta(E_1 \sqcup E_2) \models \Delta(E_1) \wedge \Delta(E_2)$.

3.2.3.2 Théorème de représentation pour la fusion

De façon similaire à la révision de croyances, on peut définir un assignement synchrétique pour faire le lien entre les postulats de Konieczny-Pino Perez et des pré-ordres totaux sur les interprétations.

Définition 3.2.3 (Assignement synchrétique). *Soient I, J deux interprétations. Un assignement synchrétique est une fonction qui associe à chaque profil de croyances E un pré-ordre $\leq_E$ sur les interprétations telle que pour tous profils de croyances E, E_1 et E_2 et pour toutes bases de croyances K, K' les conditions suivantes sont satisfaites :*

1. Si $I \models E$ et $J \models E$, alors $I \simeq_E J$.
2. Si $I \models E$ et $J \not\models E$, alors $I <_E J$.
3. Si $E_1 \leftrightarrow E_2$, alors $\leq_{E_1} = \leq_{E_2}$.
4. $\forall I \models K, \exists J \models K', J \leq_{K \sqcup K'} I$.
5. Si $I \leq_{E_1} J$ et $I \leq_{E_2} J$, alors $I \leq_{E_1 \sqcup E_2} J$.
6. Si $I <_{E_1} J$ et $I \leq_{E_2} J$, alors $I <_{E_1 \sqcup E_2} J$.

Théorème 3.2.1 (Théorème de représentation). *Un opérateur Δ est un opérateur de fusion contrainte si et seulement si il existe un assignement synchrétique qui associe à chaque profil de croyances E un pré-ordre total $\leq_E$ tel que :*

$$\text{Mod}(\Delta_{IC}(E)) = \min(\text{Mod}(IC), \leq_E)$$

3.2.4 Arbitrage et majorité

Les opérateurs de fusion peuvent être séparés en deux familles : les opérateurs d'arbitrage et les opérateurs majoritaires. Prenons un exemple pour expliquer leur différence.

Exemple 3.2.1. *Imaginons des camarades de lycée qui se retrouvent 10 ans après avoir passé leurs diplômes. La plupart sont restés proche de leur ville d'origine et ont des horaires de bureau. Cependant, l'un d'entre eux habite maintenant à 200 kilomètres de là. Ainsi, s'ils veulent se donner rendez-vous, la plupart d'entre eux vont souhaiter le faire proche de leur ville d'origine et aux heures normales de repas, alors que leur ami*

éloigné souhaitera probablement que leur rencontre ait lieu dans un lieu intermédiaire et probablement un peu plus tard pour avoir le temps d'arriver après le travail.

Un opérateur majoritaire prendra en compte l'avis de la majorité. Le repas aura lieu aux heures normales de repas, dans la ville d'origine, excluant leur ami éloigné.

Un opérateur d'arbitrage essaiera de satisfaire au mieux chacun des acteurs. Dans notre cas, peut être que leur rendez-vous se passera dans leur ville d'origine mais à une heure permettant à leur ami éloigné de les rejoindre.

De façon formelle, Konieczny et Pino-Perez [KP02] donnent les définitions suivantes :

Définition 3.2.4 (Arbitrage). *Un opérateur d'arbitrage satisfait la propriété suivante :*

$$\left. \begin{array}{l} \Delta_{IC_1}(K_1) \equiv \Delta_{IC_2}(K_2) \\ \Delta_{IC_1 \leftrightarrow IC_2}(K_1 \sqcup K_2) \leftrightarrow (IC_1 \leftrightarrow \neg IC_2) \\ IC_1 \not\models IC_2 \\ IC_2 \not\models IC_1 \end{array} \right\} \rightarrow \Delta_{IC_1 \vee IC_2}(K_1 \sqcup K_2) \equiv \Delta_{IC_1}(K_1)$$

Définition 3.2.5 (Majorité). *Un opérateur de fusion majoritaire satisfait la propriété suivante :*

$$\exists n \quad \Delta_{IC}(K_1 \sqcup K_2^n) \models \Delta_{IC}(K_2)$$

Il existe également une propriété, opposée à la propriété ci-dessus qui définit qu'un opérateur est insensible à la répétition des opinions.

Définition 3.2.6 (Indépendance à la majorité). *Un opérateur de fusion indépendant à la majorité satisfait la propriété suivante :*

$$\forall n, \Delta(K_1 \sqcup K_2^n) \equiv \Delta(K_1 \sqcup K_2)$$

3.2.5 Opérateurs

Nous allons présenter ici quelques opérateurs de fusion. Un premier opérateur de fusion intuitif peut être celui inspiré par la révision par intersection totale [Kon99]. Plus précisément, on considère que le résultat est la conjonction des bases de croyances du profil si celle-ci est cohérente et $\top$ si elle ne l'est pas.

$$\Delta(E) = \begin{cases} \bigwedge_{K_i \in E} K_i & \text{si } \bigcup K_i \text{ est cohérent} \\ \top & \text{dans le cas contraire} \end{cases}$$

Une autre approche intuitive serait de considérer la conjonction des bases de croyances à fusionner.

$$\Delta(E) = \bigwedge_{K_i \in E} K_i$$

Mais ces opérations, beaucoup trop drastiques ne produisent pas de résultat satisfaisant. Une dernière approche intuitive serait de considérer la disjonction des bases de croyances à fusionner.

$$\Delta(E) = \bigvee_{K_i \in E} K_i$$

Mais cette opération est trop lache et ne permet pas de définir clairement les croyances spécifiques au groupe.

Entre ces deux approches, de nombreux travaux ont porté sur la définition d'opérateurs de fusion selon un point de vue sémantique [KLM04] ou un point de vue syntaxique [Cho98, MGC01]. D'autres opérateurs ont été définis dans [BL02] dont une mise en œuvre a été proposée dans [GH08].

3.2.5.1 Opérateurs sémantiques

Une première tentative de définir un opérateur de fusion a été effectuée par P.R. Revesz [Rev93]. Appelé opérateur d'adéquation sémantique (*model fitting*), ces opérateurs ont pour but d'effectuer un arbitrage entre deux bases de croyances. Une autre étude intéressante est celle effectuée par Liberatore et Schaerf [LS95]; mais elle aussi ne gère que la fusion de deux bases de croyances.

Ainsi, les premiers véritables opérateurs de fusion, au sens où ils gèrent plus de deux bases de croyances, sont probablement ceux définis dans [KP99].

Ces opérateurs sont décrits comme étant des opérateurs de fusion contraintes. Leur définition est donnée indépendamment de la fonction de distance d , ils sont décrits ci-dessous.

Opérateur Somme

Pour l'opérateur Somme, la distance entre une interprétation et un profil de croyances est définie comme étant la somme des distances entre l'interprétation et les bases de croyances du profil.

Définition 3.2.7. Soient $E = \{K_1, \dots, K_n\}$ un profil de croyances et I une interprétation sur $Atome(E)$. La distance $d_{d,\Sigma}$ entre une interprétation et un profil de croyances est définie comme étant :

$$d_{d,\Sigma}(I, E) = \sum_{K_i \in E} d(I, K_i)$$

Ce qui permet de définir le pré-ordre sur les interprétations suivant :

$$I \leq_E^{d,\Sigma} J \text{ si et seulement si } d_{d,\Sigma}(I, E) \leq d_{d,\Sigma}(J, E)$$

L'opérateur $\Delta^{d,\Sigma}$ est défini par :

$$Mod(\Delta_{IC}^{d,\Sigma}(E)) = \min(Mod(IC), \leq_E^{d,\Sigma})$$

Exemple 3.2.2. Pour illustrer ces opérateurs sémantiques, nous prenons l'exemple issu de [Rev93].

Ainsi, prenons trois étudiants qui expriment leur point de vue sur les langages de bases de données qu'ils souhaitent apprendre durant cette année universitaire. Ils ont le choix entre : SQL (noté s), O_2 (noté o), Datalog (noté d). Le premier souhaite étudier SQL ou O_2 mais pas Datalog ($K_1 = \{\neg d, s \vee o\}$). Le second veut étudier seulement Datalog

ou O_2 mais pas les deux ($K_2 = \{\neg s, d \vee o, \neg o \vee \neg d\}$). Le troisième veut étudier les trois ($K_3 = \{s, d, o\}$).

Dans cet exemple, le décideur est le professeur, et les trois groupes d'étudiants peuvent être assimilés à trois agents. Le professeur doit prendre une décision en respectant au mieux le choix des étudiants selon une stratégie donnée. L'union des avis de chaque étudiant $\{\neg d, s \vee o, \neg s, d \vee o, \neg o \vee \neg d, s, d, o\}$ est incohérente, il faut donc faire un choix.

Soit $E = \{K_1, K_2, K_3\}$ un profil de croyances avec $K_1 = \{(s \vee o) \wedge \neg d\}$, $K_2 = \{(\neg s \wedge d \wedge \neg o) \vee (\neg s \wedge \neg d \wedge o)\}$ et $K_3 = \{s \wedge d \wedge o\}$. Ces bases de croyances ont les modèles suivants : $Mod(K_1) = \{\{s, \neg d, \neg o\}, \{\neg s, \neg d, o\}, \{s, \neg d, o\}\}$, $Mod(K_2) = \{\{\neg s, d, \neg o\}, \{\neg s, \neg d, o\}\}$ et $Mod(K_3) = \{\{s, d, o\}\}$ et on peut supposer qu'il n'existe pas de contraintes $IC = \top$.

En utilisant la distance de Hamming, on obtient le tableau suivant :

ω	K_1	K_2	K_3	$d_\Sigma(E)$
$\{\neg s, \neg d, \neg o\}$	1	1	3	5
$\{\neg s, \neg d, o\}$	0	0	2	2
$\{\neg s, d, \neg o\}$	2	0	2	4
$\{\neg s, d, o\}$	1	1	1	3
$\{s, \neg d, \neg o\}$	0	2	2	4
$\{s, \neg d, o\}$	0	1	1	2
$\{s, d, \neg o\}$	1	1	1	3
$\{s, d, o\}$	1	2	0	3

L'opérateur Somme réalise en quelque sorte une élection de l'interprétation la plus proche des croyances exprimées dans le profil.

Opérateur *Max*

Pour l'opérateur *Max*, la distance entre une interprétation et un profil de croyances est définie comme étant le maximum parmi les distances entre l'interprétation et les bases de croyances du profil.

Définition 3.2.8. La distance $d_{d,Max}$ entre une interprétation et un profil de croyances est définie comme étant :

$$d_{d,Max}(I, E) = \max_{K_i \in E} d(I, K_i)$$

Ce qui permet de définir le pré-ordre sur les interprétations suivant :

$$I \leq_E^{d,Max} J \text{ si et seulement si } d_{d,Max}(I, E) \leq d_{d,Max}(J, E)$$

L'opérateur $\Delta^{d,Max}$ est défini par :

$$Mod(\Delta_{IC}^{d,Max}(E)) = \min(Mod(IC), \leq_E^{d,Max})$$

Exemple 3.2.3. Nous reprenons l'exemple décrit dans 3.2.2 pour illustrer l'opérateur *Max*.

ω	K_1	K_2	K_3	$d_{Max}(E)$
$\{\neg s, \neg d, \neg o\}$	1	1	3	3
$\{\neg s, \neg d, o\}$	0	0	2	2
$\{\neg s, d, \neg o\}$	2	0	2	2
$\{\neg s, d, o\}$	1	1	1	1
$\{s, \neg d, \neg o\}$	0	2	2	2
$\{s, \neg d, o\}$	0	1	1	1
$\{s, d, \neg o\}$	1	1	1	1
$\{s, d, o\}$	1	2	0	2

Ainsi, l'interprétation choisie est celle qui minimise la distance entre l'interprétation choisie et la base de croyances la plus éloignée de cette interprétation. C'est pour cela que Revesz a donné le nom de *minimax* à l'opérateur $d_{d_H, Max}$ (où d_H est la distance de Dalal) dans [Rev93].

Opérateur $GMax$

L'opérateur $GMax$ (pour Max généralisé) est un le raffinement lexicographique de l'opérateur Max .

Définition 3.2.9. Pour chaque interprétation I , on construit la liste $(d_1^I, \dots, d_n^I)$ des distances entre cette interprétation et les n bases de croyances du profil E , i.e. $d_j^I = d(I, K_j)$. On construit L_I^E la liste obtenue à partir de $(d_1^I, \dots, d_n^I)$ en la triant dans l'ordre décroissant. On définit $d_{d, GMax}(I, E) = L_I^E$. On note $\leq_{lex}$ l'ordre lexicographique entre les vecteurs d'entiers.

On définit le pré-ordre sur les interprétations suivant :

$$I \leq_E^{d, GMax} J \text{ si et seulement si } d_{d, GMax}(I, E) \leq_{lex} d_{d, GMax}(J, E)$$

Et l'opérateur $\Delta^{d, GMax}$ est défini par :

$$Mod(\Delta_{IC}^{d, GMax}(E)) = \min(Mod(IC), \leq_E^{d, GMax})$$

Exemple 3.2.4. Nous reprenons l'exemple décrit dans 3.2.2 pour illustrer l'opérateur $GMax$.

ω	K_1	K_2	K_3	$d_{GMax}(E)$
$\{\neg s, \neg d, \neg o\}$	1	1	3	(311)
$\{\neg s, \neg d, o\}$	0	0	2	(200)
$\{\neg s, d, \neg o\}$	2	0	2	(220)
$\{\neg s, d, o\}$	1	1	1	(111)
$\{s, \neg d, \neg o\}$	0	2	2	(220)
$\{s, \neg d, o\}$	0	1	1	(110)
$\{s, d, \neg o\}$	1	1	1	(111)
$\{s, d, o\}$	1	2	0	(210)

En quelque sorte, l'opérateur $GMax$ affine les choix qui aurait été fait par l'opérateur Max . Dans l'exemple 3.2.4, Δ_{Max} aurait eu pour solution l'ensemble d'interprétations : $\{ \{s, \neg d, \neg o\}, \{ \neg s, d, \neg o\}, \{ \neg s, \neg d, o\} \}$. Et $GMax$ affine encore ce choix en ne gardant que $\{ \{ \neg s, d, \neg o\} \}$.

D'autres opérateurs ont définis comme, par exemple, l'opérateur de fusion reposant sur des notions de morphologie mathématique [BL02] et mise en œuvre grâce aux Diagrammes de Décisions Binaires [GH08].

3.2.5.2 Opérateurs syntaxiques

Les opérateurs syntaxiques déterminent un ensemble de formules cohérent à partir des formules issus du profil de croyances à fusionner.

Opérateurs de combinaison

Les premiers opérateurs syntaxiques, appelés alors opérateurs de combinaison, sont apparus dans [BKMS91] et [BKM91]. Leur définition est axée sur la recherche de sous-ensembles maximaux cohérents.

Définition 3.2.10. Soient K une base de croyances et IC une formule représentant les contraintes et E un profil de croyances. $MAXCONS(K, IC)$ dénote l'ensemble des sous-ensembles maximaux (au sens de l'inclusion ensembliste) cohérents de $K \cup IC$ et qui contiennent IC . On pose alors $MAXCONS(E, IC) = MAXCONS(\bigwedge E, IC)$ où $\bigwedge E$ est la conjonction des bases de croyances de E .

L'opérateur $MAXCONS_{card}$ qui sélectionne les sous-ensembles maximaux cohérents au sens de la cardinalité est défini comme suit :

Définition 3.2.11. Soient K une base de croyances et IC une formule représentant les contraintes et E un profil de croyances. $MAXCONS_{card}(K, IC)$ dénote l'ensemble des sous-ensembles maximaux - au sens de la cardinalité - cohérents de $K \cup IC$ et qui contiennent IC . On pose alors $MAXCONS_{card}(E, IC) = MAXCONS_{card}(\bigwedge E, IC)$.

Il est à noter que les contraintes sont intégrées dans la définition. À partir de cette définition, on peut donner celle des opérateurs de combinaison.

Définition 3.2.12. Soit E un profil de croyances et IC un ensemble de formules représentant les contraintes.

$$\begin{aligned}
 \Delta_{IC}^{C1} &= MAXCONS(E, IC) \\
 \Delta_{IC}^{C3} &= \{M : M \in MAXCONS(E, \top) \text{ et } M \wedge IC \text{ cohérent} \} \\
 \Delta_{IC}^{C4} &= \{M : M \in MAXCONS_{card}(E, IC)\} \\
 \Delta_{IC}^{C5} &= \{ \{M : M \in MAXCONS(E, \top) \text{ et } M \wedge IC \text{ cohérent} \} \\
 &\quad \text{si cet ensemble est non vide} \\
 &\quad IC \\
 &\quad \text{sinon} \}
 \end{aligned}$$

Les opérateurs de combinaison oublient la provenance des croyances. De plus, ils sont insensibles à la répétition des informations. L'opérateur Δ_{IC}^{C1} servira de base pour les opérateurs présents dans la suite.

Fonctions de sélection

De manière analogue au cadre AGM, on définit des fonctions de sélection pour la fusion de bases de croyances. Ces fonctions de sélection permettent de sélectionner des maximaux cohérents afin d'affiner encore le choix fait par les opérateurs de combinaison. On parlera ici de l'opérateur majoritaire drastique et l'opérateur Intersection.

Opérateur majoritaire drastique Une autre manière de considérer la fusion est d'essayer de rester proche des bases de croyances originales. Ainsi, l'opérateur majoritaire drastique, décrit dans [Kon99], est basé sur une définition des distances qui favorise cette orientation.

Définition 3.2.13. Soient un profil de croyances $E = \{K_1, \dots, K_n\}$ et un maximal cohérent M . La distance entre un maximal cohérent M est une base de croyances K est :

$$d_D(M, K) = \begin{cases} 0 & \text{si } M \wedge K \text{ cohérent} \\ 1 & \text{sinon} \end{cases}$$

La distance entre un maximal cohérent M et un ensemble de croyances E est alors :

$$d_D(M, E) = \sum_{1 \leq i \leq n} d_D(M, K_i)$$

Et l'opérateur majoritaire drastique est défini par :

$$\Delta_{IC}^D(E) = \{M \in \Delta_{IC}^{C1}(E) : d_D(M, E) = \min_{M_i \in \Delta_{IC}^{C1}(E)} (d_D(M_i, K))\}$$

L'opérateur majoritaire drastique choisit les maximaux cohérents qui sont cohérents avec le maximum de bases de croyances parmi le profil de croyances.

Opérateur Intersection Cet opérateur est défini dans [Kon99] à partir d'une définition de la distance qui représente la cardinalité de l'intersection entre les bases de croyances.

Définition 3.2.14. Soient un profil de croyances $E = \{K_1, \dots, K_n\}$ et un maximal cohérent M . La distance entre un maximal cohérent M est une base de croyances K est :

$$d_{\cap}(M, K) = |M \cap K|$$

La distance entre un maximal cohérent M et un profil de croyances E est alors :

$$d_{\cap}(M, E) = \sum_{1 \leq i \leq n} d_{\cap}(M, K_i)$$

Et l'opérateur majoritaire drastique est défini par :

$$\Delta_{IC}^D(E) = \{M \in \Delta_{IC}^{C1}(E) : d_{\cap}(M, E) = \max_{M_i \in \Delta_{IC}^{C1}(E)} (d_{\cap}(M_i, K))\}$$

3.2.6 Fusion réversible

Il peut, dans certains contextes, être intéressant de pouvoir revenir en arrière après une opération de fusion, de pouvoir retrouver les croyances dans les bases du profil qui a été fusionné ainsi que les préférences exprimées sur les agents. Dans [SDP06] J.Seinturier et al. proposent une méthode pour effectuer ce retour en arrière, appelé fusion réversible. Cette méthode est basée sur la représentation des préférences à l'aide de polynômes.

On rappelle que l'ensemble des polynômes de la forme $\sum_{i=0}^n p_i x^i$, $p_i \in \mathbb{R}$ est noté $\mathbb{R}[x]$. On appelle décalage vers la droite de k positions la multiplication par x^k , réciproquement, on appelle décalage vers la gauche de k positions une division par x^k .

Dans le cadre de la fusion de bases de croyances, il est possible que des préférences soient exprimées sur les agents (préférences externes) ou sur les croyances exprimées par les agents (préférences internes). On définit donc des fonctions de pondération pour représenter ces préférences.

3.2.6.1 Pré-ordre externe

Ainsi, on associe une fonction de pondération à tout profil de croyances :

Définition 3.2.15. Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances, une fonction de pondération externe est une application telle que :

$$\begin{aligned} q : E &\longrightarrow \mathbb{R} \\ K_i &\longrightarrow q(K_i) \end{aligned}$$

À partir de cette pondération, on définit un pré-ordre externe sur les profils qui est noté $\leq_E$ et représenté par le polynome $q(x) = \sum_{i=1}^n q(K_i)x^{i-1}$

Définition 3.2.16. Un pré-ordre externe sur un profil de croyances E muni d'une fonction de pondération q est défini par :

$$\forall K_i, K_j \in E, K_i \leq_E K_j \text{ si et seulement si } q(K_i) \leq q(K_j)$$

3.2.6.2 Pré-ordre interne

De la même manière, nous définissons une pondération interne aux bases de croyances afin de pouvoir définir un pré-ordre interne.

Définition 3.2.17. Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances et $\mathcal{W}$ l'ensemble des interprétations du langage. Une fonction de pondération interne est une application telle que :

$$\begin{aligned} p_{K_i} : \mathcal{W} &\longrightarrow \mathbb{R}[x] \\ \omega &\longrightarrow p_{K_i}(\omega) \end{aligned}$$

Définition 3.2.18. Un pré-ordre interne sur les interprétations d'une base de croyances K_i munie d'une fonction de pondération p_{K_i} est définie par :

$$\forall \omega_j, \omega_k \in \mathcal{W}, \omega_j \leq_{K_i} \omega_k \text{ si et seulement si } p_{K_i}(\omega_j) \leq p_{K_i}(\omega_k)$$

3.2.6.3 Pré-ordre global

À partir des pré-ordres internes et des pré-ordres externes, on construit un pré-ordre global.

Plusieurs bases de croyances peuvent avoir la même pondération, il est nécessaire de les différencier pour pouvoir les retrouver lors de l'opération inverse. Une fonction inversible de rang absolu est ainsi définie dans ce but.

Définition 3.2.19. Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. Une fonction de rang absolu, notée r est une fonction de E dans $\mathbb{N}$ qui associe à chaque base propositionnelle K_i un rang absolu $r(K_i)$ tel que :

- Si $K_i <_E K_j$ alors $r(K_i) < r(K_j)$
- Si $K_i =_E K_j$ et $i < j$ alors $r(K_i) < r(K_j)$

Définition 3.2.20. Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. Soit $p_{K_i}(\omega)$, avec $1 \leq i \leq n$ les poids internes, q la fonction de pondération externe et r la fonction de rang absolu pour E . Le poids global, noté $p_{K_1 \oplus \dots \oplus K_n}(\omega)$, pour une interprétation ω est tel que :

$$p_{K_1 \oplus \dots \oplus K_n}(\omega) = \sum_{i=1}^n p_{K_i}(\omega) x^{\sum_{j=1}^{r(K_i)-1} MAX_{r^{-1}(j)}}$$

avec

$$MAX_{r^{-1}(j)} = \max_{\omega' \in \mathcal{W}} (deg(p_{r^{-1}(j)}(\omega')) + 1)$$

Dans la suite, nous écrirons $p_{\oplus}(\omega)$ au lieu de $p_{K_1 \oplus \dots \oplus K_n}(\omega)$.

De la même manière que pour les pré-ordres internes et externes, la définition d'une fonction de pondération globale permet la définition d'un pré-ordre global sur les interprétations :

Définition 3.2.21. Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. Soit ω et ω' deux interprétations de poids globaux respectifs $p_{\oplus}(\omega)$ et $p_{\oplus}(\omega')$. Le pré-ordre global, noté $\leq_{K_1 \oplus \dots \oplus K_n}$ est tel que :

$$\omega \leq_{K_1 \oplus \dots \oplus K_n} \omega' \text{ si et seulement si } p_{\oplus}(\omega) \leq p_{\oplus}(\omega')$$

Exemple 3.2.5. Pour illustrer ceci prenons en considération le profil de croyances $E = \{K_1, K_2, K_3\}$ avec $K_1 = \{(s \vee o \wedge \neg d)\}$, $K_2 = \{(\neg s \wedge d \neg o) \vee (\neg s \wedge \neg d \wedge o)\}$ et $K_3 = \{s \wedge o \wedge d\}$.

Les poids externes sont $q(K_1) = 2$, $q(K_2) = 2$ et $q(K_3) = 1$. Le poids externe global est donc : $q_{\oplus} = 2 + 2x + 1x^2$ et le rang absolu est donc : $r(K_1) = 2$, $r(K_2) = 3$ et $r(K_3) = 1$. Le tableau suivant rassemble les poids internes de chaque interprétation :

À partir des polynômes représentant les poids internes, sont déduites les valeurs $MAX_{r^{-1}(1)} = MAX_{K_3} = 2$, $MAX_{r^{-1}(2)} = MAX_{K_1} = 3$ et $MAX_{r^{-1}(3)} = MAX_{K_2} = 2$. Ainsi, la définition du poids global donne :

$$p_{\oplus}(\omega_i) = p_{K_3}(\omega_i) + p_{K_1}(\omega_i)x^2 + p_{K_2}(\omega_i)x^4$$

ce qui donne les poids globaux suivants pour les interprétations :

Interprétations	K_1	K_2	K_3
$\omega_0 = \{\neg s, \neg d, \neg o\}$	x	1	$1 + x$
$\omega_1 = \{\neg s, \neg d, o\}$	1	0	x
$\omega_2 = \{\neg s, d, \neg o\}$	x	0	x
$\omega_3 = \{\neg s, d, o\}$	x	1	1
$\omega_4 = \{s, \neg d, \neg o\}$	1	x^2	x
$\omega_5 = \{s, \neg d, o\}$	0	1	1
$\omega_6 = \{s, d, \neg o\}$	x	1	1
$\omega_7 = \{s, d, o\}$	x	x^2	0

TABLE 3.1 – Poids internes des interprétations

Interprétations	K_1	K_2	K_3	$p_{\oplus}$
$\omega_0 = \{\neg s, \neg d, \neg o\}$	x	1	$1 + x$	$1 + x + x^3 + x^4$
$\omega_1 = \{\neg s, \neg d, o\}$	1	0	x	$x + x^2$
$\omega_2 = \{\neg s, d, \neg o\}$	x	0	x	$x + x^3$
$\omega_3 = \{\neg s, d, o\}$	x	1	1	$1 + x^3 + x^4$
$\omega_4 = \{s, \neg d, \neg o\}$	1	x^2	x	$x + x^2 + x^6$
$\omega_5 = \{s, \neg d, o\}$	0	1	1	$1 + x^4$
$\omega_6 = \{s, d, \neg o\}$	x	1	1	$1 + x^3 + x^4$
$\omega_7 = \{s, d, o\}$	x	x^2	0	$x^3 + x^6$

TABLE 3.2 – Poids globaux des interprétations

3.2.6.4 Reversibilité

Il est facile, dans l'exemple précédent, de retrouver les préférences internes de chaque base pour chaque interprétation. Par exemple, les éléments du polynome en x^0 et x correspondent aux préférences exprimés dans K_3 et celle en x^2 et x^3 correspondent aux préférences exprimées dans K_1 . Plus formellement, les poids internes peuvent être retrouvés grâce à la formule :

$$p_{K_i}(\omega) = \frac{p_{\oplus}(\omega) \bmod x^{\sum_{l=1}^{r(K_i)} (\max_{r^{-1}(l)} + 1)}}{x^{\sum_{k=1}^{r(K_i)} (\max_{r^{-1}(k)} + 1)}}$$

Exemple 3.2.6. Reprenons l'exemple 3.2.5. Le polynome $q_{\oplus} = 2 + 2x + 1x^2$ permet de déduire que 3 bases ont servi à l'opération et que leur poids respectif étaient $q(K_1) = q(K_2) = 2$ et $q(K_3) = 1$ et donc le rang absolu de chaque base : $r(1) = 2$, $r(2) = 3$ et $r(3) = 1$.

Par exemple l'interprétation ω_3 , avec $p_{\oplus}(\omega_3) = 1 + x^3 + x^4$, nous obtenons $p_{K_1}(\omega_3) = (1 + x^3 + x^4 \bmod x^4) / x^2 = x$. De la même manière $p_{K_2}(\omega_3) = 1$ et $p_{K_3}(\omega_3) = 1$.

3.2.7 Fusion possibiliste

Nous nous intéressons maintenant à la fusion de bases de croyances possibilistes. Nous présentons en particulier les travaux de [BDKP02a, BDKP00].

Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances possibilistes. Chaque base de croyances mène à une distribution de possibilité π_i . Ainsi à chaque interprétation ω , on peut associer un vecteur $\nu_E(\omega)$ constitué des degrés de possibilité $\pi_i(\omega)$ de K_i pour l'interprétation ω . $\nu_E(\omega)$ est défini de la manière suivante :

$$\nu_E(\omega) = \langle \pi_1(\omega), \dots, \pi_n(\omega) \rangle$$

Il faut maintenant agréger les degrés de possibilités du vecteur $\nu_E(\omega)$ afin d'obtenir un degré unique $\pi_E(\omega)$ qui doit respecter les propriétés suivantes :

- $\forall \omega \in \mathcal{W}$, si $\forall K_i \in E, \pi_i(\omega) = 1$ alors $\pi_E(\omega) = 1$;
- $\forall \omega, \omega' \in \mathcal{W}$, si $\forall K_i \in E, \pi_i(\omega) \geq \pi_i(\omega')$ alors $\pi_E(\omega) \geq \pi_E(\omega')$.

À partir de cette définition, il est possible de définir plusieurs types d'opérateurs de fusion possibiliste. Notamment les opérateurs conjonctifs et disjonctifs.

Définition 3.2.22. Un opérateur de fusion possibiliste, noté $\Delta^{\oplus}$, est dit conjonctif si et seulement s'il repose sur un opérateur d'agréation, noté $\oplus$, tel que :

$$\forall a \in [0, 1], \oplus(a, 1) = \oplus(1, a) = a$$

Il est dit disjonctif si et seulement s'il repose sur un opérateur d'agréation, noté $\oplus$, tel que :

$$\forall a \in [0, 1], \oplus(a, 1) = \oplus(1, a) = 1$$

3.2.8 Fusion de bases de croyances exprimées en logique possibiliste

Une méthode pour la fusion de bases de croyances possibiliste a été proposée par Guilin Qi [Qi07]. Cette méthode de fusion donne pour résultat une distribution de possibilité. Sa contre-partie syntaxique a également été donnée dans le même article.

3.2.8.1 Algorithme sémantique

Soient E un profil de croyances exprimées en logique possibiliste. On suppose que $E = K_1 \sqcup \dots \sqcup K_n = \{(f_1 \alpha_1), \dots, (f_m \alpha_m)\}$. On réorganise les différents poids apparaissant dans E tels que $\beta_1 > \beta_2 > \dots > \beta_l > 0$ où les $\beta_{1 \leq i \leq l}$ sont l'ensemble des poids apparaissant dans E . Soit IC la base de croyances représentant les contraintes pesant sur l'opération de fusion $\Delta_{IC}(E)$. On note K_j^i les formules de K_j ayant β_i pour valeur de nécessité. De même, $K^i = K_1^i \sqcup \dots \sqcup K_n^i$.

L'algorithme fixe la distribution de possibilité pour l'ensemble des interprétations en partant des interprétations les moins considérées vers les interprétations les plus considérées. Dans un premier temps, on considère chaque interprétation ω qui falsifie les contraintes IC de l'opération et on lui attribue la possibilité $\pi_{E,IC,\Delta}(\omega) = 0$. Puis, on considère chaque interprétation ω à laquelle on a toujours pas attribué de possibilité et qui falsifie au moins une formule de K^1 en lui attribuant la possibilité $\pi_{E,IC,\Delta}(\omega) = 1 - \beta_1$. Ensuite, on recommence l'opération pour les β_2 en attribuant la possibilité $1 - \beta_2$ aux interprétations non traitées falsifiant au moins une formule de K^2 et on reproduit l'opération pour tous les K^i jusqu'à K^n . Les interprétations qui ne falsifient aucune formule de E auront la possibilité 1.

Algorithme 8 Algorithme sémantique

```

 $EC_1 \leftarrow Mod(IC), i = 1;$ 
pour tout  $\omega \not\models IC$  faire
     $\pi_{E,IC,\Delta}(\omega) = 0;$ 
fin pour
tantque  $i \leq l$  faire
     $(K_k)^i = \{(f_{k,j} \alpha_{k,j}) \in K_k \mid \alpha_{k,j} = \beta_i\}$  pour tout  $k \in 1, \dots, n;$ 
     $K^i = K_1^i \sqcup \dots \sqcup K_n^i;$ 
     $EC_{i+1} \leftarrow mod(\Delta_{EC_i}(K^i));$ 
    pour tout  $\omega \in EC_i \setminus EC_{i+1}$  faire
         $\pi_{E,IC,\Delta}(\omega) = 1 - \beta_i;$ 
    fin pour
     $i \leftarrow i + 1;$ 
fin tantque
pour tout  $\omega \in EC_{l+1}$  faire
     $\pi_{E,IC,\Delta}(\omega) = 1;$ 
fin pour
renvoyer  $\pi_{E,IC,\Delta};$ 

```

3.2.8.2 Algorithme syntaxique

Le résultat de l'opération précédente est une distribution de possibilité sur l'ensemble des interprétations. Un autre algorithme permet de donner le résultat d'une opération de fusion de bases de croyances possibilistes sous forme syntaxique [Qi07].

Pour l'algorithme suivant, on considère le profil de croyances $E = \{K_1, \dots, K_n\}$, IC la base des croyances représentant les contraintes et un opérateur de fusion syntaxique $\Delta_{IC}(E)$.

L'algorithme 9 construit l'ensemble des formules au fur et à mesure. Pour chaque β_i , les croyances déduites précédemment sont considérées comme des contraintes IC_i . Au départ, seules les contraintes font partie de l'ensemble des solutions $IC_1 = IC$. Pour chaque β_i , l'opérateur de fusion $\Delta_{IC_i}(K^i)$ fournit une interprétation satisfaisant au mieux les formules de K^i et qui soit cohérente avec les formules de IC_i . Puis les formules de K^i satisfaisant cette interprétation sont ajoutées à l'ensemble des solutions $K_{E,IC,\Delta}$.

Algorithme 9 Algorithme syntaxique

```

 $IC_1 \leftarrow IC, i = 1;$ 
 $K_{E,IC,\Delta} \leftarrow \{(f \ 1) : f \in IC_1\};$ 
tantque  $i \leq l$  faire
   $(K_k)^i = \{(f_{k,j} \ \alpha_{k,j}) \in K_k \mid \alpha_{k,j} = \beta_i\}$  pour tout  $k \in 1, \dots, n;$ 
   $K^i = K_1^i \sqcup \dots \sqcup K_n^i;$ 
   $IC_{i+1} \leftarrow \Delta_{IC_i}((K_1)^k, \dots, (K_n)^k);$ 
   $K_{E,IC,\Delta} \leftarrow K_{E,IC,\Delta} \cup \{(f \ \beta_i) : f \in IC_{i+1} \setminus IC_i\};$ 
   $i = i + 1;$ 
fin tantque
renvoyer  $K_{E,IC,\Delta};$ 

```

3.2.9 Résultat d'équivalence

Les deux méthodes précédentes sont équivalentes. Formellement :

Proposition 3.2.1. *Soient un profil de croyances possibiliste $E = \{K_1, \dots, K_n\}$ et une base de croyances IC représentant les contraintes. On suppose que $\pi_{E,IC,\Delta}$ est la distribution de possibilité obtenue grâce à l'algorithme 8 et $K_{E,IC,\Delta}$ la base de croyances possibiliste obtenue à l'aide de l'algorithme 9. Alors, $\forall \omega \in \mathcal{W}, \pi_{E,IC,\Delta}(\omega) = \pi_{K_{E,IC,\Delta}}(\omega)$*

3.2.9.1 Généralisation du problème de révision

Le problème de fusion est bien évidemment lié au problème de révision. Dans les deux cas, le principe est de choisir un état épistémique représentant le plus proche possible d'un monde. Dans le cas de la révision ce choix est contraint par la nouvelle information alors que dans le cas de la fusion, il est contraint par la contrainte. Ainsi dans [KLM02], le résultat suivant est donné :

Théorème 3.2.2. *Soit $\circ$ un opérateur de révision défini à partir de Δ un opérateur de fusion tel que $K \circ IC = \Delta_{IC}(\{K\})$. Alors :*

$$si \Delta \text{ satisfait } (IC0) - (IC8), \text{ alors } \circ \text{ satisfait } (R1) - (R6)$$

Une autre approche, présentée dans [DDL06], effectue le lien entre la fusion de croyances et la révision itérée.

Conclusion

Nous avons présenté ici les approches liées à la révision de croyances. Tout d'abord l'approche consistant à réviser des théories dans laquelle ont été définis les postulats AGM et les opérateurs basés sur l'identité de Levi. Ensuite, nous avons montré comment les postulats ont été redéfinis par Katsuno et Mendelzon dans le cas où les théories sont représentées par une seule formule. Dans ce cadre, nous avons montré l'opérateur de révision selon Dalal. Puis nous avons traité différentes définitions de révision d'ensembles finis de formules de la logique propositionnelle avec notamment la révision par R-ensembles. Nous avons montré les travaux de Darwiche and Pearl sur la révision itérée.

Ensuite, nous avons présenté les principaux travaux la fusion de croyances. Nous avons montré les travaux de Konieczny et Pino-Perez sur la caractérisation des opérateurs de fusion. Nous avons présenté quelques opérateurs dont les opérateurs utilisant les stratégies Σ , Max et $Gmax$ que nous reprendrons dans notre cadre. Nous avons également introduit les travaux sur la fusion réversible de Seinturier et al. ainsi que la fusion possibiliste défini par Kaci et son implantation réalisé par Qi.

Nous allons maintenant présenter la fusion par R-ensembles, qui est une opération de fusion syntaxique. Mais, contrairement aux travaux précédents où l'opération de fusion consistait à rechercher des sous-ensembles maximaux cohérents de formules, nous choisissons de prendre le point de vue dual consistant à sélectionner les ensembles minimaux de formules permettant de restaurer la cohérence.

Deuxième partie

Fusion par R-ensembles

Introduction

De nos jours, les systèmes informatiques s'orientent de plus en plus vers la distribution de l'information, spécialement dans le domaine des bases de données. Ces sources sont rarement synchronisées et elles sont généralement conflictuelles. Ainsi, la question de l'interrogation et du raisonnement à partir de ces sources d'information est une question cruciale de l'intelligence artificielle. Le problème de fusion de croyances cherche ainsi à obtenir un point de vue global représentant les croyances du groupe en exploitant la complémentarité des bases, en résolvant les conflits potentiels et en réduisant les redondances.

Grâce à son solide cadre formel, la logique propositionnelle fournit un cadre privilégié pour l'étude de ce problème [BKMS91, Rev93, Lin96, Rev97, Cho98]. Pour caractériser le comportement de ces opérateurs, des postulats ont été énoncés [KP99, LM98, Rev93] et de nombreux opérateurs ont été proposés dans le cas où des priorités implicites ou explicites sont disponibles [KP98, LL00, DDL06, RGJM86]. Ces opérateurs peuvent fournir comme solution des ensembles d'interprétations [Kon00], on parle alors d'opérateurs sémantiques, ou des ensembles de formules [MGC01], on parle alors d'opérateurs syntaxiques. Parmi ces travaux, seuls ceux présentés dans [BL02] ont conduit à une mise en œuvre présentée dans [GH08].

Nous proposons ici une méthode de fusion syntaxique basée sur une généralisation de la révision par R-ensembles (voir 3.1.3.3) vers la fusion de croyances. Nous nous plaçons dans le cadre où il n'existe pas de priorités explicites entre les bases de croyances. Dans ce cadre, il est possible d'adapter les stratégies classiques de fusion. Une mise en œuvre de cette méthode de fusion à l'aide d'un programme logique avec sémantique des modèles stables permettant une mise en œuvre, une implantation et une expérimentation.

Dans un premier temps, nous verrons la méthode dans le cas où les bases de croyances sont constituées uniquement de clauses, nous montrerons comment traduire ce problème en un programme logique. Ensuite, nous étendrons notre étude au cas où les croyances sont exprimées par des formules quelconques de la logique propositionnelle. Puis, nous adapterons les stratégies de fusion classiques à notre cadre. Finalement, nous montrerons la méthode que nous proposons pour l'implantation à l'aide de programmes logiques avec sémantique des modèles stables dans ces deux variantes : la prise en compte des stratégies par une modification de l'algorithme *smodels* ou par une traduction utilisant n'importe quel solveur ASP permettant d'utiliser certaines instructions d'optimisation.

Chapitre 4

Définition de l'opération de fusion par R-ensembles

Un concepteur sait qu'il a atteint la perfection, non quand il n'y a rien à ajouter, mais quand il n'y a rien à retirer. *Antoine de Saint-Exupéry*

Introduction

La révision par R-ensembles (voir la section 3.1.3.3) et sa mise œuvre en programmation logique avec sémantique des modèles stables a donné de bons résultats sur l'expérimentation de problèmes de taille réelle comme dans le cas du projet REVIGIS [BBNJ⁺05]. À partir de là, il semblait naturel de vouloir généraliser cette méthode à la fusion de bases de croyances. De manière duale aux approches syntaxiques qui essayent de construire des ensembles de formules maximaux cohérents, l'idée est de déterminer les sous-ensemble de formules à retirer de l'union des bases afin de restaurer la cohérence.

Nous nous plaçons dans le cas où les bases de croyances sont commensurables et où elles sont constituées d'un ensemble fini de formules propositionnelles quelconques. Nous présentons ici la définition de l'opération de Fusion par R-ensembles dans le cas sans contrainte ainsi que dans le cas avec contraintes. Nous présentons plusieurs opérateurs classiques que nous adaptons à notre cadre et nous présentons leurs propriétés. Nous donnons la contrepartie sémantique des opérateurs de fusion proposés. Nous montrons que la fusion par R-ensembles est une généralisation de la Révision par R-ensembles.

4.1 Principes et définitions

La fusion par R-ensembles est une approche syntaxique de fusion de croyances dans le cadre de la logique propositionnelle. Les méthodes classiques de fusion cherchent à conserver des sous-ensembles maximaux cohérents de formules. De façon duale, la fusion par R-ensembles cherche des ensembles minimaux de formules à retirer afin de restaurer la cohérence. Nous commençons donc par chercher les ensembles de formules permettant

de restaurer la cohérence avant de sélectionner parmi ceux-ci les ensembles minimaux.

Cependant, alors que la notion de minimalité dans le cadre de la révision syntaxique est directe, celle-ci s'avère plus subtile dans le cadre de la fusion. La minimalité peut ainsi être déclinée selon différentes stratégies selon que la répétition ou la répartition des formules soient prises en compte ou non. Ces stratégies permettent finalement de réaliser l'opération de fusion à proprement parler.

Nous donnons tout d'abord les définitions de la fusion par R-ensembles dans le cas où il n'existe pas de contraintes sur l'opération de fusion — fusion pure — avant de définir la fusion par R-ensembles dans le cas où il existe des contraintes.

4.1.1 Fusion pure par R-ensembles

La fusion pure d'un profil de croyances $E = \{K_1, \dots, K_n\}$ est le cas particulier où les contraintes s'appliquant sur l'opération sont égales à $IC = \top$. Cette opération, de manière purement théorique, se déroule en deux étapes. La première étape consiste à rechercher les sous-ensembles de $K_1 \sqcup \dots \sqcup K_n$ qui lorsqu'ils sont retirés permettent de restaurer la cohérence, que nous appelons R-ensembles potentiels. La seconde étape, parmi ces sous-ensembles, sélectionne ceux qui sont minimaux, que nous appelons R-ensembles.

Chaque stratégie conduit à une définition différente de la fusion par R-ensembles. Pour cette raison, il serait incorrect de parler de R-ensembles de manière générale ; nous ne pouvons parler que de R-ensembles selon une certaine stratégie.

Dans un premier temps, nous donnons la définition de R-ensembles potentiels.

Définition 4.1.1 (R-ensemble potentiel). *Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances tel que $K_1 \sqcup \dots \sqcup K_n$ soit incohérent, $X \subseteq K_1 \sqcup \dots \sqcup K_n$ est un R-ensemble potentiel de E si et seulement si $(K_1 \sqcup \dots \sqcup K_n) \setminus X$ est cohérent.*

Exemple 4.1.1. *Nous reprenons l'exemple 3.2.2. Soient les bases de croyances propositionnelles suivantes, $K_1 = \{\neg d, s \vee o\}$, $K_2 = \{\neg s, d \vee o, \neg d \vee \neg o\}$ et $K_3 = \{s, d, o\}$.*

Les R-ensembles potentiels de $E = \{K_1, K_2, K_3\}$ sont au nombre de 245. Nous n'en

exhibons que quelques-uns ; les autres sont des sur-ensembles de ceux-ci.

$$\begin{aligned}
R_1 &= \{s \vee o, d \vee o, s, d, o\} & R_2 &= \{s, d\} \\
R_3 &= \{\neg d, s \vee o, s, o\} & R_4 &= \{\neg d, \neg o \vee \neg d, s\} \\
R_5 &= \{\neg s, d \vee o, d, o\} & R_6 &= \{\neg s, d\} \\
R_7 &= \{\neg d, \neg s, o\} & R_8 &= \{s \vee o, \neg s, \neg o \vee \neg d\} \\
R_9 &= \{\neg d, \neg s, d\} & R_{10} &= \{s \vee o, \neg s, d\} \\
R_{11} &= \{\neg d, s \vee o, \neg s, d\} & R_{12} &= \{\neg d, \neg s, d \vee o, d\} \\
R_{13} &= \{\neg d, \neg s, \neg o \vee \neg d, d\} & R_{14} &= \{\neg d, \neg s, d, o\} \\
R_{15} &= \{s \vee o, \neg s, d \vee o, d\} & R_{16} &= \{s \vee o, \neg s, \neg o \vee \neg d, d\} \\
R_{17} &= \{s \vee o, \neg s, d, o\} & R_{18} &= \{\neg s, d \vee o, \neg o \vee \neg d, d\} \\
R_{19} &= \{\neg s, \neg o \vee \neg d, d, o\} & R_{20} &= \{s, d, o\} \\
R_{21} &= \{\neg o \vee \neg d, s, d\} & R_{22} &= \{d \vee o, s, d\} \\
R_{23} &= \{\neg s, s, d\} & R_{24} &= \{s \vee o, s, d\} \\
R_{25} &= \{\neg d, s, d\} & R_{26} &= \{\neg o \vee \neg d, s, d, o\} \\
R_{27} &= \{d \vee o, s, d, o\} & R_{28} &= \{\neg s, s, d, o\} \\
R_{29} &= \{s \vee o, s, d, o\} & R_{30} &= \{\neg d, s, d, o\} \\
R_{31} &= \{d \vee o, \neg o \vee \neg d, s, d, o\} & R_{32} &= \{\neg s, \neg o \vee \neg d, s, d, o\} \\
R_{33} &= \{s \vee o, \neg o \vee \neg d, s, d, o\} & R_{34} &= \{\neg d, \neg o \vee \neg d, s, d, o\} \\
R_{35} &= \{\neg s, d \vee o, s, d, o\} & R_{36} &= \{\neg d, d \vee o, s, d, o\} \\
R_{37} &= \{s \vee o, \neg s, s, d, o\} & R_{38} &= \{\neg d, \neg s, s, d, o\} \\
R_{39} &= \{\neg d, s \vee o, s, d, o\} & R_{40} &= \{\neg s, d \vee o, \neg o \vee \neg d, s, d, o\} \\
R_{41} &= \{\neg d, d \vee o, \neg o \vee \neg d, s, d, o\} & R_{42} &= \{\neg d, s \vee o, \neg o \vee \neg d, s, d, o\} \\
R_{43} &= \{\neg d, s \vee o, \neg s, s, d, o\} & R_{44} &= \{\neg d, s \vee o, d \vee o, s, d, o\} \\
R_{45} &= \{\neg d, \neg s, d \vee o, s, d, o\} & R_{46} &= \{s \vee o, \neg s, d \vee o, s, d, o\} \\
R_{47} &= \{\neg d, \neg s, \neg o \vee \neg d, s, d, o\} & R_{48} &= \{s \vee o, \neg s, \neg o \vee \neg d, s, d, o\} \\
R_{49} &= \{s \vee o, d \vee o, \neg o \vee \neg d, s, d, o\} & R_{50} &= \{s \vee o, \neg s, d \vee o, \neg o \vee \neg d, s, d, o\} \\
R_{51} &= \{\neg d, \neg s, d \vee o, \neg o \vee \neg d, s, d, o\} & R_{52} &= \{\neg d, s \vee o, d \vee o, \neg o \vee \neg d, s, d, o\} \\
R_{53} &= \{\neg d, s \vee o, \neg s, \neg o \vee \neg d, s, d, o\} & R_{54} &= \{\neg d, s \vee o, \neg s, d \vee o, s, d, o\}
\end{aligned}$$

Le nombre de R-ensembles potentiels pour un profil de croyances donné est exponentiel par rapport au nombre de formules dans le profil. Ainsi, il va être nécessaire de ne sélectionner que les R-ensembles potentiels qui sont minimaux au sens de l'inclusion et qui sont minimaux au sens de la stratégie choisie. Pour matérialiser cette notion de minimalité, les stratégies sont encodées sous la forme d'un pré-ordre total sur les R-ensembles potentiels. Ainsi, une stratégie, notée P , sera codée par un pré-ordre $\leq_P$. Soient X et X' deux R-ensembles potentiels, $X \leq_P X'$ signifie que X est préféré à X' selon la stratégie P .

Formellement, un R-ensemble de E selon une stratégie P est défini comme suit :

Définition 4.1.2 (R-ensemble). *Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances tel que $K_1 \sqcup \dots \sqcup K_n$ est incohérent. Soit P une stratégie de fusion. X est un R-ensemble de E selon P si et seulement si :*

- X est un R-ensemble potentiel de E ;
- Il n'existe pas de R-ensemble potentiel $X' \subseteq K_1 \sqcup \dots \sqcup K_n$ tel que $X' \subset X$;
- Il n'existe pas de R-ensemble potentiel $X' \subseteq K_1 \sqcup \dots \sqcup K_n$ tel que $X' <_P X$.

L'ensemble des R-ensembles de E selon la stratégie P est noté $\mathcal{F}_P \mathcal{R}(E)$. Par la suite, nous utilisons comme dans 3.2.5.1 les stratégies Σ , $Card$, Max , $Gmax$.

Exemple 4.1.2. Prenons, par exemple, la stratégie la plus simple qui consiste à conserver le maximum de formules sans tenir compte de leur répartition, la stratégie *Card* (cf 4.2.1. Dans ce cas, $X \leq_{Card} X'$ si et seulement si $|X| \leq |X'|$.

Il est clair que parmi les *R*-ensembles potentiels mentionnés dans l'exemple 4.1.1 que les minimaux sont :

$$\mathcal{F}_{Card}\mathcal{R}(E) = \{\{\neg s, d\}, \{s, d\}\}$$

Tous les autres *R*-ensembles potentiels possèdent au moins 3 formules.

Grâce à cette définition, nous pouvons définir l'opération de fusion par *R*-ensembles. Ainsi, le résultat de la fusion de E selon la stratégie P est l'union, d'un ensemble de formules construit pour chaque *R*-ensemble X : l'ensemble des formules de E privé des formules de X .

Définition 4.1.3 (Fusion par *R*-ensembles). Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. Soit P une stratégie. Soit $\mathcal{F}_P\mathcal{R}(E)$ l'ensemble des *R*-ensembles de E selon P . Nous définissons l'opération de fusion de la manière suivante :

$$\Delta_P^{RSF}(E) = \bigvee_{X \in \mathcal{F}_P\mathcal{R}(E)} \{(K_1 \sqcup \dots \sqcup K_n) \setminus X\}$$

Exemple 4.1.3. En reprenant l'exemple décrit en 4.1.1, $\mathcal{F}_{Card}\mathcal{R}(E)$ possède deux *R*-ensembles et le résultat de la fusion de E selon la stratégie *Card*, $\Delta_{Card}^{RSF}(E)$, est constitué de deux ensembles de formules obtenus en retirant les *R*-ensembles aux formules du profil. À savoir :

$$\Delta_{Card}^{RSF}(E) = \{\neg d, s \vee o, d \vee o, \neg o \vee \neg d, s, o\} \vee \{\neg d, s \vee o, \neg s, d \vee o, \neg o \vee \neg d, o\}.$$

Nous rajoutons maintenant l'hypothèse qu'il existe des contraintes pesant sur l'opération $\Delta_P^{RSF}(E)$.

4.1.2 Fusion avec contraintes

Nous venons de définir l'opération de fusion pure par *R*-ensembles. C'est à dire, l'opération de fusion par *R*-ensembles lorsque qu'aucune contrainte n'est exprimée sur le résultat de l'opération. Nous allons maintenant définir l'opération de fusion contrainte par *R*-ensembles. Les contraintes seront représentées par un ensemble de formules *IC*. Dans la littérature, les contraintes sont gérées, principalement, de deux manières différentes :

- passives : le résultat de l'opération de fusion ne doit pas être contradictoire avec les contraintes.
- actives : le résultat de l'opération de fusion doit impliquer positivement les contraintes.

Dans notre cas, nous avons choisi la deuxième solution en incluant les contraintes dans la base résultant de l'opération. Les définitions précédentes sont donc redonnées dans le cas de la fusion contraintes.

Définition 4.1.4 (R-ensemble potentiel). Soient $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un ensemble de formules représentant les contraintes pesant sur l'opération tels que $K_1 \sqcup \dots \sqcup K_n \sqcup IC$ est incohérent. $X \subseteq K_1 \sqcup \dots \sqcup K_n$ est un R-ensemble potentiel de E contraint par IC si et seulement si $((K_1 \sqcup \dots \sqcup K_n) \setminus X) \sqcup IC$ est cohérent.

Exemple 4.1.4. Reprenons l'exemple 4.1.1 auquel on rajoute la condition qu'il est impossible d'apprendre en même temps le SQL et le DATALOG (pour des questions de temps dirons-nous). Autrement dit, $IC = \{\neg s \vee \neg d\}$

Les R-ensembles potentiels sont identiques à ceux de l'exemple 4.1.1 à l'exception de ceux conduisant à la déduction de $s \wedge d$. C'est à dire :

$$\begin{array}{ll} R_3 = \{\neg d, s \vee o, s, o\} & R_4 = \{\neg d, \neg o \vee \neg d, s\} \\ R_5 = \{\neg s, d \vee o, d, o\} & R_6 = \{\neg s, d\} \\ R_7 = \{\neg d, \neg s, o\} & R_8 = \{s \vee o, \neg s, \neg o \vee \neg d\} \\ R_9 = \{\neg d, \neg s, d\} & R_{10} = \{s \vee o, \neg s, d\} \\ R_{11} = \{\neg d, s \vee o, \neg s, d\} & R_{12} = \{\neg d, \neg s, d \vee o, d\} \\ R_{13} = \{\neg d, \neg s, \neg o \vee \neg d, d\} & R_{14} = \{\neg d, \neg s, d, o\} \\ R_{15} = \{s \vee o, \neg s, d \vee o, d\} & R_{16} = \{s \vee o, \neg s, \neg o \vee \neg d, d\} \\ R_{17} = \{s \vee o, \neg s, d, o\} & R_{18} = \{\neg s, d \vee o, \neg o \vee \neg d, d\} \\ R_{19} = \{\neg s, \neg o \vee \neg d, d, o\} & \end{array}$$

Certains R-ensembles potentiels ont été retirés par rapport au cas de fusion pure, ce qui simplifie quelque peu le problème.

Ainsi la définition d'un R-ensemble d'un profil de croyances E lorsqu'il est contraint par un ensemble de formules IC est la suivante :

Définition 4.1.5 (R-ensemble). Soient $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un ensemble de formules représentant les contraintes pesant sur l'opération tels que $(K_1 \sqcup \dots \sqcup K_n) \sqcup IC$ est incohérent. Soit P une stratégie de fusion. X est un R-ensemble de E contraint par IC selon la stratégie P si et seulement si :

- X est un R-ensemble potentiel de E contraint par IC ;
- Il n'existe pas de R-ensemble potentiel $X' \subseteq K_1 \sqcup \dots \sqcup K_n$ tel que $X' \subset X$;
- Il n'existe pas de R-ensemble potentiel $X' \subseteq K_1 \sqcup \dots \sqcup K_n$ tel que $X' <_P X$.

L'ensemble des R-ensembles de E contraint par IC selon la stratégie P est noté $\mathcal{F}_{P,IC}\mathcal{R}(E)$.

Exemple 4.1.5. Nous reprenons l'exemple 4.1.4. Parmi les R-ensembles potentiels restants, le R-ensemble potentiel qui possède le moins de formules est le suivant :

$$\mathcal{F}_{Card,IC}\mathcal{R}(E) = \{\{s, d\}\}$$

Le R-ensemble potentiel $\{\neg s, d\}$ ne peut pas faire partie des solutions car il n'est pas cohérent avec la contrainte IC .

Définition 4.1.6 (Fusion par R-ensembles). Soient $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un ensemble de formules représentant les contraintes. Soit P une stratégie. Soit $\mathcal{F}_{P,IC}\mathcal{R}(E)$ l'ensemble des R-ensembles de E contraint par IC selon P . Nous définissons l'opération de fusion de la manière suivante :

$$\Delta_{P,IC}^{RSF}(E) = \bigvee_{X \in \mathcal{F}_{P,IC}\mathcal{R}(E)} \{((K_1 \sqcup \dots \sqcup K_n) \setminus X) \sqcup IC\}$$

Exemple 4.1.6. *Etant donné l'opération de fusion de E , décrite dans la section e4.1.4, avec les contraintes $IC = \{\neg s \wedge \neg d\}$ selon la stratégie $Card$, $\mathcal{F}_{Card,IC} \mathcal{R}(E)$ ne possède plus qu'un seul R-ensemble le résultat de l'opération de fusion $\Delta_{Card,IC}^{RSF}(E)$ sera l'ensemble de formules obtenu en retirant le R-ensemble aux formules du profil. À savoir :*

$$\Delta_{Card,IC}^{RSF}(E) = \{\neg d, s \vee o, \neg s, d \vee o, \neg o \vee \neg d, o\}.$$

4.1.3 Réécriture des postulats de Konieczny-Pino Perez

Les postulats de Konieczny-Pino Perez ont été écrits dans le contexte de la fusion sémantique de bases de croyances. Nous proposons ici une réécriture de ces postulats adaptée au cadre syntaxique et plus particulièrement à la fusion par R-ensembles. Afin de faciliter leur compréhension, nous introduisons une nouvelle notation. Nous notons $\mathcal{F}_{P,IC}^{RSF} \mathcal{F}(E)$ l'ensemble des éléments du flock résultat de la fusion de E contraint par IC selon la stratégie P . Ainsi $\mathcal{F}_{P,IC}^{RSF} \mathcal{F}(E) = \{(E \setminus X) \mid X \in \mathcal{F}_{P,IC}^{RSF} \mathcal{R}(E)\}$. Soient E, E_1 et E_2 des profils de croyances. Soient K_1 et K_2 des bases de croyances. Soient IC, IC_1 et IC_2 des ensembles de formules représentant des contraintes. Nous proposons d'adapter les postulats de la manière suivante :

- (IC0) $IC \in Cons(\Delta_{P,IC}^{RSF}(E))$
- (IC1) Si IC est cohérent, alors $\Delta_{P,IC}^{RSF}(E)$ est cohérent.
- (IC2) Si $E \sqcup IC$ est cohérent, alors il existe un élément F unique dans $\mathcal{F}_{P,IC}^{RSF} \mathcal{F}(E)$ tel que $F = E$.
- (IC3) Si $E_1 \leftrightarrow E_2$ ¹ et $IC_1 \equiv IC_2$, alors $\Delta_{P,IC_1}^{RSF}(E_1) \equiv \Delta_{P,IC_2}^{RSF}(E_2)$.
- (IC4) Si $IC \in Cons(K_1)$ et $IC \in Cons(K_2)$, alors $\Delta_{P,IC}^{RSF}(K_1 \sqcup K_2) \sqcup K_1$ est cohérent implique $\Delta_{P,IC}^{RSF}(K_1 \sqcup K_2) \sqcup K_2$ est cohérent.
- (IC5) $\forall F_1 \in \mathcal{F}_{P,IC}^{RSF} \mathcal{F}(E_1)$ et $\forall F_2 \in \mathcal{F}_{P,IC}^{RSF} \mathcal{F}(E_2)$. Si $(F_1 \sqcup F_2) \sqcup IC$ est cohérent alors $F_1 \sqcup F_2 \in \mathcal{F}_{P,IC}^{RSF} \mathcal{F}(E_1 \sqcup E_2)$
- (IC6) Si il existe $F_1 \in \mathcal{F}_{P,IC}^{RSF} \mathcal{F}(E_1)$ et il existe $F_2 \in \mathcal{F}_{P,IC}^{RSF} \mathcal{F}(E_2)$ tels que $(F_1 \sqcup F_2) \sqcup IC$ est cohérent alors $\forall F \in \mathcal{F}_{P,IC}^{RSF} \mathcal{F}(E_1 \sqcup E_2)$ on a $F \cap E_1 \in \mathcal{F}_{P,IC}^{RSF} \mathcal{F}(E_1)$ et $F \cap E_2 \in \mathcal{F}_{P,IC}^{RSF} \mathcal{F}(E_2)$
- (IC7) $\forall F \in \mathcal{F}_{P,IC_1}^{RSF} \mathcal{F}(E)$ tel que $F \sqcup (IC_1 \wedge IC_2)$ est cohérent alors, $F \in \mathcal{F}_{P,IC_1 \wedge IC_2}^{RSF} \mathcal{F}(E)$
- (IC8) Si il existe $F_1 \in \mathcal{F}_{P,IC_1}^{RSF} \mathcal{F}(E)$ tel que $F_1 \sqcup (IC_1 \wedge IC_2)$ est cohérent, alors $\forall F \in \mathcal{F}_{P,IC_1 \wedge IC_2}^{RSF} \mathcal{F}(E)$ implique que $F \in \mathcal{F}_{P,IC_1}^{RSF} \mathcal{F}(E)$.

4.2 Différents opérateurs de fusion syntaxique

Nous allons examiner les opérateurs de fusion obtenus selon les différentes stratégies

1. L'équivalence entre deux profils de croyances est donnée dans la définition 3.2.2.

4.2.1 Somme

La stratégie *Somme* vise la réduction du nombre total de formules retirées dans le profil de croyances en ne tenant aucun compte de leur répartition dans les différentes bases.

Les stratégies sont représentées par des pré-ordres sur les R-ensembles potentiels. Nous donnons le pré-ordre pour la stratégie *Somme*.

Définition 4.2.1 (Pré-ordre $\leq_\Sigma$). Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. Soient X et X' deux R-ensembles potentiels de E . Le pré-ordre $\leq_\Sigma$ est défini de la manière suivante :

$$X \leq_\Sigma X' \text{ si et seulement si } \sum_{1 \leq i \leq n} |X \cap K_i| \leq \sum_{1 \leq i \leq n} |X' \cap K_i|$$

Grâce à ce pré-ordre, il est possible de définir le R-ensemble selon *Somme* de la manière suivante :

Définition 4.2.2 (R-ensemble selon Σ). Soient $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un ensemble de formules représentant les contraintes pesant sur l'opération tel que $K_1 \sqcup \dots \sqcup K_n \sqcup IC$ est incohérent. X est un R-ensemble de E contraint par IC selon la stratégie Σ si et seulement si :

- X est un R-ensemble potentiel de E contraint par IC ;
- Il n'existe pas de R-ensemble potentiel $X' \subseteq K_1 \sqcup \dots \sqcup K_n$ tel que $X' \subset X$.
- Il n'existe pas de R-ensemble potentiel $X' \subseteq K_1 \sqcup \dots \sqcup K_n$ tel que $\sum_{1 \leq i \leq n} |X' \cap K_i| < \sum_{1 \leq i \leq n} |X \cap K_i|$.

Cela conduit à l'opérateur de fusion suivant :

Définition 4.2.3 (Fusion selon Σ). Soient $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un ensemble de formules représentant les contraintes. Soit $\mathcal{F}_{\Sigma, IC}\mathcal{R}(E)$ l'ensemble des R-ensembles de E contraint par IC selon Σ . L'opération de fusion est définie comme suit :

$$\Delta_{\Sigma, IC}^{RSF}(E) = \bigvee_{X \in \mathcal{F}_{\Sigma, IC}\mathcal{R}(E)} \{((K_1 \sqcup \dots \sqcup K_n) \setminus X) \sqcup IC\}$$

Dans le cas de la fusion pure, où $IC = \top$, nous notons indifféremment $\Delta_{\Sigma, \top}^{RSF}(E)$ ou $\Delta_{\Sigma}^{RSF}(E)$.

Exemple 4.2.1. Nous reprenons l'exemple 4.2.3 pour illustrer cette stratégie.

Le R-ensemble selon la stratégie Σ est $R_1 = \{\neg s, d\}$. Ainsi, $\Delta_{\Sigma, \top}^{RSF}(K_1 \sqcup K_2 \sqcup K_3) = \{\neg d, s \vee o, d \vee o, \neg o \vee \neg d, s, o\}$

4.2.1.1 Postulats KP

Nous allons montrer les propriétés de cet opérateur au regard des postulats KP reformulés dans notre cadre.

Proposition 4.2.1. *Soient $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un ensemble de formules représentant les contraintes. L'opération de fusion $\Delta_{\Sigma, IC}^{RSF}(E)$ respecte les postulats suivants : **(IC0)**, **(IC1)**, **(IC2)**, **(IC5)**, **(IC6)**, **(IC7)**, **(IC8)**.*

Nous donnons maintenant la preuve de la proposition postulat par postulat. Pour simplifier l'écriture des preuves, nous définissons $\mathcal{F}_{P, IC}^{\Pi RSF, s} \mathcal{F}(E)$ comme étant : $\mathcal{F}_{P, IC}^{\Pi RSF, s} \mathcal{F}(E) = \bigcup_{X \in \mathcal{F}_{P, IC}^{\Pi RSF, s} \mathcal{R}(E)} (E \setminus X)$.

(IC0)

L'opération de fusion $\Delta_{\Sigma, IC}^{RSF}(E)$ est définie de la manière suivante :

$$\Delta_{\Sigma, IC}^{RSF}(E) = \bigvee_{X \in \mathcal{F}_{\Sigma, IC} \mathcal{R}(E)} \{((K_1 \sqcup \dots \sqcup K_n) \setminus X) \sqcup IC\}$$

Ainsi, par définition de l'opération $\Delta_{\Sigma, IC}^{RSF}(E)$, **(IC0)** est vérifié.

(IC1)

Tous les éléments de $F \in \mathcal{F}_{\Sigma, IC} \mathcal{F}(E)$ sont définis tels que $F \sqcup IC$ est cohérent. Ainsi, par définition de l'opération $\Delta_{\Sigma, IC}^{RSF}(E)$, **(IC1)** est vérifié.

(IC2)

Si $E \sqcup IC$ est cohérent, alors par définition $\mathcal{F}_{\Sigma, IC} \mathcal{R}(E) = \{\emptyset\}$:

- $E \sqcup IC$ est cohérent ;
- $\emptyset$ est minimal pour l'inclusion ;
- Et il ne peut exister de R-ensemble potentiel plus petit au sens de $\leq_{\Sigma}$.

et donc il n'y qu'un seul élément dans $\mathcal{F}_{\Sigma, IC} \mathcal{F}(E)$ qui est égal à E .

(IC3)

Nous allons donner un contre-exemple invalidant le postulat **(IC3)**.

$$\begin{aligned} K_1 &= \{a, b\} & K_2 &= \{c\} \\ K_3 &= \{\neg a\} & K_4 &= \{a \wedge b\} \\ E_1 &= K_1 \sqcup K_2 \sqcup K_3 & E_2 &= K_2 \sqcup K_3 \sqcup K_4 \end{aligned}$$

Dans ce cas, on obtient :

$$\mathcal{F}_{\Sigma, \top}^{RSF} \mathcal{F}(E_1) = \{\{a, b, c\}, \{\neg a, b, c\}\}$$

$$\mathcal{F}_{\Sigma, \top}^{RSF} \mathcal{F}(E_2) = \{\{a \wedge b, c\}, \{\neg a, c\}\}$$

On peut constater que les deux ensembles ne sont pas équivalents. Par exemple, $\neg b \wedge c$ est cohérent avec $\{\neg a, c\}$ mais avec aucun élément du flock $\mathcal{F}_{\Sigma, \top}^{RSF} \mathcal{F}(E_1)$.

Le postulat **(IC3)** n'est donc pas vérifié.

(IC4)

Nous allons donner un contre-exemple invalidant le postulat **(IC4)**.

$$IC = \top$$

$$K_1 = \{\neg b\}$$

$$K_2 = \{b \vee \neg c, b \vee \neg d, c, d\}$$

Dans ce contexte, $\Delta_{\Sigma, \top}^{RSF}(K_1 \sqcup K_2) = K_1$ et $\Delta_{\Sigma, \top}^{RSF}(K_1 \sqcup K_2) \sqcup K_1$ est cohérent alors que $\Delta_{\Sigma, \top}^{RSF}(K_1 \sqcup K_2) \sqcup K_2$ est incohérent.

Le postulat **(IC4)** n'est donc pas vérifié.

(IC5)

Soient $E_1 = \{K_1, \dots, K_m\}$ et $E_2 = \{K_{m+1}, \dots, K_n\}$ deux profils de croyances. Soient $F_1 \in \mathcal{F}_{\Sigma, \top}^{RSF} \mathcal{F}(E_1)$ et $F_2 \in \mathcal{F}_{\Sigma, \top}^{RSF} \mathcal{F}(E_2)$ tels que $F_1 \sqcup F_2$ est cohérent. Nous notons X_1 le R-ensemble correspondant à F_1 et X_2 le R-ensemble correspondant à F_2 .

Supposons que $F_1 \sqcup F_2 \notin \mathcal{F}_{\Sigma, \top}^{RSF} \mathcal{F}(E_1 \sqcup E_2)$. Il peut y avoir trois raisons à cela :

1. $F_1 \sqcup F_2$ est incohérent ;
2. $\exists X \subset (X_1 \sqcup X_2)$ tel que $(E_1 \sqcup E_2) \setminus X$ est cohérent.
3. $\exists X$ tel que $(E_1 \sqcup E_2) \setminus X$ est cohérent et tel que $\sum_{i \in \{1, \dots, n\}} |X \cap K_i| < \sum_{i \in \{1, \dots, n\}} |(X_1 \sqcup X_2) \cap K_i|$

Nous montrons qu'aucune de ces possibilités n'est compatible avec les hypothèses.

1. $F_1 \sqcup F_2$ est cohérent par hypothèse.
2. Nous avons dans ce cas, on a soit $X \cap E_1 \subset X_1$ soit $X \cap E_2 \subset X_2$:
 - $X \cap E_1 \subset X_1$. Or, $((E_1 \sqcup E_2) \setminus X) \sqcup IC$ est cohérent. Donc, $(E_1 \setminus X) \sqcup IC$ est cohérent. Donc, $(E_1 \setminus (X \cap E_1)) \sqcup IC$ est cohérent. Mais dans ce cas, X_1 ne pourrait pas être un R-ensemble de E_1 selon Σ
 - $X \cap E_2 \subset X_2$. La preuve au similaire au cas $X \cap E_1 \subset X_1$.
3. Nous avons alors dans ce cas soit $\sum_{i \in \{1, \dots, m\}} |X \cap K_i| < \sum_{i \in \{1, \dots, m\}} |X_1 \cap K_i|$ soit $\sum_{i \in \{m+1, \dots, n\}} |X \cap K_i| < \sum_{i \in \{m+1, \dots, n\}} |X_2 \cap K_i|$.
 - $\sum_{i \in \{1, \dots, m\}} |X \cap K_i| < \sum_{i \in \{1, \dots, m\}} |X_1 \cap K_i|$. Comme nous l'avons montré précédemment, $(E_1 \setminus (X \cap E_1)) \sqcup IC$ est cohérent, cette hypothèse interdirait à X_1 d'être un R-ensemble de E_1 contraint par IC selon Σ .
 - $\sum_{i \in \{m+1, \dots, n\}} |X \cap K_i| < \sum_{i \in \{m+1, \dots, n\}} |X_2 \cap K_i|$. La preuve au similaire au cas $\sum_{i \in \{1, \dots, m\}} |X \cap K_i| < \sum_{i \in \{1, \dots, m\}} |X_1 \cap K_i|$.

Le postulat **(IC5)** est donc vérifié.

(IC6)

Soient $E_1 = \{K_1, \dots, K_m\}$ et $E_2 = \{K_{m+1}, \dots, K_n\}$ deux profils de croyances. Supposons qu'il existe $F_1 \in \mathcal{F}_{\Sigma, \top}^{RSF} \mathcal{F}(E_1)$ et $F_2 \in \mathcal{F}_{\Sigma, \top}^{RSF} \mathcal{F}(E_2)$ tels que $F_1 \sqcup F_2$ est cohérent. Nous notons X_1 le R-ensemble correspondant à F_1 et X_2 le R-ensemble correspondant à F_2 .

Supposons également qu'il existe $F \in \mathcal{F}_{\Sigma, \top}^{RSF} \mathcal{F}(E_1 \sqcup E_2)$ tel que $F \cap E_1 \notin \mathcal{F}_{\Sigma, \top}^{RSF} \mathcal{F}(E_1)$ (Le cas $F \cap E_2 \notin \mathcal{F}_{\Sigma, \top}^{RSF} \mathcal{F}(E_2)$ est symétrique). Nous notons X le R-ensemble correspondant à F . Il peut y avoir trois raisons à cela :

1. $(E_1 \setminus (X \cap E_1)) \sqcup IC$ est incohérent.
2. Il existe X' tel que $(E_1 \setminus X') \sqcup IC$ est cohérent et $X' \subset (X \cap E_1)$.
3. Il existe X' tel que $(E_1 \setminus X') \sqcup IC$ est cohérent et

$$\sum_{i \in \{1, \dots, m\}} |X' \cap K_i| < \sum_{i \in \{1, \dots, m\}} |(X \cap E_1) \cap K_i|$$

Nous montrons qu'aucune de ces possibilités n'est compatible avec les hypothèses.

1. D'après les hypothèses, $(E_1 \sqcup E_2) \setminus X \sqcup IC$ est cohérent. Donc, $(E_1 \setminus X) \sqcup IC$ est cohérent. Donc, $(E_1 \setminus (X \cap E_1)) \sqcup IC$ est cohérent.
2. Découle du cas suivant.
3. X_1 et X_2 étant minimaux au sens de $\leq_\Sigma$, nous avons :

$$\sum_{i \in \{1, \dots, m\}} |X_1 \cap K_i| < \sum_{i \in \{1, \dots, m\}} |(X \cap E_1) \cap K_i|$$

et

$$\sum_{i \in \{m+1, \dots, n\}} |X_2 \cap K_i| < \sum_{i \in \{m+1, \dots, n\}} |(X \cap E_2) \cap K_i|$$

ce qui implique que

$$\sum_{i \in \{1, \dots, n\}} |(X_1 \sqcup X_2) \cap K_i| < \sum_{i \in \{1, \dots, n\}} |X \cap K_i|$$

ce qui est incompatible avec l'hypothèse selon laquelle X est un R-ensemble de E car $F_1 \sqcup F_2$ est cohérent par hypothèse.

Le postulat **(IC6)** est donc validé.

(IC7)

Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. Supposons qu'il existe $F \in \mathcal{F}_{\Sigma, IC_1}^{RSF} \mathcal{F}(E)$ tel que $F \sqcup (IC_1 \wedge IC_2)$ est cohérent. Nous notons X le R-ensemble correspondant à F .

Supposons que $F \notin \mathcal{F}_{\Sigma, IC_1 \wedge IC_2}^{RSF} \mathcal{F}(E)$. Il peut y avoir trois raisons à cela :

1. $F \sqcup (IC_1 \wedge IC_2)$ est incohérent ;

2. Il existe X' tel que $(E \setminus X') \sqcup IC_1 \wedge IC_2$ est cohérent et $X' \subset X$;
3. Il existe X' tel que $(E \setminus X') \sqcup IC_1 \wedge IC_2$ est cohérent et

$$\sum_{i \in \{1, \dots, n\}} |X' \cap K_i| < \sum_{i \in \{1, \dots, n\}} |(X \cap K_i)|$$

Nous montrons qu'aucune de ces possibilités n'est compatible avec les hypothèses.

1. $F \sqcup (IC_1 \wedge IC_2)$ est incohérent est impossible par hypothèse.
2. $(E \setminus X') \sqcup (IC_1 \wedge IC_2)$ est cohérent donc $(E \setminus X') \sqcup IC_1$ est cohérent aussi. Ce qui serait incompatible avec l'hypothèse selon laquelle X serait un R-ensemble de E contraint par IC_1 selon la stratégie Σ .
3. Démonstration similaire au cas précédent.

Le postulat **(IC7)** est donc validé.

(IC8)

Supposons qu'il existe $F_1 \in \mathcal{F}_{\Sigma, IC_1 \wedge IC_2}^{RSF}(E)$ tel que $F_1 \wedge (IC_1 \wedge IC_2)$ est cohérent. Nous notons X_1 le R-ensemble correspondant à F_1 .

Supposons qu'il existe un $F \in \mathcal{F}_{\Sigma, IC_1 \wedge IC_2}^{RSF}(E)$ tel que $F \notin \mathcal{F}_{\Sigma, IC_1}^{RSF}(E)$. Il peut y avoir trois raisons à cela :

1. $F \sqcup IC_1$ est incohérent ;
2. Il existe X' tel que $(E \setminus X') \sqcup IC_1$ est cohérent et $X' \subset X$;
3. Il existe X' tel que $(E \setminus X') \sqcup IC_1$ est cohérent et

$$\sum_{i \in \{1, \dots, n\}} |X' \cap K_i| < \sum_{i \in \{1, \dots, n\}} |(X \cap K_i)|.$$

Nous montrons qu'aucune de ces possibilités n'est compatible avec les hypothèses.

1. $F \sqcup (IC_1 \wedge IC_2)$ est cohérent, donc $F \sqcup IC_1$ est cohérent également.
2. Découle du cas suivant.
3. X_1 est minimal au sens de $\leq_\Sigma$. Ainsi, $\sum_{i \in \{1, \dots, n\}} |X_1 \cap K_i| \leq \sum_{i \in \{1, \dots, n\}} |(X' \cap K_i)|$. Or, $\sum_{i \in \{1, \dots, n\}} |X' \cap K_i| < \sum_{i \in \{1, \dots, n\}} |(X \cap K_i)|$ implique que $\sum_{i \in \{1, \dots, n\}} |X_1 \cap K_i| < \sum_{i \in \{1, \dots, n\}} |(X \cap K_i)|$. Or $(E \setminus X_1) \sqcup IC_1$ est cohérent ce qui interdirait à X d'être un R-ensemble de E contraint par IC_1 selon Σ .

Le postulat **(IC8)** est donc validé.

4.2.1.2 Équivalence avec l'opérateur Intersection

L'opération de fusion $\Delta_{\Sigma, IC}^{RSF}(E)$ est équivalente à l'opérateur Intersection $\Delta^{\cap, \Sigma}(E)$ décrit au paragraphe 3.2.5.2.

Proposition 4.2.2. *Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances IC un ensemble de formules représentant les contraintes. L'égalité suivante est vérifiée :*

$$\Delta_{\Sigma, IC}^{RSF}(E) = \Delta_{IC}^{\cap, \Sigma}(E).$$

Preuve.

Nous allons montrer la double inclusion :

$\Delta_{IC}^{\cap, \Sigma}(E) \subseteq \Delta_{\Sigma, IC}^{RSF}(E)$ Supposons qu'il existe un élément M de $\mathcal{F}_{\Sigma, IC}^{RSF}(E)$ tel que $M \in \mathcal{F}_{\Sigma, IC}^{RSF}(E)$ et $M \notin \Delta_{IC}^{\cap, \Sigma}(E)$. Si $M \in \mathcal{F}_{\Sigma, IC}^{RSF}(E)$, cela signifie que pour tout sous-ensemble M' de E tel que $M' \sqcup IC$ est cohérent, $\sum_{K_i \in E} |K_i \cap M'| \leq \sum_{K_i \in E} |K_i \cap M|$. Or, si $M \notin \Delta_{IC}^{\cap, \Sigma}(E)$, cela signifie qu'il existe au moins un sous-ensemble M' de E tel que $M' \sqcup IC$ est cohérent et tel que $\sum_{K_i \in E} |K_i \cap M| < \sum_{K_i \in E} |K_i \cap M'|$. Ce qui est incohérent avec l'hypothèse.

$\Delta_{\Sigma, IC}^{RSF}(E) \wedge IC \subseteq \Delta_{IC}^{\cap, \Sigma}(E)$ La démonstration est similaire avec $M \notin \mathcal{F}_{\Sigma, IC}^{RSF}(E)$ et $M \in \Delta_{IC}^{\cap, \Sigma}(E)$. $\square$

4.2.2 Card

La stratégie *Card* a d'abord été définie dans [BKMS91] dans le cadre des bases de croyances stratifiées. Elle est semblable à la stratégie *Somme* mais ne tient pas compte de la répétition des formules. Autrement dit, toutes les formules présentes en plusieurs exemplaires sont considérées comme si elles n'apparaissaient qu'une seule fois.

Du point de vue des motivations, cette stratégie peut se justifier en considérant l'exemple suivant issu de la théorie du choix social. Considérons le tournage d'un film, le scénario d'une scène passe mal et plusieurs options sont proposées pour résoudre le problème. Maintenant, considérons que nous disposons, d'un côté, des votes exprimés par les acteurs du film et, de l'autre côté, par les membres de l'équipe technique. Si leurs différents avis sont fusionnés en ajoutant les deux bases, nous prenons le risque de voir le vote de personnes qui seraient à la fois acteurs et membres de l'équipe de tournage exprimés deux fois.

Définition 4.2.4 (Pré-ordre $\leq_{Card}$). *Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. Soient X et X' deux R-ensembles potentiels de E . Le pré-ordre $\leq_{Card}$ est défini de la manière suivante :*

$$X \leq_{Card} X' \text{ si et seulement si } |X| \leq |X'|.$$

Définition 4.2.5 (R-ensemble selon *Card*). *Soient $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un ensemble de formules représentant les contraintes pesant sur l'opération tels que $K_1 \sqcup \dots \sqcup K_n \sqcup IC$ est incohérent. X est un R-ensemble de E contraint par IC selon la stratégie *Card* si et seulement si :*

- X est un R-ensemble potentiel de E contraint par IC ;
- Il n'existe pas de R-ensemble potentiel $X' \subseteq K_1 \sqcup \dots \sqcup K_n$ tel que $X' \subset X$.
- Il n'existe pas de R-ensemble potentiel $X' \subseteq K_1 \sqcup \dots \sqcup K_n$ tel que $|X'| < |X|$.

Cela conduit à l'opérateur de fusion suivant :

Définition 4.2.6 (Fusion selon *Card*). Soient $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un ensemble de formules représentant les contraintes. Soit $\mathcal{F}_{Card, IC}\mathcal{R}(E)$ l'ensemble des R -ensembles de E selon *Card*. L'opération de fusion est définie comme suit :

$$\Delta_{Card, IC}^{RSF}(E) = \bigvee_{X \in \mathcal{F}_{Card, IC}\mathcal{R}(E)} \{((K_1 \sqcup \dots \sqcup K_n) \setminus X) \sqcup IC\}.$$

Dans le cas de la fusion pure, où $IC = \top$, nous notons indifféremment $\Delta_{Card, \top}^{RSF}(E)$ ou $\Delta_{Card}^{RSF}(E)$.

Exemple 4.2.2. Nous reprenons l'exemple 4.2.3 pour illustrer cette stratégie.

Le R -ensemble selon la stratégie *Card* est $R_1 = \{\neg s, d\}$. Ainsi, $\Delta_{Card}^{RSF}(K_1 \sqcup K_2 \sqcup K_3) = \{\neg d, s \vee o, d \vee o, \neg o \vee \neg d, s, o\}$.

4.2.2.1 Postulats KP

Nous allons montrer les propriétés de cet opérateur au regard des postulats KP reformulés dans notre cadre.

Proposition 4.2.3. Soient $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un ensemble de formules représentant les contraintes. L'opération de fusion $\Delta_{Card, IC}^{RSF}(E)$ respecte les postulats suivants : **(IC0)**, **(IC1)**, **(IC2)**, **(IC7)** et **(IC8)**.

Nous donnons maintenant la preuve de la proposition postulat par postulat.

(IC0), (IC1), (IC2), (IC3), (IC4)

Pour les postulats **(IC0)**, **(IC1)**, **(IC2)**, les démonstrations sont similaires à celles données pour $\Delta_{\Sigma, IC}^{RSF}(E)$. De même, pour les postulats **(IC3)**, **(IC4)**, les contre-exemples sont similaires à ceux donnés pour $\Delta_{\Sigma, IC}^{RSF}(E)$.

(IC5)

Nous allons donner un contre-exemple invalidant le postulat **(IC5)**.

$$\begin{aligned} K_1 &= \{a, \neg a \vee b, b\} \\ K_2 &= \{\neg b \vee c, \neg c, \neg b \vee d, \neg d, \neg b \vee e, \neg e\} \\ K_3 &= \{f, \neg f \vee b, g, \neg g \vee b\} \\ E_1 &= K_1 \sqcup K_2 \quad E_2 = K_2 \sqcup K_3 \end{aligned}$$

Dans ce cas, nous obtenons :

$$\mathcal{F}_{Card, \top}^{RSF}\mathcal{R}(E_1) = \{\{a, b\}, \{\neg a \vee b, b\}\}$$

$$\mathcal{F}_{Card, \top}^{RSF}(\mathcal{R}(E_2)) = \{\{f, g\}, \{\neg f \vee b, g\}, \{\neg f \vee b, \neg g \vee b\}, \{f, \neg g \vee b\}\}$$

Avec ces ensembles, nous construisons $F_1 \in \mathcal{F}_{Card, \top}^{RSF}(\mathcal{F}(E_1))$ et $F_2 \in \mathcal{F}_{Card, \top}^{RSF}(\mathcal{F}(E_2))$ tels que :

$$F_1 = \{\neg a \vee b, \neg b \vee c, \neg c, \neg b \vee d, \neg d, \neg b \vee e, \neg e\}$$

$$F_2 = \{\neg b \vee c, \neg c, \neg b \vee d, \neg d, \neg b \vee e, \neg e, \neg f \vee b, \neg g \vee b\}$$

Or $F_1 \sqcup F_2$ ne fait pas partie de $\Delta_{Card, \top}^{RSF}(E_1 \sqcup E_2)$

Le postulat **(IC5)** n'est donc pas vérifié.

(IC6)

Reprenons le contre-exemple pour le postulat **(IC5)**. Dans ce cas, $\{\neg c, \neg d, \neg e\}$ fait partie des R-ensembles de $E_1 \sqcup E_2$ selon *Card* mais $\{\neg c, \neg d, \neg e\} \cap E_1 = \{\neg c, \neg d, \neg e\}$ ne fait pas partie des R-ensembles de E_1 selon *Card*.

Le postulat **(IC6)** n'est donc pas vérifié.

(IC7) et (IC8)

Pour les postulats **(IC7)**, **(IC8)**, les démonstrations sont similaires à celles données pour $\Delta_{\Sigma, IC}^{RSF}(E)$.

4.2.2.2 $\Delta_{Card, IC}^{RSF}(E)$ est indépendant de la majorité

L'opérateur *Card* ne prenant pas en compte la répétition des formules à l'intérieur des profils de croyances lors de l'opération $\Delta_{Card, IC}^{RSF}(E)$, il est normal que cette opération soit indépendante de la majorité.

Proposition 4.2.4. *Soient $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un ensemble de formules représentant les contraintes. L'opération de fusion $\Delta_{Card, IC}^{RSF}(E)$ est indépendante de la majorité.*

Preuve.

Par définition de la stratégie *Card*, $\Delta_{Card, IC}^{RSF}(K_1 \sqcup K_2^n) = \Delta_{Card, IC}^{RSF}(K_1 \sqcup K_2)$. □

4.2.3 Max

La stratégie *Max* a pour objectif de résoudre l'incohérence de façon à repartir le mieux possible les formules retirées. Formellement, l'objectif est de retirer le moins possible de formules dans la base de croyances dans laquelle le plus de formules sont retirées. Cet opérateur se trouve déjà, dans une forme légèrement différente dans [BKMS91].

Définition 4.2.7 (Pré-ordre $\leq_{Max}$). *Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. Soient X et X' deux R-ensembles potentiels de E . Le pré-ordre $\leq_{Max}$ est défini de la manière suivante :*

$$X \leq_{Max} X' \text{ si et seulement si } \max_{1 \leq i \leq n} |X \cap K_i| \leq \max_{1 \leq i \leq n} |X' \cap K_i|.$$

Définition 4.2.8 (R-ensemble selon *Max*). Soient $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un ensemble de formules représentant les contraintes pesant sur l'opération tels que $K_1 \sqcup \dots \sqcup K_n \sqcup IC$ est incohérent. X est un R-ensemble de E contraint par IC selon la stratégie *Max* si et seulement si :

- X est un R-ensemble potentiel de E contraint par IC ;
- Il n'existe pas de R-ensemble potentiel $X' \subseteq K_1 \sqcup \dots \sqcup K_n$ tel que $X' \subset X$.
- Il n'existe pas de R-ensemble potentiel $X' \subseteq K_1 \sqcup \dots \sqcup K_n$ tel que $\max_{1 \leq i \leq n} |X' \cap K_i| < \max_{1 \leq i \leq n} |X \cap K_i|$.

Cela conduit à l'opérateur de fusion suivant :

Définition 4.2.9 (Fusion selon *Max*). Soient $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un ensemble de formules représentant les contraintes. Soit $\mathcal{F}_{Max, IC} \mathcal{R}(E)$ l'ensemble des R-ensembles de E selon *Max*. L'opération de fusion est définie comme suit :

$$\Delta_{Max, IC}^{RSF}(E) = \bigvee_{X \in \mathcal{F}_{Max, IC} \mathcal{R}(E)} \{((K_1 \sqcup \dots \sqcup K_n) \setminus X) \sqcup IC\}.$$

Dans le cas de la fusion pure, où $IC = \top$, nous notons indifféremment $\Delta_{Max, \top}^{RSF}(E)$ ou $\Delta_{Max}^{RSF}(E)$.

Exemple 4.2.3. Nous reprenons ici l'exemple décrit dans l'exemple 3.2.2 que nous adaptons à la situation des opérateurs syntaxiques. L'ensemble des R-ensembles potentiels est décrit dans l'exemple 4.1.1.

Les R-ensembles potentiels minimaux au sens de la stratégie *Max* sont $R_1 = \{s \vee o, \neg s, d\}$, $R_2 = \{\neg d, \neg s, d\}$, $R_3 = \{\neg s, d\}$. Ainsi, $\Delta_{Max, IC}^{RSF}(K_1 \sqcup K_2 \sqcup K_3) = \{\neg d, d \vee o, \neg o \vee \neg d, s, o\} \vee \{s \vee o, d \vee o, \neg o \vee \neg d, s, o\} \vee \{\neg d, s \vee o, d \vee o, \neg o \vee \neg d, s, o\}$.

Nous constatons que R_1 et R_2 sont des sur-ensembles de R_3 , ce qui laisse à penser que la stratégie *Max* n'est pas optimale. La stratégie *GMax* répond à ce problème.

4.2.3.1 Postulats KP

Nous allons montrer les propriétés de cet opérateur au regard des postulats KP reformulés dans notre cadre.

Proposition 4.2.5. Soient $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un ensemble de formules représentant les croyances. L'opération de fusion $\Delta_{Max, IC}^{RSF}(E)$ respecte les postulats suivants : (IC0), (IC1), (IC2), (IC5), (IC7), (IC8).

Nous donnons maintenant la preuve de la proposition postulat par postulat.

(IC0), (IC1), (IC2), (IC3), (IC4), (IC5)

Pour les postulats (IC0), (IC1), (IC2), (IC5), les démonstrations sont similaires à celles données pour $\Delta_{\Sigma, IC}^{RSF}(E)$. De même, pour les postulats (IC3), (IC4), les contre-exemples sont similaires à ceux donnés pour $\Delta_{\Sigma, IC}^{RSF}(E)$.

(IC6)

Nous allons donner un contre-exemple invalidant le postulat **(IC6)**.

$$K_1 = \{b\}$$

$$K_2 = \{a, \neg a \vee \neg b, c, \neg b \vee \neg c\}$$

$$E_1 = \{K_1, K_2\}$$

Dans ce cas, nous obtenons :

$$\mathcal{F}_{Max, \top}^{RSF} \mathcal{R}(E_1) = \{\{b\}\}$$

$$\Delta_{Max, \top}^{RSF}(E_1) = K_2$$

$$K_3 = \{e, h\}$$

$$K_4 = \{d, \neg d \vee \neg e, f, \neg f \vee \neg e, g, \neg g \vee \neg h\}$$

$$E_2 = \{K_3, K_4\}$$

Dans ce cas, nous obtenons : $\mathcal{F}_{Max, \top}^{RSF} \mathcal{R}(E_2) = \{ \{e, h\}, \{e, g, i\}, \{e, g, \neg h \vee \neg i\}, \{e, \neg g \vee \neg h, \neg h \vee \neg i\}, \{h, d, f\}, \{h, \neg d \vee \neg e, f\}, \{h, d, \neg e \vee \neg f\}, \{h, \neg d \vee \neg e, \neg e \vee \neg f\} \}$. Ainsi, le résultat de la fusion par R-ensembles de E_1 est :

$$\Delta_{Max, \top}^{RSF}(E_1) = \bigvee_{X \in \mathcal{F}_{Max, \top}^{RSF} \mathcal{R}(E_2)} \{(E_2 \setminus X)\}$$

Nous pouvons voir que $\{a, c, e, h\} \in \mathcal{F}_{Max, \top}^{RSF} \mathcal{R}(E_1 \sqcup E_2)$

Le postulat **(IC6)** n'est donc pas vérifié.

(IC7), (IC8)

Pour les postulats **(IC7)**, **(IC8)**, les démonstrations sont similaires à celles données pour $\Delta_{\Sigma, IC}^{RSF}(E)$.

4.2.3.2 $\Delta_{Max, IC}^{RSF}(E)$ est indépendant de la majorité

La stratégie *Max* ne s'intéresse qu'à la base la plus touchée dans le profil. Dans ce contexte, la répétition d'une base à l'intérieur ne donne pas plus de poids à celle-ci et $\Delta_{Max, IC}^{RSF}(E)$ est indépendant de la majorité.

Proposition 4.2.6. *Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un ensemble de formules représentant les contraintes. L'opération de fusion $\Delta_{Max, IC}^{RSF}(E)$ est indépendante de la majorité.*

Preuve.

Considérons deux profils $E = \{K_1, \dots, K_n\}$ et $E' = \{K_1, \dots, K_n^m\}$. Considérons deux R-ensembles potentiels X et X' issus de E tel que $\max_{1 \leq i \leq n} |X \cap K_i| < \max_{1 \leq i \leq n} |X' \cap K_i|$. Cette comparaison est donc valable pour E' avec $m = 1$.

Maintenant supposons que $\max_{1 \leq i \leq n} |X \cap K_i| < \max_{1 \leq i \leq n} |X' \cap K_i|$ pour une valeur $m = k$. La comparaison est toujours valable pour $m = k + 1$.

Ainsi, quelle que soit la valeur de m , $X <_{Max} X'$. $\square$

4.2.4 GMax

L'opérateur $GMax$ est un raffinement lexicographique de l'opérateur Max , nous cherchons à repartir de la manière la plus équitable possible les formules qui sont retirées.

Définition 4.2.10. *Pour chaque R-ensemble potentiel X et chaque base de croyances K_i , soit $p_X^i = |X \cap K_i|$ alors L_X^E est définie comme étant la séquence $(p_X^1, \dots, p_X^n)$ du R-ensemble potentiel X sur le profil de croyances E ordonnée de façon décroissante.*

Nous représentons le pré-ordre lexicographique par le symbole $\leq_{lex}$.

Définition 4.2.11 (Pré-ordre $\leq_{GMax}$). *Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. Soient X et X' deux R-ensembles potentiels de E . Le pré-ordre $\leq_{GMax}$ est défini de la manière suivante :*

$$X \leq_{GMax} X' \text{ si et seulement si } L_X^E \leq_{lex} L_{X'}^E.$$

Définition 4.2.12 (R-ensemble selon $GMax$). *Soient $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un ensemble de formules représentant les contraintes pesant sur l'opération tels que $K_1 \sqcup \dots \sqcup K_n \sqcup IC$ est incohérent. X est un R-ensemble de E contraint par IC selon la stratégie $GMax$ si et seulement si :*

- X est un R-ensemble potentiel de E contraint par IC ;
- Il n'existe pas de R-ensemble potentiel $X' \subseteq K_1 \sqcup \dots \sqcup K_n$ tel que $X' \subset X$.
- Il n'existe pas de R-ensemble potentiel $X' \subseteq K_1 \sqcup \dots \sqcup K_n$ tel que $L_{X'}^E <_{lex} L_X^E$.

Cela conduit à l'opérateur de fusion suivant :

Définition 4.2.13 (Fusion selon $GMax$). *Soient $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un ensemble de formules représentant les contraintes. Soit $\mathcal{F}_{GMax, IC} \mathcal{R}(E)$ l'ensemble des R-ensembles de E selon $GMax$. L'opération de fusion est définie comme suit :*

$$\Delta_{GMax, IC}^{RSF}(E) = \bigvee_{X \in \mathcal{F}_{GMax, IC} \mathcal{R}(E)} \{((K_1 \sqcup \dots \sqcup K_n) \setminus X) \sqcup IC\}.$$

Dans le cas de la fusion pure, où $IC = \top$, nous notons indifféremment $\Delta_{GMax, \top}^{RSF}(E)$ ou $\Delta_{GMax}^{RSF}(E)$.

Exemple 4.2.4. *Nous reprenons l'exemple 4.2.3 pour illustrer cette stratégie.*

Le R-ensemble de E selon la stratégie $GMax$ est $R_1 = \{\neg s, d\}$. Ainsi, $\Delta_{GMax, \top}^{RSF}(K_1 \sqcup K_2 \sqcup K_3) = \{\neg d, s \vee o, d \vee o, \neg o \vee \neg d, s, o\}$.

4.2.4.1 Postulats KP

Nous allons montrer les propriétés de cet opérateur au regard des postulats KP reformulés dans notre cadre.

Proposition 4.2.7. *Soient $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un ensemble de formules représentant les croyances. L'opération de fusion $\Delta_{GMax, IC}^{RSF}(E)$ respecte les postulats suivants : (IC0), (IC1), (IC2), (IC5), (IC6), (IC7), (IC8).*

Pour les postulats (IC0), (IC1), (IC2), (IC5), (IC6), (IC7), (IC8), les démonstrations sont similaires à celles données pour $\Delta_{\Sigma, IC}^{RSF}(E)$. De même, pour les postulats (IC3), (IC4), les contre-exemples sont similaires à ceux donnés pour $\Delta_{\Sigma, IC}^{RSF}(E)$.

4.2.4.2 $\Delta_{GMax, IC}^{RSF}(E)$ est indépendant de la majorité

Globalement, la stratégie $GMax$ se comporte comme la stratégie Max vis-à-vis de la répétition des base de croyances. Ainsi, $\Delta_{GMax, IC}^{RSF}(E)$ est indépendant de la majorité.

Proposition 4.2.8. *Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un ensemble de formules représentant les contraintes. L'opération de fusion $\Delta_{GMax, IC}^{RSF}(E)$ est indépendante de la majorité.*

Preuve.

Considérons deux profils de croyances $E = \{K_1, \dots, K_i, \dots, K_n\}$ et $E' = \{K_1, \dots, K_i^m, \dots, K_n\}$. Considérons deux R-ensembles potentiels X et X' issus de E tels que — afin de simplifier l'écriture de la preuve — $|K_1 \cap X| > \dots > |K_i \cap X| > \dots > |K_n \cap X|$. Considérons de la même manière $|K_1 \cap X'| > \dots > |K_i \cap X'| > \dots > |K_n \cap X'|$. Considérons également que $X <_{GMax} X'$

Par hypothèse, $L_X^E <_{lex} L_{X'}^E$, ce qui signifie que $L_X^{E'} <_{lex} L_{X'}^{E'}$ lorsque $m = 1$.

Maintenant supposons que $L_X^{E'} <_{lex} L_{X'}^{E'}$ lorsque $m = k$. Nous savons alors que $(p_X^1, \dots, p_X^i, \dots, p_X^n) < (p_{X'}^1, \dots, p_{X'}^i, \dots, p_{X'}^n)$. Or, cela implique que l'inégalité $(p_X^1, \dots, p_X^i, p_X^i, \dots, p_X^n) < (p_{X'}^1, \dots, p_{X'}^i, p_{X'}^i, \dots, p_{X'}^n)$ est également vraie.

Ainsi, quelle que soit la valeur de m , $X <_{GMax} X'$. □

4.3 Contrepartie sémantique

Une question intéressante est de définir la contrepartie sémantique de l'opération de fusion syntaxique que nous avons définie. Selon le point de vue sémantique, le résultat de la fusion est un ensemble d'interprétations.

Afin de pouvoir définir la contrepartie sémantique de la fusion par R-ensembles, il est nécessaire de modifier sa définition afin qu'elle prenne en compte les conséquences des formules appartenant $\Delta_{P, IC}^{RSF}(E)$. Ainsi, pour cette section, nous considérerons la définition suivante :

Définition 4.3.1. *Soient $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un ensemble de formules représentant les contraintes. Soit P une stratégie de fusion. Soit $\mathcal{F}_{P, IC}\mathcal{R}(E)$*

l'ensemble des R -ensembles de E contraint par IC selon P . Nous définissons l'opération de fusion de la manière suivante :

$$\Delta_{P,IC}^{RSF}(E) = \bigvee_{X \in \mathcal{F}_{P,IC}\mathcal{R}(E)} Cons(\{((K_1 \sqcup \dots \sqcup K_n) \setminus X) \sqcup IC\}).$$

Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. Soit ω une interprétation. Nous notons $\mathcal{NS}_E(\omega)$ l'ensemble des formules de E qui sont falsifiées par ω . Nous définissons un pré-ordre sur les interprétations pour une stratégie P .

Définition 4.3.2. Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. Soient ω_i et ω_j deux interprétations. Soit P une stratégie de fusion. Nous définissons le pré-ordre total $\leq_{P,sem}^E$ de la manière suivante :

$$\omega_i \leq_{P,sem} \omega_j \text{ si et seulement si } \mathcal{NS}_E(\omega_i) \leq_P \mathcal{NS}_E(\omega_j).$$

Grâce à ce pré-ordre, nous pouvons définir la contrepartie sémantique de l'opération de fusion, que nous notons $\blacktriangle_{P,IC}^{RSF}(E)$. Les interprétations qui seront retenues comme résultats de l'opération de fusion seront celles qui sont minimales au sens de la stratégie P choisie. Ainsi :

Définition 4.3.3. Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un ensemble de formules représentant les contraintes. Soit P une stratégie de fusion.

$$\blacktriangle_{P,IC}^{RSF}(E) = \min(Mod(IC), \leq_{P,sem})$$

L'équivalence entre les opérateurs sémantiques et syntaxiques est donnée par la proposition suivante.

Proposition 4.3.1. Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un ensemble de formules représentant les contraintes. Soit P une stratégie de fusion.

$$Mod(\Delta_{P,IC}^{RSF}(E)) = \blacktriangle_{P,IC}^{RSF}(E)$$

Preuve.

$$\boxed{Mod(\Delta_{P,IC}^{RSF}(E)) \subseteq \blacktriangle_{P,IC}^{RSF}(E)}$$

$\forall \omega \in Mod(\Delta_{P,IC}^{RSF}(E))$, il existe au moins un R -ensemble $X \in \mathcal{F}_{P,IC}^{RSF}\mathcal{R}(E)$ tel que $\omega \in Mod((E \setminus X) \sqcup IC)$. Soit $\mathcal{NS}_E(\omega)$ l'ensemble des formules de E non satisfaites par ω . X étant un R -ensemble, nous savons que $\mathcal{NS}_E(\omega) = X$.

Si $\omega \notin \min(Mod(IC), \leq_P)$, alors il existe une interprétation $\omega' \in Mod(IC)$ tel que $\mathcal{NS}_E(\omega') <_P \mathcal{NS}_E(\omega)$. Or $\mathcal{NS}_E(\omega') <_P \mathcal{NS}_E(\omega) =_P X$ n'est pas compatible avec le fait que X soit un R -ensemble. Ainsi, $Mod(\Delta_{P,IC}^{RSF}(E)) \subseteq \blacktriangle_{P,IC}^{RSF}(E)$.

$$\boxed{\blacktriangle_{P,IC}^{RSF}(E) \subseteq \text{Mod}(\Delta_{P,IC}^{RSF}(E))}$$

$\forall \omega \in \min(\text{Mod}(IC), \leq_P)$, nous supposons que $\mathcal{NS}_E(\omega) \neq \emptyset$ car $K_1 \sqcup \dots \sqcup K_n$ est incohérent. $\mathcal{NS}_E(\omega) \subseteq E$ et $\omega \models IC$ et $\omega \models (E \setminus \mathcal{NS}_E(\omega))$. Donc $(E \setminus \mathcal{NS}_E(\omega))$ est cohérent.

$\forall \omega' \in \text{Mod}(IC)$, on a $\mathcal{NS}_E(\omega) \leq_P \mathcal{NS}_E(\omega')$ car dans le cas contraire ω n'appartient pas à $\min(\text{Mod}(IC), \leq_P)$. Ainsi, $\mathcal{NS}_E(\omega)$ est un R-ensemble de E contraint par IC selon la stratégie P et $\blacktriangle_{P,IC}^{RSF}(E) \subseteq \text{Mod}(\Delta_{P,IC}^{RSF}(E))$. $\square$

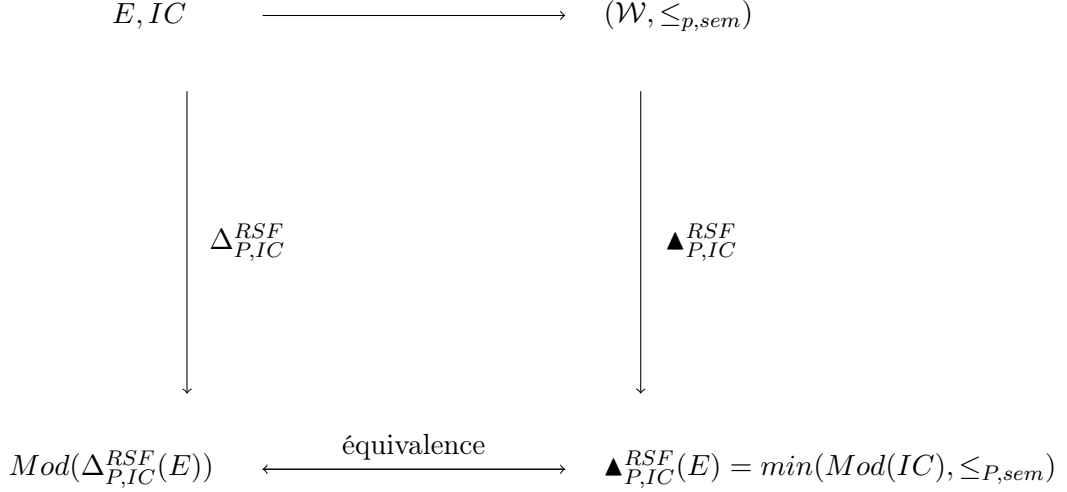


FIGURE 4.1 – Liens entre la Fusion par R-ensembles et sa contrepartie sémantique

4.3.1 Contrepartie sémantique de l'opérateur *Somme*

Nous définissons le pré-ordre pour la contrepartie sémantique de la stratégie *Somme* de la manière suivante :

Définition 4.3.4 (Pré-ordre $\leq_{\Sigma, sem}$). Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. Soient ω_i et ω_j deux interprétations. Le pré-ordre $\leq_{\Sigma, sem}$ est défini de la manière suivante :

$$\omega_i \leq_{\Sigma, sem} \omega_j \text{ si et seulement si } \sum_{1 \leq i \leq n} |\mathcal{NS}_E(\omega_i) \cap K_i| \leq \sum_{1 \leq i \leq n} |\mathcal{NS}_E(\omega_j) \cap K_i|.$$

À partir de la définition précédente, nous pouvons définir la contrepartie sémantique de l'opération de fusion selon la stratégie *Somme*.

Définition 4.3.5 (Fusion selon Σ). Soient $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un ensemble de formules représentant les contraintes. La contrepartie sémantique de

l'opération de fusion de E contraint par IC selon la stratégie *Somme* est définie de la manière suivante :

$$\blacktriangle_{\Sigma, IC}^{RSF}(E) = \min(\text{Mod}(IC), \leq_{\Sigma, \text{sem}}).$$

Nous donnons un exemple pour illustrer la stratégie *Somme*.

Exemple 4.3.1. Reprenons l'exemple 3.2.2. Soient 3 bases de croyances propositionnelles, $K_1 = \{\neg d, s \vee o\}$, $K_2 = \{\neg s, d \vee o, \neg d \vee \neg o\}$, $K_3 = \{s, d, o\}$.

Nous donnons un tableau qui récapitule les différentes interprétations et leur valeur selon la stratégie *Somme*. Les valeurs grisées correspondent aux minimaux selon $\leq_{\Sigma}$.

ω_i	K_1	K_2	K_3	Σ
$\{\neg s, \neg d, \neg o\}$	1	1	3	5
$\{\neg s, \neg d, o\}$	0	0	2	2
$\{\neg s, d, \neg o\}$	2	0	2	4
$\{\neg s, d, o\}$	1	1	1	3
$\{s, \neg d, \neg o\}$	0	2	1	3
$\{s, \neg d, o\}$	0	1	1	2
$\{s, d, \neg o\}$	1	1	1	3
$\{s, d, o\}$	1	2	0	3

Nous voyons donc que $\blacktriangle_{\Sigma, IC}^{RSF}(E) = \{\{\neg s, \neg d, o\}, \{s, \neg d, o\}\}$.

4.3.2 Contrepartie sémantique de l'opérateur *Card*

Nous définissons le pré-ordre pour la contrepartie sémantique de la stratégie *Card* de la manière suivante :

Définition 4.3.6 (Pré-ordre $\leq_{Card, \text{sem}}$). Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. Soient ω_i et ω_j deux interprétations. Le pré-ordre $\leq_{Card, \text{sem}}$ est défini de la manière suivante :

$$\omega_i \leq_{Card, \text{sem}} \omega_j \text{ si et seulement si } |\mathcal{NS}_E(\omega_i)| \leq |\mathcal{NS}_E(\omega_j)|.$$

À partir de la définition précédente, nous pouvons définir la contrepartie sémantique de l'opération de fusion selon la stratégie *Card*.

Définition 4.3.7 (Fusion selon *Card*). Soient $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un ensemble de formules représentant les contraintes. La contrepartie sémantique de l'opération de fusion de E contraint par IC selon la stratégie *Card* est définie de la manière suivante :

$$\blacktriangle_{Card, IC}^{RSF}(E) = \min(\text{Mod}(IC), \leq_{Card, \text{sem}}).$$

Nous donnons un exemple pour illustrer la stratégie *Card*.

Exemple 4.3.2. Reprenons l'exemple 4.3.1. Nous donnons un tableau qui récapitule les différentes interprétations et leur valeur selon la stratégie *Card*. Les valeurs grisées correspondent aux minimaux selon $\leq_{Card}$.

ω_i	K_1	K_2	K_3	$Card$
$\{\neg s, \neg d, \neg o\}$	1	1	3	5
$\{\neg s, \neg d, o\}$	0	0	2	2
$\{\neg s, d, \neg o\}$	2	0	2	4
$\{\neg s, d, o\}$	1	1	1	3
$\{s, \neg d, \neg o\}$	0	2	1	3
$\{s, d, \neg o\}$	0	1	1	2
$\{s, d, o\}$	1	1	1	3
$\{s, d, o\}$	1	2	0	3

Nous voyons donc que $\blacktriangle_{Card, IC}^{RSF}(E) = \{\{\neg s, \neg d, o\}, \{s, \neg d, o\}\}$.

4.3.3 Contrepartie sémantique de l'opérateur *Max*

Nous définissons le pré-ordre pour la contrepartie sémantique de la stratégie *Max* de la manière suivante :

Définition 4.3.8 (Pré-ordre $\leq_{Max, sem}$). Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. Soient ω_i et ω_j deux interprétations. Le pré-ordre $\leq_{Max, sem}$ est défini de la manière suivante :

$$\omega_i \leq_{Max, sem} \omega_j \text{ si et seulement si } \max_{1 \leq i \leq n} |\mathcal{NS}_E(\omega_i) \cap K_i| \leq \max_{1 \leq i \leq n} |\mathcal{NS}_E(\omega_j) \cap K_i|.$$

À partir de la définition précédente, nous pouvons définir la contrepartie sémantique de l'opération de fusion selon la stratégie *Max*.

Définition 4.3.9 (Fusion selon *Max*). Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un ensemble de formules représentant les contraintes. La contrepartie sémantique de l'opération de fusion selon la stratégie *Max* est définie de la manière suivante :

$$\blacktriangle_{Max, IC}^{RSF}(E) = \min(\text{Mod}(IC), \leq_{Max, sem}).$$

Nous donnons un exemple pour illustrer la stratégie *Max*.

Exemple 4.3.3. Reprenons l'exemple 4.3.1. Nous donnons un tableau qui récapitule les différentes interprétations et leur valeur selon la stratégie *Max*. Les valeurs grisées correspondent aux minimaux selon $\leq_{Max}$.

ω_i	K_1	K_2	K_3	Max
$\{\neg s, \neg d, \neg o\}$	1	1	3	3
$\{\neg s, \neg d, o\}$	0	0	2	2
$\{\neg s, d, \neg o\}$	2	0	2	2
$\{\neg s, d, o\}$	1	1	1	1
$\{s, \neg d, \neg o\}$	0	2	1	2
$\{s, \neg d, o\}$	0	1	1	1
$\{s, d, \neg o\}$	1	1	1	1
$\{s, d, o\}$	1	2	0	2

Nous voyons donc que $\blacktriangle_{Max, IC}^{RSF}(E) = \{\{\neg s, d, o\}, \{s, \neg d, \neg o\}, \{s, \neg d, o\}\}$.

4.3.4 Contrepartie sémantique de l'opérateur $GMax$

Nous rappelons que pour chaque R-ensemble potentiel X et chaque base de croyances K_i , nous définissons que $p_X^i = |X \cap K_i|$ alors L_X^E est définie comme étant la séquence $(p_X^1, \dots, p_X^n)$ du R-ensemble potentiel X sur le profil de croyances E ordonnée de façon décroissante. Nous définissons le pré-ordre pour la contrepartie sémantique de la stratégie $GMax$ de la manière suivante :

Définition 4.3.10 (Pré-ordre $\leq_{GMax, sem}$). Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. Soient ω_i et ω_j deux interprétations. Le pré-ordre $\leq_{GMax, sem}$ est défini de la manière suivante :

$$\omega_i \leq_{GMax, sem} \omega_j \text{ si et seulement si } L_{\mathcal{NS}_E(\omega_i)}^E <_{lex} L_{\mathcal{NS}_E(\omega_j)}^E.$$

À partir de la définition précédente, nous pouvons définir la contrepartie sémantique de l'opération de fusion selon la stratégie $GMax$.

Définition 4.3.11 (Fusion selon $GMax$). Soient $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un ensemble de formules représentant les contraintes. La contrepartie sémantique de l'opération de fusion de E contraint par IC selon la stratégie $GMax$ est définie de la manière suivante :

$$\blacktriangle_{GMax, IC}^{RSF}(E) = \min(\text{Mod}(IC), \leq_{GMax, sem}).$$

Nous donnons un exemple pour illustrer la stratégie $GMax$.

Exemple 4.3.4. Reprenons l'exemple 4.3.1. Nous donnons un tableau qui récapitule les différentes interprétations et leur valeur selon la stratégie $GMax$. Les valeurs grisées correspondent aux minimaux selon $\leq_{GMax}$.

ω_i	K_1	K_2	K_3	$GMax$
$\{\neg s, \neg d, \neg o\}$	1	1	3	(3, 1, 1)
$\{\neg s, \neg d, o\}$	0	0	2	(2, 0, 0)
$\{\neg s, d, \neg o\}$	2	0	2	(2, 2, 0)
$\{\neg s, d, o\}$	1	1	1	(1, 1, 1)
$\{s, \neg d, \neg o\}$	0	2	1	(2, 1, 0)
$\{s, \neg d, o\}$	0	1	1	(1, 1, 0)
$\{s, d, \neg o\}$	1	1	1	(1, 1, 1)
$\{s, d, o\}$	1	2	0	(2, 1, 0)

Nous voyons donc que $\blacktriangle_{GMax, IC}^{RSF}(E) = \{\{s, \neg d, o\}\}$.

4.4 Généralisation de la Révision par R-ensembles

La révision par R-ensembles, rappelée dans la section 3.1.3.3 et détaillée dans [BnBPW04], est définie dans le cas où les formules sont sous Forme Normale Conjonctive. L'opération de révision est définie de la manière suivante. Soit K et A deux ensembles cohérents de clauses et soit $\mathcal{R}(K \cup A)$ l'ensemble des R-ensembles de $K \cup A$: $K \circ_{RSR} A = \bigvee_{R \in \mathcal{R}(K \cup A)} Cons(\{(K \setminus R) \cup A\})$.

Afin de montrer que la fusion par R-ensembles est une généralisation de l'opération de révision par R-ensembles, nous modifions légèrement la définition de RSF :

$$\Delta_{\Sigma, IC}^{RSF}(E) = \bigvee_{X \in \mathcal{F}_{\Sigma, IC} \mathcal{R}(E)} Cons(\{((K_1 \sqcup \dots \sqcup K_n) \setminus X) \sqcup IC\}).$$

C'est à dire que l'opération de fusion par R-ensembles du singleton $E = \{K_1\}$ contraint par IC est équivalente à la révision par R-ensembles de K_1 par IC .

Proposition 4.4.1. *Soit $E = \{K_1\}$ un profil de croyances constitué d'un singleton et IC un ensemble de formules représentant les contraintes, alors :*

$$\Delta_{Card, IC}^{RSF}(E) \equiv K_1 \circ_{RSR} IC.$$

Preuve.

$$\begin{aligned} \Delta_{Card, IC}^{RSF}(E) &\equiv Cons(\{K_1 \setminus X \mid \text{tel que } (K_1 \setminus X) \sqcup IC \text{ est cohérent et} \\ &\quad \forall X' \text{ tel que } (K_1 \setminus X') \sqcup IC \text{ est cohérent} \\ &\quad \text{alors } |X| < |X'| \}) \\ &\equiv K_1 \circ_{RSR} IC. \end{aligned}$$

□

Nous donnons un exemple pour illustrer cette proposition.

Exemple 4.4.1. Soit $K = \{\neg c, a \vee c, b \vee c\}$ et $A = \{\neg a, \neg b, d \vee e, \neg d \vee \neg e\}$ deux bases de croyances. Nous obtenons pour la révision par R-ensembles :

$$K \circ_{RSR} A = Cons(\{a \vee c, b \vee c, \neg a, \neg b, d \vee e, \neg d \vee \neg e\}).$$

Et nous obtenons pour la fusion par R-ensembles :

$$\Delta_{Card,A}^{RSF}(K) = Cons(\{a \vee c, b \vee c, \neg a, \neg b, d \vee e, \neg d \vee \neg e\}).$$

Nous constatons que les résultats sont similaires.

Conclusion

Nous avons présenté dans ce chapitre l'opération de fusion par R-ensembles ainsi que l'adaptation des opérateurs classiques (*Somme*, *Card*, *Max* et *GMax*) à notre cadre. Nous avons montré les propriétés de ces opérateurs par rapport aux postulats de Konieczny et Pino-Perez. Nous avons également montré que les opérateurs de fusion par R-ensembles $\Delta_{Card,IC}^{RSF}(E)$, $\Delta_{Max,IC}^{RSF}(E)$ et $\Delta_{GMax,IC}^{RSF}(E)$ étaient indépendants de la majorité. Nous avons montré que l'opération de fusion par R-ensembles est une généralisation de la révision par R-ensembles. Nous avons également présenté une caractérisation sémantique des opérations de fusion par R-ensembles. Ce résultat a fait l'objet de publications dans [HPW07b, HPW07a, HPW08b, HPW08a].

Nous allons maintenant montrer une méthode pour mettre en œuvre la fusion par R-ensembles à l'aide de la programmation logique avec sémantique des modèles stables.

Chapitre 5

Mise en œuvre

On ne retient vraiment que ce qui s'intègre dans l'ensemble de ses connaissances. *Monique Corriveau, Compagnon du soleil*

Introduction

Nous allons maintenant proposer une méthode pour mettre en œuvre de la fusion par R-ensembles. Cette méthode est basée sur la traduction du problème de fusion en un programme logique dont les modèles stables permettent de donner effectivement les résultats de l'opération de fusion.

Cette traduction est composée de deux parties. La première partie calcule l'ensemble des R-ensembles potentiels de E contraint par IC . La seconde partie s'attèle à la sélection des R-ensembles potentiels qui sont minimaux en fonction de la stratégie choisie.

Nous montrons tout d'abord la traduction du problème de fusion du profil E contraint par IC en un programme logique $\Pi_{E,IC}$. Puis nous montrons que les modèles stables de ce programme correspondent aux R-ensembles potentiels du profil. Ensuite, nous donnons la définition de modèles stables préférés avant de montrer que les R-ensembles de E contraint par IC selon la stratégie P sont les modèles stables préférés d'un programme $\Pi_{E,IC}$ selon cette même stratégie.

5.1 Traduction en programme logique avec sémantique des modèles stables

Le programme que nous allons décrire dans cette section se base sur le constat qu'il est possible de définir les R-ensembles potentiels à partir des interprétations. Ainsi nous allons créer un programme qui génère l'ensemble des interprétations puis, à chacune de ces interprétations nous allons associer un R-ensemble potentiel — qui sera simplement l'ensemble des formules falsifiées par l'interprétation. Cette méthode fournit l'ensemble des R-ensembles qui sont minimaux au sens de l'inclusion.

Soient un profil de croyances $E = \{K_1, \dots, K_n\}$ et IC un ensemble de croyances représentant les contraintes. Nous construisons un programme $\Pi_{E,IC}$ dont les solutions correspondent aux R-ensembles potentiels E contraint par IC . Dans le cas où $IC = \top$, nous écrivons indifféremment $\Pi_{E,\top}$ ou Π_E .

Cette traduction s'effectue en plusieurs étapes :

- la première étape génère les interprétations sur lesquelles vont se baser la construction des R-ensembles potentiels ;
- la deuxième étape construit les R-ensembles potentiels basés sur ces interprétations ;
- la troisième étape permet de construire les R-ensembles restants à construire ;
- la quatrième étape permet de représenter les contraintes IC et d'exclure les R-ensembles potentiels en contradiction avec elles.

Le programme $\Pi_{E,IC}$ que nous construisons permet d'associer à chaque modèle stable un R-ensemble potentiel.

Nous représentons les atomes de E dans $\Pi_{E,IC}$. Un atome a dans le profil de croyances E sera représenté par un atome a dans $\Pi_{E,IC}$. L'ensemble de tous les littéraux positifs de $\Pi_{E,IC}$ est noté V^+ .

À partir de chaque atome a de V^+ , nous construisons un atome a' représentant la négation de a . L'ensemble de tous les littéraux négatifs de $\Pi_{E,IC}$ est noté V^- .

Nous présentons maintenant les quatre parties constitutives de notre traduction.

5.1.1 Première étape : génération des interprétations

Dans la première étape, nous introduisons des règles permettant de construire une correspondance entre les modèles stables de $\Pi_{E,IC}$ et les interprétations de V^+ . Pour chaque atome, $a \in V^+$ nous introduisons deux règles :

- $a \leftarrow \text{not } a'$.
- $a' \leftarrow \text{not } a$.

Exemple 5.1.1. Soit $E = \{K_1, K_2, K_3\}$ le profil de croyances défini dans l'exemple 4.1.4. Les interprétations de Π_E sont générées par les règles suivantes :

$$\begin{array}{lll} s \leftarrow \text{not } s' & s' \leftarrow \text{not } s & d \leftarrow \text{not } d' \\ d' \leftarrow \text{not } d & o \leftarrow \text{not } o' & o' \leftarrow \text{not } o \end{array}$$

5.1.2 Deuxième étape : détermination des R-ensembles potentiels correspondants

Dans la seconde étape, nous supprimons les formules qui ne sont pas cohérentes avec l'interprétation courante. Formellement, nous excluons les modèles stables S qui correspondent aux interprétations qui ne sont pas des modèles de $(K_1 \sqcup \dots \sqcup K_n) \setminus C_i$ avec $C_i = \{f \mid r_f \in S\}$.

L'ensemble de tous les atomes représentant les formules est défini par $R^+ = \{r_f^i \mid f \in K_i\}$ et $F_O(r_f^i)$ représente les formules de K_i correspondant à r_f^i dans $\Pi_{E,IC}$, autrement dit $\forall r_f^i \in R^+, F_O(r_f^i) = f$. Ces atomes représentant les formules sont appelés atomes de

règle. L'absence d'une formule dans le R-ensemble potentiel est représentée par l'atome r_f^i qui est, en quelque sorte, une négation de r_f^i . L'ensemble des atomes de règle dans un R-ensemble potentiel R est noté R^- .

5.1.2.1 Le cas clausal

Dans ce cas, les formules sont mises sous forme CNF.

Pour chaque clause c de K_j telle que $c = \neg b_0 \vee \dots \vee \neg b_n \vee b_{n+1} \vee \dots \vee b_m$, nous introduisons la règle suivante :

$$- r_c^j \leftarrow b_0, \dots, b_n, b'_{n+1}, \dots, b'_m.$$

Exemple 5.1.2. Soit $E = \{K_1, K_2, K_3\}$ le profil de croyances défini dans l'exemple 4.1.4. Les atomes suivants seront utilisés dans le programme logique Π_E en fonction de leur utilité :

- $V^+ = \{s, d, o\}$;
- $V^- = \{s', d', o'\}$;
- $R^+ = \{r_{\neg d}^1, r_{s \vee o}^1, r_{\neg s}^2, r_{d \vee o}^2, r_{\neg d \vee \neg o}^2, r_s^3, r_d^3, r_o^3\}$.
- $R^- = \{r_{\neg d}^1, r_{s \vee o}^1, r_{\neg s}^2, r_{d \vee o}^2, r_{\neg d \vee \neg o}^2, r_s^3, r_d^3, r_o^3\}$.

Les atomes de règles de chaque modèle stable de Π_E sont générés à l'aide des règles suivantes :

$$\begin{array}{lll} r_{\neg d}^1 \leftarrow d & r_{s \vee o}^1 \leftarrow s', o' & r_{\neg s}^2 \leftarrow s \\ r_{d \vee o}^2 \leftarrow d', o' & r_{\neg d \vee \neg o}^2 \leftarrow d, o & r_s^3 \leftarrow s' \\ r_d^3 \leftarrow d' & r_o^3 \leftarrow o' & \end{array}$$

Les règles du programme Π_E décrites dans les exemples 5.1.1 et 5.1.2 génèrent les modèles stables suivants (entre parenthèses les R-ensembles potentiels correspondants) :

- $S_1 = \{s', d', o', r_{s \vee o}^1, r_{d \vee o}^2, r_s^3, r_d^3, r_o^3\}$, ($R_1 = \{s \vee o, d \vee o, s, d, o\}$) ;
- $S_2 = \{s, d', o', r_{\neg s}^2, r_{d \vee o}^2, r_d^3, r_o^3\}$, ($R_2 = \{\neg s, d \vee o, d, o\}$) ;
- $S_3 = \{s', d', o, r_d^3, r_s^3\}$, ($R_3 = \{d, s\}$) ;
- $S_4 = \{s, d', o, r_{\neg s}^2, r_d^3\}$, ($R_4 = \{\neg s, d\}$) ;
- $S_5 = \{s', d, o', r_{\neg d}^1, r_{s \vee o}^1, r_s^3, r_o^3\}$, ($R_5 = \{\neg d, s, o, s \vee o\}$) ;
- $S_6 = \{s, d, o', r_{\neg d}^1, r_{\neg s}^2, r_o^3\}$, ($R_6 = \{\neg d, o, \neg s\}$) ;
- $S_7 = \{s', d, o, r_{\neg d}^1, r_{\neg d \vee \neg o}^2, r_s^3\}$, ($R_7 = \{\neg d, s, \neg d \vee \neg o\}$) ;
- $S_8 = \{s, d, o, r_{\neg d}^1, r_{\neg s}^2, r_{\neg d \vee \neg o}^2\}$, ($R_8 = \{\neg d, \neg s, \neg d \vee \neg o\}$) ;

Nous pouvons constater que ce programme calcule l'ensemble minimal de clauses à retirer pour chaque interprétation. Les R-ensembles potentiels non-minimaux (dont la plupart sont inutilement gros) ne sont pas calculés. Les règles décrites dans 5.1.3 permettent de les calculer.

5.1.2.2 Cas des bases de croyances composées de formules quelconques

Soient $f, f^1, \dots, f^n$ des formules de K_i . L'ensemble des littéraux positifs (resp. négatifs) de Π_E est noté V^+ (resp. V^-). Soit $f \in K_i$ une formule, nous notons r_f^i l'atome de règle qui correspond à f . L'ensemble des tous les atomes représentant les formules, appelés atomes de règle, est défini par $R^+ = \{r_f^i \mid f \in K_i\}$. La traduction du problème

nécessite la création d'atomes intermédiaires représentant les sous-formules de f . Nous notons ρ_f^j l'atome intermédiaire représentant f^j qui est une sous-formule de $f \in K_i$. De plus, nous notons $F_O(r_f^i)$ la formule de K_i correspondant à r_f^i dans Π_E . En d'autres termes $\forall r_f^i \in R^+, F_O(r_f^i) = f$. La création de Π_E se déroule en plusieurs étapes. Les étapes sont similaires à celles décrites dans 5.1, à l'exception de 5.1.2 qui doit représenter la complexité des formules bien formées quelconques.

Cette étape introduit les atomes de règle qui permettent de déterminer les R-ensembles potentiels associés aux interprétations. La présence de l'atome de règle r_f^i dans le modèle stable S signifie que f est dans le R-ensemble potentiel correspondant. Quelle que soit la formule f , les règles suivantes sont introduites selon la syntaxe de f :

1. Si $f \equiv a$, la règle correspondante est $r_f^i \leftarrow \text{not } a$;
2. Si $f \equiv \neg f^1$, la règle correspondante est $r_f^i \leftarrow \text{not } \rho_{f^1}$;
3. Si $f \equiv f^1 \vee \dots \vee f^m$, la règle correspondante est $r_f^i \leftarrow \rho_{f^1}, \dots, \rho_{f^m}$;
4. Si $f \equiv f^1 \wedge \dots \wedge f^m$, il est alors nécessaire de rajouter plusieurs règles au programme. Les règles correspondantes sont $\forall 1 \leq j \leq m, r_f^i \leftarrow \rho_{f^j}$.

Exemple 5.1.3. Soit $E = \{K_1, K_2, K_3\}$ le profil de croyances constitué des bases $K_1 = \{a \vee b, b\}$, $K_2 = \{a \leftrightarrow b, b\}$ et $K_3 = \{\neg a \wedge \neg b, \neg a \vee \neg b\}$. Le programme logique correspondant est le suivant :

$$\Pi_E = \left\{ \begin{array}{ll} a \leftarrow \text{not } a'. & a' \leftarrow \text{not } a. \\ b \leftarrow \text{not } b'. & b' \leftarrow \text{not } b. \\ r_{a \vee b}^1 \leftarrow \text{not } a, \text{not } b. & r_b^1 \leftarrow \text{not } b. \\ r_{a \leftrightarrow b}^2 \leftarrow \rho_1^2. & r_{a \leftrightarrow b}^2 \leftarrow \rho_2^2. \\ \rho_1^2 \leftarrow a, \text{not } b. & \rho_2^2 \leftarrow \text{not } a, b. \\ r_b^2 \leftarrow \text{not } b. & r_{\neg a \wedge \neg b}^3 \leftarrow \rho_1^3. \\ r_{\neg a \wedge \neg b}^3 \leftarrow \rho_2^3. & \rho_1^3 \leftarrow a. \\ \rho_2^3 \leftarrow b. & r_{\neg a \vee \neg b}^3 \leftarrow a, b. \end{array} \right\}$$

Le programme Π_E possède les modèles stables suivants (entre parenthèses, le R-ensemble potentiel correspondant) :

- $S_1 = \{a, b, r_{\neg a \vee \neg b}^3, \rho_2^3, \rho_1^3, r_{\neg a \wedge \neg b}^3\}$, ($R_1 = \{\neg a \vee \neg b, \neg a \wedge \neg b\}$) ;
- $S_2 = \{a', b, \rho_2^3, r_{\neg a \wedge \neg b}^3, \rho_2^2, r_{a \leftrightarrow b}^2\}$, ($R_2 = \{\neg a \wedge \neg b, a \leftrightarrow b\}$) ;
- $S_3 = \{a, b', \rho_1^3, r_{\neg a \wedge \neg b}^3, r_b^2, \rho_1^2, r_{a \leftrightarrow b}^2, r_b^1\}$, ($R_3 = \{\neg a \wedge \neg b, b, a \leftrightarrow b, b\}$) ;
- $S_4 = \{a', b', r_b^2, r_{a \vee b}^1, r_b^1\}$, ($R_4 = \{b, a \vee b, b\}$) ;

5.1.3 Troisième étape : génération des R-ensembles restants

Les deux précédentes étapes dessinent un programme qui génère l'ensemble des R-ensembles potentiels minimaux au sens de la cardinalité. Mais ceux qui ne sont pas minimaux ne sont pas tous générés. Cependant s'il est nécessaire de les générer le programme suivant le permet.

Dans la dernière étape, nous générons les modèles stables potentiels restants. Pour chaque atome de règle $r_i^j \in R^+$, nous introduisons deux règles : $r_i^j \leftarrow \text{not } r_i'^j$ et $r_i'^j \leftarrow \text{not } r_i^j$ où $r_i'^j \in R^-$ est l'atome négatif correspondant à r_i^j .

Exemple 5.1.4. *Nous continuons l'exemple 4.1.4. Les règles suivantes permettent de générer les R-ensembles potentiels restants :*

$$\begin{array}{ll}
 r_{\neg d}^1 \leftarrow \text{not } r_{\neg d}^1 & r_{\neg d}^1 \leftarrow \text{not } r_{\neg d}'^1 \\
 r_{s \vee o}^1 \leftarrow \text{not } r_{s \vee o}^1 & r_{s \vee o}^1 \leftarrow \text{not } r_{s \vee o}'^1 \\
 r_{\neg s}^2 \leftarrow \text{not } r_{\neg s}^2 & r_{\neg s}^2 \leftarrow \text{not } r_{\neg s}'^2 \\
 r_{d \vee o}^2 \leftarrow \text{not } r_{d \vee o}^2 & r_{d \vee o}^2 \leftarrow \text{not } r_{d \vee o}'^2 \\
 r_{\neg d \vee \neg o}^2 \leftarrow \text{not } r_{\neg d \vee \neg o}^2 & r_{\neg d \vee \neg o}^2 \leftarrow \text{not } r_{\neg d \vee \neg o}'^2 \\
 r_s^3 \leftarrow \text{not } r_s^3 & r_s^3 \leftarrow \text{not } r_s'^3 \\
 r_d^3 \leftarrow \text{not } r_d^3 & r_d^3 \leftarrow \text{not } r_d'^3 \\
 r_o^3 \leftarrow \text{not } r_o^3 & r_o^3 \leftarrow \text{not } r_o'^3
 \end{array}$$

L'union des règles présentées dans les parties précédentes et illustrées dans les exemples 5.1.1 et 5.1.2 et de celle-ci génère l'ensemble des R-ensembles potentiels. Néanmoins, les R-ensembles potentiels supplémentaires générés grâce à ces règles ne sont pas minimaux aux sens de l'inclusion. Nous pouvons donc considérer qu'ils ne constituent pas des solutions souhaitables.

5.1.4 Quatrième étape : Représentation des contraintes IC

Une contrainte est une formule que tout résultat de l'opération se doit de respecter. En d'autres termes, une formule qui ne peut pas être retirée. Pour cette raison, les contraintes sont représentées par des formules de règles spécifiques qui ne seront pas prises en compte par la stratégie lors de la sélection des R-ensembles.

Ainsi pour une contrainte f , représentée par la règle r_f , nous introduisons les règles permettant de déduire l'atome de règle correspondant, que nous nommons r_f^0 (tel que décrits dans 5.1.2) ainsi que la règle :

$$\leftarrow r_f^0$$

5.2 Équivalence entre modèle stable et R-ensemble potentiel

Nous donnons les propriétés dans le cas où les formules appartenant aux bases de croyances sont quelconques. Bien évidemment, le cas clausal en est un cas particulier. Nous rappelons que Π^S représente la réduction de Gelfond-Lifschitz [GL88] du programme Π . Nous notons $F_O(r)$ la formule correspondant à l'atome de règle r . Une première proposition montre qu'un atome de règle r n'est déduit, pour une interprétation donnée S , que si la formule $F_O(r)$ est effectivement invalidée par l'interprétation S .

Pour simplifier les preuves, nous nous limitons au cas où $IC = \top$ sans perte de généralité.

Proposition 5.2.1. *Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. Soit $S \subseteq V$ un ensemble d'atomes. Soit r un atome de règle ou un atome intermédiaire. $r \in CN(\Pi_E^S)$ si et seulement si $I_S \not\models F_O(r)$.*

Preuve.

Dans ce qui suit, I_S peut être construit à partir de S en considérant que $a \in S$ si et seulement si $a \in I_S$ et $a' \in S$ si et seulement si $\neg a \in I_S$. Les r_i représentent les sous-formules de r et a est un atome.

Dans le cas où $F_O(r)$ se réduit à un simple atome ($F_O(r) = a$), la seule règle permettant sa traduction est $r \leftarrow \text{not } a$. Ainsi, nous retirons r si et seulement si a est faux.

Dans le cas où la formule $F_O(r)$ est une simple négation : $F_O(r) = \neg r_1$. La seule règle permettant la déduction de r est $r \leftarrow \text{not } r_1$. Ainsi, r est vrai si et seulement si r_1 ne l'est pas.

Dans le cas où la formule $F_O(r)$ est une conjonction : $F_O(r) = r_1 \wedge \dots \wedge r_k$. Les seules règles permettant la déduction de r sont $r \leftarrow r_1 \dots r \leftarrow r_k$. Ainsi, r est vrai si et seulement si au moins un des atomes $r_1, \dots, r_k$ est vrai.

Dans le cas où la formule $F_O(r)$ est une disjonction : $F_O(r) = r_1 \vee \dots \vee r_k$. La seule règle permettant la déduction de r est $r \leftarrow r_1, \dots, r_k$. Ainsi, r est vrai si et seulement si tous les atomes $r_1, \dots, r_k$ sont vrais.

Dans le cas où la formule $F_O(r)$ est une implication : $F_O(r) = r_1 \rightarrow r_2$. Les seules règles permettant la déduction de r sont $r \leftarrow \text{not } r_1$ et $r \leftarrow r_2$. Ainsi, r est vrai si et seulement si r_2 est vrai ou si r_1 est faux.

Dans le cas où la formule $F_O(r)$ est une équivalence : $F_O(r) = r_1 \leftrightarrow r_2$. Les seules règles permettant la déduction de r sont $r \leftarrow r_1, r_2$ et $r \leftarrow \text{not } r_1, \text{not } r_2$. Ainsi, r est vrai si et seulement si r_1 et r_2 sont tous les deux vrais ou si ils sont tous les deux faux.

En conséquence, si $I_S \models F_O(r)$ alors $r \notin CN(\Pi_E^S)$ et si $r \in CN(\Pi_E^S)$ alors $I_S \not\models F_O(r)$. $\square$

Soit un S un modèle stable de Π_E , nous construisons l'interprétation I_S de la manière suivante : si $a \in S$, alors $a \in I_S$ et si $a' \in S$ alors $\neg a \in I_S$. La proposition suivante montre que pour tout modèle stable S de Π_E , une interprétation I_S satisfait l'ensemble de formules $(K_1 \sqcup \dots \sqcup K_n) \setminus F_O(R^+ \cap S)$

Proposition 5.2.2. *Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. Soit $S \subseteq V$ un ensemble d'atomes. S est un modèle stable de Π_E si et seulement si I_S est une interprétation de V^+ qui satisfait $(K_1 \sqcup \dots \sqcup K_n) \setminus F_O(R^+ \cap S)$.*

Preuve.

$\Rightarrow$ Si S est un modèle stable de Π_E , alors S correspond à l'interprétation I_S qui satisfait $(K_1 \sqcup \dots \sqcup K_n) \setminus F_O(S \cap R^+)$.

Les atomes de règles n'interviennent pas dans cette partie de la preuve.

Nous montrons tout d'abord que I_S est une interprétation de V^+ .

Exclusion mutuelle. Il est impossible que :

- $a \notin S$ et $a' \notin S$ car si $a' \notin S$, alors la règle $a \leftarrow \text{not } a'$ s'applique et $a \in S$ ce qui est contradictoire avec l'hypothèse.

- $a \in S$ et $a' \in S$ car la seule règle qui permette de déduire a est $a \leftarrow \text{not } a'$. Ceci implique que $a' \notin S$ ce qui est contradictoire avec l'hypothèse.

Il s'ensuit donc que I_S est une interprétation totale de V^+ .

Nous montrons ensuite que I_S satisfait $(K_1 \sqcup \dots \sqcup K_n) \setminus F_O(S \cap R^+)$. Utilisons une *reductio ad absurdum*. Supposons que I_S satisfait pas $(K_1 \sqcup \dots \sqcup K_n) \setminus F_O(S \cap R^+)$. Il s'ensuit que $\exists f$ une formule telle que $f \in (K_1 \sqcup \dots \sqcup K_n) \setminus F_O(S \cap R^+)$ et $I_S \not\models f$. Grâce à la proposition 5.2.1, nous savons que $I_S \not\models f$ implique $r_f \in CN(\Pi_E^S)$ et donc que $f \in F_O(S \cap R^+)$ ce qui est contradictoire avec l'hypothèse.

$\square$ Si I_S est une interprétation de V^+ qui satisfait $(K_1 \sqcup \dots \sqcup K_n) \setminus F_O(S \cap R^+)$, alors S est un modèle de Π_E^S .

S se construit de la manière suivante à partir de I_S . Si $a \in I_S$ alors $a \in S$ et si $\neg a \in I_S$ alors $a' \in S$. Ainsi, nous ajoutons tout atome de règle tel que $S \not\models F_O(n)$.

$S \subseteq CN(\Pi_E^S)$. Soit $a \in S$:

- $a \in V^+$. I_S est une interprétation, ainsi $\neg a \notin V^+$ et $a' \notin S$. À cause de la règle $a \leftarrow \text{not } a'$, nous avons $a \in CN(\Pi_E^S)$.
- $a \in V^-$. Avec la même démonstration.

Un autre point est que, grâce à la Proposition 5.2.1, nous savons que $S \not\models n$, alors $n \in CN(\Pi_E^S)$. Ainsi si $n \in S$ alors $n \in CN(\Pi_E^S)$.

$CN(\Pi_E^S) \subseteq S$. Supposons qu'il existe $a \in CN(\Pi_E^S) \cap (V \setminus R^+)$ tel que $a \notin S$. Il s'ensuit que :

- $a \in V^+$, $a \notin S$ implique $a' \in S$ (par la règle $a' \leftarrow \text{not } a$). Nous avons montré que $S \subseteq CN(\Pi_E^S)$ qui implique que $a' \in CN(\Pi_E^S)$. Nous savons également que la seule règle permettant de déduire a est $a \leftarrow \text{not } a'$ ce qui implique que $a' \notin CN(\Pi_E^S)$ et mène à une contradiction
- $a \in V^-$. Avec la même démonstration.

Supposons qu'il existe un atome de règle $n \in CN(\Pi_E^S) \cap R^+$ et $n \notin S$. Grâce à la Proposition 5.2.1, nous savons que $n \in CN(\Pi_E^S) \cap R^+$ implique $I_S \models F_O(n)$. Mais si $n \notin S$ cela signifie que $I_S \models F_O(n)$ et cela mène à une contradiction.

Nous avons montré que $S \subseteq CN(\Pi_E^S)$ et que $CN(\Pi_E^S) \subseteq S$, donc $CN(\Pi_E^S) = S$. $\square$

Proposition 5.2.3. Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. Soit $S \subseteq V$ un ensemble d'atomes. Soit $Y \subseteq (K_1 \sqcup \dots \sqcup K_n)$ tel que $(K_1 \sqcup \dots \sqcup K_n) \setminus Y$ est cohérent. Alors, il existe un ensemble d'atomes S tel que S est un modèle stable de Π_E et $F_O(S \cap R^+) \subseteq Y$.

Preuve.

Si $(K_1 \sqcup \dots \sqcup K_n) \setminus Y$ est cohérent, il existe une interprétation I_S qui la satisfasse (i.e. $\forall f \in (K_1 \sqcup \dots \sqcup K_n) \setminus Y, I_S \models f$). À partir de I_S nous pouvons construire un ensemble d'atomes $S = \{a \mid a \in I_S\} \cup \{a' \mid \neg a \in I_S\} \cup \{r_f \mid I_S \not\models c\}$ et S est un modèle stable. Nous savons que toutes les formules non satisfaites par I_Y sont dans Y (s'il en existe) mais il peut exister des clauses satisfaites dans Y . C'est pour cela que tous les R-ensembles potentiels ne conduisent pas à un modèle stable, mais nous pouvons exhiber un modèle stable qui est un sous-ensemble de Y . $\square$

Corollaire 5.2.1. Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. Soit $S \subseteq V$ un ensemble d'atomes. Soit $Y \subseteq (K_1 \sqcup \dots \sqcup K_n)$ tel que $(K_1 \sqcup \dots \sqcup K_n) \setminus Y$ est cohérent et

il n'existe pas de $Y' \subseteq Y$ tel que $(K_1 \sqcup \dots \sqcup K_n) \setminus Y'$ est cohérent. Alors, il existe S un ensemble d'atomes tel que S est un modèle stable de Π_E et $F_O(S \cap R^+) = Y$.

À partir de Y , nous construisons une interprétation $I_Y = \{a \mid a \in Y\} \cup \{\neg a \mid a' \in Y\}$. À partir de la proposition 5.2.1, nous savons que tous les r_f qui n'apparaissent pas dans Y ont leur formule correspondante $F_O(r_f)$ satisfaite par I_Y .

5.3 Définition des modèles stables préférés

Les modèles stables du programme $\Pi_{E,IC}$ correspondent aux R-ensembles potentiels de E contraint par IC . À partir de ces modèles stables, il nous faudra calculer ceux qui sont préférés au sens de la stratégie choisie. Nous les appelons des modèles stables préférés.

Définition 5.3.1 (Modèles stables préférés). Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un ensemble de formules représentant les contraintes. Soient X et X' deux ensembles d'atomes de $\Pi_{E,IC}$. X est un modèle stable préféré de $\Pi_{E,IC}$ selon la stratégie P si seulement si :

- X est un modèle stable de $\Pi_{E,IC}$;
- Il n'existe pas de modèle stable X' de $\Pi_{E,IC}$ tel que $X' \subset X$.
- Il n'existe pas de modèle stable X' de $\Pi_{E,IC}$ tel que $X' <_P X$.

Ainsi pour chaque stratégie, nous donnons la définition du pré-ordre sur les modèles stables.

Définition 5.3.2 (Pré-ordre $\leq_\Sigma$). Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. Soient X et X' deux modèles stables de Π_E . Le pré-ordre $\leq_\Sigma$ est défini de la manière suivante :

$$X \leq_\Sigma X' \text{ si et seulement si } \sum_{1 \leq i \leq n} |(X \cap R^+) \cap K_i| \leq \sum_{1 \leq i \leq n} |(X' \cap R^+) \cap K_i|.$$

Définition 5.3.3 (Pré-ordre $\leq_{Card}$). Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. Soient X et X' deux R-ensembles potentiels de E . Le pré-ordre $\leq_{Card}$ est défini de la manière suivante :

$$X \leq_{Card} X' \text{ si et seulement si } |(X \cap R^+)| \leq |(X' \cap R^+)|.$$

Définition 5.3.4 (Pré-ordre $\leq_{Max}$). Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. Soient X et X' deux R-ensembles potentiels de E . Le pré-ordre $\leq_{Max}$ est défini de la manière suivante :

$$X \leq_{Max} X' \text{ si et seulement si } \max_{1 \leq i \leq n} |(X \cap R^+) \cap K_i| \leq \max_{1 \leq i \leq n} |(X' \cap R^+) \cap K_i|.$$

Définition 5.3.5 (Pré-ordre $\leq_{GM_{ax}}$). Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. Soient X et X' deux R-ensembles potentiels de E . Le pré-ordre $\leq_{GM_{ax}}$ est défini de la manière suivante :

$$X \leq_{GM_{ax}} X' \text{ si et seulement si } L_{(X \cap R^+)}^E \leq_{lex} L_{(X' \cap R^+)}^E{}^1.$$

1. La définition de L_X^E est donnée dans 4.2.10

Nous illustrons maintenant cette notion de préférences entre modèles stables à l'aide d'un exemple.

Exemple 5.3.1. Nous reprenons l'exemple 4.1.4 dont la traduction Π_E possède les modèles stables donnés dans l'exemple 5.1.2 :

- $S_1 = \{s', d', o', r_{s \vee o}^1, r_{d \vee o}^2, r_s^3, r_d^3, r_o^3\}$, $(R_1 = \{s \vee o, d \vee o, s, d, o\})$;
- $S_2 = \{s, d', o', r_{\neg s}^2, r_{d \vee o}^2, r_d^3, r_o^3\}$, $(R_2 = \{\neg s, d \vee o, d, o\})$;
- $S_3 = \{s', d', o, r_d^3, r_s^3\}$, $(R_3 = \{d, s\})$;
- $S_4 = \{s, d', o, r_{\neg s}^2, r_d^3\}$, $(R_4 = \{\neg s, d\})$;
- $S_5 = \{s', d, o', r_{\neg d}^1, r_{s \vee o}^1, r_s^3, r_o^3\}$, $(R_5 = \{\neg d, s, o, s \vee o\})$;
- $S_6 = \{s, d, o', r_{\neg d}^1, r_{\neg s}^2, r_o^3\}$, $(R_6 = \{\neg d, o, \neg s\})$;
- $S_7 = \{s', d, o, r_{\neg d}^1, r_{\neg d \vee \neg o}^2, r_s^3\}$, $(R_7 = \{\neg d, s, \neg d \vee \neg o\})$;
- $S_8 = \{s, d, o, r_{\neg d}^1, r_{\neg s}^2, r_{\neg d \vee \neg o}^2\}$, $(R_8 = \{\neg d, \neg s, \neg d \vee \neg o\})$;

Nous donnons un tableau récapitulant les modèles stables de Π_E minimaux au sens de l'inclusion ainsi que leur valeur en fonction des différentes stratégies. Les colonnes K_1, K_2, K_3 donnent les valeurs de $|(S_i \cap R^+) \cap K_i|$.

S_i	K_1	K_2	K_3	Σ	$Card$	Max	$GMax$
S_1	1	1	3	5	5	3	(3, 1, 1)
S_2	0	2	2	4	4	2	(2, 2, 0)
S_3	0	0	2	2	2	2	(2, 0, 0)
S_4	0	1	1	2	2	1	(1, 1, 0)
S_5	2	0	2	4	4	2	(2, 2, 0)
S_6	1	1	1	3	3	1	(1, 1, 1)
S_7	1	1	1	3	3	1	(1, 1, 1)
S_8	1	2	0	3	3	2	(2, 1, 0)

Nous pouvons en déduire les modèles stables préférés pour chacune de nos stratégies classiques de fusion.

Pour les stratégies Somme et Card, nous avons les modèles stables préférés suivants : $\{\{s', d', o, r_d^3, r_s^3\}, \{s, d', o, r_{\neg s}^2, r_d^3\}\}$.

Pour la stratégie Max, nous avons les modèles stables préférés suivants : $\{\{s, d', o, r_{\neg s}^2, r_d^3\}, \{s, d, o', r_{\neg d}^1, r_{\neg s}^2, r_o^3\}, \{s', d, o, r_{\neg d}^1, r_{\neg d \vee \neg o}^2, r_s^3\}\}$.

Pour la stratégie GMax, nous avons les modèles stables préférés suivants : $\{\{s, d', o, r_{\neg s}^2, r_d^3\}\}$.

5.4 Équivalence entre modèles stables préférés et R-ensembles

Nous montrons maintenant qu'à chaque R-ensemble de E selon la stratégie P , il correspond un et un seul modèle stable préféré de Π_E selon la stratégie P .

Proposition 5.4.1. Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. Soit $S \subseteq V$ un ensemble d'atomes. Soit P une stratégie de fusion. S est un R-ensemble de E selon la stratégie P si et seulement si il existe un modèle stable préféré S de Π_E selon P tel que $F_O(R^+ \cap S) = S$.

Preuve.

La proposition précédente peut être réexprimée comme $\{F_O(S \cap R^+) \mid S \text{ est un modèle stable préféré selon } P\} = \mathcal{F}_P\mathcal{R}(E)$. La preuve ne sera donnée que pour *Card*, la trame reste identique pour les stratégies Σ , *Max* et *GMax*.

Nous cherchons donc à prouver que $\{F_O(S \cap R^+) \mid S \text{ est un modèle stable préféré selon } P\} = \mathcal{F}_{Card}\mathcal{R}(E)$.

Dans un premier temps, on montre que $\{F_O(S \cap R^+) \mid S \text{ est un modèle stable préféré selon } Card\} \subseteq \mathcal{F}_{Card}\mathcal{R}(E)$. Pour cela, nous supposons qu'il existe un ensemble S appartenant à l'ensemble des modèles stables préférés de Π_E selon *Card* et qu'il est impossible que le R-ensemble potentiel correspondant ne fasse pas partie des R-ensembles de E selon *Card*. Ainsi, nous posons $X = F_O(S \cap R^+)$. Par la proposition 5.2.1, $(K_1 \sqcup \dots \sqcup K_n) \setminus X$ est cohérent. Supposons maintenant que X n'appartienne pas à $\mathcal{F}_{Card}\mathcal{R}(E)$, il existe donc X' tel que $(K_1 \sqcup \dots \sqcup K_n) \setminus X'$ est cohérent et $X' <_{Card} X$. Nous pouvons en déduire qu'il existe une interprétation I_X sur l'ensemble des atomes de E qui satisfait $(K_1 \sqcup \dots \sqcup K_n) \setminus X'$. X' est le modèle stable associé à S' et $S' = \{a \mid a \in I_X\} \cup \{a' \mid \neg a \in I_X\} \cup \{r_f \mid f \in X'\}$. Nous savons que $X' = F_O(S' \cap R^+)$ et, par la proposition 1, que S' est un modèle stable de P_E . Cela entraîne que X' est préféré à X car $|X| = |S \cap R^+|$ ce qui est contradictoire avec l'hypothèse que X est un modèle stable préféré.

Nous montrons maintenant que $\mathcal{F}_{Card}\mathcal{R}(E) \subseteq \{F_O(S \cap R^+) \mid S \text{ est un modèle stable préféré selon } P\}$. S est un R-ensemble de E donc $(K_1 \sqcup \dots \sqcup K_n) \setminus X$ est cohérent. Il existe donc une interprétation I_X sur l'ensemble des atomes de E qui satisfait $(K_1 \sqcup \dots \sqcup K_n) \setminus X$. Nous notons S le modèle stable de Π_E correspondant à X . Par la proposition 1, nous savons que S est un modèle stable de Π_E . Supposons que S ne soit pas un modèle stable préféré de Π_E selon *Card*, cela suppose qu'il existe S' tel que $|F_O(S' \cap R^+)| = |S' \cap R^+| < |F_O(S \cap R^+)| = |S \cap R^+|$. Mais $S' = F_O(S' \cap R^+)$ et $(K_1 \sqcup \dots \sqcup K_n) \setminus S'$ est cohérent. S n'appartiendrait donc pas à $\mathcal{F}_{Card}\mathcal{R}(E)$ ce qui est contraire à l'hypothèse. $\square$

Nous rappelons maintenant l'exemple que nous avons développé plus haut afin de

Exemple 5.4.1. Nous reprenons l'exemple 5.3.1. Les modèles stables préférés de $\Pi_{E,IC}$ sont :

- Pour les stratégies *Somme* et *Card*, nous avons les modèles stables préférés suivants : $\{\{s', d', o, r_d^3, r_s^3\}, \{s, d', o, r_{\neg s}^2, r_d^3\}\}$.
- Pour la stratégie *Max*, nous avons les modèles stables préférés suivants : $\{\{s, d', o, r_{\neg s}^2, r_d^3\}, \{s, d, o', r_{\neg d}^1, r_{\neg s}^2, r_o^3\}, \{s', d, o, r_{\neg d}^1, r_{\neg d \vee \neg o}^2, r_s^3\}\}$.
- Pour la stratégie *GMax*, nous avons les modèles stables préférés suivants : $\{\{s, d', o, r_{\neg s}^2, r_d^3\}\}$.

Les R-ensembles de E dans cet exemple sont :

- Pour les stratégies *Somme* et *Card*, nous avons les R-ensembles suivants : $\{\{d, s\}, \{\neg s, d\}\}$.
- Pour la stratégie *Max*, nous avons les R-ensembles suivants : $\{\{\neg s, d\}, \{\neg d, \neg s, o\}, \{\neg d, \neg d \vee \neg o, s\}\}$.
- Pour la stratégie *Gmax*, nous avons les R-ensembles suivants : $\{\{\neg s, d\}\}$.

Nous observons que les R-ensembles de E pour chacune des stratégies correspondent bien aux modèles stables préférés tels qu'ils sont donnés dans l'exemple 5.3.1.

Conclusion

Dans ce chapitre, nous avons montré tout d'abord la traduction du problème de fusion du profil E contraint par IC en un programme logique $\Pi_{E,IC}$. Puis, nous avons montré que les modèles stables de ce programme correspondent aux R-ensembles potentiels du profil. Nous avons également donné la définition de modèles stables préférés avant de montrer que les R-ensembles de E contraint par IC selon la stratégie P sont les modèles stables préférés d'un programme $\Pi_{E,IC}$ selon cette même stratégie. Le cas clausal a donné lieu à publication dans [HPW07b, HPW07a] alors que le cas où les bases sont constituées de formules quelconque a fait l'objet de publication dans [HPW08b, HPW08a].

Il reste maintenant à donner dans le chapitre suivant une méthode de calcul des modèles stables quelle que soit la stratégie.

Chapitre 6

Le calcul des R-ensembles

Ce que les hommes veulent en fait, ce n'est pas la connaissance, c'est la certitude. *Bertrand Russell*

Introduction

Nous avons montré l'équivalence entre R-ensembles de E contraint par IC selon la stratégie P et les modèles stables préférés de $\Pi_{E,IC}$ selon P , nous allons présenter plusieurs méthodes permettant de sélectionner les modèles stables préférés en fonction de la stratégie choisie. Une première méthode est l'adaptation de l'algorithme Smodels. La seconde méthode utilise les instructions supplémentaires des solveurs ASP et permet une mise en œuvre indépendante du solveur.

6.1 Adaptation de l'algorithme Smodels

Nous avons proposé un algorithme, que nous appelons *rsf*, pour calculer les modèles stables préférés correspondant aux R-ensembles. Cet algorithme est une modification de Smodels. Contrairement à ce dernier qui fournit l'ensemble des modèles stables, *rsf* ne fournit que les modèles stables qui vérifient une condition passée en paramètre. En l'occurrence, les modèles stables préférés selon la stratégie P choisie. Il construit, étape par étape, une collection de modèles stables candidats. À la fin du calcul, cette collection contient tous les modèles stables préférés correspondant aux R-ensembles du profil pour la stratégie choisie.

La sélection des modèles stables préférés est réalisée grâce à la fonction $condition_P(A)$, où A est un ensemble d'atomes. Cette fonction compare le modèle stable candidat A en cours à ceux qui ont déjà été calculés. Les trois comportements possibles de la fonction $condition_P(A)$ sont :

1. A ne peut plus conduire à aucun modèle stable préféré. Dans ce cas, le calcul est arrêté et l'algorithme revient en arrière ;

2. L'interprétation qui correspond à A est complète et A est aussi préféré que les meilleurs modèles stables déjà calculés. Dans ce cas, A est ajouté à la collection des modèles stables candidats ;
3. L'interprétation qui correspond à A est complète et A est préféré aux modèles stables précédemment calculés. Dans ce cas, la collection constituée par le singleton A remplace la collection des modèles stables candidats.

6.2 L'heuristique de choix

L'heuristique originale de Smodels a pour but de maximiser le nombre d'atomes déduits après chaque choix. Cette heuristique n'est pas pertinente pour le type de programme logique que *rsf* produit. Elle est même contre-productive. En maximisant le nombre d'atomes, elle maximise le nombre d'atomes de règles alors que le but de notre algorithme est de minimiser leur quantité.

Nous avons modifié cette heuristique afin de sélectionner les atomes qui minimisent le nombre d'atomes déduits. De cette manière, le premier modèle stable aura plus de chances d'être un modèle stable préféré d'après la stratégie choisie et l'arbre de recherche sera fortement réduit.

Nous appelons cette nouvelle fonction $mheuristic(A)$.

6.3 Le nouvel algorithme

Le déroulement de l'algorithme est celui d'un algorithme de parcours d'arbre classique sauf que nous rajoutons des conditions afin d'élaguer l'arbre de manière plus efficace. De plus, nous mettons en place des conditions permettant de stocker les R-ensembles calculés au fur et à mesure de leur découverte :

1. si l'interprétation partielle conduit à des R-ensembles potentiels moins préférés au sens de la stratégie choisie, il n'est pas nécessaire de descendre plus loin dans l'arbre ;
2. si l'interprétation totale permet de construire un R-ensemble potentiel équivalent, au sens de la stratégie choisie, à ceux déjà construits, alors ce R-ensemble potentiel est ajouté à la liste des R-ensembles ;
3. si l'interprétation totale permet de construire un R-ensemble potentiel préféré, au sens de la stratégie choisie, à ceux déjà construits, alors le singleton constitué de ce R-ensemble potentiel devient la liste des R-ensembles.

Les adaptations de l'algorithme original de Smodels consistent à :

- ne pas calculer des sous-ensembles de R^+ conduisant à des modèles stables qui retirent plus de clauses que les meilleurs modèles déjà calculés ;
- ne pas calculer plusieurs fois les mêmes sous-ensembles de littéraux de R^+ ;
- tirer profit d'élagages dans l'arbre de recherche.

Algorithme 10 $rsf(A)$

```

 $A \leftarrow expand(A)$ 
si  $conflict(A)$  alors
  renvoyer  $Faux$ 
fin
si  $(1) condition_P(A) = 1$  alors
  renvoyer  $Faux$ 
sinon si  $A$  conduit à un R-ensemble déjà calculé alors
  renvoyer  $Faux$ 
sinon si  $A$  couvre  $Atome(E)$  alors
  si  $(2) condition_P(A) = 0$  alors
     $A$  est ajouté à l'ensemble des solutions
    renvoyer  $Vrai$ 
  sinon
     $(3)\{A\}$  devient l'ensemble des solutions
    renvoyer  $Vrai$ 
  fin
fin
 $x \leftarrow mheuristic(A)$ 
 $rsf(A \cup \{x\})$ 
 $rsf(A \cup \{not\ x\})$ 

```

Bien que correspondant à nos attentes, l'adaptation de *smodels* pour l'algorithme *rsf* reste dépendante de ce solveur dont les performances sont dépassées par des solveurs plus récents comme CLASP ou NoMore. C'est pour cette raison que nous souhaitons disposer d'une méthode pour mettre en œuvre la fusion de croyances indépendamment du solveur.

6.4 Sélection Off The Shelf

Une méthode pour réaliser la fusion par R-ensembles (RSF) à l'aide de n'importe quel solveur ASP consiste à rajouter au programme $\Pi_{E,IC}$, des règles et des instructions dont le but est de compter les atomes de règle présents dans chaque modèle stable afin de sélectionner les R-ensembles potentiels minimaux selon la stratégie P choisie.

Nous présentons ici les programmes logiques permettant de coder les stratégies *Somme*, *Card*, *Max*, *GMax*. Nous prouvons également que les résultats obtenus grâce à ces programmes logiques sont bien ceux attendus.

6.4.1 Stratégie *Somme*

L'ensemble des R-ensembles pour la stratégie *Somme* est l'ensemble de R-ensembles potentiels possédant le moins possible de formules. Ainsi, l'ensemble des modèles stables

préférés de $\Pi_{E,IC}$ sera l'ensemble des modèles stables de $\Pi_{E,IC}$ possédant le minimum d'atomes de règles. Nous notons indifféremment Π_E ou $\Pi_{E,\top}$ dans le cas où $IC = \top$.

Le principe est relativement simple puisqu'il suffit de compter le nombre total d'atomes de règle et de sélectionner les R-ensembles qui en comptent le moins. Cette sélection est réalisée par l'intermédiaire de l'instruction *minimize*{}. Ainsi, $\Pi_{E,IC}^\Sigma = \text{minimize} \{r_f^i \mid r_f^i \in R^+\} \cup \Pi_{E,IC}$.

Proposition 6.4.1. *Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. L'ensemble de modèles stables de $\Pi_{E,IC}^\Sigma$ est l'ensemble des modèles stables préférés de $\Pi_{E,IC}$ selon la stratégie Σ .*

Preuve.

$\Rightarrow$ Supposons qu'il existe un ensemble d'atomes X tel que X soit un modèle stable de $\Pi_{E,IC}^\Sigma$ mais tel que X ne soit pas un modèle stable préféré de $\Pi_{E,IC}$ selon la stratégie Σ .

Si X est un modèle stable de $\Pi_{E,IC}^\Sigma$ cela signifie qu'il s'agit d'un modèle stable de $\Pi_{E,IC}$ et qu'il n'existe pas d'autre modèle stable Y de $\Pi_{E,IC}$ tel que $|Y| < |X|$, grâce à l'instruction *minimize*. Mais si X n'est pas un modèle stable préféré de $\Pi_{E,IC}$, cela signifie qu'il existe un ensemble d'atome Y tel que Y est un modèle stable de $\Pi_{E,IC}$ et $|Y| < |X|$. Ce qui est contradictoire avec l'hypothèse. Nous avons donc montré que l'ensemble de modèles stables de $\Pi_{E,IC}^\Sigma$ est inclus dans l'ensemble des modèles stables préférés de $\Pi_{E,IC}$ selon la stratégie Σ .

$\Leftarrow$ Supposons qu'il existe un ensemble d'atomes X tel que X soit modèle stable préféré de $\Pi_{E,IC}$ selon la stratégie Σ mais tel que X ne soit pas un modèle stable de $\Pi_{E,IC}^\Sigma$.

Si X est un modèle stable préféré de $\Pi_{E,IC}$, cela signifie qu'il n'existe un ensemble d'atome Y tel que Y soit un modèle stable de $\Pi_{E,IC}$ et $|Y| < |X|$. Mais si X n'est pas un modèle stable de $\Pi_{E,IC}^\Sigma$ cela signifie qu'il s'agit d'un modèle stable de $\Pi_{E,IC}$ et qu'il existe d'autre modèle stable Y de $\Pi_{E,IC}$ tel que $|Y| < |X|$, grâce à l'instruction *minimize*. Ce qui est contradictoire avec l'hypothèse.

Nous avons donc montré que l'ensemble des modèles stables préférés de $\Pi_{E,IC}$ selon la stratégie Σ est inclus dans l'ensemble de modèles stables de $\Pi_{E,IC}^\Sigma$. $\square$

Exemple 6.4.1. *Soit $E = \{K_1, K_2, K_3\}$ le profil de croyances tel que $K_1 = \{\neg d, s \vee o\}$, $K_2 = \{\neg s, d \vee o, \neg d \vee \neg o\}$ et $K_3 = \{s, d, o\}$ avec $IC = \top$*

$$\Pi_E^\Sigma = \{\text{minimize}\{r_{\neg d}^1, r_{s \vee o}^1, r_{\neg s}^2, r_{d \vee o}^2, r_{\neg d \vee \neg o}^2, r_s^3, r_d^3, r_o^3\} \cup \Pi_{E,IC}\}.$$

6.4.2 Stratégie *Card*

La stratégie *Card* est très proche de la stratégie Σ (cf *supra*). La différence se situe dans le fait que la stratégie *Card* traite les formules exprimées plusieurs fois comme si elles n'étaient présentes qu'en un seul exemplaire. La différence dans la mise en œuvre se fera donc grâce à un prétraitement où les doublons seront supprimés.

La méthode employée pour la stratégie *Card* est similaire au détail près que les doublons sont préalablement enlevés du profil. Nous notons $s(E)$ le profil de croyances où les formules présentes en plusieurs exemplaires sont réduites à un singleton.

Ainsi $\Pi_{E,IC}^{Card} = \text{minimize} \{r_f^i \mid r_f^i \in R^+\} \cup \Pi_{s(E),IC}$. La proposition suivante est vérifiée :

Proposition 6.4.2. *Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. L'ensemble de modèles stables de $\Pi_{E,IC}^{Card}$ est l'ensemble des modèles stables préférés de $\Pi_{E,IC}$ selon la stratégie *Card*.*

Preuve.

Démonstration similaire à la stratégie Σ . □

Exemple 6.4.2. *Soit $E = \{K_1, K_2, K_3\}$ le profil décrit dans l'exemple 6.4.1.*

$$\Pi_E^{Card} = \{\text{minimize}\{r_{\neg d}^1, r_{s \vee o}^1, r_{\neg s}^2, r_{d \vee o}^2, r_{\neg d \vee \neg o}^2, r_s^3, r_d^3, r_o^3\} \cup \Pi_{E,IC}\}.$$

6.4.3 Stratégie *Max*

Pour la stratégie *Max*, l'ensemble de modèles stables préférés de $\Pi_{E,IC}$ sera l'ensemble des modèles stables qui retirent le moins possible de formules dans la base où elle en retire le plus. Sa mise en pratique se fera donc en plusieurs étapes. Dans un premier temps, le nombre d'atomes de règles présents dans le modèle stable sera compté base par base ; cette valeur est stockée au travers d'un prédicat $size(U)$ qui signifie qu'il existe une base dans laquelle il est retiré U formules. Dans un deuxième temps, nous prenons la plus grande valeur de $size(U)$ qui sera stockée dans une variable $max(X)$. Et, finalement, une instruction d'optimisation sera définie afin de ne conserver que les modèles stables pour lesquelles la valeur de $max(X)$ est minimale.

La stratégie *Max* requiert une autre utilisation de l'instruction $\text{minimize}\{\}$. Soit X un modèle stable. Une première étape calcule $max_{1 \leq i \leq n}(|\{r_f^i \mid r_f^i \in X\}|)$ pour chaque X . La seconde étape utilise ce calcul et l'instruction $\text{minimize}\{\}$ pour conserver les modèles stables minimaux. Pour savoir combien de formules sont retirées dans la base où il y a le plus de retraits, il est nécessaire de commencer par calculer combien de formules sont retirées dans chaque base de croyances. Ce calcul est effectué par :

$$\Pi_E^{max,size} = \left\{ \begin{array}{l} \delta_0 : \#domain\ possible(U). \\ \delta_1 : \#domain\ base(V). \\ \delta_2 : possible(1..m). \\ \delta_3 : base(1..n). \\ \alpha : size(U) \leftarrow U\{r_f^V \mid F_0(f) \in \varphi_V\}U. \end{array} \right\}$$

où m est la taille de la plus grande base de croyances. Les règles δ_i sont les domaines de définition pour la règle α . S'il existe une base K_V où le nombre d'atomes r_f^V vrais dans X est égal à U , alors $size(U)$ sera vrai dans X . La plus grande valeur U pour laquelle $size(U)$ est vrai est calculée grâce à $\Pi_E^{max,bound}$:

$$\Pi_E^{max, bound} = \left\{ \begin{array}{l} \delta_4 : \#domain\ possible(W). \\ \beta_1 : negmax(W) \leftarrow size(U), U > W. \\ \beta_2 : max(U) \leftarrow size(U), not\ negmax(U). \end{array} \right\}$$

β_1 détermine tous les entiers W pour lesquelles il existe $U > W$ tel que $size(U)$ est vrai. Alors, $max(U)$ est vrai pour la plus grande valeur de U telle que $size(U)$ est vrai. Pour la stratégie *Max*, soit

$$\Pi_{E, IC}^{Max} = \Pi_E^{max, size} \cup \Pi_E^{max, bound} \cup \Pi_{E, IC} \cup minimize[max(1) = 1, \dots, max(m) = m].$$

La proposition suivante est vérifiée :

Proposition 6.4.3. *Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. L'ensemble des modèles stables de $\Pi_{E, IC}^{Max}$ est l'ensemble des modèles stables préférés de $\Pi_{E, IC}$ selon *Max*.*

Preuve.

1. Nous savons, au travers de la proposition 5.2.3 que X est un modèle stable de $\Pi_{E, IC}$ si et seulement si $F_O(X \cap S)$ est un R-ensemble potentiel de E contraint par *IC*.
2. Soit X un modèle stable de $\Pi_{E, IC}$, l'atome $size(U)$ est vrai dans $\Pi_{E, IC} \cup \Pi_E^{Max, size}$ si et seulement si il existe i tel que $|X \cap K_i| = U$ au travers de la règle α . Donc, $negmax(W)$ est vrai si et seulement si il existe W et il existe i tel que $|X \cap K_i| = W$ et il existe V et il existe j $|X \cap K_j| = V$ avec $V > W$. Ainsi $negmax(W)$ est vrai pour tout W tel que $size(W)$ est vrai et non maximale par les règles α et β_1 . Finalement, $max(U)$ est vrai si et seulement si $U = \max_{i=1}^n |X \cap K_i|$ par les règles α, β_1, β_2 .
3. Nous allons maintenant montrer que l'ensemble des R-ensembles de $\Pi_{E, IC}^{Max}$ est l'ensemble des modèles stables préférés de $\Pi_{E, IC}$ pour la stratégie *Max*.
 - ⊆ Si X est un R-ensemble de $\Pi_{E, IC} \cup \Pi_E^{Max, size} \cup \Pi_E^{Max, bound}$, nous notons sa valeur pour l'instruction $minimize\ val(X)$ et nous rappelons que $val(X) = \max_{i=1}^n |X \cap K_i|$. Supposons qu'il existe X tel que $F_O(X \cap S) \subseteq \mathcal{F}_{Max, IC}^{RSF} \mathcal{R}(E)$ et tel que X ne soit pas un modèle stable préféré de $\Pi_{E, IC}$ selon *Max*. Il existe donc X' tel que $val(X') < val(X)$. Par construction, si $val(X') < val(X)$ cela implique que $\max_{i=1}^n |X' \cap K_i| < \max_{i=1}^n |X \cap K_i|$ et donc qu'il est impossible que $F_O(X \cap S)$ appartienne à $\mathcal{F}_{Max, IC}^{RSF} \mathcal{R}(E)$.
 - ⊇ Supposons qu'il existe X tel que X soit un modèle stable préféré de $\Pi_{E, IC}$ selon *Max* et tel que $F_O(X \cap S) \not\subseteq \mathcal{F}_{Max, IC}^{RSF} \mathcal{R}(E)$. Il existe donc X' tel que $\max_{i=1}^n |X' \cap K_i| < \max_{i=1}^n |X \cap K_i|$ cela implique donc $val(X') < val(X)$ ce qui est contradictoire avec le fait que X soit un modèle stable préféré de $\Pi_{E, IC}$ selon *Max*.

□

Nous illustrons cette traduction à l'aide de l'exemple suivant :

Exemple 6.4.3. Soit $E = \{K_1, K_2, K_3\}$ le profil décrit dans l'exemple 6.4.1.

$$\Pi_E^{Max, size} = \left\{ \begin{array}{l} \#domaine\ possible(U) \quad \#domaine\ base(V) \\ possible(1..3) \quad base(1..3) \\ size(U) \leftarrow U\{r_{\neg d}^1, r_{s \vee o}^1\}U. \\ size(U) \leftarrow U\{r_{\neg s}^2, r_{d \vee o}^2, r_{\neg d \vee \neg o}^2\}U. \\ size(U) \leftarrow U\{r_s^3, r_d^3, r_o^3\}U. \end{array} \right\}$$

$$\Pi_E^{Max, bound} = \left\{ \begin{array}{l} \#domaine\ possible(W) \quad negmax(W) \leftarrow size(U), U > W. \\ max(U) \leftarrow size(U), not\ negmax(W). \end{array} \right\}$$

$$\Pi_E^{Max} = \left\{ \begin{array}{l} \Pi_E^{Max, size} \cup \Pi_E^{Max, bound} \cup minimize[max(1) = 1, max(2) = 2, max(3) = 3] \\ \cup \Pi_{E, IC} \end{array} \right\}$$

6.4.4 Stratégie *GMax*

Pour la stratégie *GMax*, il est nécessaire de compter le nombre d'atomes de règles pour chacune des bases de croyances. Une fois ces valeurs obtenues, nous les rangeons par ordre décroissant. Ainsi, un modèle stable préféré sera un modèle stable tel qu'il soit minimal selon l'ordre lexicographique.

La stratégie *GMax* compare les R-ensembles potentiels d'après la séquence des $|X \cap K_i|$ ordonnée de manière décroissante. Comme pour *Max*, il faut connaître le nombre de formules retirées dans chacune des bases. Ce calcul est représenté par la règle suivante où m représente la taille de la base φ_V .

Si $size(V, U)$ est vrai, dans la base φ_V , le modèle stable en cours contient U atomes de règle. Une fois calculées pour toutes les bases K_V , ces valeurs sont ordonnées par $\Pi_E^{GMax, bound}$ où X_1 est la plus grande valeur de $size()$ et X_n la plus petite. Finalement, le polynôme $X_n + X_{n-1} \times m + \dots + X_1 \times m^{n-1}$ est optimisé.

$$\Pi_E^{GMax, size} = \{\gamma_1 : size(V, U) \leftarrow U\{r_1^V, \dots, r_m^V\}U.\}$$

$$\Pi_E^{GMax, bound} = \left\{ \begin{array}{l} \alpha_0 : \quad max(X_1, \dots, X_n) \leftarrow size(Y_1, X_1), \dots, size(Y_n, X_n), \\ \quad X_1 \geq X_2, \dots, X_{n-1} \geq X_n, neq(Y_1, \dots, Y_n). \\ \alpha_1 : \quad max_1(X_1) \leftarrow max(X_1, \dots, X_n), X_1 \geq X_2, \dots, X_{n-1} \geq X_n. \\ \alpha_i : \quad \dots \\ \alpha_n : \quad max_3(X_3) \leftarrow max(X_1, \dots, X_n), X_1 \geq X_2, \dots, X_{n-1} \geq X_n. \end{array} \right\}$$

Pour la stratégie *GMax*, soit

$$\begin{aligned} \Pi_E^{GMax} &= \Pi_E^{GMax, size} \cup \Pi_E^{GMax, bound} \cup \Pi_{E, IC} \\ &\cup minimize[max_n(1) = 1, max_n(2) = 2, \dots, max_i(1) = m^{n-i}, \\ &\quad max_i(2) = 2 \times m^{n-i}, \dots, max_1(n) = n \times m^{n-1}]. \end{aligned}$$

La proposition suivante est vérifiée :

Proposition 6.4.4. *Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. L'ensemble des R-ensembles de $\Pi_{E,IC}^{GMax}$ est l'ensemble des modèles stables préférés de $\Pi_{E,IC}$ pour la stratégie $GMax$.*

Preuve.

1. Nous savons, au travers de la proposition 5.2.3 que X est un modèle stable de $\Pi_{E,IC}$ si et seulement si $F_O(X \cap S)$ est un R-ensemble potentiel de E contraint par IC .
2. Soit X un modèle stable de $\Pi_{E,IC}$, l'atome $size(V, U)$ est vrai dans $\Pi_{E,IC} \cup \Pi_E^{GMax, size}$ si et seulement si $|X \cap K_V| = U$ au travers des règles de $\Pi_E^{GMax, size}$ avec $U = p_X^V$. Ainsi, au travers de la règle α_0 , on sait que $max(X_1, \dots, X_n)$ est vrai si et seulement si $L_X^E = (X_1, \dots, X_n)$ puisque qu'il réalise l'ordonnancement des p_X^i calculé par la règle γ_1 . Finalement, les règles α_i séparent les valeurs de X_1 pour qu'elles puissent être utiliser dans l'instruction minimize.
3. Nous allons maintenant montrer que l'ensemble des R-ensembles de $\Pi_{E,IC}^{GMax}$ est l'ensemble des modèles stables préférés de $\Pi_{E,IC}$ pour la stratégie $GMax$.
 - $\subseteq$ Si X est un R-ensemble de $\Pi_E^{GMax, size} \cup \Pi_E^{GMax, bound} \cup \Pi_{E,IC}$, nous notons sa valeur pour l'instruction minimize $val(X)$ et nous rappelons que $val(X) = \sum_{i=1}^n (p_X^i) \times (m+1)^{n-i}$. Supposons qu'il existe X tel que $F_O(X \cap S) \subseteq \mathcal{F}_{GMax, IC}^{RSF} \mathcal{R}(E)$ et tel que X ne soit pas un modèle stable préféré de $\Pi_{E,IC}$ selon $GMax$. Il existe donc X' tel que $val(X') < val(X)$. Par construction, si $val(X') < val(X)$ cela implique que :
 - $\exists i$ tel que $p_{X'}^i < p_X^i$;
 - $\forall j < i$ alors $p_{X'}^j = p_X^j$.
 ce qui implique que $(p_{X'}^1, \dots, p_{X'}^n) <_{lex} (p_X^1, \dots, p_X^n)$ et qu'il est impossible que $F_O(X \cap S)$ appartienne à $\mathcal{F}_{GMax, IC}^{RSF} \mathcal{R}(E)$.
 - $\supseteq$ Supposons qu'il existe X tel que X soit un modèle stable préféré de $\Pi_{E,IC}$ selon $GMax$ et tel que $F_O(X \cap S) \not\subseteq \mathcal{F}_{GMax, IC}^{RSF} \mathcal{R}(E)$. Il existe donc X' tel que $(p_{X'}^1, \dots, p_{X'}^n) <_{lex} (p_X^1, \dots, p_X^n)$ cela implique que :
 - $\exists i$ tel que $p_{X'}^i < p_X^i$;
 - $\forall j < i$ alors $p_{X'}^j = p_X^j$.
 et donc que $val(X') < val(X)$ ce qui est contradictoire avec le fait que X soit un modèle stable préféré de $\Pi_{E,IC}$ selon $GMax$.

□

Exemple 6.4.4. *Soit $E = \{K_1, K_2, K_3\}$ le profil décrit dans l'exemple 6.4.1.*

$$\Pi_E^{GMax, size} = \left\{ \begin{array}{cc} \#domaine\ possible(U) & \#domaine\ base(V) \\ \#domaine\ possible(W) & \#domaine\ possible(X_1) \\ \#domaine\ base(X_2) & \#domaine\ possible(X_3) \\ possible(1..3) & base(1..3) \\ size(1, U) : -U\{r_{-d}^1, r_{s \vee o}^1\}U. \\ size(2, U) : -U\{r_{-s}^2, r_{d \vee o}^2, r_{-d \vee -o}^2\}U. \\ size(3, U) : -U\{r_s^3, r_d^3, r_o^3\}U. \end{array} \right\}$$

$$\Pi_E^{GMax, bound} = \left\{ \begin{array}{l} max(X_1, X_2, X_3) : - \quad size(1, X_1), size(2, X_2), \\ \quad \quad \quad \quad \quad \quad \quad size(3, X_3), X_1 \geq X_2, X_2 \geq X_3. \\ max_1(X_1) : - \quad max(X_1, X_2, X_3), X_1 \geq X_2, X_2 \geq X_3. \\ max_2(X_2) : - \quad max(X_1, X_2, X_3), X_1 \geq X_2, X_2 \geq X_3. \\ max_3(X_3) : - \quad max(X_1, X_2, X_3), X_1 \geq X_2, X_2 \geq X_3. \end{array} \right\}$$

$$\Pi_E^{GMax} = \left\{ \begin{array}{l} \Pi_E^{GMax, size} \cup \Pi_E^{GMax, bound} \cup \Pi_{E, IC} \\ minimize[max_1(1) = 16, max_1(2) = 32, max_1(3) = 48, max_2(1) = 4, \\ max_2(2) = 8, max_2(3) = 12, max_3(1) = 1, max_3(2) = 2, max_3(3) = 3] \end{array} \right\}$$

Conclusion

Nous avons dans ce chapitre montré plusieurs méthodes pour calculer les R-ensembles selon les différentes stratégies (*Somme*, *Card*, *Max* et *Gmax*) de fusion. Nous avons montré pour chacune de ces stratégies qu'il existait une correspondance entre les modèles stables des programmes construits et les R-ensembles selon cette stratégie. L'adaptation de l'algorithme Smodels a donné lieu à publication dans [HPW07b, HPW07a] alors que l'adaptation indépendante du solveur a fait l'objet de publication dans [HPW08b, HPW08a].

Des tests utilisant le solveur ASP ont permis de vérifier expérimentalement ces résultats dans le chapitre suivant.

Chapitre 7

Expérimentations

Sans oubli, pas s'abstraction, pas de pensée, pas de vie non plus. L'oubli est une condition nécessaire de la vie. *Denis Guedj*

Introduction

Nous présentons ici une série de tests effectués sur la mise en œuvre de la fusion par R-ensembles décrite dans le chapitre précédent. Dans un premier temps, nous présentons une expérimentation basée sur la génération de profils de croyances aléatoires afin d'avoir une première évaluation du comportement global de la méthode. Cette expérimentation sur des bases aléatoires portera également sur une comparaison avec la méthode basée sur les Diagrammes de Décisions Binaires (ou BDD) de N.Gorogiannis et A. Hunter présentée dans [GH08].

Dans une seconde partie nous présenterons l'adaptation faite à la méthode de fusion par R-ensembles afin de répondre au besoin du projet européen VENUS. La méthode de fusion par R-ensembles y a été adaptée pour la gestion d'incohérences dans les relevés en archéologie sous-marine.

7.1 Expérimentations sur des bases aléatoires

Nous présentons ici une étude expérimentale des performances de notre implantation de la fusion par R-ensembles.

Les tests ont été conduits selon le protocole suivant. Les bases de croyances sont générées aléatoirement selon deux critères : le nombre d'atomes, que l'on note na , et le nombre de formules, que l'on note nf . Les formules sont des formules quelconques d'une profondeur inférieure ou égale à trois. Les profils de croyances sont ensuite générés en fonction du nombre de bases choisi, que l'on note nb .

Les tests ont été effectués sur un Centrino Duo cadencé à 1.73Ghz et équipé de 2Go de RAM

7.1.1 Stratégie *Somme*

Des tests sont lancés avec des profils contenant, 2, 3, 4 et 5 bases de croyances. Chaque base de croyances est cohérente. Pour chacune de ces valeurs, des tests sont lancés pour un nombre d'atomes égal à 30, 35 et 40. Et pour chaque nb et pour chaque na , une série de mille tests est lancée pour une série de ratio de (nf/na) allant de 4 à 7 avec un pas de 0,2.

Le graphe de la figure 7.1 présente l'évolution du temps d'exécution moyen en fonction de l'évolution du nombre de bases.

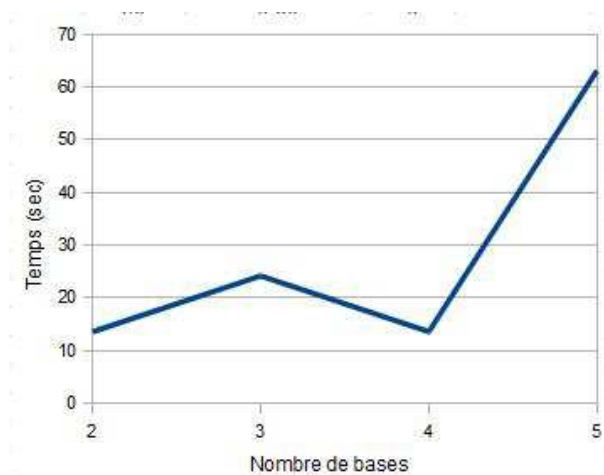


FIGURE 7.1 – Temps d'exécution par rapport au nombre de bases

Le graphe de la figure 7.2 présente l'évolution du temps d'exécution moyen en fonction de l'évolution du ratio (nf/na) .

Nous pouvons constater que l'augmentation du temps d'exécution est relativement lente jusqu'à un ratio qui semble se situer autour de 5,8 et qu'il se produit une accélération autour de $nf = 6,4$.

7.1.2 Stratégie *Max*

Le protocole utilisé dans cette sous-section est identique à celui décrit pour la stratégie *Somme*.

Le graphe de la figure 7.3 présente l'évolution du temps d'exécution moyen en fonction de l'évolution du nombre de bases.

Nous pouvons observer que le temps d'exécution moyen décroît avec l'augmentation du nombre de bases.

Le graphe de la figure 7.4 présente l'évolution du temps moyen en fonction de l'évolution du ratio (nf/na) .

De la même manière que pour la stratégie *Somme*, on peut constater que l'augmentation du temps d'exécution est relativement lente jusqu'à un ratio qui semble se

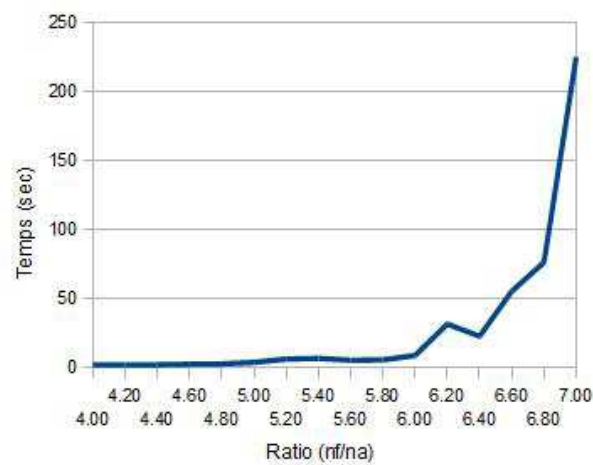
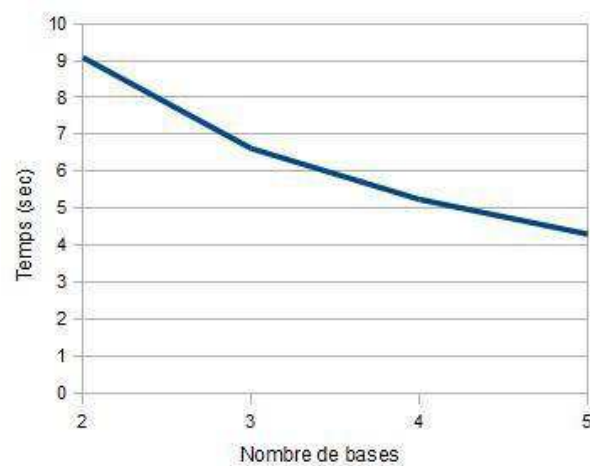
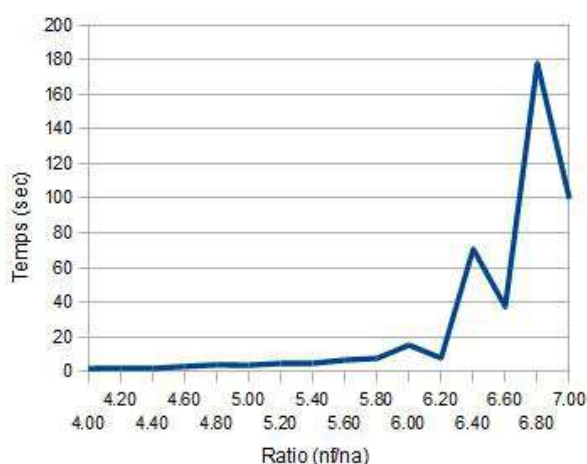
FIGURE 7.2 – Temps d'exécution en fonction du rapport (nf/na)

FIGURE 7.3 – Temps d'exécution par rapport au nombre de bases

FIGURE 7.4 – Temps d'exécution en fonction du rapport (nf/na)

situer autour de 5,8 et qu'il se produit une augmentation rapide du temps d'exécution de $nf = 6,4$.

7.1.3 Comparaison avec la méthode de N. Gorogiannis et A.Hunter

Nous comparons ici les résultats d'une étude expérimentale permettant de comparer la mise en œuvre de la fusion par R-ensembles avec une mise en œuvre de la fusion basée sur des opérateurs de morphologie mathématique utilisant des Diagrammes de Décisions Binaires (ou BDD) présentée dans [GH08].

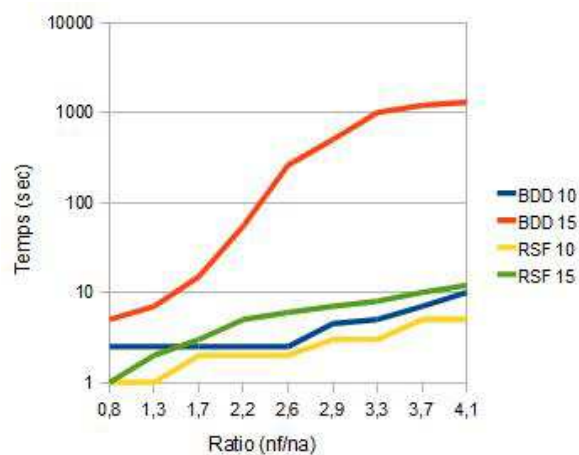
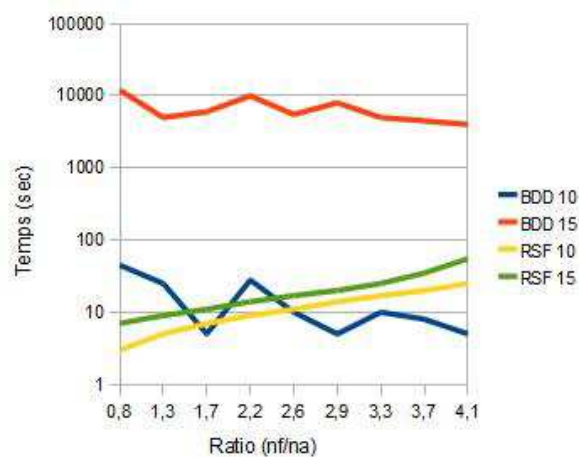
Un protocole proche de celui utilisé dans [GH08] est utilisé ici. Chaque profil de croyances est composé de sept bases de croyances. Toutes les bases de croyances sont générées indépendamment en fonction de deux critères : le nombre d'atomes, que l'on note na , et le nombre de formules, que l'on note nf . Chaque base de croyances est cohérente.

Les tests sont présentés pour les stratégies *Max* et *Somme*.

Il ressort de l'analyse que la fusion par R-ensembles se comporte mieux lors de l'augmentation du nombre d'atomes. L'augmentation du nombre d'atomes de dix à quinze multiplie le temps d'exécution par deux pour la fusion par R-ensembles alors que dans le cas de la méthode basée sur les Diagrammes de Décision Binaires la même augmentation multiplie le temps d'exécution par cent ; et ceci quelle que soit la stratégie.

Mais d'un autre côté, la fusion par R-ensembles est plus sensible à l'augmentation du nombre de formules, notamment dans le cas de la stratégie *Max* où l'augmentation du nombre de formules réduit le temps d'exécution pour la mise en œuvre utilisant des BDD. Alors que celui de la fusion par R-ensembles augmente.

La comparaison entre ces deux méthodes est d'une valeur relative car la fusion par R-ensembles se situe dans le cadre syntaxique alors que la méthode basée sur les BDD

FIGURE 7.5 – Temps d'exécution pour la stratégie *Somme*FIGURE 7.6 – Temps d'exécution pour la stratégie *Max*

se situe dans le cadre sémantique.

7.2 Gestion de l'incohérence sur le site de Pianosa

Présentation du projet VENUS

Nos travaux se situent dans le cadre du projet européen VENUS (Virtual Exploration of Underwater Sites) no # FP6-2005-IST-5¹, dont le but est de fournir des méthodes scientifiques et des outils technologiques pour l'exploration virtuelle de sites archéologiques sous-marins.

L'une des tâches de ce projet est l'étude de la représentation des connaissances. Lors des missions du projet sur les sites archéologiques sous-marins, la photogrammétrie est utilisée pour l'acquisition des données archéologiques et la connaissance sur les artefacts provient du domaine de l'archéologie sous-marine mais aussi du processus d'acquisition des données.

La connaissance générique est fournie sous la forme d'une description des amphores fournie par les archéologues et les observations proviennent des relevés photogrammétriques du site. En raison des conditions souvent difficiles (difficulté d'accès, profondeurs des sites, etc.), les observations des amphores sur le site peuvent être incohérentes avec la description générique et nécessitent le développement d'une méthode de gestion de l'incohérence. Nous avons utilisé la fusion par R-ensembles pour la gestion de l'incohérence et notre expérimentation repose sur le schéma de la figure 7.7

1. Description de la connaissance générique dans le Modèle Entité par un expert.
2. Relevé photogrammétrique utilisant l'interface ARPENTEUR par un expert.
3. Chargement des contraintes en logique des prédicats.
4. Sauvegarde des résultats du relevé dans la base de données.
5. Export en XML des instances d'amphores.
6. Modification possible des instances à partir du module de Réalité Virtuelle et chargement des instances à visualiser.
7. Chargement du modèle de terrain.
8. Chargement des photographies.
9. Intégration des résultats finaux dans la Réalité Virtuelle.
10. Interaction avec les données grâce au module de Réalité Virtuelle par un expert.

Les informations génériques et les observations sont fournies au module de raisonnement reposant sur la fusion par R-ensembles qui pointe les amphores mesurés incohérentes avec la description générique et qui les transmet à un module de réalité virtuelle qui permet à l'archéologue de les visualiser.

1. <http://www.venus-project.eu>

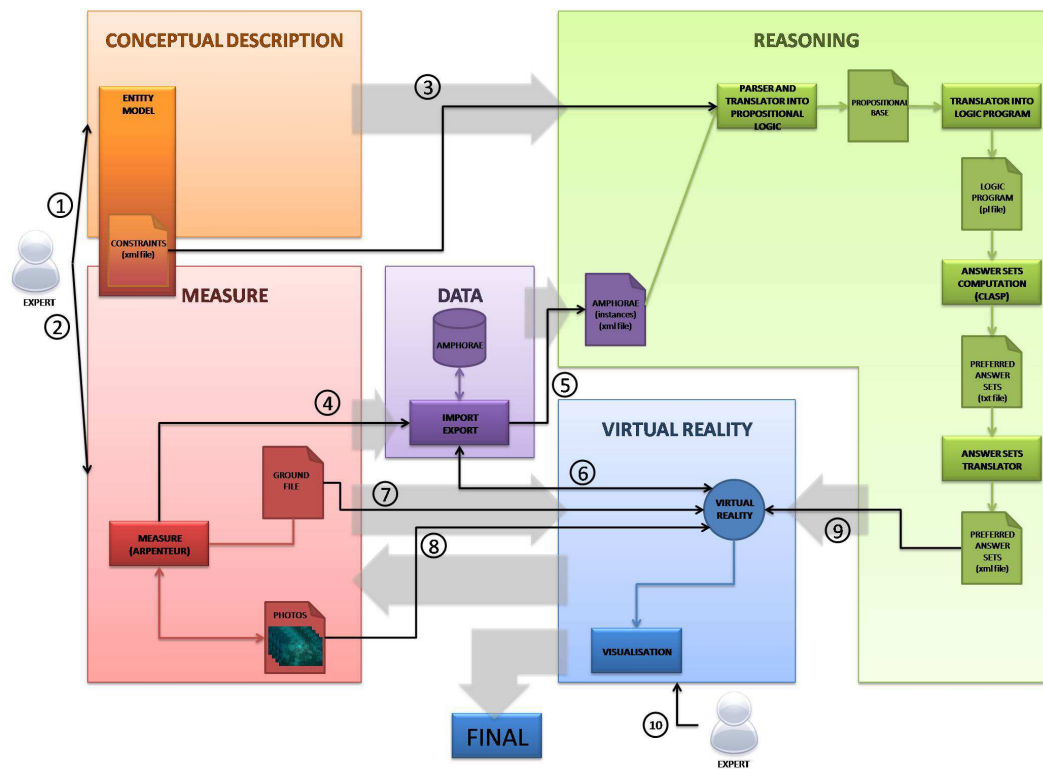


FIGURE 7.7 – Schéma descriptif du prototype de gestion de l'incohérence dans VENUS

7.2.0.1 Description conceptuelle

Nous représentons la connaissance générique selon le modèle Entité qui a été proposé dans [Sei06]. Le modèle Entité est représenté par $\mathcal{E} = \{\mathcal{C}, \mathcal{V}^d, \mathcal{C}^I, \mathcal{R}, \mathcal{C}^E\}$ où $\mathcal{C}$ est un concept (ou une classe), $\mathcal{V}^d$ est l'ensemble des valeurs par défaut des attributs, $\mathcal{C}^I$ est un ensemble de contraintes sur les attributs, $\mathcal{R}$ est un ensemble de relations entre les objets et $\mathcal{C}^E$ est un ensemble de contraintes sur les relations appartenant à $\mathcal{R}$. Dans le projet VENUS, nous nous limitons à un sous-ensemble du modèle Entité constitué de $\mathcal{E} = \{\mathcal{C}, \mathcal{V}^d, \mathcal{C}^I\}$.

Les concepts sont des types d'amphores observés sur le site archéologique. Par exemple, dans le cas du site de Pianosa, les types d'amphores sont au nombre de huit : *Dressel2_4_Courte*, *Dressel2_4_longue*, *Dressel7_11*, *Dressel_20*, *Gauloise_3*, *Beltram_2B*, *Haltern_70*, *Pascual_1*.

Pour chaque concept, c'est à dire que chaque type d'amphore, nous représentons neuf attributs (entre parenthèses, leur dénomination dans la base de connaissances) :

- Hauteur totale (*totalHeight*)
- Largeur totale (*totalWidth*)
- Longueur totale (*totalLength*)
- Diamètre de la ceinture (*bellyDiameter*)
- Diamètre interne (*internalDiameter*)
- Diamètre externe (*externalDiameter*)
- Diamètre du pied (*footDiameter*)
- Longueur des lèvres (*heightLips*)
- Masse (*Mass*)

Étant donné que nous travaillons sur des amphores, les attributs *totalWidth* et *totalLength* ont la même valeur car il s'agit d'objet de révolution et les seuls attributs mesurables sont *totalHeight* et *totalLength*.

Les valeurs par défaut de ces attributs prennent la forme d'une plage de valeurs et $\mathcal{V}^d$ représente l'ensemble de neuf intervalles, chaque intervalle correspondant aux valeurs possibles pour un des neuf attributs pour un type donné.

L'ensemble des contraintes sur les neuf attributs $\mathcal{C}^I$ consiste en contraintes d'intégrité. Par exemple, dans le cas du site archéologique de Pianosa, une contrainte d'intégrité spécifie que la valeur des attributs *TotalHeight* et *totalLength* (ou *totalWidth*) ne peut pas dépasser deux mètres. Un autre exemple est une contrainte de domaine stipulant qu'une amphore possède un et un seul type parmi les huit types possibles. Les contraintes conditionnelles expriment la compatibilité entre les valeurs mesurées des attributs et les valeurs par défaut des attributs pour un type donné.

La base de connaissances est constituée par le savoir générique selon le modèle Entité restreint que nous venons de décrire, stockée dans un fichier `typology.xml`, et les données provenant du relevé, stockées dans un fichier `amphorae.xml`.

7.2.0.2 Le module de réalité virtuelle

Le module de réalité virtuelle destiné aux archéologues permet aux experts d'intégrer avec les informations relevées sur le site. La structure de l'environnement virtuel est construit sur le fond marin et les artefacts mesurés par l'application ARPENTEUR [DSS⁺07]. Les artefacts sont instanciés sur le site virtuel en fonction de leur enregistrement dans la base de données archéologique. Les experts peuvent inspecter et interroger les artefacts à partir de l'environnement virtuel, par exemple, à chaque fois qu'un expert rencontre un artefact suspect (un artefact qui peut présenter un défaut dans l'enregistrement), il ou elle peut demander à l'application de vérifier la cohérence (à travers le module de raisonnement) qui peut lui proposer un nouvel ensemble d'alternatives pour l'objet suspecté (ces alternatives peuvent être de refaire la mesure à partir des photos ou donner un nouveau type plus adapté à l'artefact).

7.2.0.3 Mesures

Ce module correspond au relevé photogrammétrique du site. C'est à dire la prise de photos des amphores sur le site, ainsi que des photos du terrain sur lequel elles reposent. Puis la construction du modèle 3D des amphores et du terrain pour ARPENTEUR [DSS⁺07] à partir des photographies [DGC⁺04].

7.2.0.4 Données

Nous décrivons ici de manière plus détaillée les informations avec lesquelles nous allons travailler.

Le jeu de données est construit à partir de deux sources :

- Une typologie qui définit les types d'amphores susceptibles d'être rencontrés sur le site. La typologie contient l'ensemble des caractéristiques attendues pour une amphore. Par exemple, une amphore de type *Beltran_2B* est supposée avoir un diamètre de 378,4 millimètres.
- Un relevé des informations recueillies sur le site à proximité de l'île de Pianosa (Italie).

La typologie inclut la description de douze types d'amphores² différentes et porte sur neuf attributs mesurables : Hauteur totale, Largeur totale, Longueur totale, Diamètre de la ceinture, Diamètre interne, Diamètre externe, Diamètre du pied, Longueur des lèvres, Masse.

Les informations sont représentées sous la forme d'un fichier XML que nous allons décrire maintenant.

La typologie

Nous décrivons ici les informations fournies par la typologie. Pour chaque type d'amphore, la typologie fournit le nom, les valeurs attendues pour chaque attribut et les unités

2. Le site de Pianosa ne comporte, d'après les archéologues que huit types d'amphores différents. Cependant, la typologie fournit la description de douze types d'amphores.

dans lesquelles ces valeurs sont exprimées. Les valeurs sont entendues avec un marge de 20%.

Par exemple, une amphore de type *Beltran_2B* possède (à vingt pourcents près) une hauteur de 479.4 millimètres, une longueur de 378.4 millimètres, une largeur de 378.4 millimètres, une ceinture de 378.4 millimètres de diamètre, un diamètre interne de 115.6 millimètres, un diamètre externe de 161.0 millimètres, un pied de diamètre de 3.56 millimètres, des lèvres de 32.0 millimètres et une masse de 11.0 kilogrammes.

Ces caractéristiques sont encodées de la manière suivante dans le fichier XML.

```
<itemMesurableTypology>
  <item>
    <hierarchy
      name="Beltran_2B"/>
    <description
      dimensionUnit="mm" massUnit="kg"/>
    <defaultValues
      totalHeight="479.4" totalLength="378.4"
      totalWidth="378.4" bellyDiameter="378.4"
      internalDiameter="115.6" externalDiameter="161.0"
      footDiameter="3.56" heightLips="32.0" mass="11.0"/>
  </item>
  <item>
    ...
  </item>
</itemMesurableTypology>
```

Le champ *name* est utilisé en tant qu'identifiant. Les dimensions attendues et le poids sont groupés dans la balise *defaultValues*. Néanmoins, ces informations peuvent être exprimées dans différentes unités (m,mm,...). Nous décidons, pour les exemples, de considérer que toutes les informations de longueur sont exprimées dans une seule unité : le millimètre.

Le relevé

Nous décrivons ici les informations présentées dans le relevé du site de l'île de Pianosa. Le relevé présente, pour toutes les amphores du site, les informations mesurées sur place ainsi qu'un nom qui sert d'identifiant pour chaque amphore mesurée.

```
<survey>
  <amphora
    id="EntierE_1">
    <description
      subClass="Dressel2_4_longue"/>
    <metrology
      totalHeight="1.01" totalLength="0.30"
```

```

    totalWidth="0.30" internalDiameter="0.29"
    externalDiameter="0.089" footDiameter="0.08"
    bellyDiameter="0.31" heightLips="0.026" mass="25.5"/>
  <surveysessiondata
    dimensionUnit="m" massUnit="kg"/>
</amphora>
<amphora>
  ...
</amphora>
</survey>

```

Le champ *id* est utilisé comme identifiant. Le type supposé de l'amphore est décrit dans le champ *subClass* dans la balise *description*. Ce type doit être mis en relation avec le champ *name* donné dans la typologie. La balise *metrology* présente les dimensions de l'amphore telles qu'elles ont été mesurées sur le site. L'unité dans laquelle les informations sont représentées se situe dans le champ *surveysessiondata*.

7.2.0.5 Le module de raisonnement

L'objectif de cet outil est de fournir une assistance aux archéologues afin de détecter et de corriger les erreurs de mesure et de typologie dans la description des objets. Afin d'arriver à cet objectif, un pipeline de programmes est fourni, comme décrit dans la figure 7.7. Ces programmes sont basés sur la méthode de fusion par R-ensembles et adaptés à la détection d'incohérence dans le cadre de notre expérimentation.

Lors d'un relevé archéologique, de mauvaises conditions climatiques ou un emploi du temps trop court peuvent compliquer les mesures et plusieurs problèmes peuvent survenir. Les mesures peuvent être imprécises, l'archéologue peut avoir fait une erreur lors du choix du type des amphores ou les relevés peuvent avoir mal été transcrits. Toutes ces erreurs peuvent conduire à des incohérences entre le relevé et la typologie. Les mesures peuvent être en désaccord avec les valeurs attendues qui sont fournies par la typologie.

Pour toutes ces raisons, nous reformulons le problème de la manière suivante : est-ce que les dimensions du relevé sont en accord avec la typologie (au pourcentage de tolérance près) ? Le type associé avec l'amphore est-il judicieusement choisi ? Dans le cas contraire, nous pouvons utiliser la fusion par R-ensembles pour identifier la source de l'incohérence. D'abord, nous devons considérer les croyances venant de la typologie. Ces connaissances sont fournies par les archéologues et sont considérées comme sûres. En tant que telles, elles ne peuvent pas être remises en cause. Nous les considérons donc comme des contraintes. Puis, nous considérons ensuite les croyances provenant du relevé.

À partir de maintenant, nous notons $\mathcal{T}$ l'ensemble des types d'amphores. Nous notons $\mathcal{A}$ l'ensemble des amphores du relevé et nous notons $\mathcal{D}$ l'ensemble des attributs. Nous nous plaçons dans le cadre de la logique des prédicats instanciés.

Faire correspondre la typologie

Dans cette optique, nous avons à gérer deux types de contraintes :

- Une amphore ne peut avoir qu’un seul et unique type ;
- Les mesures du relevé doivent être en accord avec la typologie.

La contrainte d’unicité sur le type Comme nous l’avons dit, certaines croyances exprimées ne peuvent pas être remises en cause. Par exemple, chaque amphore doit être associée à un type et un seul. Une amphore ne peut être en même temps une Dressel 2-4 est une Beltran_2B.

Afin de répondre à cette question, nous créons, pour tout $x \in \mathcal{A}$ et pour tout $y \in \mathcal{T}$ le prédicat $type(x, y)$ qui représente le fait qu’une amphore x est de type y . La contrainte d’unicité est traitée grâce aux règles suivantes :

1. Pour tout $x \in \mathcal{A}$, pour tout $y \in \mathcal{T}$, pour tout $z \in \mathcal{T}$ tel que $y \neq z$, nous ajoutons la règle : $\neg type(x, y) \vee \neg type(x, z)$;
2. Pour tout $x \in \mathcal{A}$, pour tout $y_1, \dots, y_n \in \mathcal{T}$ où $y_1, \dots, y_n$ représentent tous les types connus d’amphores, nous ajoutons la règle : $type(x, y_1) \vee \dots \vee type(x, y_n)$.

La première règle exprime qu’une amphore ne peut pas posséder deux types d’amphores simultanément. La seconde partie exprime le fait qu’une amphore doit avoir au moins un type. Ces deux règles mises ensembles n’autorisent les amphores qu’à avoir exactement un type.

La contrainte d’intégrité sur les mesures Si une amphore est considérée comme étant une Dressel 2-4, alors les croyances sur les mesures issues du relevé doivent correspondre — à un pourcentage d’erreur près — aux valeurs attendues fournies par la typologie.

Pour tout $x \in \mathcal{D}$, pour tout $y \in \mathcal{T}$, pour tout $z \in \mathcal{A}$, le prédicat $cmp(x, y, z)$ représente le fait que les mesures de l’attribut x sur l’amphore z sont compatibles avec le type y tel qu’il est décrit dans la typologie.

Pour tout $x \in \mathcal{D}$, pour tout $y \in \mathcal{T}$, pour tout $z \in \mathcal{A}$, nous ajoutons la règle : $type(z, y) \rightarrow cmp(x, y, z)$.

Reconsidération des croyances du relevé

Le relevé fournit deux types de croyances pouvant être remises en cause : les dimensions et le type de l’amphore mesurée. Dans cette section, nous nous plaçons dans le cadre de l’hypothèse du monde clos — si une croyance n’est pas explicitement décrite, elle est considérée comme fausse. Par exemple, si le fait $cmp(TotalHeight, Beltran_2B, Entiere_3)$ est présent dans le programme logique, alors les mesures de l’attribut $TotalHeight$ de l’amphore $Entiere_3$ ne correspondent pas avec les valeurs attendues pour le type $Beltran_2B$; si ce fait ne fait pas partie du programme, alors ces mêmes mesures ne sont pas compatibles.

Vérification des dimensions La logique des prédicats et la programmation logique ne sont pas adéquates pour le traitement des valeurs numériques. Il est donc né-

cessaire d'effectuer un prétraitement afin de pouvoir traiter les informations numériques. C'est dans cette optique que nous utilisons le prédicat $cmp(x, y, z)$.

Pour tout $x \in \mathcal{D}$, pour tout $y \in \mathcal{T}$, pour tout $z \in \mathcal{A}$, le prédicat $cmp(x, y, z)$ est ajouté au programme logique si les mesures de l'attribut x sur l'amphore z sont compatibles avec le type y tel qu'il est décrit dans la typologie — au pourcentage de tolérance près.

Il est important de préciser qu'un attribut peut être compatible avec plusieurs types simultanément. Il peut également n'être compatible avec aucun type.

Vérification du type Un type est associé à chaque amphore. Ainsi, pour tout $x \in \mathcal{A}$, pour tout $y \in \mathcal{T}$, si l'amphore x est déclaré comme étant du type y , nous ajoutons la clause unaire : $type(x, y)$.

Exemple 7.2.1. Nous présentons ici un exemple incluant trois amphores (E_1, E_2 et E_3) et une typologie présentant deux types différents ($Dressel_2_4$ and $Beltran_2B$) et limitée à deux attributs ($TotalHeight$ and $Mass$).

La typologie

Nous exprimons d'abord la contrainte d'unicité de type.

$$K_{T_1} = \left\{ \begin{array}{l} \neg type(E_1, Dressel_2_4) \vee \neg type(E_1, Beltran_2B) \\ type(E_1, Dressel_2_4) \vee type(E_1, Beltran_2B) \\ \neg type(E_2, Dressel_2_4) \vee \neg type(E_2, Beltran_2B) \\ type(E_2, Dressel_2_4) \vee type(E_2, Beltran_2B) \\ \neg type(E_3, Dressel_2_4) \vee \neg type(E_3, Beltran_2B) \\ type(E_3, Dressel_2_4) \vee type(E_3, Beltran_2B) \end{array} \right\}$$

Nous introduisons ensuite la règle vérifiant que les mesures du relevé respectent le type tel qu'il est décrit dans la typologie.

$$K_{T_2} = \left\{ \begin{array}{l} type(E_1, Dressel_2_4) \rightarrow cmp(totalHeight, Dressel_2_4, E_1) \\ type(E_1, Dressel_2_4) \rightarrow cmp(Mass, Dressel_2_4, E_1) \\ type(E_1, Beltran_2B) \rightarrow cmp(totalHeight, Beltran_2B, E_1) \\ type(E_1, Beltran_2B) \rightarrow cmp(Mass, Beltran_2B, E_1) \\ type(E_2, Dressel_2_4) \rightarrow cmp(totalHeight, Dressel_2_4, E_2) \\ type(E_2, Dressel_2_4) \rightarrow cmp(Mass, Dressel_2_4, E_2) \\ type(E_2, Beltran_2B) \rightarrow cmp(totalHeight, Beltran_2B, E_2) \\ type(E_2, Beltran_2B) \rightarrow cmp(Mass, Beltran_2B, E_2) \\ type(E_3, Dressel_2_4) \rightarrow cmp(totalHeight, Dressel_2_4, E_3) \\ type(E_3, Dressel_2_4) \rightarrow cmp(Mass, Dressel_2_4, E_3) \\ type(E_3, Beltran_2B) \rightarrow cmp(totalHeight, Beltran_2B, E_3) \\ type(E_3, Beltran_2B) \rightarrow cmp(Mass, Beltran_2B, E_3) \end{array} \right\}$$

Le relevé

Le type de l'amphore est tout d'abord déclaré.

$$K_{RV_1} = \left\{ \begin{array}{l} type(E_1, Dressel_2_4) \\ type(E_2, Dressel_2_4) \\ type(E_3, Beltran_2B) \end{array} \right\}$$

Ensuite, la compatibilité entre les types et les mesures est présentée.

$$K_{RV_2} = \left\{ \begin{array}{l} \text{cmp}(\text{totalHeight}, \text{Dressel_2_4}, E_1) \\ \text{cmp}(\text{Mass}, \text{Dressel_2_4}, E_1) \\ \text{cmp}(\text{totalHeight}, \text{Dressel_2_4}, E_2) \\ \text{cmp}(\text{Mass}, \text{Beltran_2B}, E_2) \\ \text{cmp}(\text{totalHeight}, \text{Beltran_2B}, E_3) \end{array} \right\}$$

Les compatibilités impliquent les non-compatibilités suivantes.

$$K_{RV_3} = \left\{ \begin{array}{l} \neg \text{cmp}(\text{totalHeight}, \text{Beltran_2B}, E_1) \\ \neg \text{cmp}(\text{Mass}, \text{Beltran_2B}, E_1) \\ \neg \text{cmp}(\text{totalHeight}, \text{Beltran_2B}, E_2) \\ \neg \text{cmp}(\text{Mass}, \text{Dressel_2_4}, E_2) \\ \neg \text{cmp}(\text{totalHeight}, \text{Dressel_2_4}, E_3) \\ \neg \text{cmp}(\text{totalHeight}, \text{Dressel_2_4}, E_3) \\ \neg \text{cmp}(\text{Mass}, \text{Dressel_2_4}, E_3) \end{array} \right\}$$

Dans ce cas, la stratégie la plus judicieuse semble être la stratégie Σ . Les R -ensembles de ce problème selon la stratégie Σ sont $\{\neg \text{cmp}(\text{Mass}, \text{Dressel_2_4}, E_2)\}$. Nous pouvons donc déduire que la masse mesurée pour l'amphore E_2 ne correspond pas à la valeur attendue pour son type.

La traduction du problème

Une fois que le problème a été correctement définie en logique des prédicats instanciés, il doit être traduit dans le cadre de la Fusion par R -ensembles en un programme logique avec sémantique des modèles stables. Cette traduction est :

$$\Pi_{T_1} = \left\{ \begin{array}{l} \leftarrow \text{not type}(E_1, \text{Dressel_2_4}), \text{not type}(E_1, \text{Beltran_2B}) \\ \leftarrow \text{type}(E_1, \text{Dressel_2_4}), \text{type}(E_1, \text{Beltran_2B}) \\ \leftarrow \text{not type}(E_2, \text{Dressel_2_4}), \text{not type}(E_2, \text{Beltran_2B}) \\ \leftarrow \text{type}(E_2, \text{Dressel_2_4}), \text{type}(E_2, \text{Beltran_2B}) \\ \leftarrow \text{not type}(E_3, \text{Dressel_2_4}), \text{not type}(E_3, \text{Beltran_2B}) \\ \leftarrow \text{type}(E_3, \text{Dressel_2_4}), \text{type}(E_3, \text{Beltran_2B}) \end{array} \right\}$$

$$\Pi_{T_2} = \left\{ \begin{array}{l} \leftarrow \text{type}(E_1, \text{Dressel_2_4}), \text{not cmp}(\text{totalHeight}, \text{Dressel_2_4}, E_1) \\ \leftarrow \text{type}(E_1, \text{Beltran_2B}), \text{not cmp}(\text{totalHeight}, \text{Beltran_2B}, E_1) \\ \leftarrow \text{type}(E_1, \text{Dressel_2_4}), \text{not cmp}(\text{Mass}, \text{Dressel_2_4}, E_1) \\ \leftarrow \text{type}(E_1, \text{Beltran_2B}), \text{not cmp}(\text{Mass}, \text{Beltran_2B}, E_1) \\ \leftarrow \text{type}(E_2, \text{Dressel_2_4}), \text{not cmp}(\text{totalHeight}, \text{Dressel_2_4}, E_2) \\ \leftarrow \text{type}(E_2, \text{Beltran_2B}), \text{not cmp}(\text{totalHeight}, \text{Beltran_2B}, E_2) \\ \leftarrow \text{type}(E_2, \text{Dressel_2_4}), \text{not cmp}(\text{Mass}, \text{Dressel_2_4}, E_2) \\ \leftarrow \text{type}(E_2, \text{Beltran_2B}), \text{not cmp}(\text{Mass}, \text{Beltran_2B}, E_2) \\ \leftarrow \text{type}(E_3, \text{Dressel_2_4}), \text{not cmp}(\text{totalHeight}, \text{Dressel_2_4}, E_3) \\ \leftarrow \text{type}(E_3, \text{Beltran_2B}), \text{not cmp}(\text{totalHeight}, \text{Beltran_2B}, E_3) \\ \leftarrow \text{type}(E_3, \text{Dressel_2_4}), \text{not cmp}(\text{Mass}, \text{Dressel_2_4}, E_3) \\ \leftarrow \text{type}(E_3, \text{Beltran_2B}), \text{not cmp}(\text{Mass}, \text{Beltran_2B}, E_3) \end{array} \right\}$$

Nous donnons maintenant les connaissances qui peuvent être remises en cause. Si une des connaissances représentées ci-dessous est remise en cause, elle déclenchera la déduction d'un atome r_i qui nous permettra, lors de l'analyse de la solution de voir ce qui a posé problème.

$$\Pi_{RV_1} = \left\{ \begin{array}{l} r_1 \leftarrow \text{type}(E_1, \text{Dressel_2_4})' \\ r_2 \leftarrow \text{type}(E_2, \text{Dressel_2_4})' \\ r_3 \leftarrow \text{type}(E_3, \text{Beltran_2B})' \end{array} \right\}$$

$$\Pi_{RV_2} = \left\{ \begin{array}{l} r_4 \leftarrow \text{cmp}(\text{totalHeight}, \text{Dressel_2_4}, E_1)' \\ r_5 \leftarrow \text{cmp}(\text{Mass}, \text{Dressel_2_4}, E_1)' \\ r_6 \leftarrow \text{cmp}(\text{totalHeight}, \text{Dressel_2_4}, E_3)' \\ r_7 \leftarrow \text{cmp}(\text{Mass}, \text{Dressel_2_4}, E_3)' \\ r_8 \leftarrow \text{cmp}(\text{totalHeight}, \text{Beltran_2B}, E_3)' \end{array} \right\}$$

$$\Pi_{RV_3} = \left\{ \begin{array}{l} r_9 \leftarrow \text{cmp}(\text{totalHeight}, \text{Beltran_2B}, E_1) \\ r_{10} \leftarrow \text{cmp}(\text{Mass}, \text{Beltran_2B}, E_1) \\ r_{11} \leftarrow \text{cmp}(\text{totalHeight}, \text{Beltran_2B}, E_2) \\ r_{12} \leftarrow \text{cmp}(\text{Mass}, \text{Dressel_2_4}, E_2) \\ r_{13} \leftarrow \text{cmp}(\text{totalHeight}, \text{Dressel_2_4}, E_3) \\ r_{14} \leftarrow \text{cmp}(\text{totalHeight}, \text{Dressel_2_4}, E_3) \\ r_{15} \leftarrow \text{cmp}(\text{Mass}, \text{Dressel_2_4}, E_3) \end{array} \right\}$$

L'ensemble des interprétations sur les compatibilités est obtenu grâce aux règles suivantes :

$$\begin{aligned} & \text{cmp}(\text{totalHeight}, \text{Dressel_2_4}, E_1) \leftarrow \text{not cmp}(\text{totalHeight}, \text{Dressel_2_4}, E_1)' \\ & \text{cmp}(\text{totalHeight}, \text{Dressel_2_4}, E_1)' \leftarrow \text{not cmp}(\text{totalHeight}, \text{Dressel_2_4}, E_1) \\ & \text{cmp}(\text{Mass}, \text{Dressel_2_4}, E_1) \leftarrow \text{not cmp}(\text{Mass}, \text{Dressel_2_4}, E_1)' \\ & \text{cmp}(\text{Mass}, \text{Dressel_2_4}, E_1)' \leftarrow \text{not cmp}(\text{Mass}, \text{Dressel_2_4}, E_1) \\ & \text{cmp}(\text{totalHeight}, \text{Dressel_2_4}, E_2) \leftarrow \text{not cmp}(\text{totalHeight}, \text{Dressel_2_4}, E_2)' \\ & \text{cmp}(\text{totalHeight}, \text{Dressel_2_4}, E_2)' \leftarrow \text{not cmp}(\text{totalHeight}, \text{Dressel_2_4}, E_2) \\ & \text{cmp}(\text{Mass}, \text{Dressel_2_4}, E_2) \leftarrow \text{not cmp}(\text{Mass}, \text{Dressel_2_4}, E_2)' \\ & \text{cmp}(\text{Mass}, \text{Dressel_2_4}, E_2)' \leftarrow \text{not cmp}(\text{Mass}, \text{Dressel_2_4}, E_2) \\ & \text{cmp}(\text{totalHeight}, \text{Dressel_2_4}, E_3) \leftarrow \text{not cmp}(\text{totalHeight}, \text{Dressel_2_4}, E_3)' \\ & \text{cmp}(\text{totalHeight}, \text{Dressel_2_4}, E_3)' \leftarrow \text{not cmp}(\text{totalHeight}, \text{Dressel_2_4}, E_3) \\ & \text{cmp}(\text{Mass}, \text{Dressel_2_4}, E_3) \leftarrow \text{not cmp}(\text{Mass}, \text{Dressel_2_4}, E_3)' \\ & \text{cmp}(\text{Mass}, \text{Dressel_2_4}, E_3)' \leftarrow \text{not cmp}(\text{Mass}, \text{Dressel_2_4}, E_3) \\ & \text{cmp}(\text{totalHeight}, \text{Beltran_2B}, E_1) \leftarrow \text{not cmp}(\text{totalHeight}, \text{Beltran_2B}, E_1)' \\ & \text{cmp}(\text{totalHeight}, \text{Beltran_2B}, E_1)' \leftarrow \text{not cmp}(\text{totalHeight}, \text{Beltran_2B}, E_1) \\ & \text{cmp}(\text{Mass}, \text{Beltran_2B}, E_1) \leftarrow \text{not cmp}(\text{Mass}, \text{Beltran_2B}, E_1)' \\ & \text{cmp}(\text{Mass}, \text{Beltran_2B}, E_1)' \leftarrow \text{not cmp}(\text{Mass}, \text{Beltran_2B}, E_1) \\ & \text{cmp}(\text{totalHeight}, \text{Beltran_2B}, E_2) \leftarrow \text{not cmp}(\text{totalHeight}, \text{Beltran_2B}, E_2)' \\ & \text{cmp}(\text{totalHeight}, \text{Beltran_2B}, E_2)' \leftarrow \text{not cmp}(\text{totalHeight}, \text{Beltran_2B}, E_2) \\ & \text{cmp}(\text{Mass}, \text{Beltran_2B}, E_2) \leftarrow \text{not cmp}(\text{Mass}, \text{Beltran_2B}, E_2)' \\ & \text{cmp}(\text{Mass}, \text{Beltran_2B}, E_2)' \leftarrow \text{not cmp}(\text{Mass}, \text{Beltran_2B}, E_2) \\ & \text{cmp}(\text{totalHeight}, \text{Beltran_2B}, E_3) \leftarrow \text{not cmp}(\text{totalHeight}, \text{Beltran_2B}, E_3)' \\ & \text{cmp}(\text{totalHeight}, \text{Beltran_2B}, E_3)' \leftarrow \text{not cmp}(\text{totalHeight}, \text{Beltran_2B}, E_3) \\ & \text{cmp}(\text{Mass}, \text{Beltran_2B}, E_3) \leftarrow \text{not cmp}(\text{Mass}, \text{Beltran_2B}, E_3)' \\ & \text{cmp}(\text{Mass}, \text{Beltran_2B}, E_3)' \leftarrow \text{not cmp}(\text{Mass}, \text{Beltran_2B}, E_3) \end{aligned}$$

L'ensemble des interprétations sur les types est obtenu grâce aux règles suivantes :

```

type(E_1, Beltran_2B) ← not type(E_1, Beltran_2B)'
type(E_1, Beltran_2B)' ← not type(E_1, Beltran_2B)
type(E_2, Beltran_2B) ← not type(E_2, Beltran_2B)'
type(E_2, Beltran_2B)' ← not type(E_2, Beltran_2B)
type(E_3, Beltran_2B) ← not type(E_3, Beltran_2B)'
type(E_3, Beltran_2B)' ← not type(E_3, Beltran_2B)
type(E_1, Dressel_2_4) ← not type(E_1, Dressel_2_4)'
type(E_1, Dressel_2_4)' ← not type(E_1, Dressel_2_4)
type(E_2, Dressel_2_4) ← not type(E_2, Dressel_2_4)'
type(E_2, Dressel_2_4)' ← not type(E_2, Dressel_2_4)
type(E_3, Dressel_2_4) ← not type(E_3, Dressel_2_4)'
type(E_3, Dressel_2_4)' ← not type(E_3, Dressel_2_4)

```

Ce programme logique permet de calculer les modèles stables préférés d'après la stratégie *Somme*.

La solution Nous sommes en présence d'un profil de croyances constitué d'une base de croyances unique. Pour cette raison, la stratégie choisie est indifférente. Il y a cinq cent douze R-ensembles potentiels qui restaurent la cohérence, mais il n'y a qu'un seul modèle stable préféré. Ainsi, la solution de ce problème est :

```

Stable Model: r_12 cmp(mass,Dressel_2_4,E_2) cmp(totalHeight,Dressel_2_4,E_1)
cmp(mass,Dressel_2_4,E_1) cmp(totalHeight,Dressel_2_4,E_2)
cmp(totalHeight,Dressel_2_4,E_3)' cmp(mass,Dressel_2_4,E_3)'
cmp(totalHeight,py_4,E_1)' cmp(mass,py_4,E_1)' cmp(totalHeight,py_4,E_2)'
cmp(mass,py_4,E_2) cmp(totalHeight,py_4,E_3) cmp(mass,py_4,E_3)
type2(E_1,py_4) type2(E_2,py_4) type(E_3,py_4) type(E_1,Dressel_2_4)
type(E_2,Dressel_2_4) type2(E_3,Dressel_2_4)

```

Les atomes significatifs dans un résultat sont les r_i . Dans notre résultat, il s'agit de l'atome r_{12} qui est déclenché par l'atome $cmp(Mass, Dressel_2_4, E_2)$. Ainsi, le résultat exprime que c'est l'incompatibilité de la masse de l'amphore E_2 qui pose problème dans ce relevé. Ainsi, les archéologues doivent reconsidérer la mesure de cette amphore.

7.2.1 Experimentations et résultats

7.2.1.1 En fonction du nombre d'amphores

Nous conduisons une expérimentation afin d'observer l'évolution du temps d'exécution de notre mise œuvre en fonction du nombre d'amphores. Nous choisissons donc un ensemble, compris entre 1 et 40 amphores, issues du relevé de Pianosa, lançons notre module de raisonnement et comparons les temps d'exécution en fonction du nombre d'amphores.

Le relevé de Pianosa comporte 40 amphores, la typologie qui nous a été fournie par les archéologues comporte 12 amphores et la description des types porte sur 9 champs. Le problème a généré un programme comportant 8462 formules avec 4160 atomes.

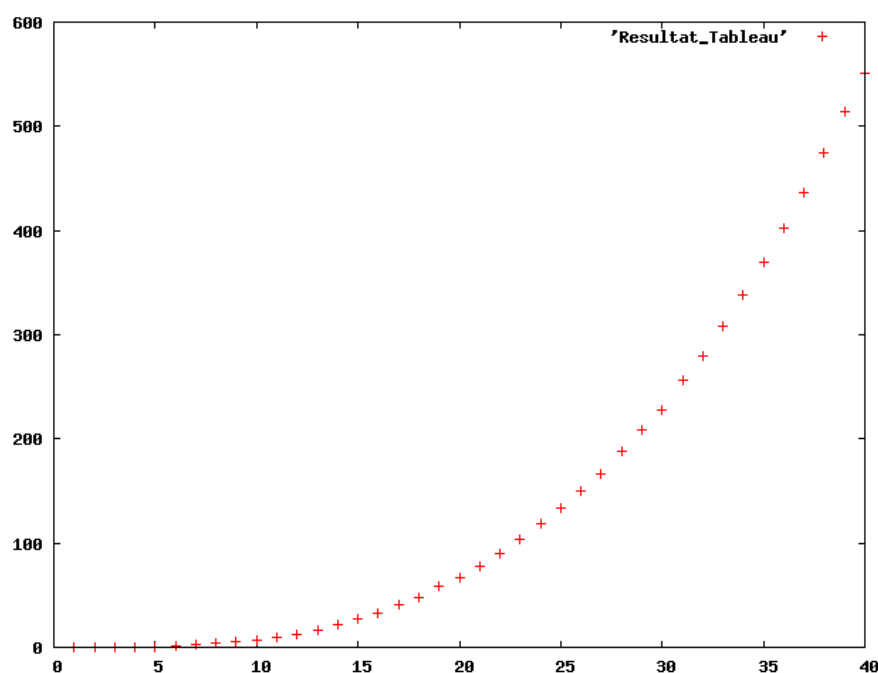


FIGURE 7.8 – Temps d'exécution ,en millisecondes, par rapport au nombre d'amphores

Un programme logique aléatoire de taille comparable serait résolu dans un temps astronomique. Cela prouve que la structuration des données d'un problème permet de réduire drastiquement la difficulté d'un problème. Étant donné le nombre de clauses et le nombre d'atomes, les temps d'exécution sont plutôt bons.

Néanmoins, le problème est cloisonné et se comporte comme l'union de problèmes indépendants les uns des autres. De plus, la plupart des clauses sont des clauses binaires ou ternaires.

7.2.1.2 En fonction du pourcentage d'erreur toléré

Nous conduisons une expérimentation afin d'observer le comportement de notre mise en œuvre de la fusion par R-ensembles en fonction de la marge d'erreur toléré. Nous conduisons cette expérimentation sur le relevé issu de Pianosa mais également sur le site de Port Miou où plus de 500 amphores ont été relevées.

Les temps d'exécution sont déclinés pour chacun des modules de notre mise en œuvre : le module de traduction, le module de raisonnement et le module d'écriture des résultats sous forme XML.

Nous pouvons voir que les temps d'exécution sont relativement stables lorsque la marge d'erreur varie. La réduction de la marge d'erreur conduit même à une légère réduction du temps d'exécution. La différence est plus marquée, par contre, dans le

%erreur	$T1(traducteur)$	$T2(raisonnement)$	$T3(XML)$	$T1 + T2 + T3$
20	50	620	950	1620
10	50	600	640	1290
5	50	610	450	1110
1	50	600	330	980

FIGURE 7.9 – Temps d'exécution, en millisecondes, en fonction de l'erreur pour le site de Pianosa

%erreur	$T1(traducteur)$	$T2(raisonnement)$	$T3(XML)$	$T1 + T2 + T3$
20	430	5260	140	5830
10	430	5060	40	5530
5	430	4990	0	5420
1	430	5060	0	5490

FIGURE 7.10 – Temps d'exécution, en millisecondes, en fonction de l'erreur pour le site de Pianosa

module d'écriture des résultats sous forme XML.

Conclusion

Nous avons présenté une série de tests aléatoires mettant en valeur le bon comportement de la méthode de fusion par R-ensemble. Les résultats montrent que notre mise en œuvre offre globalement de meilleurs résultats que la méthode basée sur les Diagrammes de Décisions Binaires (ou BDD) de N.Gorogiannis et A. Hunter.

Nous avons également présenté une adaptation de la méthode de fusion par R-ensembles à la recherche d'incohérences dans les relevés d'archéologie sous-marine du projet européen VENUS. La méthode s'avère être adaptée au besoin et les temps d'exécution tout a fait compatibles avec des problèmes de taille réelle. Pour plus de détails, voir les livrables [PWJ⁺08, PWC⁺09]

Chapitre 8

Fusion par R-ensembles en présence de préférences

Trop de connaissances ne facilitent pas les plus simples décisions *Frank Herbert, Les enfants de Dune*

Introduction

Les profils de croyances ne se situent pas toujours dans un cadre où toutes les croyances sont sur un pied d'égalité. Dans certains cas, des préférences sont exprimées parmi les bases de croyances à fusionner : certaines croyances proviennent de sources plus sûres ou plus récentes. Dans d'autres cas, des préférences sont exprimées entre les croyances à l'intérieur même d'une base de croyances : la source peut par exemple considérer certaines croyances comme étant plus sûres que d'autres. Nous proposons ici d'étudier la fusion de croyances dans le cas où des préférences sont exprimées entre les bases de croyances, ainsi que dans le cas où des préférences sont exprimées entre les bases de croyances et à l'intérieur des bases de croyances. Nous traitons ces cas en créant un pré-ordre total représentant les préférences entre les formules. Le cas où seul des préférences à l'intérieur des bases de croyances sont exprimées revient à un problème où les préférences sont représentées à l'aide d'un pré-ordre partiel, pour ce cas, nous renvoyons le lecteur à [Lag03].

8.1 Préférences externes uniquement

Nous nous limitons, dans cette section au cas où les préférences sont exprimées seulement sur les bases de croyances, que nous appelons préférence externe, et où il n'existe pas de préférences exprimées entre les formules.

Dans ce cas, le problème peut se résoudre de deux manières différentes :

- nous pouvons le considérer comme un problème de révision itérée [DDL06] ;

- nous pouvons également le considérer comme un problème de fusion classique pour lequel nous employons une stratégie basée sur l'ordre lexicographique des bases.

Nous allons commencer par la deuxième proposition avant de présenter la fusion comme une opération de révision itérée.

8.1.1 Fusion classique avec opérateur lexicographique

L'objectif ici est donc d'obtenir une base de croyances en retirant des formules issues de $K_1 \sqcup \dots \sqcup K_n$ afin de restaurer la cohérence. Le retrait de ces formules doit respecter les préférences exprimées sur les bases.

Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances sur lequel portent les préférences suivantes : $K_1 < K_2 < K_3 < \dots < K_n$. Pour le reste de ce chapitre, nous considérons que les bases de croyances sont organisées selon cet ordre. Il s'agit d'une convention de nommage et tous les cas peuvent se ramener à celui-là par permutation. Afin de respecter les préférences, les formules doivent être retirées dans les bases situées en bas de la hiérarchie. Nous proposons donc d'organiser les R-ensembles potentiels selon le pré-ordre suivant :

Définition 8.1.1. *Pour chaque R-ensemble potentiel X et chaque base de croyances K_i , nous définissons $p_X^i = |X \cap K_i|$. Nous considérons donc la séquence $(p_X^1, \dots, p_X^n)$ du R-ensemble potentiel X sur le profil de croyances E .*

Nous représentons le pré-ordre lexicographique par le symbole $\leq_{lex}$.

Définition 8.1.2 (Pré-ordre $\leq_{lexipref}$). *Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. Soient X et X' deux R-ensembles potentiels de E . Le pré-ordre $\leq_{lexipref}$ est défini de la manière suivante :*

$$X \leq_{lexipref} X' \text{ si et seulement si } (p_X^1, \dots, p_X^n) \leq_{lex} (p_{X'}^1, \dots, p_{X'}^n).$$

Nous définissons ainsi le R-ensemble de E contraint par IC selon la stratégie *lexipref* de la manière suivante :

Définition 8.1.3 (R-ensemble selon *lexipref*). *Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un ensemble de formules représentant les contraintes pesant sur l'opération tel que $K_1 \sqcup \dots \sqcup K_n \sqcup IC$ est incohérent. X est un R-ensemble de E contraint par IC selon la stratégie *lexipref* si et seulement si :*

- X est un R-ensemble potentiel de E contraint par IC ;
- Il n'existe pas de R-ensemble potentiel $X' \subseteq K_1 \sqcup \dots \sqcup K_n$ tel que $X' \subset X$.
- Il n'existe pas de R-ensemble potentiel $X' \subseteq K_1 \sqcup \dots \sqcup K_n$ tel que $(p_{X'}^1, \dots, p_{X'}^n) <_{lex} (p_X^1, \dots, p_X^n)$.

Cela conduit à l'opération de fusion suivante :

Définition 8.1.4 (Fusion selon *lexipref*). *Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un ensemble de formules représentant les contraintes. Soit $\mathcal{F}_{lexipref, IC} \mathcal{R}(E)$ l'ensemble des R-ensembles de E selon *lexipref*. L'opération de fusion est définie comme*

suit :

$$\Delta_{lexipref, IC}^{RSF}(E) = \bigvee_{X \in \mathcal{F}_{lexipref, IC} \mathcal{R}(E)} \{((K_1 \sqcup \dots \sqcup K_n) \setminus X) \sqcup IC\}.$$

Dans le cas de la fusion pure, où $IC = \top$, nous notons indifféremment $\Delta_{lexipref, \top}^{RSF}(E)$ ou $\Delta_{lexipref}^{RSF}(E)$.

Exemple 8.1.1. *Considérons le profil de croyances $E = \{K_1, K_2, K_3\}$ tel que $K_1 < K_2 < K_3$ et $IC = \top$ avec :*

$$K_1 = \{\neg e\}$$

$$K_2 = \{e \vee \neg b, e \vee \neg d\}$$

$$K_3 = \{\neg a \vee b, a, \neg c \vee d, c\}$$

Avec un opérateur de fusion classique, Σ par exemple, le profil de croyances E posséderait un seul R -ensemble : $\mathcal{F}_{\Sigma}^{RSF} \mathcal{R}(E) = \{\{\neg e\}\}$. Ce R -ensemble ne tient pas compte des préférences en retirant des formules parmi la base préférée.

Au contraire, les R -ensembles de $\mathcal{F}_{lexipref}^{RSF} \mathcal{R}(E) = \{\{a, c\}, \{a, \neg c \vee d\}, \{\neg a \vee b, c\}, \{\neg a \vee b, \neg c \vee d\}\}$ restaurent la cohérence en épargnant les bases les mieux placées dans la hiérarchie des préférences.

Le résultat de la fusion par R -ensembles selon $lexipref$ est $\Delta_{lexipref, IC}^{RSF}(E) = \{\neg e, e \vee \neg b, e \vee \neg d, \neg a \vee b, \neg c \vee d\} \vee \{\neg e, e \vee \neg b, e \vee \neg d, \neg a \vee b, c\} \vee \{\neg e, e \vee \neg b, e \vee \neg d, a, \neg c \vee d\} \vee \{\neg e, e \vee \neg b, e \vee \neg d, a, c\}$.

8.1.2 Fusion comme révision itérée

Une autre manière de voir l'opération de fusion en présence de préférences est de la considérer comme étant une opération de révision itérée [DDL06].

De manière naïve, nous pourrions définir l'opération de la manière suivante. La base de croyances la moins préférée, considérée moins fiable, est révisée par la base de croyances située juste au-dessus d'elle dans la hiérarchie. Puis le résultat de cette opération est révisé par la base au-dessus dans la hiérarchie, etc. jusqu'aux contraintes. Cette opération de fusion, notée $\Delta_{\alpha, IC}^{PRSF}(E)$, est définie à l'aide de l'opération de révision par R -ensembles $\circ_{RSR}$ que nous reformulons dans le cas de bases de croyances constituées de formules quelconques. Nous notons $\mathcal{R}(K \cup A)$ l'ensemble des R -ensembles de cette opération.

Définition 8.1.5 (Révision par R -ensembles). *Soient K une base de croyances et A une formule. La révision de K par A est définie comme :*

$$K \circ_{RSR} A = \bigvee_{R \in \mathcal{R}(K \cup A)} \{(K \cup A) \setminus R\}.$$

Nous définissons donc la fusion de croyances en présence de préférences comme une suite d'opérations de révision de la manière suivante :

Définition 8.1.6. Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un ensemble de formules représentant les contraintes.

$$\Delta_{\alpha, IC}^{PRSF}(E) = (((\dots (K_n \circ_{RSR} K_{n-1}) \circ_{RSR} \dots \circ_{RSR} K_2) \circ_{RSR} K_1) \circ_{RSR} IC).$$

Dans le cas de la fusion pure, où $IC = \top$, nous notons indifféremment $\Delta_{\alpha, \top}^{PRSF}(E)$ ou $\Delta_{\alpha}^{PRSF}(E)$. Cependant, cet opérateur ne se comporte pas de manière satisfaisante, reprenons l'exemple 8.1.1.

Exemple 8.1.2. L'opération de fusion se déroule ainsi en deux étapes :

$$K_3 \circ_{RSR} K_2 = K_2 \sqcup K_3$$

puis

$$\begin{aligned} (K_3 \circ_{RSR} K_2) \circ_{RSR} K_1 = & \{ \neg e, \neg a \vee b, a, \neg c \vee d, c \} \vee \\ & \{ \neg e, e \vee \neg d, \neg a \vee b, a, c \} \vee \\ & \{ \neg e, e \vee \neg d, \neg a \vee b, a, c \} \vee \\ & \{ \neg e, e \vee \neg b, a, \neg c \vee d, c \} \vee \\ & \{ \neg e, e \vee \neg b, e \vee \neg d, a, c \} \vee \\ & \{ \neg e, e \vee \neg b, e \vee \neg d, a, \neg c \vee d \} \vee \\ & \{ \neg e, e \vee \neg b, \neg a \vee b, \neg c \vee d, c \} \vee \\ & \{ \neg e, e \vee \neg b, e \vee \neg d, \neg a \vee b, c \} \vee \\ & \{ \neg e, e \vee \neg b, e \vee \neg d, \neg a \vee b, \neg c \vee d \}. \end{aligned}$$

Nous voyons que le résultat est différent de celui décrit dans l'exemple 8.1.1. Cela est dû au fait qu'à chaque étape, l'origine des formules est perdue dans le processus et les formules provenant de K_2 sont traitées de la même manière que celles provenant de K_3 .

Pour cette raison, il est préférable d'appliquer les opérateurs de révision dans un ordre différent pour réaliser cette opération de fusion. Nous commençons par réviser la base de croyances préférée par les contraintes. Puis nous révisons la deuxième base de la hiérarchie par le résultat de la première opération, etc. jusqu'à la base la plus basse dans la hiérarchie. De cette manière, nous décrivons l'opération $\Delta_{\beta, IC}^{PRSF}(E)$.

Définition 8.1.7. Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un ensemble de formules représentant les contraintes.

$$\Delta_{\beta, IC}^{PRSF}(E) = (K_n \circ_{RSR} (K_{n-1} \circ_{RSR} \dots \circ_{RSR} (K_2 \circ_{RSR} (K_1 \circ_{RSR} IC)) \dots)).$$

Dans le cas de la fusion pure, où $IC = \top$, nous notons indifféremment $\Delta_{\beta, \top}^{PRSF}(E)$ ou $\Delta_{\beta}^{PRSF}(E)$. Cette opération est conforme aux attentes. Ainsi, comme l'illustre 8.1.1.

Exemple 8.1.3. L'opération se déroule en deux étapes :

$$K_2 \circ_{RSR} K_1 = K_1 \sqcup K_2$$

puis

$$\begin{aligned} K_3 \circ_{RSR} (K_2 \circ_{RSR} K_1) &= \{\neg e, e \vee \neg b, e \vee \neg d, a, c\} \\ &\vee \{\neg e, e \vee \neg b, e \vee \neg d, a, \neg c \vee d\} \\ &\vee \{\neg e, e \vee \neg b, e \vee \neg d, \neg a \vee b, c\} \\ &\vee \{\neg e, e \vee \neg b, e \vee \neg d, \neg a \vee b, \neg c \vee d\}. \end{aligned}$$

Nous constatons que les résultats ici sont identiques à ceux décrits dans 8.1.1.

8.1.2.1 Equivalence entre $\Delta_{\beta, IC}^{PRSF}(E)$ et $\Delta_{lexipref, IC}^{RSF}(E)$

Nous montrons maintenant que ces opérations de fusion $\Delta_{\beta, IC}^{PRSF}(E)$ et $\Delta_{lexipref, IC}^{RSF}(E)$ en présence de préférences conduisent en fait au même résultat.

Proposition 8.1.1. *Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances tel que $K_1 < \dots < K_n$ et soit IC un ensemble de formules représentant les contraintes.*

$$\Delta_{\beta, IC}^{PRSF}(E) = \Delta_{lexipref, IC}^{RSF}(E).$$

Preuve.

Nous rappelons que pour l'opération $K \circ_{RSR} A$, l'ensemble des R-ensembles de $K \cup A$ est noté $\mathcal{R}(K \cup A)$. Dans la suite, pour l'opération de révision itérée $(K_n \circ_{RSR} \dots (K_1 \circ_{RSR} IC))$ nous notons $\mathcal{R}(K_n \cup \dots \cup K_1 \cup IC)$ l'ensemble des R-ensembles correspondant. Nous rappelons que $\mathcal{F}_{\beta, IC}^{PRSF}(E) = \cup_{X \in \mathcal{R}(E)} \mathcal{R}(E \setminus X)$

$$\boxed{\Delta_{\beta, IC}^{PRSF}(E) \subseteq \Delta_{lexipref, IC}^{RSF}(E)}$$

Supposons qu'il existe $F \in \mathcal{F}_{\beta, IC}^{PRSF}(E)$ tel que $F \notin \mathcal{F}_{lexipref, IC}^{RSF}(E)$. Nous notons X le R-ensemble correspondant à F . Cela peut avoir trois explications :

1. $(E \setminus X) \sqcup IC$ est incohérent ;
2. Il existe X' tel que $X' \subseteq X$ et $(E \setminus X') \sqcup IC$ est cohérent ;
3. Il existe X' tel que $X' <_{lexipref} X$ et $(E \setminus X') \sqcup IC$ est cohérent.

Nous allons montrer qu'aucune de ces alternatives n'est compatible avec les hypothèses.

1. F est un élément du flock de $\mathcal{F}_{\beta, IC}^{PRSF}(E)$, donc $(E \setminus X) \sqcup IC$ est cohérent.
2. Supposons que $X \cap (K_1 \cup \dots \cup K_i) = X' \cap (K_1 \cup \dots \cup K_i)$ et que $(X' \cap K_{i+1}) \subset (X \cap K_{i+1})$ cela signifierait que $X \cap K_{i+1} \notin \mathcal{R}((K_{i+1} \cup \dots \cup (K_1 \cup IC)))$, ce qui est impossible par hypothèse.
3. Supposons que $X \cap (K_1 \cup \dots \cup K_i) =_{lexipref} X' \cap (K_1 \cup \dots \cup K_i)$ et que $(X' \cap K_{i+1}) <_{lexipref} (X \cap K_{i+1})$ (c'est à dire que $|(X' \cap K_{i+1})| < |(X \cap K_{i+1})|$) cela signifierait que $X \cap K_{i+1} \notin \mathcal{R}((K_{i+1} \cup \dots \cup (K_1 \cup IC)))$, ce qui est impossible par hypothèse.

$$\boxed{\Delta_{lexipref, IC}^{RSF}(E) \subseteq \Delta_{\beta, IC}^{PRSF}(E)}$$

Nous procédons par récurrence en montrant que $\forall X \in \mathcal{F}_{lexipref, IC}^{RSF} \mathcal{R}(E)$, nous avons $X \cap K_1 \subseteq \mathcal{R}(K_1 \cup IC)$ puis que si nous supposons que $(X \cap K_i) \in \mathcal{R}((K_i \cup \dots (K_1 \cup IC)))$ alors $X \cap K_{i+1} \in \mathcal{R}((K_{i+1} \cup \dots (K_1 \cup IC)))$.

Base Supposons qu'il existe $X \in \mathcal{F}_{lexipref, IC}^{RSF} \mathcal{R}(E)$ tel que $X \cap K_1 \notin \mathcal{R}(K_1 \cup IC)$. Cela peut avoir deux explications :

1. $(IC \sqcup K_1) \setminus (X \cap K_1)$ est incohérent. Or, $(E \setminus X) \sqcup IC$ est cohérent. Donc, $(K_1 \setminus X) \sqcup IC$ et $(K_1 \setminus (X \cap K_1)) \sqcup IC$ est également cohérent.
2. Il existe X' tel que $(K_1 \setminus X') \sqcup IC$ est cohérent et tel que $|X'| < |X \cap K_1|$. C'est impossible car $X' \cup K_2 \cup \dots \cup K_n$ est un R-ensemble potentiel de E contraint par IC et $X' \cup K_2 \cup \dots \cup K_n <_{lexipref} X$ ce qui est incompatible avec le fait que X soit un R-ensemble de E contraint par IC d'après la stratégie *lexipref*.

Nous pouvons donc affirmer que $\forall X \in \mathcal{F}_{lexipref, IC}^{RSF} \mathcal{R}(E)$, $(X \cap K_1) \in \mathcal{R}(K_1 \cup IC)$.

Récurrence Supposons que $X \cap K_i \in \mathcal{R}((K_i \cup \dots (K_1 \cup IC)))$. Nous voulons montrer que $X \cap K_{i+1} \in \mathcal{R}((K_{i+1} \cup \dots (K_1 \cup IC)))$.

Supposons que $X \cap K_{i+1} \notin \mathcal{R}((K_{i+1} \cup \dots (K_1 \cup IC)))$. Cela peut avoir deux explications :

1. $(K_{i+1} \setminus X \cap K_{i+1}) \sqcup (K_i \circ_{RSR} \dots (K_1 \circ_{RSR} IC))$ est incohérent. Or, $(E \setminus X) \sqcup IC$ est cohérent. Donc $((K_1 \sqcup \dots \sqcup K_i) \setminus (X \cap (K_1 \sqcup \dots \sqcup K_i))) \sqcup (K_{i+1} \setminus (X \cap K_{i+1})) \sqcup IC$ est cohérent et donc $((K_i \circ_{RSR} \dots (K_1 \circ_{RSR} IC)) \sqcup (K_{i+1} \setminus (X \cap K_{i+1})))$ est également cohérent.
2. Il existe X' tel que $((K_i \cup \dots (K_1 \cup IC)) \sqcup (K_{i+1} \setminus X'))$ est cohérent et tel que $|X'| < |X \cap K_{i+1}|$. Or dans ce cas, $X_1 \in \mathcal{R}(K_1 \cup IC), \dots, X_i \in \mathcal{R}((K_i \cup \dots (K_1 \cup IC)))$; nous avons $X_1 \cup \dots \cup X_i \cup X' \cup K_{i+2} \cup \dots \cup K_n$ serait un R-ensemble potentiel de E contraint par IC et serait préféré à X selon *lexipref*, ce qui est incompatible avec l'hypothèse que X est un R-ensemble de E contraint par IC selon *lexipref*.

Nous pouvons donc affirmer que si $(X \cap K_i) \in \mathcal{R}((K_i \cup \dots (K_1 \cup IC)))$ alors $X \cap K_{i+1} \in \mathcal{R}((K_{i+1} \cup \dots (K_1 \cup IC)))$. Et donc que $\Delta_{lexipref, IC}^{RSF}(E) \subseteq \Delta_{\beta, IC}^{PRSF}(E)$. $\square$

8.1.3 Mise en œuvre de la stratégie *lexipref*

Comme dans le cas de la mise en œuvre de la fusion sans préférence présentée dans le chapitre 6.4, nous utilisons un programme logique avec sémantique des modèles stables $\Pi_{E, IC}$ et construisons un programme logique $\Pi_{E, IC}^{lexipref}$ dont l'ensemble des modèles stables est l'ensemble des modèles stables préférés de $\Pi_{E, IC}$ selon la stratégie *lexipref*.

Pour la stratégie *lexipref*, il est nécessaire de compter le nombre d'atomes de règles pour chacune des bases de croyances. La stratégie *lexipref* compare les R-ensembles potentiels d'après la séquence des $|X \cap K_i|$ ordonnée selon les préférences accordées aux bases de croyances K_i . Comme pour la stratégie *Max*, il est nécessaire de connaître le nombre de formules retirées dans chacune des bases. Ce calcul est représenté par la règle suivante où m représente la taille de la base K_V .

Si $size(V, U)$ est vrai, dans la base K_V , le modèle stable en cours contient U atomes de règle. Finalement, le polynôme $X_n + X_{n-1} \times m + \dots + X_1 \times m^{n-1}$ est transformé en un entier qui devra être optimisé.

$$\Pi_E^{lexipref, size} = \{ \gamma_1 : size(V, U) \leftarrow U \{f_1^V, \dots, f_m^V\} U. \}$$

Pour la stratégie *lexipref*, soit $\Pi_{E, IC}^{lexipref} = \Pi_E^{lexipref, size} \cup \Pi_{E, IC} \cup minimize[size(1, 1) = 1 \times (m+1)^{n-1}, size(1, 2) = 1 \times (m+1)^{n-1}, \dots, size(i, 1) = 1 \times (m+1)^{n-i}, size(i, 2) = 2 \times (m+1)^{n-i}, \dots, size(n, m) = m]$.

La proposition suivante est vérifiée :

Proposition 8.1.2. *Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances tel que $K_1 < \dots < K_n$. L'ensemble des R-ensembles de $\Pi_{E, IC}^{lexipref}$ est l'ensemble des modèles stables préférés de $\Pi_{E, IC}$ pour la stratégie *lexipref*.*

Preuve.

1. Nous savons, au travers de la proposition 5.2.3 que X est un modèle stable de $\Pi_{E, IC}$ si et seulement si $F_O(X \cap S)$ est un R-ensemble potentiel de E contraint par IC .
2. Soit X un modèle stable de $\Pi_{E, IC}$, l'atome $size(V, U)$ est vrai dans $\Pi_{E, IC} \cup \Pi_E^{lexipref, size}$ si et seulement si $|X \cap K_V| = U$ au travers des règles de $\Pi_E^{lexipref, size}$.
3. Nous allons maintenant montrer que l'ensemble des R-ensembles de $\Pi_{E, IC}^{lexipref}$ est l'ensemble des modèles stables préférés de $\Pi_{E, IC}$ pour la stratégie *lexipref*.

⊆ Si X est un R-ensemble de $\Pi_{E, IC} \cup \Pi_E^{lexipref, size}$, nous notons sa valeur pour l'instruction *minimize* $val(X)$ et nous rappelons que $val(X) = \sum_{i=1}^n |X \cap K_i| \times (m+1)^{n-i}$. Supposons qu'il existe X tel que $F_O(X \cap S) \subseteq \mathcal{F}_{lexipref, IC}^{RSF} \mathcal{R}(E)$ et tel que X ne soit pas un modèle stable préféré de $\Pi_{E, IC}$ selon *lexipref*. Il existe donc X' tel que $val(X') < val(X)$. Par construction, si $val(X') < val(X)$ cela implique que :

- $\exists i$ tel que $|X' \cap K_i| < |X \cap K_i|$;
- $\forall j < i$ alors $|X' \cap K_j| = |X \cap K_j|$.

ce qui implique que $(p_{X'}^1, \dots, p_{X'}^n) <_{lex} (p_X^1, \dots, p_X^n)$ et qu'il est impossible que $F_O(X \cap S)$ appartienne à $\mathcal{F}_{lexipref, IC}^{RSF} \mathcal{R}(E)$.

⊇ Supposons qu'il existe X tel que X soit un modèle stable préféré de $\Pi_{E, IC}$ selon *lexipref* et tel que $F_O(X \cap S) \not\subseteq \mathcal{F}_{lexipref, IC}^{RSF} \mathcal{R}(E)$. Il existe donc X' tel que $(p_{X'}^1, \dots, p_{X'}^n) <_{lex} (p_X^1, \dots, p_X^n)$ cela implique que :

- $\exists i$ tel que $|X' \cap K_i| < |X \cap K_i|$;
- $\forall j < i$ alors $|X' \cap K_j| = |X \cap K_j|$.

et donc que $val(X') < val(X)$ ce qui est contradictoire avec le fait que X soit un modèle stable préféré de $\Pi_{E, IC}$ selon *lexipref*.

□

Exemple 8.1.4. *Soit $E = \{K_1, K_2, K_3\}$ le profil de croyances décrit dans l'exemple 8.1.1 que nous rappelons :*

$$K_1 = \{\neg e\}$$

$$K_2 = \{e \vee \neg b, e \vee \neg d\}$$

$$K_3 = \{\neg a \vee b, a, \neg c \vee d, c\}$$

Le programme $\Pi_{E,IC}^{lexipref}$ réalisant la fusion par R -ensembles selon la stratégie *lexipref* est le suivant :

$$\begin{aligned} a &\leftarrow \text{not } a'. & a' &\leftarrow \text{not } a. \\ b &\leftarrow \text{not } b'. & b' &\leftarrow \text{not } b. \\ c &\leftarrow \text{not } c'. & c' &\leftarrow \text{not } c. \\ d &\leftarrow \text{not } d'. & d' &\leftarrow \text{not } d. \\ e &\leftarrow \text{not } e'. & e' &\leftarrow \text{not } e. \\ f_{\neg e}^1 &\leftarrow e. & f_{e \vee \neg b}^2 &\leftarrow e', b. \\ f_{e \vee \neg d}^2 &\leftarrow e', d. & f_{\neg a \vee b}^3 &\leftarrow a, b'. \\ f_{\neg c \vee d}^3 &\leftarrow c, d'. & f_a^3 &\leftarrow a'. \\ f_c^3 &\leftarrow c'. \end{aligned}$$

$$\begin{aligned} size(1, 1) &\leftarrow 1\{f_{\neg e}^1\}1. & size(1, 2) &\leftarrow 1\{f_{e \vee \neg b}^2, f_{e \vee \neg d}^2\}1. \\ size(2, 2) &\leftarrow 2\{f_{e \vee \neg b}^2, f_{e \vee \neg d}^2\}2. & size(1, 3) &\leftarrow 1\{f_{\neg a \vee b}^3, f_{\neg c \vee d}^3, f_a^3, f_c^3\}1. \\ size(2, 3) &\leftarrow 2\{f_{\neg a \vee b}^3, f_{\neg c \vee d}^3, f_a^3, f_c^3\}2. & size(3, 3) &\leftarrow 3\{f_{\neg a \vee b}^3, f_{\neg c \vee d}^3, f_a^3, f_c^3\}3. \\ size(4, 3) &\leftarrow 4\{f_{\neg a \vee b}^3, f_{\neg c \vee d}^3, f_a^3, f_c^3\}4. \end{aligned}$$

$$\begin{aligned} minimize[& size(1, 1) = 25, size(1, 2) = 5, size(2, 2) = 10, size(1, 3) = 1, \\ & size(2, 3) = 2, size(3, 3) = 3, size(4, 3) = 4] \end{aligned}$$

Pour lequel nous obtenons quatre modèles stables préférés :

$$S_1 = \{size(2, 3), a, b', c, d', e', f_{\neg a \vee b}^3, f_{\neg c \vee d}^3\}$$

$$S_2 = \{size(2, 3), a', b', c, d', e', f_a^3, f_{\neg c \vee d}^3\}$$

$$S_3 = \{size(2, 3), a, b', c', d', e', f_{\neg a \vee b}^3, f_c^3\}$$

$$S_4 = \{size(2, 3), a', b', c', d', e', f_a^3, f_c^3\}$$

qui correspondent aux quatre R -ensembles de E pour la stratégie *lexipref*, à savoir :

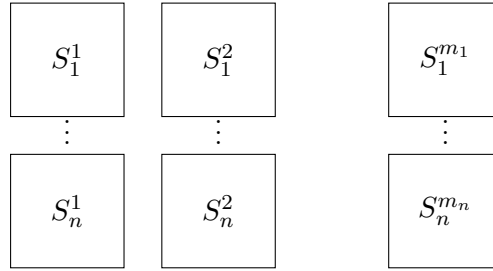
$$\begin{aligned} R_1 &= \{\neg a \vee b, \neg c \vee d\} & R_2 &= \{a, \neg c \vee d\} \\ R_3 &= \{\neg a \vee b, c\} & R_4 &= \{a, c\} \end{aligned}$$

8.2 Préférences externes et préférences internes

Nous étudions maintenant le cas où des préférences sont également exprimées à l'intérieur des bases de croyances entre les formules, que nous appelons préférences internes. Afin de les distinguer, nous notons $<_E$ la préférence externe et $<_{K_i}$ la préférence interne à la base K_i . Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances tel que $K_1 <_E \dots <_E K_n$ et à l'intérieur de chacune des base de croyances K_i , les formules sont organisées en strates. C'est à dire $K_i = S_i^1 \sqcup \dots \sqcup S_i^m$ et $\forall A \in S_i^k$ et $\forall B \in S_i^l$, $A <_{K_i} B$ si $k < l$.

Pour pouvoir choisir les formules à retirer pour restaurer la cohérence, nous devons tenir compte à la fois des préférences internes et des préférences externes. Nous construisons donc un pré-ordre total sur les formules qui permettent de représenter toutes les préférences exprimées. Ainsi, soient $A \in S_i^k$ et $B \in S_j^l$ deux formules de E , $A < B$ si et seulement si $K_i <_E K_j$ ou si $K_i =_E K_j$ et $S_i^k <_{K_i} S_j^l$.

Cette définition de la préférence entre formules crée une nouvelle stratification du profil de croyances.

FIGURE 8.1 – Décomposition en base et en strates de E

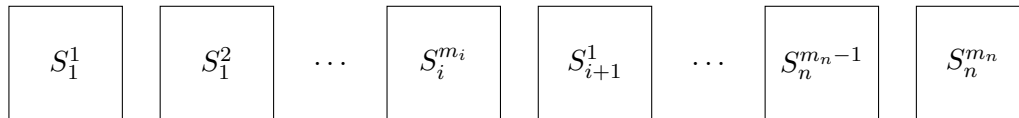
Le profil de croyances ainsi découpé est recomposé de la manière suivante.

Ce qui ramène le problème à un ensemble stratifié de formules comme dans le cas où seules des préférences externes ont été exprimées et cette recombposition permet de résoudre le problème à l'aide de la stratégie *lexipref* décrite dans la section précédente.

Comme dans le cas de la révision par R-ensembles [BnBPW04], nous étendons la fusion par R-ensembles à la fusion par R-ensembles au cas avec préférences. Nous notons $\Delta_{P,IC}^{PRSF}(E)$ l'opération de fusion du profil avec préférences E contraint par IC selon la stratégie P .

Définition 8.2.1 (R-ensemble). *Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances tel que $K_1 < K_2 < \dots < K_n$ et chaque base K_i est stratifié, $K_i = S_i^1 \cup \dots \cup S_i^{m_i}$. Soit IC un ensemble de formules représentant les contraintes pesant sur l'opération tel que $(K_1 \sqcup \dots \sqcup K_n) \sqcup IC$ est incohérent. Soit P une stratégie de fusion. X est un R-ensemble de E contraint par IC selon la stratégie P si et seulement si :*

- X est un R-ensemble potentiel de E contraint par IC ;
- Il n'existe pas de R-ensemble potentiel $X' \subseteq K_1 \sqcup \dots \sqcup K_n$ tel que $X' \subset X$;
- Il n'existe pas de R-ensemble potentiel $X' \subseteq K_1 \sqcup \dots \sqcup K_n$ tel que $X' <_P X$.

FIGURE 8.2 – Recomposition en strates de E

Ce qui conduit à l'opération de fusion suivante :

Définition 8.2.2 (Fusion par R-ensembles). *Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances tel que $K_1 < K_2 < \dots < K_n$ et chaque base K_i est stratifié, $K_i = S_i^1 \cup \dots \cup S_i^m$. Soit IC un ensemble de formules représentant les contraintes. Soit P une stratégie. Soit $\mathcal{F}_{P,IC}^{PRSF} \mathcal{R}(E)$ l'ensemble des R-ensembles de E contraint par IC selon P . Nous définissons l'opération de fusion de la manière suivante :*

$$\Delta_{P,IC}^{RSF}(E) = \bigvee_{X \in \mathcal{F}_P^{PRSF} \mathcal{R}(E)} \{((K_1 \sqcup \dots \sqcup K_n) \setminus X) \sqcup IC\}.$$

Illustrons cela à l'aide de notre exemple que nous modifions.

Exemple 8.2.1. *Soit $E = \{K_1, K_2, K_3\}$ le profil de croyances suivant :*

$$K_1 = \{\neg e\}$$

$$K_2 = \{e \vee \neg b, e \vee \neg d\}$$

$$K_3 = \{\neg a \vee b, a, \neg c \vee d, c\}$$

Pour lequel nous avons les préférences externes $K_1 < K_2 < K_3$ avec K_2 et K_3 découpées en strates. De plus $IC = \top$, ainsi :

$$S_2^1 = \{e \vee \neg b\} \quad S_2^2 = \{e \vee \neg d\}$$

avec $S_2^1 < S_2^2$ et

$$S_3^1 = \{a, c\} \quad S_3^2 = \{\neg a \vee b, \neg c \vee d\}$$

avec $S_3^1 < S_3^2$.

Les préférences sont donc ici recomposées de la manière suivante : $K_1 < S_2^1 < S_2^2 < S_3^1 < S_3^2$.

La fusion de E ainsi découpée selon $lexipref$ ne possède qu'un seul R-ensemble $\mathcal{F}_{lexipref}^{PRSF} \mathcal{R}(E) = \{\{\neg a \vee b, \neg c \vee d\}\}$ et $\Delta_{lexipref}^{PRSF}(E) = \{\neg e, e \vee \neg b, e \vee \neg d, a, c\}$.

Les autres R-ensembles donnés dans l'exemple 8.1.4 ne respectent pas les préférences internes exprimées ici. Par exemple, $\{\neg a \vee b, c\} < \{\neg a \vee b, \neg c \vee d\}$ car $c \in S_3^1$ et $\neg c \vee d \in S_3^2$.

Ainsi, à partir du moment où les bases stratifiées ont été réorganisées, la mise en œuvre de l'opération de fusion par R-ensembles dans ce cas s'effectue de façon similaire au cas où les préférences portent sur les bases de croyances seules.

Conclusion

Nous avons étudié la fusion par R-ensembles de croyances dans le cas où des préférences sont exprimées entre les bases de croyances, ainsi que dans le cas où des préférences sont exprimées entre les bases de croyances et à l'intérieur des bases de croyances. Nous avons proposé un opérateur de fusion utilisant une nouvelle stratégie *lexipref* permettant

de tenir compte des préférences. Nous avons également étudié la possibilité de considérer le problème de la fusion de croyances dans le cas de croyances totalement pré-ordonnées en tant que problème de révision itérée et nous avons montré que ces deux méthodes étaient équivalentes. Finalement, nous avons proposé une mise en œuvre de la stratégie *lexipref*.

Pour le cas où seules les préférences internes sont exprimées, une étude approfondie peut être trouvée dans [YBL⁺08].

Chapitre 9

Fusion de bases de croyances exprimées sous forme de programme logique

La connaissance progresse en intégrant en elle l'incertitude, non en l'exorcisant. *Edgar Morin, La méthode*

Introduction

L'intérêt d'utiliser la logique propositionnelle comme langage pour la représentation de croyances réside dans sa simplicité, dans sa décidabilité, ainsi que dans les nombreuses mises en œuvre de ce formalisme. Elle possède malgré tout quelques limites en termes d'expressivité. En particulier, elle ne peut pas représenter de croyances non-monotones. La programmation logique avec sémantique des modèles stables semble plus adaptée pour représenter des problèmes de raisonnement non-monotone (planification, raisonnement en cas d'exceptions, etc.). Nous l'avons utilisée pour effectuer la mise en œuvre de la fusion par R-ensembles de bases de croyances représentées par des formules de la logique propositionnelle. Il nous a semblé naturel d'utiliser la programmation logique pour représenter les bases de croyances.

Nous étendons l'approche de fusion par R-ensembles au cas où les bases de croyances sont représentées par des programmes logiques. Nous considérons d'abord la sémantique des modèles stables de Gelfond et Lifschitz (décrite dans 2.4) et nous proposons une opération de fusion que nous appelons opération forte de fusion par R-ensembles et nous en étudions les propriétés. Nous utilisons ensuite une autre sémantique basée sur la logique du Here-and-There (décrite dans 2.4.2.1) pour définir une opération faible de fusion dont nous montrons les avantages par rapport à l'opération forte. Il est à noter que récemment J. Delgrande et T. Schaub [DSTW08] ont également utilisé cette sémantique pour la révision de programmes logiques. Nous montrons les propriétés de l'opération faible de fusion par R-ensembles.

9.1 Première approche : opération forte de fusion par R-ensembles

Nous nous plaçons dans cette chapitre dans le cadre où les croyances sont représentées sous forme de programmes logiques. Nous considérons donc un profil de croyances $E = \{K_1, \dots, K_n\}$ où chaque K_i est un base de croyances composée de règles de la programmation logique. L'objectif d'une opération de fusion est d'obtenir un ensemble de formules (ici, ce sera des règles) cohérente. Il convient donc de définir la notion de cohérence pour un programme logique. La définition directe de la cohérence est la suivante :

Définition 9.1.1. *Soit K une base de croyances représentée par un programme logique avec sémantique des modèles stables. K est cohérente si et seulement si elle possède au moins un modèle stable.*

La définition de l'opération forte de R-ensembles se fait en trois étapes. D'abord, nous définissons les R-ensembles potentiels qui correspondent aux ensembles de règles, qui lorsqu'ils sont retirés permettent de restaurer la cohérence. Puis, parmi cette multitude de R-ensembles potentiels, nous choisissons ceux qui sont préférés au sens de la stratégie choisie. Finalement, ces R-ensembles permettent de définir l'opération de fusion par R-ensembles.

Définition 9.1.2 (R-ensemble potentiel fort). *Soient $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un programme logique représentant les contraintes tels que $K_1 \sqcup \dots \sqcup K_n \sqcup IC$ est incohérent. Soit X un sous-ensemble de règles de $K_1 \sqcup \dots \sqcup K_n$. X est un R-ensemble potentiel fort de E contraint par IC si et seulement si $((K_1 \sqcup \dots \sqcup K_n) \setminus X) \sqcup IC$ est cohérent.*

Nous donnons un exemple afin d'illustrer cette définition.

Exemple 9.1.1. *Prenons pour illustrer notre opération forte de fusion, le profil de croyances $E = \{K_1, K_2\}$ suivant avec :*

$$K_1 = \left\{ \begin{array}{ll} a \leftarrow \text{not } b. & c \leftarrow \text{not } a. \\ e. & d \leftarrow f. \end{array} \right\}$$

$$K_2 = \left\{ \begin{array}{ll} b \leftarrow \text{not } c. & \neg d. \\ d \leftarrow e. & f. \end{array} \right\}$$

Les R-ensembles potentiels forts de ce profil sont :

$$\begin{array}{ll} \{\neg d. & a \leftarrow \text{not } b.\} & \{\neg d. & b \leftarrow \text{not } c.\} \\ \{\neg d. & c \leftarrow \text{not } a.\} & \{e. & f. & a \leftarrow \text{not } b.\} \\ \{e. & f. & b \leftarrow \text{not } c.\} & \{e. & f. & c \leftarrow \text{not } a.\} \\ \{e. & d \leftarrow f. & a \leftarrow \text{not } b.\} & \{e. & d \leftarrow f. & b \leftarrow \text{not } c.\} \\ \{e. & d \leftarrow f. & c \leftarrow \text{not } a.\} & \{d \leftarrow e. & f. & a \leftarrow \text{not } b.\} \\ \{d \leftarrow e. & f. & b \leftarrow \text{not } c.\} & \{d \leftarrow e. & f. & c \leftarrow \text{not } a.\} \\ \{d \leftarrow e. & d \leftarrow f. & a \leftarrow \text{not } b.\} & \{d \leftarrow e. & d \leftarrow f. & b \leftarrow \text{not } c.\} \\ \{d \leftarrow e. & d \leftarrow f. & c \leftarrow \text{not } a.\} & \end{array}$$

ainsi que tous les sur-ensembles de règles incluant ces ensembles-là.

Nous définissons les R-ensembles forts comme étant les R-ensembles potentiels forts préférés au sens de la stratégie P choisie.

Définition 9.1.3 (R-ensemble fort). *Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un programme logique représentant les contraintes tels que $K_1 \sqcup \dots \sqcup K_n \sqcup IC$ est incohérent. Soit P la stratégie de fusion. Soit X un sous-ensemble de règles de $K_1 \sqcup \dots \sqcup K_n$ est un R-ensemble fort de E contraint par IC selon la stratégie P si et seulement si :*

1. X est un R-ensemble potentiel fort de E contraint par IC ;
2. Il n'existe pas de R-ensemble potentiel fort Y de E contraint par IC tel que $Y \subset X$.
3. Il n'existe pas de R-ensemble potentiel fort Y de E contraint par IC tel que $Y <_P X$.

Exemple 9.1.2. Reprenons le profil E de l'exemple 9.1.1.

Ainsi, peu importe la stratégie P , les R-ensembles forts de E selon P sont :

$$\begin{aligned} &\{\neg d. \quad a \leftarrow \text{not } b.\} \quad \{\neg d. \quad b \leftarrow \text{not } c.\} \\ &\{\neg d. \quad c \leftarrow \text{not } a.\} \end{aligned}$$

L'ensemble des R-ensembles forts de E contraint par IC selon la stratégie P est noté $\mathcal{F}_{P,IC}^{\Pi RSF,s} \mathcal{R}(E)$. L'opération forte de fusion est définie de la manière suivante.

Définition 9.1.4 (Opération forte de fusion). *Soient $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un programme logique représentant les contraintes tels que $K_1 \sqcup \dots \sqcup K_n \sqcup IC$ est incohérent. Soit P une stratégie de fusion. L'opération forte de fusion $\Delta_{P,IC}^{\Pi RSF,s}(E)$ est définie par :*

$$\Delta_{P,IC}^{\Pi RSF,s}(E) = \bigvee_{X \in \mathcal{F}_{P,IC}^{\Pi RSF,s} \mathcal{R}(E)} \{((K_1 \sqcup \dots \sqcup K_n) \setminus X) \sqcup IC\}$$

Exemple 9.1.3. Terminons l'exemple 9.1.1. L'opération forte de fusion de E quelle que soit la stratégie P aura pour résultat :

$$\begin{aligned} \Delta_{P,IC}^{\Pi RSF,s}(E) &= \left\{ \begin{array}{lll} c \leftarrow \text{not } a. & e. & d \leftarrow f. \\ b \leftarrow \text{not } c. & d \leftarrow e. & f. \end{array} \right\} \\ &\vee \left\{ \begin{array}{lll} a \leftarrow \text{not } b. & c \leftarrow \text{not } a. & e. \\ d \leftarrow f. & d \leftarrow e. & f. \end{array} \right\} \\ &\vee \left\{ \begin{array}{lll} a \leftarrow \text{not } b. & e. & d \leftarrow f. \\ b \leftarrow \text{not } c. & d \leftarrow e. & f. \end{array} \right\} \end{aligned}$$

L'étude des propriétés de notre opération forte de fusion forte nécessite une définition de l'inférence entre programmes logiques. Nous en proposons donc maintenant une définition basée sur la logique du Here-and-There.

9.1.1 Relation d'inférence entre programmes logiques

En logique propositionnelle, les conséquences d'un ensemble de formules sont clairement définies. Soit A un ensemble de formules. Une interprétation I est un modèle de A si et seulement si toutes les formules de A sont satisfaites par I (i.e. $I \models A$). Et, grâce à cette définition, nous savons qu'un ensemble F de formules est la conséquence de A si et seulement si l'ensemble des interprétations qui satisfont F satisfont également A (i.e. $A \models F$ si et seulement si $\forall I, (I \models A) \text{ implique } (I \models F)$). Si les conséquences d'un programme logique sont définies dans le cadre de la sémantique des modèles stables, le fait qu'un programme logique est la conséquence d'un autre programme logique n'est pas clairement défini.

Exemple 9.1.4. *Le postulat (IC0) signifie qu'un opérateur de fusion sous contraintes doit respecter ces dernières (i.e. $IC \in \text{Cons}(\Delta_{P,IC}^{\Pi RSF,s}(E))$). Considérons $E = \{\{a.\}\}$ et $IC = \{c \leftarrow \text{not } a. \quad \neg b.\}$.*

La base de croyances $E \sqcup IC$ possède des modèles stables. Ainsi, quelle que soit la stratégie choisie : $\Delta_{P,IC}^{\Pi RSF,s}(E) = E \sqcup IC$. L'opération de fusion du profil E contraint par IC contient bien les règles de IC . Cependant, le modèle stable de IC est $\{c, \neg b\}$ alors que le modèle stable de $\Delta_{P,IC}^{\Pi RSF,s}(E)$ est $\{a, \neg b\}$.

Nous constatons donc que malgré le fait que les règles appartenant à IC soient incluses dans $E \sqcup IC$, les conséquences de IC ne sont pas incluses dans les conséquences de $E \sqcup IC$ et les conséquences de $E \sqcup IC$ ne sont pas incluses dans les conséquences de IC .

Cet exemple montre qu'avec la définition de l'inférence entre programmes logiques en termes d'inclusion de modèles stables, le postulat (IC0) n'est pas vérifié alors que $\Delta_{P,IC}^{\Pi RSF,s}(E)$ respecte les contraintes. Cette définition de l'inférence n'est pas satisfaisante. Pour ces raisons, nous proposons une définition de l'inférence entre programmes logiques qui ne soit pas directement basée sur les modèles stables mais sur la logique, monotone, du Here-and-There présentée dans la section 2.4.2.1.

Définition 9.1.5 (Inférence). *Soient Π_1 et Π_2 des programmes logiques. Nous définissons que Π_1 implique Π_2 (noté $\Pi_1 \models \Pi_2$) si et seulement si pour tout (X, Y) tel que $(X, Y) \in HT(\Pi_1)$ alors $(X, Y) \in HT(\Pi_2)$.*

Nous illustrons cette définition grâce à l'exemple suivant :

Exemple 9.1.5. *Soient Π_1 et Π_2 deux programmes logiques tels que :*

$$\Pi_1 = \{a \leftarrow \text{not } b. \quad b \leftarrow \text{not } a.\}$$

$$\Pi_2 = \{a \leftarrow \text{not } b.\}$$

L'ensemble des HT-modèles de Π_1 est $HT(\Pi_1) = \{(\{b\}, \{b\}), (\{a\}, \{a\}), (\emptyset, \{a, b\}), (\{a\}, \{a, b\}), (\{b\}, \{a, b\}), (\{a, b\}, \{a, b\})\}$. L'ensemble des HT-modèles de Π_2 est $HT(\Pi_2) = \{(\emptyset, \{b\}), (\{b\}, \{b\}), (\{a\}, \{a\}), (\emptyset, \{a, b\}), (\{a\}, \{a, b\}), (\{b\}, \{a, b\}), (\{a, b\}, \{a, b\})\}$.

Il est clair que si $(X, Y) \in HT(\Pi_2)$ alors $(X, Y) \in HT(\Pi_1)$ ce qui implique que $\Pi_2 \models \Pi_1$ selon notre définition.

Cette définition permet de définir une relation d'inférence permettant de prendre en compte les spécificités de la négation par échec dans la programmation logique. La monotonie de la logique du Here-and-There permet d'assurer que si toutes les règles d'un programme Π_1 sont incluses dans un programme Π_2 alors $\Pi_1 \models \Pi_2$.

9.1.1.1 Opérateurs de fusion forte et postulats IC

Les opérateurs classiques de fusion que nous avons décrits dans le chapitre 4 peuvent également être utilisés dans le contexte de la fusion par R-ensembles de croyances exprimées sous forme de programme logique.

Nous rappelons ici la définition des pré-ordres associés à chaque stratégie afin de les illustrer par des exemples. Puis, nous donnerons les propriétés de ces opérateurs dans ce cadre en fonction des postulats de Konieczny et Pino-Perez tels que nous les avons reformulés.

9.1.1.2 La stratégie *Somme*

Nous rappelons la définition du pré-ordre associé à la stratégie *Somme*.

Définition 9.1.6 (Pré-ordre $\leq_\Sigma$). *Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. Soient X et X' deux R-ensembles potentiels forts de E . Le pré-ordre $\leq_\Sigma$ est défini de la manière suivante :*

$$X \leq_\Sigma X' \text{ si et seulement si } \sum_{1 \leq i \leq n} |X \cap K_i| \leq \sum_{1 \leq i \leq n} |X' \cap K_i|.$$

Nous donnons un exemple pour illustrer la stratégie *Somme* dans le cadre de la fusion forte par R-ensembles de bases de croyances exprimées sous forme de programme logique.

Exemple 9.1.6. *Soient $E = \{K_1, K_2, K_3\}$ un profil de croyances tel que :*

$$K_1 = \left\{ \begin{array}{cc} \neg a. & b \leftarrow a \\ d \leftarrow \text{not } a. & e \leftarrow \text{not } a. \\ c \leftarrow a & \end{array} \right\}$$

$$K_2 = \{ a. \quad \neg b. \}$$

$$K_3 = \left\{ \begin{array}{cc} b \leftarrow d. & c \leftarrow e. \\ \neg c. & \end{array} \right\}$$

*Le seul R-ensemble fort pour la stratégie *Somme* est :*

$$\mathcal{F}_{\Sigma, IC}^{\Pi RSF, s} \mathcal{R}(E) = \{ \neg a. \quad b \leftarrow a. \}$$

qui enlève deux règles dans la même base de croyances et qui donne pour résultat de l'opération forte de fusion :

$$\Delta_{\Sigma, IC}^{\Pi RSF, s}(E) = \left\{ \begin{array}{ll} d \leftarrow not\ a. & e \leftarrow not\ a. \\ c \leftarrow a & a. \\ \neg b. & b \leftarrow d. \\ c \leftarrow e. & \neg c. \end{array} \right\}$$

Nous étudions maintenant les propriétés de l'opération forte de fusion $\Delta_{\Sigma, IC}^{\Pi RSF, s}(E)$ dans ce contexte.

Proposition 9.1.1. *L'opération forte de fusion $\Delta_{\Sigma, IC}^{\Pi RSF, s}(E)$ respecte les postulats de fusion **IC0**, **IC1** et **IC2**.*

Stratégies	(IC0)	(IC1)	(IC2)	(IC3)	(IC4)	(IC5)	(IC6)	(IC7)	(IC8)
Somme	✓	✓	✓	×	×	×	×	×	×

Preuves et contre-exemples

Nous donnons maintenant les preuves et les contre-exemples de cette proposition postulat par postulat. Nous rappelons que : $\mathcal{F}_{P, IC}^{\Pi RSF, s}\mathcal{F}(E) = \cup_{X \in \mathcal{F}_{P, IC}^{\Pi RSF, s}\mathcal{R}(E)}(E \setminus X)$.

(IC0)

Nous savons que $HT(\Pi \cup \Pi') = HT(\Pi) \cap HT(\Pi')$ par l'intermédiaire de la proposition 2.4.1. Nous savons également que

$$\Delta_{P, IC}^{\Pi RSF, s}(E) = \bigvee_{X \in \mathcal{F}_{P, IC}^{\Pi RSF, s}\mathcal{R}(E)} \{((K_1 \sqcup \dots \sqcup K_n) \setminus X) \sqcup IC\}.$$

Par construction, nous pouvons déduire que $IC \subseteq \Delta_{\Sigma, IC}^{\Pi RSF, s}(E)$ et donc $HT(IC) \subseteq HT(\Delta_{\Sigma, IC}^{\Pi RSF, s}(E))$.

Le postulat **(IC0)** est donc vérifié.

(IC1)

Tous les éléments de $F \in \mathcal{F}_{\Sigma, IC}^{\Pi RSF, s}\mathcal{F}(E)$ sont définis tel que $F \sqcup IC$ est cohérent. Ainsi, par définition de l'opération $\Delta_{\Sigma, IC}^{\Pi RSF, s}(E)$, $\Delta_{\Sigma, IC}^{\Pi RSF, s}(E)$ est cohérent si IC est cohérent.

Le postulat **(IC1)** est donc vérifié.

(IC2)

Si $E \sqcup IC$ est cohérent, alors par définition $\mathcal{F}_{\Sigma, IC}^{\Pi RSF, s}\mathcal{R}(E) = \{\emptyset\}$:

- $E \sqcup IC$ est cohérent ;
- $\emptyset$ est minimal pour l'inclusion ;
- Et il ne peut exister de R-ensemble potentiel plus petit que $\emptyset$ au sens de $\leq_{\Sigma}$.

et donc il n'y qu'un seul élément dans $\mathcal{F}_{\Sigma, IC}^{\Pi RSF, s}\mathcal{F}(E)$ qui est égal à E .

(IC3)

Nous donnons un contre-exemple invalidant **(IC3)**.

Soient E_1 et E_2 deux profils de croyances tels que $E_1 = \{K_1, K_3\}$ et $E_2 = \{K_2, K_3\}$ et $IC = \top$ avec :

$$K_1 = \{a \leftarrow \text{not } b. \quad c \leftarrow a. \quad a \leftarrow c.\}$$

$$K_2 = \{c \leftarrow \text{not } b. \quad c \leftarrow a. \quad a \leftarrow c.\}$$

$$K_3 = \{\neg a.\}$$

Les ensembles de HT-modèles de K_1 et K_2 sont identiques à savoir $HT(K_1) = HT(K_2) = \{ (\emptyset, \{a, b, c\}) (\{b\}, \{a, b, c\}) (\{a, c\}, \{a, c\}) (\emptyset, \{b\}) (\{b\}, \{b\}) \}$. E_1 et E_2 sont donc équivalents puisqu'il existe une bijection $f(K_1) = K_2$ et $f(K_3) = K_3$ entre les deux profils telle que $f(K_1) \equiv K_2$ et $f(K_3) \equiv K_3$.

Cependant les résultats obtenus par $\Delta_{\Sigma, IC}^{\Pi RSF, s}(E_1)$ et $\Delta_{\Sigma, IC}^{\Pi RSF, s}(E_2)$ ne sont pas équivalents, à savoir :

$$\Delta_{\Sigma, IC}^{\Pi RSF, s}(E_1) = \{c \leftarrow a. \quad a \leftarrow c. \quad \neg a.\}$$

et

$$\begin{aligned} \Delta_{\Sigma, IC}^{\Pi RSF, s}(E_2) &= \{ \{c \leftarrow a. \quad a \leftarrow c. \quad \neg a.\} \\ &\quad \vee \{c \leftarrow \text{not } b. \quad c \leftarrow a. \quad \neg a.\} \} \end{aligned}$$

Le postulat **(IC3)** n'est donc pas vérifié.

(IC4)

Nous donnons un contre-exemple invalidant **(IC4)**.

Soient $K_1 = \{a \leftarrow \text{not } b. \quad b \leftarrow \text{not } a. \quad c \leftarrow a. \quad c \leftarrow b.\}$, $K_2 = \{\neg c.\}$ et $IC = \top$.

Nous avons ici $\Delta_{\Sigma, IC}^{\Pi RSF, s}(K_1 \sqcup K_2) = K_1$ ainsi $\Delta_{\Sigma, IC}^{\Pi RSF, s}(K_1 \sqcup K_2) \sqcup K_1$ est cohérent mais $\Delta_{\Sigma, IC}^{\Pi RSF, s}(K_1 \sqcup K_2) \sqcup K_2$ n'est pas cohérent.

Le postulat **(IC4)** n'est donc pas vérifié.

(IC5) et (IC6)

Nous donnons un contre-exemple invalidant **(IC5)** et **(IC6)**.

Soient $E_1 = \{K_1, K_2\}$ et $E_2 = \{K_3, K_4\}$ deux profils de croyances et $IC = \top$ avec les bases de croyances suivantes :

$$K_1 = \{ a \leftarrow \text{not } \neg h. \quad c \leftarrow b. \}$$

$$K_2 = \left\{ \begin{array}{l} c \leftarrow a. \quad b \leftarrow \text{not } \neg h. \\ \neg c. \end{array} \right\}$$

$$K_3 = \{ d \leftarrow \text{not } \neg c. \quad h \leftarrow e. \}$$

$$K_4 = \left\{ \begin{array}{l} h \leftarrow d. \quad e \leftarrow \text{not } \neg c. \\ \neg h. \end{array} \right\}$$

$\Delta_{\Sigma, \top}^{\Pi RSF, s}(E_1) = \{a \leftarrow \text{not } \neg h. \quad b \leftarrow \text{not } \neg h. \quad c \leftarrow a. \quad c \leftarrow b.\}$ et tous les HT-modèles du programme $\Delta_{\Sigma, \top}^{\Pi RSF, s}(E_1)$ ont c dans le premier ensemble d'atomes de la paire.

$\Delta_{\Sigma, \top}^{\Pi RSF, s}(E_2) = \{d \leftarrow \text{not } \neg c. \quad e \leftarrow \text{not } \neg c. \quad h \leftarrow d. \quad h \leftarrow e.\}$ et tous les HT-modèles du programme $\Delta_{\Sigma, \top}^{\Pi RSF, s}(E_2)$ ont h dans le premier ensemble d'atomes de la paire.

$\Delta_{\Sigma, \top}^{\Pi RSF, s}(E_1 \sqcup E_2) = E_1 \cup E_2$ et $\neg c$ et $\neg h$ sont dans le premier ensemble d'atomes de la paire de tous les HT-modèles du programme.

Ainsi, pour tout $(X, Y) \in HT(\Delta_{\Sigma, \top}^{\Pi RSF, s}(E_1))$, nous avons $(X, Y) \notin HT(\Delta_{\Sigma, \top}^{\Pi RSF, s}(E_1 \sqcup E_2))$ ce qui invalide le postulat **(IC5)**.

Et pour tout $(X, Y) \in HT(\Delta_{\Sigma, \top}^{\Pi RSF, s}(E_1 \sqcup E_2))$, nous avons $(X, Y) \notin HT(\Delta_{\Sigma, \top}^{\Pi RSF, s}(E_2))$ ce qui invalide le postulat **(IC6)**.

Les postulats **(IC5)** et **(IC6)** ne sont donc pas vérifiés.

(IC7) et (IC8)

Soit $E = \{K_1, K_2\}$ un profil de croyances et $IC_1 = \top$ et $IC_2 = \{c.\}$ avec les bases de croyances suivantes :

$$K_1 = \left\{ \begin{array}{l} d \leftarrow b. \quad b \leftarrow \text{not } c. \end{array} \right\}$$

$$K_2 = \left\{ \begin{array}{l} a \leftarrow \text{not } c. \quad \neg d. \quad d \leftarrow a. \end{array} \right\}$$

$\Delta_{\Sigma, \top}^{\Pi RSF, s}(E) = \{a \leftarrow \text{not } c. \quad b \leftarrow \text{not } c. \quad d \leftarrow a. \quad d \leftarrow b.\}$. Tous les HT-modèles de $\Delta_{\Sigma, \top}^{\Pi RSF, s}(E)$ ont d dans le premier ensemble d'atomes de la paire. Tous les HT-modèles de $\Delta_{\Sigma, \top \wedge IC_2}^{\Pi RSF, s}(E)$ ont $\neg d$ dans le premier ensemble d'atomes de la paire.

Ainsi, pour tout $(X, Y) \in HT(\Delta_{\Sigma, \top}^{\Pi RSF, s}(E))$, nous avons $(X, Y) \notin HT(\Delta_{\Sigma, IC_2}^{\Pi RSF, s}(E))$ ce qui invalide le postulat **(IC7)**. Et pour tout $(X, Y) \in HT(\Delta_{\Sigma, IC_2}^{\Pi RSF, s}(E))$, nous avons $(X, Y) \notin HT(\Delta_{\Sigma, \top}^{\Pi RSF, s}(E))$ ce qui invalide le postulat **(IC8)**.

Les postulats **(IC7)** et **(IC8)** ne sont donc pas vérifiés.

9.1.1.3 La stratégie Card

Nous rappelons la définition du pré-ordre associé à la stratégie Card.

Définition 9.1.7 (Pré-ordre $\leq_{Card}$). Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. Soient X et X' deux R -ensembles potentiels forts de E . Le pré-ordre $\leq_{Card}$ est défini de la manière suivante :

$$X \leq_{Card} X' \text{ si et seulement si } |X| \leq |X'|.$$

Nous donnons maintenant un exemple permettant d'illustrer le comportement de l'opération $\Delta_{Card,IC}^{\Pi RSF,s}(E)$ définie à partir de ce pré-ordre.

Exemple 9.1.7. Soit $E = \{K_1, K_2, K_3\}$ un profil de croyances tel que :

$$K_1 = \{\neg a. \quad a \leftarrow not\ b.\}$$

$$K_2 = \{\neg c. \quad a \leftarrow not\ b.\}$$

$$K_3 = \{c \leftarrow a. \quad a \leftarrow not\ b.\}$$

Pour ce profil, la stratégie Somme aurait deux R-ensembles ayant un cardinal égal à 2 : $\mathcal{F}_{\Sigma}^{\Pi RSF,s}\mathcal{R}(E) = \{\{\neg a. \quad \neg c.\}\{\neg a. \quad c \leftarrow a.\}\}$ car l'importance de la règle $a \leftarrow not\ b.$ répétés trois fois est trop important.

Alors que la stratégie Card ne tenant pas compte de la répétition des règles possède un seul R-ensemble de cardinal égal à 1 : $\mathcal{F}_{\Sigma}^{\Pi RSF,s}\mathcal{R}(E) = \{a \leftarrow not\ b.\}$

Nous étudions maintenant les propriétés de l'opération forte de fusion $\Delta_{Card,IC}^{\Pi RSF,s}(E)$ dans ce contexte.

Proposition 9.1.2. L'opération forte de fusion $\Delta_{Card,IC}^{\Pi RSF,s}(E)$ respecte les postulats de fusion **IC0**, **IC1** et **IC2**.

Strategies	(IC0)	(IC1)	(IC2)	(IC3)	(IC4)	(IC5)	(IC6)	(IC7)	(IC8)
Card	✓	✓	✓	×	×	×	×	×	×

Preuves et contre-exemples

Les preuves et les contre-exemples sont similaires à ceux donnés pour l'opération $\Delta_{\Sigma,IC}^{\Pi RSF,s}(E)$.

9.1.1.4 La stratégie Max

Nous rappelons la définition du pré-ordre associé à la stratégie Max.

Définition 9.1.8 (Pré-ordre $\leq_{Max}$). Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. Soient X et X' deux R-ensembles potentiels forts de E . Le pré-ordre $\leq_{Max}$ est défini de la manière suivante :

$$X \leq_{Max} X' \text{ si et seulement si } \max_{1 \leq i \leq n} |X \cap K_i| \leq \max_{1 \leq i \leq n} |X' \cap K_i|.$$

Et nous donnons un exemple pour illustrer la stratégie Max dans le cadre de la fusion forte par R-ensembles de bases de croyances exprimées sous forme de programmes logiques.

Exemple 9.1.8. Reprenons l'exemple 9.1.6.

Le R-ensemble pour la stratégie Somme ($\{\neg a. \quad b \leftarrow a.\}$) enlève deux règles dans la même base de croyances. Il existe d'autres R-ensembles potentiels qui restaurent la cohérence en repartissant les règles retirées de manière plus équitable.

Par exemple, les R -ensembles pour la stratégie Max sont : $\mathcal{F}_{Max,IC}^{\Pi RSF,s}(\mathcal{R}(E)) = \{ \{ \neg a. \neg b. \neg c. \}, \{ c \leftarrow not\ a. \ a. \ b \leftarrow d. \}, \{ d \leftarrow not\ a. \ a. \ c \leftarrow e. \} \}$ qui enlève une règle dans chaque base de croyances.

Nous étudions maintenant les propriétés de l'opération de fusion $\Delta_{Max,IC}^{\Pi RSF,s}(E)$ dans ce contexte.

Proposition 9.1.3. *L'opération forte de fusion $\Delta_{Max,IC}^{\Pi RSF,s}(E)$ respecte les postulats de fusion IC0, IC1 et IC2.*

Strategies	(IC0)	(IC1)	(IC2)	(IC3)	(IC4)	(IC5)	(IC6)	(IC7)	(IC8)
Max	✓	✓	✓	×	×	×	×	×	×

Preuves et contre-exemples

Les preuves et les contre-exemples sont les mêmes que ceux donnés pour l'opération $\Delta_{\Sigma,IC}^{\Pi RSF,s}(E)$.

9.1.1.5 La stratégie $GMax$

Nous rappelons la définition du pré-ordre associé à la stratégie $GMax$.

Définition 9.1.9 (Pré-ordre $\leq_{GMax}$). *Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. Soient X et X' deux R -ensembles potentiels forts de E . Le pré-ordre $\leq_{GMax}$ est défini de la manière suivante :*

$$X \leq_{GMax} X' \text{ si et seulement si } L_X^E <_{lex} L_{X'}^E.$$

Nous donnons maintenant un exemple permettant d'illustrer le comportement de l'opération $\Delta_{GMax,IC}^{\Pi RSF,s}(E)$ défini à partir de ce pré-ordre.

Exemple 9.1.9. *Soit $E = \{K_1, K_2, K_3\}$ un profil de croyances tel que :*

$$K_1 = \{ \neg a. \ a \leftarrow not\ b. \}$$

$$K_2 = \{ \neg c. \ a \leftarrow not\ b. \}$$

$$K_3 = \{ c \leftarrow a. \ a \leftarrow not\ b. \}$$

Pour ce profil, la stratégie $GMax$ aurait trois R -ensembles forts : $\mathcal{F}_{Max}^{\Pi RSF,s}(\mathcal{R}(E)) = \{ \{ \neg a. \neg c. \}, \{ \neg a. \ c \leftarrow a. \}, \{ a \leftarrow not\ b. \ a \leftarrow not\ b. \ a \leftarrow not\ b. \} \}$.

Alors que la stratégie $GMax$ permet de remarquer que le troisième R -ensemble fort retire plus d'information que les deux autres : $\mathcal{F}_{GMax}^{\Pi RSF,s}(\mathcal{R}(E)) = \{ \{ \neg a. \neg c. \}, \{ \neg a. \ c \leftarrow a. \} \}$.

Nous donnons un tableau permettant d'illustrer cet exemple. Chaque case de ce tableau contient le nombre de règles d'un R -ensemble fort S_i appartenant à une base de croyances K_j . Nous notons les R -ensembles potentiels forts : $S_1 = \{ \neg a. \neg c. \}$, $S_2 = \{ \neg a. \ c \leftarrow a. \}$, $S_3 = \{ a \leftarrow not\ b. \ a \leftarrow not\ b. \ a \leftarrow not\ b. \}$. Les R -ensembles potentiels forts grisés sont les R -ensembles forts suivant la stratégie correspondant à la colonne.

S_i	K_1	K_2	K_3	Max	$GMax$
S_1	1	1	0	1	(1,1,0)
S_2	1	0	1	1	(1,1,0)
S_3	1	1	1	1	(1,1,1)

Nous étudions maintenant les propriétés de l'opération forte de fusion $\Delta_{GMax,IC}^{\Pi RSF,s}(E)$ dans ce contexte.

Proposition 9.1.4. *L'opération forte de fusion $\Delta_{GMax,IC}^{\Pi RSF,s}(E)$ respecte les postulats de fusion IC0, IC1 et IC2.*

Strategies	(IC0)	(IC1)	(IC2)	(IC3)	(IC4)	(IC5)	(IC6)	(IC7)	(IC8)
$GMax$	✓	✓	✓	×	×	×	×	×	×

Preuves et contre-exemples

Les preuves et les contre-exemples sont similaires ceux donnés pour l'opération $\Delta_{\Sigma,IC}^{\Pi RSF,s}(E)$.

9.2 Deuxième approche : fusion faible par R-ensembles

Certains ensembles de règles ne possèdent pas de modèles stables car elles impliquent des ensembles d'atomes incohérents et d'autres ne possèdent pas de modèles stables car il est impossible de trouver une justification pour leurs conséquences. Par exemple, le programme $\Pi = \{\neg a. \quad a \leftarrow b. \quad b.\}$ a pour ensemble de conséquences immédiates $\{a, \neg a, b\}$ qui est incohérent. Nous pouvons appeler ce type d'incohérences, des incohérences fortes. En revanche, le programme logique $\Pi' = \{a \leftarrow not \ b. \quad b \leftarrow not \ c. \quad c \leftarrow not \ a.\}$ n'implique pas d'ensembles incohérents d'atomes mais ne possède pas de modèles stables car Π' ne peut pas déduire d'ensembles d'atomes ayant une justification. Nous pouvons appeler ce type d'incohérences, des incohérences faibles.

Définition 9.2.1. *Soit K une base de croyances représenté par un programme logique avec sémantique des modèles stables. K est faiblement cohérente si et seulement si elle possède au moins un HT-modèle.*

Dans le cas du programme Π , il n'existe pas d'ensemble de règles K tel que $\Pi \cup K$ possède au moins un modèle stable. Ainsi, la seule possibilité de restaurer la cohérence dans Π est de retirer quelques règles. D'un autre côté, le programme Π' n'est pas intrinsèquement incohérent. Il est possible de restaurer la cohérence sans perdre de croyances, par exemple, l'union $\Pi' \cup \{a.\}$ possède le modèle stable $\{a, b\}$.

Exemple 9.2.1. *Nous illustrons les notions liées à l'opérateur faible de fusion grâce à l'exemple suivant. Soit le profil de croyances $E = \{K_1, K_2\}$ avec :*

$$K_1 = \left\{ \begin{array}{ll} a \leftarrow not \ b. & rc \leftarrow not \ a. \\ e. & d \leftarrow f. \end{array} \right\}$$

$$K_2 = \left\{ \begin{array}{ll} b \leftarrow \text{not } c. & \neg d. \\ d \leftarrow e. & f. \end{array} \right\}$$

Le programme $K_1 \sqcup K_2$ est incohérent. Il peut être séparé en deux parties indépendantes qui sont chacune incohérente. Ainsi :

- Les règles $\{a \leftarrow \text{not } b. \quad c \leftarrow \text{not } a. \quad b \leftarrow \text{not } c.\}$ ne trouvent pas de justifications ;
- Les règles $\{e. \quad d \leftarrow f. \quad \neg d. \quad d \leftarrow e. \quad f.\}$ conduisent à la déduction d'un ensemble d'atomes incohérents : $\{d, \neg d, e, f\}$.

Quelle que soit la stratégie P choisie, les R -ensembles forts de $\Delta_{P, \top}^{\Pi R S F, s}(K_1 \sqcup K_2)$ sont $\{\neg d. \quad a \leftarrow \text{not } b.\}$, $\{\neg d. \quad b \leftarrow \text{not } c.\}$ et $\{\neg d. \quad c \leftarrow \text{not } a.\}$ qui résolvent la seconde incohérence grâce au retrait de $\neg d$ et qui résolvent l'incohérence faible grâce au retrait de la deuxième règle.

Nous proposons ici une définition alternative de la notion de R -ensembles, et a fortiori d'opération de fusion, afin de ne pas perdre de croyances inutilement par le retrait de règles qui ne résoudraient qu'une incohérence faible. Cette définition est basée sur une notion de l'incohérence plus faible que celle employée pour l'opération forte de fusion. Nous proposons de considérer comme incohérent un programme Π qui ne possède pas de HT-modèles.

Proposition 9.2.1. *Soit Π un programme logique. Si Π est faiblement incohérent, alors Π est incohérent.*

Preuve.

Trivial au regard de la proposition 2.4.1. □

Définition 9.2.2 (R -ensemble potentiel faible). *Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un programme logique représentant les contraintes tels que $K_1 \sqcup \dots \sqcup K_n \sqcup IC$ est incohérent. Soit X un sous-ensemble de formules de $K_1 \sqcup \dots \sqcup K_n$. X est R -ensemble potentiel faible de E contraint par IC si et seulement si $((K_1 \sqcup \dots \sqcup K_n) \setminus X) \sqcup IC$ possède au moins un HT-modèle.*

Exemple 9.2.2. *Reprenons le profil E de l'exemple 9.2.1. Les R -ensembles potentiels forts de ce profil sont :*

$$\begin{array}{ll} \{\neg d. \quad a \leftarrow \text{not } b.\} & \{\neg d. \quad b \leftarrow \text{not } c.\} \\ \{\neg d. \quad c \leftarrow \text{not } a.\} & \{e. \quad f. \quad a \leftarrow \text{not } b.\} \\ \{e. \quad f. \quad b \leftarrow \text{not } c.\} & \{e. \quad f. \quad c \leftarrow \text{not } a.\} \\ \{e. \quad d \leftarrow f. \quad a \leftarrow \text{not } b.\} & \{e. \quad d \leftarrow f. \quad b \leftarrow \text{not } c.\} \\ \{e. \quad d \leftarrow f. \quad c \leftarrow \text{not } a.\} & \{d \leftarrow e. \quad f. \quad a \leftarrow \text{not } b.\} \\ \{d \leftarrow e. \quad f. \quad b \leftarrow \text{not } c.\} & \{d \leftarrow e. \quad f. \quad c \leftarrow \text{not } a.\} \\ \{d \leftarrow e. \quad d \leftarrow f. \quad a \leftarrow \text{not } b.\} & \{d \leftarrow e. \quad d \leftarrow f. \quad b \leftarrow \text{not } c.\} \\ \{d \leftarrow e. \quad d \leftarrow f. \quad c \leftarrow \text{not } a.\} & \end{array}$$

ainsi que tous les ensembles de règles incluant ces règles-là.

Alors que les R-ensembles potentiels faibles sont :

$$\begin{array}{ccc} \{\neg d.\} & \{e. \quad f.\} & \{e. \quad d \leftarrow f.\} \\ \{d \leftarrow e. \quad f.\} & \{d \leftarrow e. \quad d \leftarrow f.\} & \end{array}$$

ainsi que tous les ensembles de règles incluant ces règles-là. Nous pouvons voir que, n'ayant pas à résoudre l'incohérence faible, les R-ensembles potentiels faibles sont plus petits que les R-ensembles potentiels forts.

Nous définissons les R-ensembles faibles comme étant les R-ensembles potentiels faibles préférés au sens de la stratégie P choisie.

Définition 9.2.3 (R-ensemble faible). Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un programme logique représentant les contraintes tels que $K_1 \sqcup \dots \sqcup K_n \sqcup IC$ est incohérent. Soit P la stratégie de fusion. $X \subseteq K_1 \sqcup \dots \sqcup K_n$ est un R-ensemble faible de $K_1 \sqcup \dots \sqcup K_n$ contraint par IC selon la stratégie P si et seulement si

1. X est un R-ensemble potentiel faible de E contraint par IC ;
2. Il n'existe pas de R-ensemble potentiel faible Y de E contraint par IC tel que $Y \subset X$.
3. Il n'existe pas de R-ensemble potentiel faible Y de E contraint par IC tel que $Y <_P X$.

Exemple 9.2.3. Reprenons le profil E de l'exemple 9.2.1.

Ainsi, peu importe la stratégie P , les R-ensembles forts de E selon P sont : $\{\{\neg d. \quad a \leftarrow \text{not } b.\}, \{\neg d. \quad b \leftarrow \text{not } c.\}, \{\neg d. \quad c \leftarrow \text{not } a.\}\}$

Alors que l'ensemble des R-ensembles faibles de E selon P est $\{\{\neg d.\}\}$

La collection des R-ensembles faibles de E contraint par IC selon la stratégie P est notée $\mathcal{F}_{P,IC}^{\Pi RSF,w} \mathcal{R}(E)$

Définition 9.2.4 (Opération faible de fusion). Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances et IC un programme logique représentant les contraintes tels que $K_1 \sqcup \dots \sqcup K_n \sqcup IC$ est incohérent. Soit P une stratégie de fusion. L'opération faible de fusion $\Delta_{P,IC}^{\Pi RSF,w}(E)$ est définie par :

$$\Delta_{P,IC}^{\Pi RSF,w}(E) = \bigvee_{X \in \mathcal{F}_{P,IC}^{\Pi RSF,w} \mathcal{R}(E)} \{((K_1 \sqcup \dots \sqcup K_n) \setminus X) \sqcup IC\}$$

Exemple 9.2.4. Terminons l'exemple 9.2.1. L'opération faible de fusion de E quelle que soit la stratégie P aura pour résultat :

$$\Delta_{P,IC}^{\Pi RSF,w}(E) = \left\{ \begin{array}{ccc} a \leftarrow \text{not } b. & c \leftarrow \text{not } a. & e. \\ d \leftarrow f. & b \leftarrow \text{not } c. & d \leftarrow e. \\ & f. & \end{array} \right\}.$$

9.2.1 Operation de fusion faible et postulats IC

Nous étudions maintenant, stratégie par stratégie, les propriétés de l'opération de fusion $\Delta_{P,IC}^{\Pi RSF,w}(E)$.

9.2.1.1 La stratégie Somme

Nous considérons maintenant la stratégie *Somme* et nous utilisons le pré-ordre défini en 9.1.1.2.

Proposition 9.2.2. *Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. L'opération de fusion $\Delta_{\Sigma,IC}^{\Pi RSF,w}(E)$ respecte les postulats de fusion **IC0**, **IC1** et **IC2**.*

Strategies	(IC0)	(IC1)	(IC2)	(IC3)	(IC4)	(IC5)	(IC6)	(IC7)	(IC8)
<i>Somme</i>	✓	✓	✓	×	×	×	×	×	×

Preuves et contre-exemples

Nous donnons maintenant les preuves et les contre-exemples de cette proposition postulat par postulat. Nous rappelons que : $\mathcal{F}_{P,IC}^{\Pi RSF,w} \mathcal{F}(E) = \cup_{X \in \mathcal{F}_{P,IC}^{\Pi RSF,w} \mathcal{R}(E)} (E \setminus X)$.

(IC0)

Nous savons que $HT(\Pi \sqcup \Pi') = HT(\Pi) \cap HT(\Pi')$ par l'intermédiaire de la proposition 2.4.1. Nous savons également que

$$\Delta_{P,IC}^{\Pi RSF,w}(E) = \bigvee_{X \in \mathcal{F}_{P,IC}^{\Pi RSF,w} \mathcal{R}(E)} \{((K_1 \sqcup \dots \sqcup K_n) \setminus X) \sqcup IC\}.$$

Par construction, nous pouvons déduire que $IC \subseteq \Delta_{\Sigma,IC}^{\Pi RSF,w}(E)$ et donc $HT(IC) \subseteq HT(\Delta_{\Sigma,IC}^{\Pi RSF,w}(E))$.

Le postulat **(IC0)** est donc vérifié.

(IC1)

Tous les éléments de $F \in \mathcal{F}_{\Sigma,IC}^{\Pi RSF,w} \mathcal{F}(E)$ sont définis tel que $F \sqcup IC$ est cohérent. Ainsi, par définition de l'opération $\Delta_{\Sigma,IC}^{\Pi RSF,w}(E)$, $\Delta_{\Sigma,IC}^{\Pi RSF,w}(E)$ est cohérent si IC est faiblement cohérent.

Le postulat **(IC1)** est donc vérifié.

(IC2)

Si $E \sqcup IC$ est faiblement cohérent, alors par définition $\mathcal{F}_{\Sigma,IC}^{\Pi RSF,w} \mathcal{R}(E) = \{\emptyset\}$:

- $E \sqcup IC$ est faiblement cohérent ;
- $\emptyset$ est minimal pour l'inclusion ;
- Et il ne peut exister de R-ensemble potentiel plus petit que $\emptyset$ au sens de $\leq_{\Sigma}$.

et donc il n'y qu'un seul élément dans $\mathcal{F}_{\Sigma,IC}^{\Pi RSF,w} \mathcal{F}(E)$ qui est égal à E .

(IC3)

Nous donnons un contre-exemple invalidant **(IC3)**.

Soient E_1 et E_2 deux profils de croyances tels que $E_1 = \{K_1, K_3\}$ et $E_2 = \{K_2, K_3\}$ et $IC = \top$ avec :

$$K_1 = \{a \leftarrow \text{not } b. \quad c \leftarrow a. \quad a \leftarrow c.\}$$

$$K_2 = \{c \leftarrow \text{not } b. \quad c \leftarrow a. \quad a \leftarrow c.\}$$

$$K_3 = \{\neg a.\}$$

Les ensembles de HT-modèles de K_1 et K_2 sont identiques à savoir $HT(K_1) = HT(K_2) = \{ (\emptyset, \{a, b, c\}) (\{b\}, \{a, b, c\}) (\{a, c\}, \{a, c\}) (\emptyset, \{b\}) (\{b\}, \{b\}) \}$. E_1 et E_2 sont donc équivalents puisqu'il existe une bijection $f(K_1) = K_2$ et $f(K_3) = K_3$ entre les deux profils telle que $f(K_1) \equiv K_2$ et $f(K_3) \equiv K_3$.

Cependant les résultats obtenus par $\Delta_{\Sigma, IC}^{\Pi RSF, w}(E_1)$ et $\Delta_{\Sigma, IC}^{\Pi RSF, w}(E_2)$ ne sont pas équivalents, à savoir :

$$\Delta_{\Sigma, IC}^{\Pi RSF, w}(E_1) = \{c \leftarrow a. \quad a \leftarrow c. \quad \neg a.\}$$

et

$$\begin{aligned} \Delta_{\Sigma, IC}^{\Pi RSF, w}(E_2) &= \{ \{c \leftarrow a. \quad a \leftarrow c. \quad \neg a.\} \\ &\quad \vee \quad \{c \leftarrow \text{not } b. \quad c \leftarrow a. \quad \neg a.\} \} \end{aligned}$$

Le postulat **(IC3)** n'est donc pas vérifié.

(IC4)

Nous donnons un contre-exemple invalidant **(IC4)**.

Soient $K_1 = \{a \leftarrow \text{not } b. \quad b \leftarrow \text{not } a. \quad c \leftarrow a. \quad c \leftarrow b.\}$, $K_2 = \{\neg c.\}$ et $IC = \top$.

Nous avons ici $\Delta_{\Sigma, IC}^{\Pi RSF, w}(K_1 \sqcup K_2) = K_1$ ainsi $\Delta_{\Sigma, IC}^{\Pi RSF, w}(K_1 \sqcup K_2) \sqcup K_1$ est faiblement cohérent mais $\Delta_{\Sigma, IC}^{\Pi RSF, w}(K_1 \sqcup K_2) \sqcup K_2$ n'est pas faiblement cohérent.

Le postulat **(IC4)** n'est donc pas vérifié.

(IC5) et (IC6)

Nous donnons un contre-exemple invalidant **(IC5)** et **(IC6)**.

Soient $E_1 = \{K_1, K_2\}$ et $E_2 = \{K_3, K_4\}$ deux profils de croyances et $IC = \top$ avec les bases de croyances suivantes :

$$K_1 = \{ \quad a \leftarrow \text{not } \neg h. \quad c \leftarrow b. \quad \}$$

$$K_2 = \left\{ \begin{array}{l} c \leftarrow a. \quad b \leftarrow \text{not } \neg h. \\ \neg c. \end{array} \right\}$$

$$K_3 = \{ \quad d \leftarrow \text{not } \neg c. \quad h \leftarrow e. \quad \}$$

$$K_4 = \left\{ \begin{array}{l} h \leftarrow d. \quad e \leftarrow \text{not } \neg c. \\ \neg h. \end{array} \right\}$$

$\Delta_{\Sigma, \top}^{\Pi RSF, w}(E_1) = \{a \leftarrow \text{not } \neg h. \quad b \leftarrow \text{not } \neg h. \quad c \leftarrow a. \quad c \leftarrow b.\}$ et tous les HT-modèles du programme $\Delta_{\Sigma, \top}^{\Pi RSF, w}(E_1)$ ont c dans le premier ensemble d'atomes de la paire.

$\Delta_{\Sigma, \top}^{\Pi RSF, w}(E_2) = \{d \leftarrow \text{not } \neg c. \quad e \leftarrow \text{not } \neg c. \quad h \leftarrow d. \quad h \leftarrow e.\}$ et tous les HT-modèles du programme $\Delta_{\Sigma, \top}^{\Pi RSF, w}(E_2)$ ont h dans le premier ensemble d'atomes de la paire.

$\Delta_{\Sigma, \top}^{\Pi RSF, w}(E_1 \sqcup E_2) = E_1 \cup E_2$ et $\neg c$ et $\neg h$ sont dans le premier ensemble d'atomes de la paire de tous les HT-modèles du programme.

Ainsi, pour tout $(X, Y) \in HT(\Delta_{\Sigma, \top}^{\Pi RSF, w}(E_1))$, nous avons $(X, Y) \notin HT(\Delta_{\Sigma, \top}^{\Pi RSF, w}(E_1 \sqcup E_2))$ ce qui invalide le postulat **(IC5)**.

Et pour tout $(X, Y) \in HT(\Delta_{\Sigma, \top}^{\Pi RSF, w}(E_1 \sqcup E_2))$, nous avons $(X, Y) \notin HT(\Delta_{\Sigma, \top}^{\Pi RSF, w}(E_2))$ ce qui invalide le postulat **(IC6)**.

Les postulats **(IC5)** et **(IC6)** ne sont donc pas vérifiés.

(IC7) et (IC8)

Soit $E = \{K_1, K_2\}$ un profil de croyances et $IC_1 = \top$ et $IC_2 = \{c.\}$ avec les bases de croyances suivantes :

$$K_1 = \left\{ \begin{array}{l} d \leftarrow b. \quad b \leftarrow \text{not } c. \end{array} \right\}$$

$$K_2 = \left\{ \begin{array}{l} a \leftarrow \text{not } c. \quad \neg d. \quad d \leftarrow a. \end{array} \right\}$$

$\Delta_{\Sigma, \top}^{\Pi RSF, w}(E) = \{a \leftarrow \text{not } c. \quad b \leftarrow \text{not } c. \quad d \leftarrow a. \quad d \leftarrow b.\}$. Tous les HT-modèles de $\Delta_{\Sigma, \top}^{\Pi RSF, w}(E)$ ont d dans le premier ensemble d'atomes de la paire. Tous les HT-modèles de $\Delta_{\Sigma, \top \wedge IC_2}^{\Pi RSF, w}(E)$ ont $\neg d$ dans le premier ensemble d'atomes de la paire.

Ainsi, pour tout $(X, Y) \in HT(\Delta_{\Sigma, \top}^{\Pi RSF, w}(E))$, nous avons $(X, Y) \notin HT(\Delta_{\Sigma, IC_2}^{\Pi RSF, w}(E))$ ce qui invalide le postulat **(IC7)**. Et pour tout $(X, Y) \in HT(\Delta_{\Sigma, IC_2}^{\Pi RSF, w}(E))$, nous avons $(X, Y) \notin HT(\Delta_{\Sigma, \top}^{\Pi RSF, w}(E))$ ce qui invalide le postulat **(IC8)**.

Les postulats **(IC7)** et **(IC8)** ne sont donc pas vérifiés.

9.2.1.2 La stratégie Card

Nous considérons maintenant la stratégie *Card* et nous utilisons le pré-ordre défini en 9.1.1.3.

Proposition 9.2.3. *Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. L'opération de fusion $\Delta_{Card, IC}^{\Pi RSF, w}(E)$ respecte les postulats de fusion **IC0**, **IC1** et **IC2**.*

Strategies	(IC0)	(IC1)	(IC2)	(IC3)	(IC4)	(IC5)	(IC6)	(IC7)	(IC8)
<i>Card</i>	✓	✓	✓	×	×	×	×	×	×

Preuves et contre-exemples

Les preuves et les contre-exemples sont similaires à ceux donnés pour l'opération $\Delta_{\Sigma, IC}^{\Pi RSF, w}(E)$ sauf pour les postulats **(IC5)** et **(IC6)** que nous donnons ici.

(IC5) et (IC6)

Soient $E_1 = \{K_1, K_2, K_3\}$ et $E_2 = \{K_4, K_5, K_6\}$ deux profils de croyances et $IC = \top$ tels que :

$$K_1 = \{a \leftarrow not\ b. \quad b \leftarrow not\ a. \quad d \leftarrow a. \quad d \leftarrow b. \quad \neg d\}$$

$$K_2 = \{a \leftarrow not\ b. \quad b \leftarrow not\ a. \quad d \leftarrow a. \quad d \leftarrow b. \quad \neg e\}$$

$$K_3 = \{a \leftarrow not\ b. \quad b \leftarrow not\ a. \quad d \leftarrow a. \quad d \leftarrow b. \quad \neg f\}$$

$$K_4 = \{g \leftarrow not\ h. \quad h \leftarrow not\ g. \quad d \leftarrow g. \quad d \leftarrow h. \quad \neg d\}$$

$$K_5 = \{g \leftarrow not\ h. \quad h \leftarrow not\ g. \quad e \leftarrow g. \quad e \leftarrow h. \quad \neg e\}$$

$$K_6 = \{g \leftarrow not\ h. \quad h \leftarrow not\ g. \quad f \leftarrow g. \quad f \leftarrow h. \quad \neg f\}$$

$\Delta_{Card, \top}^{\Pi RSF, w}(E_1) = \{d \leftarrow a. \quad d \leftarrow b. \quad \neg d. \quad d \leftarrow a. \quad d \leftarrow b. \quad \neg e. \quad d \leftarrow a. \quad d \leftarrow b. \quad \neg f\}$ et tous les HT-modèles de $\Delta_{Card, \top}^{\Pi RSF, w}(E_1)$ contiennent $\neg d, \neg e$ et $\neg f$ dans le premier ensemble d'atomes de la paire.

$\Delta_{Card, \top}^{\Pi RSF, w}(E_2) = \{d \leftarrow g. \quad d \leftarrow h. \quad \neg d. \quad d \leftarrow g. \quad d \leftarrow h. \quad \neg e. \quad d \leftarrow g. \quad d \leftarrow h. \quad \neg f\}$ et tous les HT-modèles de $\Delta_{Card, \top}^{\Pi RSF, w}(E_2)$ contiennent $\neg d, \neg e$ et $\neg f$ dans le premier ensemble d'atomes de la paire.

$\Delta_{Card, \top}^{\Pi RSF, w}(E_1 \sqcup E_2) = \{a \leftarrow not\ b. \quad b \leftarrow not\ a. \quad d \leftarrow a. \quad d \leftarrow b. \quad e \leftarrow a. \quad e \leftarrow b. \quad f \leftarrow a. \quad f \leftarrow b. \quad g \leftarrow not\ h. \quad h \leftarrow not\ g. \quad d \leftarrow g. \quad d \leftarrow h. \quad e \leftarrow g. \quad e \leftarrow h. \quad f \leftarrow g. \quad f \leftarrow h.\}$ et tous les HT-modèles de $\Delta_{Card, \top}^{\Pi RSF, w}(E_2)$ contiennent d, e et f dans le premier ensemble d'atomes de la paire.

Les postulats **(IC5)** et **(IC6)** ne sont donc pas vérifiés.

9.2.1.3 La stratégie *Max*

Nous considérons maintenant la stratégie *Max* et nous utilisons le pré-ordre défini en 9.1.1.4.

Proposition 9.2.4. *Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. L'opération de fusion $\Delta_{Max, IC}^{\Pi RSF, w}(E)$ respecte les postulats de fusion **IC0**, **IC1** et **IC2**.*

Stratégies	(IC0)	(IC1)	(IC2)	(IC3)	(IC4)	(IC5)	(IC6)	(IC7)	(IC8)
<i>Max</i>	✓	✓	✓	×	×	×	×	×	×

Preuves et contre-exemples

Les preuves et les contre-exemples sont similaires à ceux donnés pour l'opération $\Delta_{\Sigma, IC}^{\Pi RSF, w}(E)$.

9.2.1.4 La stratégie *GMax*

Nous considérons maintenant la stratégie *GMax* et nous utilisons le pré-ordre défini en 9.1.1.5.

Proposition 9.2.5. *Soit $E = \{K_1, \dots, K_n\}$ un profil de croyances. L'opération de fusion $\Delta_{GMax, IC}^{\Pi RSF, w}(E)$ respecte les postulats de fusion **IC0**, **IC1** et **IC2**.*

Strategies	(IC0)	(IC1)	(IC2)	(IC3)	(IC4)	(IC5)	(IC6)	(IC7)	(IC8)
<i>GMax</i>	✓	✓	✓	×	×	×	×	×	×

Preuves et contre-exemples

Les preuves et les contre-exemples sont similaires à ceux donnés pour l'opération $\Delta_{\Sigma, IC}^{\Pi RSF, w}(E)$.

Conclusion

Dans ce chapitre, nous avons proposé une définition de la fusion par R-ensembles adaptée à la fusion de croyances exprimées sous forme de programmes logiques. La non-monotonie de la programmation logique, nous a permis de considérer différemment la définition de cohérence. Ainsi, nous avons proposé une définition de la cohérence basée sur les HT-modèles. Nous avons appelé cette cohérence, la cohérence faible et à partir de cette définition, nous avons également proposé une autre opération de fusion, que nous avons appelée opération de fusion faible. Celle-ci permet de réduire le nombre de formules perdues lors de l'opération de fusion. Pour chacune des opérations, faibles et fortes, de fusion nous avons adapté les stratégies classiques de fusion (*Somme*, *Card*, *Max* et *GMax*). Ce travail a fait l'objet de publications dans [HPW09b, HPW09a].

Conclusion et perspectives

Ce manuscrit a présenté une méthode originale de fusion syntaxique. Nous avons pris le parti de considérer les ensembles de formules à retirer d'un profil de croyances pour y restaurer la cohérence. Dans un premier temps, nous avons posé les bases formelles de l'opération de fusion par R-ensembles, appelée RSF, en présence ou pas de contraintes lorsque les bases de croyances sont des ensembles finis de formules propositionnelles [HPW07b, HPW07a, HPW08b, HPW08a]. Cette opération de fusion peut être adaptée selon différentes stratégies que nous avons étudiées (*Somme*, *Card*, *Max* et *GMax*). Nous avons étudié leurs propriétés vis-à-vis des postulats de Konieczny et Pino-Perez. Nous avons proposé une contrepartie sémantique pour tous les opérateurs et nous avons montré l'équivalence entre les opérateurs syntaxiques et leur contrepartie sémantique.

Ensuite, nous avons proposé une mise en œuvre de l'opération de fusion par R-ensembles basée sur la traduction du problème de fusion de profil E contraint par IC en un programme logique avec sémantique des modèles stables $\Pi_{E,IC}$. Nous avons montré que le programme logique issu de cette traduction permettait de déterminer correctement le résultat de l'opération de fusion par R-ensembles. C'est à dire que l'équivalence entre les modèles stables préférés de $\Pi_{E,IC}$ selon une stratégie P et les R-ensembles de E selon P . Nous avons implanté cette traduction de deux manières différentes : en adaptant l'algorithme Smodels [HPW07b, HPW07a] à nos besoins et en utilisant les instructions fournies par Lparse/Smodels et Gringo/CLASP de manière à fournir une implantation indépendante du solveur ASP [HPW08b, HPW08a].

Cette mise en œuvre de la fusion par R-ensembles a été illustrée par des expérimentations. Dans une première étape, nous avons étudié son comportement à l'aide de profils de croyances générés aléatoirement. Nous avons notamment comparé notre mise en œuvre avec celle présentée par Gorogiannis et Hunter. Dans une seconde étape, nous avons montré comment la fusion par R-ensembles a été adaptée dans un module de gestion de l'incohérence dans un projet européen VENUS portant sur le relevé en archéologie sous-marine. Tous ces résultats ont montré que la fusion par R-ensembles se comportait de manière satisfaisante.

Nous avons également proposé une généralisation de la fusion par R-ensembles au cas où des préférences sont exprimées entre les bases de croyances ainsi que dans le cas où il existe des préférences entre les bases de croyances et entre les formules appartenant à chaque base de croyances. Nous avons proposé de mettre en œuvre la fusion par R-

ensembles dans ce cadre à l'aide d'une nouvelle stratégie et nous avons montré que cette stratégie était équivalente à une opération de révision itérée.

Nous avons finalement proposé une extension de la fusion par R-ensembles au cas où les bases de croyances sont exprimées sous forme de programmes logiques [HPW09b, HPW09a]. En effet, nous avons au départ considéré les bases de croyances sous forme d'ensembles finis de formules en logique propositionnelle. Et nous avons utilisé la programmation logique avec sémantique des modèles stables pour calculer le résultat de la fusion. Il nous a semblé naturel d'utiliser également la programmation logique pour représenter les bases de croyances à fusionner. La représentation propositionnelle des bases de croyances paraît adaptée pour représenter des observations mais la programmation logique offre un cadre plus expressif pour représenter des règles, voire des règles avec exception. Nous avons étudié les stratégies classiques de fusion dans ce cadre. Nous avons constaté que, d'un certain point de vue, la définition directe de la fusion par R-ensembles conduisait à supprimer de façon inutile certaines croyances. Nous avons proposé une définition alternative de l'inférence entre programmes logiques et de la cohérence d'un programme logique basée sur la logique du Here-and-There. Nous avons également étudié les propriétés des opérations fortes et faibles que nous avons définies.

Les travaux que nous avons développés dans cette thèse ouvre plusieurs perspectives.

Une première perspective serait bien évidemment de mettre en oeuvre la fusion par R-ensembles de programmes logiques. Une première mise en oeuvre utilisant la traduction du problème de fusion en programme logique a été proposé dans [HPW09b, HPW09a]. De la même manière, une mise en oeuvre des opérateurs de fusion avec préférences *PRSF* pourrait être envisagée. La programmation logique possibiliste semble fournir un cadre privilégié pour cette adaptation. Une étude expérimentale sur les opérateurs de fusion avec préférences *PRSF* ainsi que de fusion forte ou faible pour des programmes logiques $\Pi RSF, s$ et $\Pi RSF, w$ devrait être menée. Cette expérimentation permettrait de comparer le comportement théorique et pratique des opérations de fusion par R-ensembles que nous avons définies. De la même manière, il serait intéressant de comparer durant cette étude le comportement des opérateurs *RSF* et ceux de ΠRSF .

Les postulats de Konieczny et Pino-Perez pour la fusion ont été décrits dans le cadre de bases de croyances propositionnelles et nous avons vu dans le cadre de notre étude de la fusion par R-ensembles de croyances exprimées sous forme de programmes logiques que ces postulats ne sont pas adaptés aux opérations de fusion qui sont décrites dans le cadre de logique non-monotone. Une réflexion portant sur l'analyse critique et la reformulation éventuelle des postulats dans un cadre plus général pourrait être menée.

Il serait intéressant d'étudier la complexité théorique des opérations de fusion que nous avons décrits dans cette thèse. En particulier, pour la stratégie *GMax* dont la complexité semble être plus élevée que pour les autres stratégies.

Lorsque les bases de croyances sont représentées par des programmes logiques une direction consistant à déterminer non pas l'ensemble de règles à retirer mais plutôt l'en-

semble des règles à ajouter pour effectuer la fusion pourrait être étudiée.

Récemment, Everaere et al. [EKM08] ont proposé un postulat de tempérance visant à assurer que chacun des croyances de chacun des agents soit cohérent avec le résultat de la fusion. Il serait intéressant de mener une étude sur la complexité des opérateurs respectant ce postulat ainsi que sur sa possible implantation dans notre cadre.

L'expérimentation de la fusion par R-ensembles que nous avons menée dans le cadre du projet VENUS ne portait pas sur la fusion mais sur la gestion de l'incohérence. Un domaine pour lequel la fusion semble particulièrement appropriée pourrait être celui de la construction d'un système expert. Lors de la construction d'un système expert, la première étape est l'interrogation des experts pour qu'ils décrivent leur conception du métier à modéliser. Puis, après avoir modélisé ces interrogations dans des bases de croyances, une deuxième phase d'interrogation nécessite de réunir les différents experts afin de résoudre les incohérences relevées dans la première phase. Ainsi, ces réunions, coûteuses en temps, pourraient être remplacées avantageusement par les opérations de fusion. L'expérimentation à partir des données d'un système expert permettrait de comparer les résultats obtenus avec ceux qui sont ressortis des réunions entre experts.

La phase d'expérimentation a également permis de mettre en relief les difficultés à comparer deux mises en œuvre de fusion. La création et la mise à la disposition de la communauté d'un ensemble de profils de croyances pourrait simplifier cette comparaison. Une idée pourrait être d'utiliser les instances incohérentes des collections de problèmes SAT comme SATLIB [HS00] et de les séparer en bases de croyances afin de créer une collection de profils de croyances à fusionner.

Index

- $<_B$ -préservation, 48
- $<_D$ -préservation, 48
- $<_W$ -préservation, 48
 - contre-modèle, 22
 - modèle, 22
- Algorithme ASSAT, 56
- Algorithme de Quine, 18
- Algorithme DPLL, 19
- arbitrage, 73
- arbre fermé, 17
- arité, 33
- ASSAT, 54
- atome, 21
- atome possibiliste, 58
- base de croyances, 66
- base de Herbrand, 24
- boucle, 55
- branche fermée, 17
- clause, 16
- clôture déductive, 38, 66
- CNF, 16
- Coloriage R-V, 53
- complétion de clark, 55
- conséquence, 13
- conséquences immédiates possibilistes, 59
- contraction, 62
- Contrepartie sémantique, 108
- corps, 34
- couverture, 49
- distance de Hamming, 66
- distribution de possibilité, 25
- déduction, 14
- démonstration, 14
- ensemble, 11
- ensemble non-fondé, 38
- fait, 32
- fait universel, 34
- flock, 70
- foncteur, 33
- formule, 21
- formule close, 21
- formule de boucle, 55
- formule fermée, 21
- Full meet contraction, 64
- fusion
 - opérateur de combinaison, 77
 - postulats FP, 72
 - postulats KP, 71
- fusion contrainte, 70
- Fusion de croyances, 70
- fusion possibiliste, 83
- fusion pure, 70
- fusion reversible, 79
- Graphe de dépendance de règles, 50
- Graphe de dépendance étendu, 53
- généralisation, 35
- HT-logique, 42
- HT-modèle, 42
- identité, 35
- identité de Harper, 63
- identité de Levi, 63
- indépendance à la majorité, 73
- instance, 33
- instanciation, 33, 35
- interprétation, 22, 37

- Kernel Normal Form, 52
- Kernel-EDG, 53
- littéral, 16
- logique des prédicats, 21
 - cohérente, 22
 - incohérence, 22
 - insatisfaisabilité, 22
 - satisfaisable, 22
 - tautologie, 22
- logique des prédicats
 - cohérente, 190
- logique du premier ordre, 21
- logique possibiliste, 24
- logique propositionnelle
 - cohérente, 13
 - contre-modèle, 13
 - déduction sémantique, 12
 - incohérence, 13
 - insatisfaisabilité, 13
 - modèle, 12, 13
 - satisfaisable, 13
 - tautologie, 13
- Lparse, 49
- majorité, 73
- Maxi choice contraction, 64
- mesure de nécessité, 24
- mesure de possibilité, 24
- modus ponens, 14, 34
- modèle bien-fondé, 40
- modèle minimal, 37
- modèle partiel bien-fondé, 39
- modèle stable, 40
- modèle stable possibiliste, 58, 60
- modèles stables, 41
- nogoods, 57
- noMoRe, 50
- négation par echec, 35
- opérateur
 - majoritaire drastique, 78
- ordre, 11
 - strict, 11
- Partial meet contraction, 64
- plus petit ensemble clos, 38
- postulats
 - AGM, 63
 - KM, 64
- profil de croyances, 70
- programme basique, 38
- programme général, 38
- programme logique, 34
 - cohérence, 170
 - cohérence faible, 179
- programme logique possibiliste, 58, 59
- programme étendu, 38
- PROLOG, 31
 - sémantique déclarative, 37
 - sémantique dénotationnelle, 37
 - sémantique opérationnelle, 35
- pré-ordre, 11
- préprocesseur, 43
- question, 35
- R-ensemble, 68, 93, 95, 165
- R-ensemble faible, 181
- R-ensemble fort, 171
- R-ensemble potentiel, 92, 95
- R-ensemble potentiel faible, 180
- R-ensemble potentiel fort, 170
- relation, 11
 - antisymétrique, 11
 - irreflexive, 11
 - partielle, 11
 - reflexive, 11
 - symétrique, 11
 - totale, 11
 - transitive, 11
 - équivalence, 11
- revision, 62
- règle, 34
- règle possibiliste, 58
- règles génératrices, 47
- réduction, 41
- résolution unitaire, 19
- résolvante, 20

- révision, 62
 - assignement fidèle, 65
 - itérée, 68
 - par R-ensembles, 68
 - postulats DP, 69
 - R-ensembles, 68
 - théorème de représentation, 65
- semi-décidabilité, 23
- SLD-résolution, 36
- Smodels, 49
- solveur, 43
- stratégie, 92
- substitution, 14, 33
- système de Herbrand, 24
- système formel, 14
- séquence, 47
- séquence fondée, 47
- terme, 21, 33
- terme composé, 33
- terme simple, 33
- tete, 34
- théorie, 63
- théorème, 15
- théorème de déduction, 15
- théorème de Herbrand, 24
- transformation de Gelfond-Lifschitz, 41
- univers de Herbrand, 24
- variable, 32
- variable libre, 21
- variable liée, 21
- équivalence sémantique, 13

Bibliographie

- [AGM85] C. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change : Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50(2) :510–530, 1985.
- [AKL02] C. Anger, K. Konczak, and T. Linke. Nomore : Non-monotonic reasoning with logic programs. In *Proceedings of JELIA02*, pages 521–524, 2002.
- [BBNJ⁺05] S. Benferhat, J. Ben-Naim, R. Jeansoulin, M. Khelfallah, S. Lagrue, O. Papini, N. Wilson, and E. Würbel. Belief revision of gis systems : The results of rev!gis. In *Proceedings of ECSQARU05*, volume Volume 3571, pages 452–464, 2005.
- [BDKP00] S. Benferhat, D. Dubois, S. Kaci, and H. Prade. Encoding information fusion in possibilistic logic : A general framework for rational syntactic merging. In *Proceedings of ECAI00*, pages 3–7. Humboldt University, 2000.
- [BDKP02a] S. Benferhat, D. Dubois, S. Kaci, and H. Prade. Bipolar representation and fusion of preferences in the possibilistic logic framework. In *Proceedings of 8th International Conference on Principles of Knowledge Representation and Reasoning KR'2002*, pages 421–432. Morgan Kaufmann Publishers, Inc., 2002.
- [BDKP02b] S. Benferhat, D. Dubois, S. Kaci, and H. Prade. Possibilistic Merging and Distance-based Fusion of Propositional Information. *Proceedings of AMAI02*, 34((1-3)) :217–252, 2002.
- [BDLP02] S. Benferhat, D. Dubois, S. Lagrue, and O. Papini. Making revision reversible : a method based on polynomials. *Fundamenta Informaticae*, 53(3-4) :251–280, 2002.
- [BE99] G. Brewka and T. Eiter. Preferred answer sets for extended logic programs. *Artificial Intelligence*, 109 :297–356, 1999.
- [BK03] S. Benferhat and S. Kaci. Fusion of possibilistic knowledge bases from a postulate point of view. *International Journal of Approximate Reasoning*, 33 :255–285, 2003.
- [BKM91] C. Baral, S. Kraus, and J. Minker. Combining multiple knowledge bases. *IEEE Trans. on Knowl. and Data Eng.*, 3(2) :208–220, 1991.

- [BKMS91] C. Baral, S. Kraus, J. Minker, and V. S. Subrahmanian. Combining knowledge bases consisting of first order theories. In *Proceedings of ISMIS*, pages 92–101, 1991.
- [BL02] I. Bloch and J. Lang. Towards mathematical morpho-logics. In *Technologies for constructing intelligent systems : tools*, pages 367–380, 2002.
- [BnBPW04] J. Ben-naim, S. Benferhat, O. Papini, and E. Würbel. An answer set programming encoding of prioritized removed sets revision : application to gis. In *Proceedings of JELIA'04*, pages 604–616, 2004.
- [Bor85] A. Borgida. Language Features for Flexible Handling of Exceptions in Information Systems. *ACM Transactions on DataBase System*, 10, 1985.
- [Bou93] C. Boutilier. Revision Sequences and Nested Conditionals. In *Proceedings of IJCAI93*, pages 519–531, 1993.
- [Cho98] L. Cholvy. Reasoning about merging information. *Handbook of DRUMS*, 3 :233–263, 1998.
- [Chu36] A. Church. A note on the entscheidungsproblem. *Journal of Symbolic Logic*, 1(1) :40–41, 1936.
- [Coo71] S. Cook. The complexity of theorem-proving procedures. In *STOC*, pages 151–158, 1971.
- [Dal88] M. Dalal. Investigations into a theory of knowledge base revision. In *Proceedings of AAAI88*, pages 475–479, 1988.
- [Dav80] H. David. Logic programming and compiler writing. *Softw., Pract. Exper.*, 10(2) :97–125, 1980.
- [DDC09] S. Destercke, D. Dubois, and E. Chojnacki. Possibilistic information fusion using maximal coherent subsets. *IEEE Transactions on Fuzzy Systems*, 17(1) :79–92, 2009.
- [DDL06] J. Delgrande, D. Dubois, and J. Lang. Iterated revision as prioritized merging. In *Proceedings of KR06*, pages 210–220, 2006.
- [Del97] J. Delgrande. Compiling reasoning with and about preferences into default logic. In *Proceedings of IJCAI97*, pages 168–174, 1997.
- [DGC⁺04] P. Drap, P. Grussenmeyer, P.-Y. Curtinot, J. Seinturier, and G. Gaillard. Presentation of the web based arpenteur tools : Towards a photogrammetry based heritage information system. In Orhan Altan, editor, *Proceedings of ISPRS04*, volume Vol. XXXV, part B6, pages 123–129. ISPRS organisation committee, 2004.
- [DLP94] D. Dubois, J. Lang, and H. Prade. Possibilistic Logic. *Handbook of Logic in Artificial Intelligence and Logic Programming*, 3 :439–513, 1994.
- [DP60] M. Davis and H. Putnam. A computing procedure for quantification theory. *J. ACM*, 7(3) :201–215, 1960.

- [DP92] D. Dubois and H. Prade. Belief change and possibility theory. In P. Gärdenfors Eds, editor, *Belief Revision*, pages 142–182. Cambridge University Press, 1992.
- [DP94] D. Dubois and H. Prade. Possibilistic logic. In *Handbook of logic in Artificial Intelligence and logic programming*, pages 439–513, 1994.
- [DP97a] A. Darwiche and J. Pearl. On the logic of iterated belief revision. *Artificial intelligence*, 89 :5–23, 1997.
- [DP97b] D. Dubois and H. Prade. A synthetic view of belief revision with uncertain inputs in the framework of possibility theory. *Int. J. Approx. Reasoning*, 17 :295–324, 1997.
- [DSS⁺07] P. Drap, J. Seinturier, D. Scaradozzi, P. Gambogi, L. Long, and F. Gauch. Photogrammetry for virtual exploration of underwater archeological sites. In Andreas Georgopoulos, editor, *Proceedings of CIPA07*, pages 281–286, Athens, Greece, 2007.
- [DST02] J. Delgrande, T. Schaub, and H. Tompits. A framework for compiling preferences in logic programs. In *Theory and Practice of Logic Programming*, 2002.
- [DSTW08] J. Delgrande, T. Schaub, H. Tompits, and S. Woltran. Belief revision of logic programs under answer set semantics. In *Proceedings of KR08*, pages 411–421, 2008.
- [EKM08] P. Everaere, S. Konieczny, and P. Marquis. Propositional merging operators based on set-theoretic closeness. In *Proceedings of ECAI08*, pages 737–738, 2008.
- [ELR05] E. Erdem, V. Lifschitz, and D. Ringe. Temporal phylogenetic networks and logic programming. *CoRR*, pages 539–558, 2005.
- [Fag94] F. Fages. Consistency of clark’s completion and existence of stable models, 1994.
- [FH99] N. Friedman and J. Halpern. Belief revision : a critique. *Journal of Logic, Language, and Information*, 8 :401–420, 1999.
- [FJK⁺86] S. Feferman, J. Dawson Jr., S. Kleene, G. Moore, R. Solovay, and J. van Heijenoort, editors. *Kurt Godel : collected works. Vol. 1 : Publications 1929-1936*. Oxford University Press, Inc., New York, NY, USA, 1986.
- [FUV83] R. Fagin, J. Ullman, and M. Vardi. On the semantics of updates in databases. In *Proceedings of PODS83*, pages 352–365, 1983.
- [GH08] N. Gorogiannis and A. Hunter. Implementing semantic merging operators using binary decision diagrams. *International Journal on Approximative Reasoning*, 49(1) :234–251, 2008.
- [GJ79] M. Garey and D. Johnson. *Computers and Intractability : A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)*. W. H. Freeman and Co Ltd, 1979.

- [GKNS07a] M. Gebser, B. Kaufmann, A. Neumann, and T. Schaub. clasp : A conflict-driven answer set solver. In *Proceedings of LPNMR07*, pages 260–265, 2007.
- [GKNS07b] M. Gebser, B. Kaufmann, A. Neumann, and T. Schaub. Conflict-driven answer set solving. In *Proceedings of IJCAI07*, pages 386–, 2007.
- [GKO⁺09] M. Gebser, R. Kaminski, M. Ostrowski, T. Schaub, and S. Thiele. On the input language of asp grounder gringo. In *Proceedings of LPNMR09*, pages 502–508, 2009.
- [GL88] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *Proceedings of ICLP*, pages 1070–1080, 1988.
- [GLM04] E. Giunchiglia, Y. Lierler, and M. Maratea. Sat-based answer set programming. In *Proceedings of AAAI04*, pages 25–29, 2004.
- [GM05] G. Grossi and M. Marchi. A new algorithm for answer set computation. In *Answer Set Programming*, 2005.
- [Gro88] A. Grove. Two modellings for theory change. *Journal of Philosophical Logic*, 17(2) :157–170, 1988.
- [Har75] William L. Harper. Rational belief change, popper functions and counterfactuals. *Synthese*, 30(1-2), 1975.
- [Hey30] A. Heyting. Die formalen regeln der intuitionistischen logik. *Sitzungsberichte der Preussischen Akademie von Wissenschaften. Physikalisch-mathematische Klasse*, pages 42–56, 1930.
- [HPW07a] J. Hue, O. Papini, and E. Würbel. Fusion de bases propositionnelles : une approche basée sur les r-ensembles. In *Annales du LAMSADE*, pages 175–187, 2007.
- [HPW07b] J. Hue, O. Papini, and E. Würbel. Syntactic propositional belief bases fusion with removed sets. In *Proceedings of ECSQARU07*, pages 66–77, 2007.
- [HPW08a] J. Hué, O. Papini, and E. Würbel. Fusion par r-ensembles : une méthode complète de fusion. In *actes de IAF08*, pages 101–109, 2008.
- [HPW08b] J. Hué, O. Papini, and E. Würbel. Removed sets fusion : Performing off the shelf. In *Proceedings of ECAI08*, pages 94–98, 2008.
- [HPW09a] J. Hué, O. Papini, and E. Würbel. Fusion de bases de croyances exprimées sous la forme de programmes logiques. In *IAF09*, 2009.
- [HPW09b] J. Hué, O. Papini, and E. Würbel. Merging belief bases represented by logic programs. In *Proceedings of ECSQARU09*, pages 371–382, 2009.
- [HS00] H. Hoos and T. Stutzle. Satlib : An online resource for research on sat. In *Proceedings of SAT2000*, pages 283–292. IOS Press, 2000.
- [KLM02] S. Konieczny, J. Lang, and P. Marquis. Distance-based merging : a general framework and some complexity results. In *Proceedings of KR02*, pages 97–108, 2002.
- [KLM04] S. Konieczny, J. Lang, and P. Marquis. Da2 merging operators. *Artificial Intelligence*, 157(1-2) :49–79, 2004.

- [KM91] H. Katsuno and A. Mendelzon. Propositional knowledge base revision and minimal change. *Artificial Intelligence*, 52(3) :263–294, 1991.
- [Kon99] S. Konieczny. *Sur la logique du changement : révision et fusion de bases de connaissance*. PhD thesis, Université de Lille, 1999.
- [Kon00] S. Konieczny. On the difference between merging knowledge bases and combining them. In *Proceedings of KR00*, pages 135–144, 2000.
- [Kow88] R. Kowalski. The early years of logic programming. *Commun. ACM*, 31(1) :38–43, 1988.
- [KP98] S. Konieczny and R. Pino Pérez. On the logic of merging. In *Proceedings of KR98*, pages 488–498, 1998.
- [KP99] S. Konieczny and R. Pino Pérez. Merging with integrity constraints. *Proceedings of ECSQARU99*, 1638 :233–244, 1999.
- [KP02] S. Konieczny and R. Pino Perez. On the frontier between arbitration and majority. In *Proceedings of KR02*, pages 109–118, 2002.
- [Lag03] S. Lagrue. *Gestion d’informations partiellement ordonnées : raisonnement, révision et information géographique*. PhD thesis, Université du Sud-Toulon Var, 2003.
- [Leh95] D. Lehmann. Belief revision, revised. In *Proceedings of IJCAI95*, pages 1534–1540, 1995.
- [Lev80] Isaac Levi. *The enterprise of knowledge*. MIT Press, 1980.
- [Lew73] D. Lewis. Counterfactuals. *Journal of philosophical logic*, 1973.
- [Lif99] V. Lifschitz. Action languages, answer sets and planning. In *In The Logic Programming Paradigm : a 25-Year Perspective*, pages 357–373. Springer Verlag, 1999.
- [Lif08] V. Lifschitz. What is answer set programming? In *Proceedings of AAAI08*, pages 1594–1597, 2008.
- [Lin96] J. Lin. Integration of weighted knowledge bases. *Artificial Intelligence*, 83 :363–378, 1996.
- [LL00] C. Lafage and J. Lang. Logical representation of preferences for group decision making. In *Proceedings of KR00*, pages 457–468, 2000.
- [LL03] J. Lee and V. Lifschitz. Loop formulas for disjunctive logic programs. In *Proceedings of ICLP03*, pages 451–465. Springer, 2003.
- [LM98] J. Lin and A. O. Mendelzon. Merging databases under constraints. *International Journal of Cooperative Information Systems*, 7(1) :55–76, 1998.
- [LS95] P. Liberatore and M. Schaerf. Arbitration : A commutative operator for belief revision. In *Proceedings of WOCFAI95*, pages 217–228, 1995.
- [Luk64] J. Lukasiewicz. *Elements of mathematical logic*, 1964.
- [Luo97] V. P. Luong. Computing stable models of datalog programs with negation. *Actes des Journées francophones de programmation logique et programmation par contraintes*, pages 101–115, 1997.

- [LZ02] F. Lin and Y. Zhao. Assat : Computing answer sets of a logic program by sat solvers. In *Artificial Intelligence*, pages 112–117, 2002.
- [LZ04] F. Lin and Y. Zhao. Assat : computing answer sets of a logic program by sat solvers. *Artificial Intelligence*, 157(1-2) :115–137, 2004.
- [Men94] P. Menzies. [knowledge in flux : Modeling the dynamics of epistemic states. (peter gardenfors)]. *The Philosophical Review*, 103(1) :159–163, 1994.
- [MGC01] T. Meyer, A. Ghose, and S. Chopra. Syntactic representations of semantic merging operations. In *Proceedings of IJCAI01*, pages 87–91, 2001.
- [Neb91] B. Nebel. Belief Revision and Default Reasoning : Syntax-based Approach. In *Proceedings of KR91*, pages 417–427, 1991.
- [NFPS94] A. C. Nayak, N. Y. Foo, M. Pagnucco, and A. Sattar. Entrenchment kinematics. In *Proceedings of the Seventh Australian Joint Conference on Artificial Intelligence*, pages 157–164, 1994.
- [NFPS96] A. Nayak, N. Foo, M. Pagnucco, and A. Sattar. Changing conditional beliefs unconditionally. In *Proceedings of TARK96*, pages 119–135, 1996.
- [NGS05] P. Nicolas, L. Garcia, and I. Stéphan. Possibilistic stable models. In *Proceedings of IJCAI05*, pages 248–253, 2005.
- [Nie99] I. Niemelä. Logic programs with stable model semantics as a constraint programming paradigm. *Proceedings of AMAI99*, 1999.
- [Pap01] O. Papini. Iterated revision operations stemming from the history of an agent’s observations. *Frontiers of Belief Revision*, pages 279–301, 2001.
- [Pea96] J. Pearl. Qualitative probabilities for default reasoning, belief revision, and causal modeling. *Artificial Intelligence*, 84(1-2) :57–112, 1996.
- [PWC⁺09] O. Papini, E. Wurbel, O. Curé, P. Drap, M. Sérayet, J. Hué, J. Seinturier, B. Fertil, and D. Roussel. D3.6 reasoning with archaeological ontologies. technical report and prototype of software for the reversible fusion operations., July 2009.
- [PWJ⁺08] O. Papini, E. Wurbel, R. Jeansoulin, O. Curé, P. Drap, M. Sérayet, J. Hue, J. Seinturier, and L. Long. D3.4 representation of archaeological ontologies 1., July 2008.
- [Qi07] G. Qi. A model-based approach for merging prioritized knowledge bases in possibilistic logic. In *Proceedings of AAAI07*, pages 471–476, 2007.
- [Qui52] W. V. Quine. *Methods of logic*. Routledge and Kegan Paul, London :, 1952.
- [Rei80] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1-2) :81–132, 1980.
- [Rev93] P. Z. Revesz. On the semantics of theory change : arbitration between old and new information. 12th *ACM SIGACT-SGMIT-SIGART symposium on Principles of Databases*, pages 71–92, 1993.
- [Rev97] P. Z. Revesz. On the semantics of arbitration. *Journal of Algebra and Computation*, 7(2) :133–160, 1997.

- [RGJM86] R.Fagin, G.M.Kuper, J.D.Ullman, and M.Y.Vardi. Updating logical databases, 1986.
- [Rob65] J. A. Robinson. A machine-oriented logic based on the resolution principle. *J. ACM*, 12(1) :23–41, 1965.
- [SDP06] J. Seinturier, P. Drap, and O. Papini. Un cadre réversible pour la fusion de bases propositionnelles. In *actes de RFIA06*, page 139, Tours. France, 2006.
- [Sei06] J. Seinturier. Fusion de connaissances : Applications aux relevés photogrammétriques de fouilles archéologiques sous-marines, December 2006.
- [Sim02] P. Simons. Extending and implementing the stable model semantics. *Artificial Intelligence*, pages 181–234, 2002.
- [SK94] P. Smets and R. Kennes. The transferable belief model. *Artificial Intelligence*, 66(2) :191–234, 1994.
- [SN98] T. Soininen and I. Niemela. Formalizing configuration knowledge using rules with choices. In *Proceedings of NMR98*, 1998.
- [Som94] L. Sombé. *Revision and updating in knowledge bases*. John Wiley and Sons, 1994.
- [Spo88a] W. Spohn. Ordinal conditional functions : A dynamic theory of epistemic states. In *Causation in decision, belief change, and statistics*, volume II, pages 105–134. W. Harper and B. Skyrms, 1988.
- [Spo88b] W. Spohn. Ordinal conditional functions : a dynamic theory of epistemic states. *Causation in Decision, Belief Change, and Statistics*, pages 105–134, 1988.
- [Spo90] W. Spohn. A general non-probabilistic theory of inductive reasoning. In *Proceedings of UAI88*, pages 149–158, 1990.
- [SS94] L. Shapiro and E. Sterling. *The Art of PROLOG : Advanced Programming Techniques*. The MIT Press, 1994.
- [Syr99] T. Syrjänen. A rule-based formal model for software configuration. Research report, Helsinki University of Technology, 1999.
- [Syr02] T. Syrjanen. lparse user’s manual, 2002.
- [Tur03] H. Turner. Strong equivalence made easy : nested expressions and weight constraints. *Theory and Practice of Logic Programming*, pages 609–622, 2003.
- [vGRS91] A. van Gelder, K. Ross, and J. Schlipf. The well-founded semantics for general logic programs. *Journal of the ACM*, 38 :620–650, 1991.
- [Wil94] M. A. Williams. Transmutations of Knowledge Systems. In *Proceedings of 4th Principle of Knowledge Representation and Reasoning Conference*, pages 619–629, 1994.
- [Wil95] M. A. Williams. Iterated Theory Base Change :A Computational Model. In *Proceedings of IJCAI95*, pages 1541–1547, 1995.

- [WZL00] K. Wang, L. Zhou, and F. Lin. Alternating fixpoint theory for logic programs with priority. In *Proceedings of CL00*, pages 164–178, 2000.
- [YBL⁺08] S. Yahi, S. Benferhat, S. Lagrue, M. Sérayet, and O. Papini. A lexicographic inference for partially preordered belief bases. In *Proceedings of KR08*, ISBN 978-1-57735-384-3, pages 507–516, 2008.
- [YLH07] A. Yue, W. Liu, and A. Hunter. Approaches to constructing a stratified merged knowledge base. In *Proceedings of ECSQARU07*, pages 54–65, 2007.
- [Zad65] L. Zadeh. Fuzzy sets. *Information Control*, 8 :338–353, 1965.