

Université des Sciences et de la Technologie Houari Boumediene
Faculté d'Électronique et d'Informatique

THESE

présentée pour l'obtention du diplôme de doctorat en:
INFORMATIQUE

Spécialité:
Intelligence Artificielle

Par:
Asma BELHADI

Vers la logique possibiliste multi-agents

Soutenue publiquement le 27 octobre 2016 devant le jury composé de:

Professeur	Aicha	AISSANI-MOKHTARI	USTHB	Présidente
Professeur	Hadja Faiza	KHELLAF-HANED	USTHB	Directrice de thèse
Directeur de recherche	Henri	PRADE	Université Paul Sabatier	Co-directeur de thèse
Professeur	Yassine	DJOUADI	USTHB	Examineur
Maître de conférences A	Mohand Cherif	BOUKALA	USTHB	Examineur
Professeur	Allel	HADJALI	ISAE-ENSMA	Examineur

A la mémoire de Amira.

Remerciements

Je rends grâce à Dieu, miséricordieux de m'avoir soutenu et donné la volonté, la persévérance et l'obstination pour réaliser ce travail.

Je remercie le professeur Aicha AISSANI-MOKHTARI qui me fait l'honneur de présider ce jury et d'avoir bien voulu juger ce travail.

Mes remerciements vont également au professeur Yassine DJOUADI d'avoir bien voulu participer au jury de soutenance.

Je remercie vivement le docteur Mohand Cherif BOUKALA d'avoir bien voulu accepter d'examiner ce travail.

Que le professeur Allel HADJALI trouve ma gratitude d'avoir accepté de faire partie de ce jury.

Je tiens à exprimer ma gratitude à Madame Hadja Faiza KHELLAF-HANED de m'avoir accordé sa confiance en voulant bien être ma directrice de thèse et d'avoir été à l'origine de mes premiers pas dans le monde de la recherche.

Je suis infiniment reconnaissante à Monsieur PRADE Henri de l'université Paul Sabatier de Toulouse de m'avoir proposé ce sujet de thèse, d'avoir dirigé cette thèse et de m'avoir initié à la théorie des possibilités.

Je tiens à remercier spécialement Monsieur Didier DUBOIS de l'université Paul Sabatier de Toulouse pour sa collaboration. Qu'il trouve ici ma profonde reconnaissance.

Je dédie cette thèse à ma famille: à mon époux pour son soutien et ses encouragements essentiels et à mes chères parents. Je remercie enfin mes beaux-parents qui m'ont toujours encouragé dans ce travail, ainsi que mes sœurs qui ont, chacune à sa manière, embelli ces dernières années.

Je remercie aussi tous ceux qui, de près ou de loin, par leur amitié ou leur gentillesse, ont contribué à l'aboutissement de ce travail.

Table des matières

Table des matières	1
Table des figures	7
Liste des tables	9
Introduction Générale	11
I État de l’art	17
1 La logique propositionnelle	21
1.1 Introduction	21
1.2 Aspect syntaxique	21
1.3 Aspect sémantique	22
1.4 Aspect axiomatique	25
1.5 Principales propriétés	26
1.6 Résolution	27
1.7 Le problème SAT et la méthode de Davis-Putnam-Logemann-Loveland .	28
1.7.1 Principe	29
1.7.2 Illustration	30
1.8 Conclusion	31
2 Les logiques multi-agents	33
2.1 Introduction	33
2.2 Logique modale normale	33
2.3 Logique modale normale pour les connaissances et les croyances	35
2.4 Théorie de l’intention : Cohen et Levesque	36
2.5 Croyance, désire et intention : Rao et Georgeff	38
2.6 Logique de coalition	39
2.7 Logique temporelle du temps en alternance	41
2.8 La logique individuelle VACQ	43
2.9 Conclusion	44
3 La Logique possibiliste	45
3.1 Introduction	45
3.2 Langage	45
3.3 Le treillis des valuations	46
3.4 Sémantique de la logique possibiliste	46
3.4.1 Les distributions de possibilité normalisées comme sémantique de la logique possibiliste	46

3.4.1.1	Distribution de possibilités	46
3.4.1.2	Les mesures de possibilités et de nécessité	47
3.4.1.3	Principe du minimum de spécificité	49
3.4.1.4	Satisfaction d'une base de connaissance possibiliste par la distribution de possibilité	49
3.4.1.5	Conséquence logique	50
3.4.1.6	Le problème de la déduction	50
3.4.1.7	Le modèle possibiliste le moins spécifique	51
3.4.2	Inconsistances partielles en logique possibiliste	52
3.4.2.1	Introduction	52
3.4.2.2	La sémantique sous normalisée de la logique possibiliste	52
3.4.2.3	Conséquence logique en logique possibiliste sous norma- lisée	53
3.4.2.4	La distribution de possibilité normalisée ou non, la moins spécifique parmi les distributions les moins sous-normalisées	53
3.4.2.5	Déduction sous inconsistance partielle	54
3.4.2.6	Déduction triviale vs déduction non-triviale	54
3.5	Axiomatique de la logique possibiliste	55
3.6	Exemple illustratif	56
3.7	La résolution possibiliste	57
3.7.1	Motivations et définitions	57
3.8	Mise sous forme clausale dans la logique possibiliste	57
3.8.1	Les règles de résolution	57
3.8.2	Correction des règles de la logique possibiliste	58
3.8.3	Réfutation par résolution	58
3.8.3.1	La réfutation par résolution dans la logique possibiliste	59
3.8.3.2	Correction et complétude de la réfutation par résolution	59
3.8.4	Stratégie de résolution	60
3.8.5	Stratégie par α -coupes dichotomique	61
3.8.6	Stratégies linéaires de type <i>pseudo</i> – A^* en logique possibiliste .	61
3.8.6.1	Réfutations par résolution possibilistes linéaires	61
3.8.6.2	Stratégies linéaires informées	62
3.8.6.3	Une famille de fonctions d'évaluation admissibles	63
3.9	Exemple illustratif	65
3.10	Conclusion	65

II Contributions 67

4	La logique multi-agents	71
4.1	Introduction	71
4.2	Langage	71
4.3	Le treillis des sous-ensembles	73
4.4	Sémantique possibiliste de la logique multi-agents	74
4.4.1	Les distributions de possibilité multi-agents multi-agents-normalisées comme sémantique de la logique multi-agents	74
4.4.1.1	Définitions de base	74
4.4.1.2	Satisfaction d'une base de connaissances possibiliste par la distribution de possibilité	77
4.4.1.3	Conséquence logique	78

4.4.1.4	Le problème de déduction	79
4.4.1.5	Le modèle multi-agents le moins spécifique	79
4.4.1.6	Déduction en logique multi-agents	80
4.4.2	Les distributions de possibilité multi-agents individuellement-normalisées comme sémantique de la logique multi-agents	81
4.4.2.1	La sémantique individuellement normalisée de la logique multi-agents	81
4.4.2.2	Conséquence logique en logique multi-agents individuel- lement normalisée	83
4.4.2.3	La distribution de possibilité multi-agents, multi-agents normalisée ou non, la moins spécifique parmi les distri- butions les moins individuellement-normalisées	83
4.4.2.4	Déduction sous individuelle inconsistance	85
4.4.2.5	Déduction multi-agents triviale vs déduction multi-agents non-triviale	86
4.5	L'ensemble d'interprétations individuelles	88
4.6	Axiomatique de la logique multi-agents	90
4.7	Exemple pour la logique multi-agents	91
4.8	Conclusion	93
5	La logique possibiliste multi-agents	95
5.1	Introduction	95
5.2	Langage	95
5.3	Le treillis des sous-ensembles flous	98
5.4	Sémantique de la logique possibiliste multi-agents	98
5.4.1	Les distributions de possibilité multi-agents multi-agents-normalisées comme sémantique de la logique multi-agents	98
5.4.1.1	Définitions de base	98
5.4.1.2	Satisfaction d'une base de connaissance multi-agents pos- sibiliste par la distribution de possibilité multi-agents graduée	101
5.4.1.3	Conséquence logique	102
5.4.1.4	Le problème de déduction	103
5.4.1.5	Le modèle possibiliste multi-agents le moins spécifique .	103
5.4.1.6	Déduction en logique possibiliste multi-agents	104
5.4.2	Les distributions de possibilité multi-agents graduée partiellement individuellement normalisées comme sémantique de la logique pos- sibiliste multi-agents	105
5.4.2.1	La sémantique partiellement individuellement normali- sée de la logique possibiliste multi-agents	105
5.4.2.2	Conséquence logique en logique possibiliste multi-agents partiellement-individuellement normalisée	107
5.4.2.3	La distribution de possibilité multi-agents graduée, multi- agents normalisée ou non, la moins spécifique parmi les distributions les moins partiellement individuellement- normalisées	108
5.4.2.4	Déduction sous la graduelle individuelle inconsistance .	109
5.4.2.5	Déduction possibiliste multi-agents triviale vs déduc- tion possibiliste multi-agents non-triviale	110
5.5	Axiomatique de la logique possibiliste multi-agents	113

5.6	Exemple pour la logique possibiliste multi-agents	115
5.7	Conclusion	117
6	La résolution multi-agents	119
6.1	Introduction	119
6.2	Motivations et définitions	119
6.3	Mise sous forme clausale dans la logique multi-agents	120
6.4	Les règles de résolution	121
6.4.1	Les règles de la logique multi-agents	121
6.4.2	Correction des règles de la logique multi-agents	121
6.5	Réfutation par résolution	122
6.5.1	La réfutation par résolution dans la logique multi-agents	122
6.5.2	Correction et complétude de la réfutation par résolution	124
6.6	Stratégie de résolution	126
6.6.1	Stratégies linéaires de type <i>pseudo</i> – A^* en logique multi-agents	127
6.6.1.1	Réfutations par résolution multi-agents linéaires	127
6.6.1.2	Stratégies linéaires informées	129
6.6.1.3	Une famille de fonctions d'évaluation admissibles	132
6.6.1.4	Exemple	133
6.7	Conclusion	135
7	La résolution possibiliste multi-agents	137
7.1	Introduction	137
7.2	Motivations et définitions	137
7.3	Mise sous forme clausale dans la logique possibiliste multi-agents	138
7.4	Les règles de résolution	139
7.4.1	Les règles de la logique possibiliste multi-agents	139
7.4.2	Correction des règles de la logique possibiliste multi-agents	139
7.5	Réfutation par résolution	140
7.5.1	La réfutation par résolution dans la logique possibiliste multi-agents	140
7.5.2	Correction et complétude de la réfutation par résolution	141
7.6	Stratégie de résolution	143
7.6.1	Stratégie informée en logique possibiliste multi-agents	144
7.6.1.1	Réfutations par résolution possibiliste multi-agents linéaires	144
7.6.1.2	Stratégies linéaires informées	145
7.6.1.3	Une famille de fonctions d'évaluation admissibles	147
7.6.1.4	Exemple	148
7.7	Conclusion	150
8	Implémentation et Expérimentation	153
8.1	Introduction	153
8.2	Implémentation	153
8.3	Validation de l'application	156
8.3.1	Tests sur des bases de connaissances de petites tailles	156
8.3.2	Tests sur des bases de connaissances générées aléatoirement	160
8.4	Conclusion	164

9	Vers des applications multi-agents/ possibilistes multi-agents	165
9.1	Introduction	165
9.2	Révision de croyances en logique propositionnelle	165
9.2.1	L'approche AGM	166
9.2.2	Postulats AGM	167
9.2.3	Postulats KM	168
9.3	Approches évaluées de la révision	168
9.4	Esquisse d'une révision des croyances possibiliste multi-agents	169
9.4.1	Motivation	169
9.4.2	Le problème de la révision possibiliste multi-agents	170
9.4.2.1	Contrepartie sémantique de la révision possibiliste multi-agents	170
9.4.2.2	Contrepartie syntaxique de la révision possibiliste multi-agents	170
9.5	Esquisse de la notion de confiance	171
9.5.1	Formalisation de la confiance	173
9.5.2	Application liées au concept de confiance	174
9.6	Conclusion	174
	Conclusion Générale	175
	Bibliographie	177

Table des figures

1.1	Structure en arbre associée à l'exemple 1.3	23
1.2	Exemple d'arbre de recherche pour le problème SAT	30
3.1	Arbre de recherche de l'exemple illustratif 3.9	65
4.1	Les bases de connaissances associées à chaque groupe des médecins spécialistes	73
4.2	Distribution de possibilités multi-agents associées a l'exemple 4.3	75
4.3	Distributions de possibilités multi-agents non normalisée associées à l'exemple 4.9	82
4.4	Les différents sous-ensembles d'agents selon leurs comportements envers <i>a</i>	89
5.1	La base de connaissances associée aux différents groupes des nouveaux médecins spécialistes	97
5.2	Distribution de possibilités multi-agents associée à l'exemple 5.4	99
5.3	Distributions de possibilités multi-agents graduée non normalisées de l'exemple 5.9	107
6.1	Les deux réfutations possibles.	124
6.2	Illustration de la construction de l'arbre de réfutation	128
6.3	Illustration de l'application d'une stratégie linéaire	129
6.4	Arbre de recherche -1-	134
6.5	Arbre de recherche -2-	134
6.6	Arbre de recherche -3-	135
6.7	Arbre de recherche -4-	136
6.8	Arbre de recherche -5-	136
7.1	Réfutation optimale vs réfutation non optimale de l'exemple 7.3.	142
7.2	Arbre de recherche multi-agents possibiliste -1-	149
7.3	Arbre de recherche multi-agents possibiliste -2-	149
7.4	Arbre de recherche multi-agents possibiliste -3-	150
7.5	Arbre de recherche multi-agents possibiliste -4-	150
7.6	Arbre de recherche -5-	151
8.1	La classe <code>group_agent</code>	154
8.2	La classe <code>Clause</code>	154
8.3	La classe <code>DPLL</code>	154
8.4	La classe <code>refutation</code>	154
8.5	Temps CPU (Sec) du processus de réfutation en utilisant des bases de connaissances multi-agents de petites tailles.	161

8.6	Temps CPU (Sec) du processus de réfutation en utilisant des bases de connaissances multi-agents de tailles moyennes.	161
8.7	Temps CPU (Sec) du processus de réfutation en utilisant des bases de connaissances multi-agents de grandes tailles.	162
8.8	Temps CPU (Sec) du processus de réfutation en utilisant des bases de connaissances possibilistes multi-agents de petites tailles.	162
8.9	Temps CPU (Sec) du processus de réfutation en utilisant des bases de connaissances possibilistes multi-agents de tailles moyennes.	163
8.10	Temps CPU (Sec) du processus de réfutation en utilisant des bases de connaissances possibilistes multi-agents de grandes tailles.	163
8.11	Comparaison entre la réfutations par résolution multi-agents linéaires et sa contrepartie possibiliste multi-agents en termes de temps d'exécution (Sec)	164
9.1	Transitions entre statuts épistémiques.	167

Liste des tableaux

1.1	Table de vérité du calcul propositionnel	23
1.2	Table de vérité	24
1.3	Les équivalences de la logique propositionnelle	24
3.1	Calcul détaillé de la distribution de possibilité de l'exemple illustratif 3.6	56
4.1	Calcul détaillé de la distribution de possibilité multi-agents de l'exemple de la section 4.7	92
5.1	Calcul détaillé de la distribution de possibilité multi-agents graduée - 1 de l'exemple de la section 5.6	116
5.2	Calcul détaillé de la distribution de possibilité multi-agents graduée - 2 de l'exemple de la section 5.6	116

Introduction Générale

Un enjeu central, activement étudié dans le domaine de l'intelligence artificielle, réside sans conteste dans le développement des formalismes permettant de représenter les connaissances et de fournir des méthodes permettant de raisonner sur ces connaissances. Les solutions mises en œuvre doivent prendre en compte et tirer parti de la nature complexe du monde.

La logique mathématique est l'un des formalismes les plus adaptés à ces objectifs. En effet, elle est souvent utilisée comme base de représentation car elle fournit un cadre formel permettant non seulement de coder des connaissances dans un langage évolué et expressif mais aussi de les exploiter grâce à ses méthodes d'inférences. Cependant, la logique propositionnelle et la logique des prédicats, dites logiques classiques, ont un pouvoir expressif limité contrairement au langage naturel.

Parmi les limitations des la logiques classiques figure l'impossibilité de représenter des connaissances imparfaites. En effet, les connaissances du monde dont nous disposons sont souvent imparfaites. Ceci est dû à plusieurs facteurs liés soit à la manière dont ces connaissances sont récoltées, soit à la nature même de ces connaissances. Les imperfections des connaissances sont nombreuses. Les principales imperfections considérées sont : l'incertitude et l'imprécision [Bloch 96] [Dubois, Prade 94 a] [Dubois, Prade 94 b].

L'incertitude caractérise le degré de conformité à la réalité d'une information, ou encore l'assurance d'une source en l'information fournie. Une information incertaine décrit donc une connaissance partielle de la réalité, alors que ce qui est certain entraîne nécessairement la connaissance de toute la réalité. Pour mesurer l'incertitude, la réalité doit donc être connue. A titre d'exemple, considérons l'énoncé 'cette lettre arrivera demain'. Le fait que la réalité (le futur) ne soit pas connue, ne nous permet pas d'affirmer que la lettre arrivera demain. Cette information est donc incertaine.

L'imprécision d'une information est caractérisée par le contenu de l'information. Elle est relative à l'information ou à la source. Elle mesure donc un défaut quantitatif de connaissance. Deux sortes d'imprécision sont distinguées [Smets 97] : l'imprécision sans erreur (par exemple : il pleut beaucoup) et l'imprécision avec erreur (par exemple : il pleut $10 \text{ mm}^3/h$ alors qu'il pleut $15 \text{ mm}^3/h$). Dans le premier cas, l'imprécision est difficile à mesurer et elle reste relative à l'idée que l'expert s'en fait, alors que dans le second cas, elle se mesure par l'erreur de l'information fournie. La réalité doit donc être connue ou estimée.

L'imprécision d'une information est souvent par erreur, confondue avec l'incertitude [Bloch 03]. Or, l'imprécision est un défaut quantitatif de la connaissance, alors que l'incertitude est un défaut qualitatif de la connaissance. Ces deux aspects (imprécision et

incertitude) peuvent alors coexister mais ils sont distincts.

Dans le but de modéliser et de simuler des raisonnements dus aux connaissances imparfaites, la communauté de l'intelligence artificielle s'est intéressée à des logiques dites non classiques. Plusieurs modèles ont émergé, reflétant les nouveaux concepts prenant en compte les différents aspects de l'imperfection.

Ainsi, afin de modéliser la notion d'imprécision, Zadeh a proposé de modéliser le mécanisme de la pensée humaine par un raisonnement approximatif basé sur des variables linguistiques : la théorie des sous-ensembles flous [Zadeh 65] est alors apparue. L'idée de base étant de manipuler des classes floues représentées par des ensembles dont les contours sont mal définis (c'est-à-dire flous). Un ensemble flou est donc défini par une fonction d'appartenance qui n'est plus binaire, mais partielle.

La théorie des sous-ensembles flous a donné naissance à la théorie de possibilités [Zadeh 78] qui offre un caractère incertain et non probabilisé des connaissances. L'idée de base est d'évaluer en fonction de la connaissance disponible, incomplète et entachée d'incertitude, la possibilité et la certitude d'une proposition (non floue en général) par des nombres appartenant à l'intervalle $[0, 1]$ qui ne sont pas des degrés de vérité étant donné que la proposition ne peut être, en définitive, que vraie ou fausse. La logique possibiliste se propose alors de traiter de telles formules évaluées par des degrés de possibilités et de certitude. Elle fût développée à partir de 1986 par Dubois et Prade [Dubois, Prade 87 a] [Dubois, Prade 87 b].

Problématique :

Les humains, semble-il, possèdent des connaissances et mènent des raisonnements. Connaissances et raisonnement sont également importants pour les agents artificiels. En effet, les agents fondés sur les connaissances peuvent tirer parti de connaissances exprimées sous forme très générale grâce à leurs capacités à combiner et à recombinaison des informations pour les adapter à une multitude de problèmes. C'est ainsi, par exemple, qu'un médecin ausculte un patient (autrement dit qu'il infère un état pathologique qui n'est pas directement observable) avant de choisir un traitement.

Si un agent veut agir de façon "intelligente" dans un environnement donné, il est important que cet agent ait des connaissances relatives à cet environnement. Pour concevoir un agent doué d'une "intelligence artificielle", deux problèmes majeurs se posent à nous : comment, d'une part, représenter les connaissances de l'agent ? et comment d'autre part raisonner à partir de ces connaissances ? Ces problèmes deviennent plus complexes lorsque les connaissances disponibles des agents sur le monde sont entachées d'incertitude et qui ne sont donc plus incontestables.

Nous proposons dans cette thèse, dans un premier temps, une extension multi-agents de la logique propositionnelle qui est la logique multi-agents. Cette extension repose sur la logique propositionnelle pour laquelle, un sous ensemble d'agents est attaché à chaque proposition exprimant qu'au moins tous les agents de cet ensemble croient la proposition. La logique multi-agents a pour but de modéliser les croyances d'un ensemble d'agents.

Dans un second temps, nous utilisons cette logique multi-agents dans le but de proposer une autre extension multi-agents, cette fois ci dans le cadre de la logique pos-

sibiliste collective graduée. Cette deuxième extension a donné naissance à la logique possibiliste multi-agents. Nous nous sommes ainsi intéressés au cas où les propositions sont associées à la fois à des degrés de certitude et à des sous-ensembles d'agents. La logique possibiliste multi-agents permet donc de décrire les croyances graduées d'un sous-ensemble d'agents.

Ces modélisations formelles sont très utiles à plus d'un titre et peuvent être exploitées de diverses manières :

- Elles peuvent être vues comme un rapprochement entre les systèmes multi-agents et la théorie des possibilités.
- Elles permettent de représenter les croyances des groupes d'agents. Celle-ci définit la représentation que le groupe d'agents se fait du monde. De ce fait, ces croyances peuvent être tout à fait différentes des croyances des autres groupes d'agents. En effet, les groupes d'agents peuvent s'entendre entre eux comme ils peuvent se contredire.
- Elles permettent de représenter les croyances graduées des groupes d'agents. En effet, chaque groupe d'agents peuvent être plus au moins certain en vue de ses croyances. De plus, ce qui est considéré certain pour un groupe d'agents peut ne pas l'être pour un autre. En effet, les agents peuvent avoir des points de vue différents.
- Il est alors possible maintenant d'exprimer que tous les agents dans un ensemble donné partagent les mêmes croyances.
- Il est également possible de spécifier que tous les agents dans un ensemble possèdent des croyances particulières.
- Elles permettent d'effectuer un raisonnement avec les croyances détenues par un ensemble d'agents. Ceci est dû à la logique propositionnelle qui, avec son solide socle formel pour l'inférence, fournit un cadre privilégié pour raisonner sur les croyances des groupes d'agents.
- Elles permettent de raisonner sur les croyances graduées des ensembles d'agents. En effet, la logique possibiliste multi-agents est basée sur la logique possibiliste et par conséquent la représentation qu'un ensemble d'agents emploie doit donc lui permettre de raisonner à partir d'informations incertaines. Ceci dans le but d'en dériver les conclusions lui permettant d'agir de façon appropriée.

Principales contributions :

Nos travaux apportent un nouveau formalisme de représentation et de raisonnement pour la logique possibiliste.

Du point de vue de la représentation des connaissances ;

- Nous nous sommes intéressés dans [Belhadi et al. 13] à la logique propositionnelle multi-agents où les formules sont des paires de la forme : (a, A) construites avec la proposition a et un ensemble d'agents A .

La formule (a, A) signifie que : au moins tous les agents qui sont dans A croient

que a est vraie. Nous avons proposé d'abord l'aspect syntaxique de la logique multi-agents, ensuite son aspect sémantique et enfin ses résultats de complétude et de correction. En effet, la logique multi-agent est correcte et complète.

- Nous nous sommes également intéressés dans [Belhadi et al. 13] à l'extension de la logique propositionnelle multi-agents en terme de logique possibiliste qui est : la logique possibiliste multi-agents. Les formules de la logique possibiliste multi-agents sont de la forme : $(a, \alpha/A)$ où a est une proposition, α un nombre réel entre 0 et 1 et A un ensemble d'agents.
Ces formules expriment que : tous les agents dans A croient que a est vraie à au moins un certain degré α . Nous présentons sa syntaxe et ensuite sa sémantique définie cette fois ci en termes d'ensembles flous d'agents qui trouvent une interprétation plus au moins possible.
- Nous avons traité dans [Belhadi et All. 13] le problème de raisonnement sur les opinions des groupes d'agents. En effet, les agents ayant leurs propres croyances et opinions peuvent être en conflits avec d'autres groupes d'agents n'ayant pas les même croyances et opinions sur le même état du monde.
- Dans [Belhadi et All. 15], nous avons étudié l'aspect de la complétude et de la correction de la logique possibiliste multi-agents. En se basant sur les résultats de complétude et de correction de la logique possibiliste [Dubois et al. 94 b], nous avons prouvé que la logique possibiliste multi-agents est effectivement complète et correcte vis-à-vis de sa sémantique.

Du point de vue du raisonnement ;

- Nous avons développé dans [Belhadi et All. 14] un algorithme de réfutation par résolution pour une base de connaissances multi-agents. Ce dernier est une adaptation de l'algorithme A^* de la recherche de chemins dans un espace d'état.
- Nous avons aussi développé dans [Belhadi et All. 14] un l'algorithme d'inférence de la réfutation par résolution pour la logique possibiliste multi-agents. Ce dernier est une généralisation de l'algorithme d'inférence de la logique multi-agents. Cette généralisation nous permet de prendre en compte les degrés de certitudes.

Sur le plan de l'implémentation, nous avons mis en œuvre les deux algorithmes proposés. Ensuite, dans [Belhadi et All. 16] nous avons mené quelques expériences exploitant les procédures développées. Ces expériences ont porté sur des petits exemples ainsi que sur des profils de croyances générés aléatoirement.

Lecture de la thèse :

D'une manière plus détaillée, le manuscrit se compose de neuf chapitres.

Le premier chapitre introduit les notions essentielles de la logique propositionnelle. Nous présentons dans un premier temps son aspect syntaxique, son aspect sémantique et son aspect axiomatique. Puis, nous présentons ses résultats de complétude et de correction. Nous rappelons par la suite la méthode de résolution. Finalement, nous présentons le problème SAT ainsi que la méthode de Davis- Putnam- Logemann-Loveland.

Le deuxième chapitre est consacré à faire un tour d'horizon sur les logiques multi-agents en rappelant les principaux travaux réalisés dans le domaine étudié. D'une manière détaillée, nous présentons les différentes logiques modales utilisées dans les systèmes multi-agents en fonction de leurs différents objectifs.

Dans le troisième chapitre, nous nous intéressons à la logique possibiliste standard où nous détaillerons les différentes notions reliées à cette logique. La suite du chapitre sera consacrée à présenter les concepts de base sur la résolution possibiliste. Les quatre chapitres suivants s'étaleront sur nos contributions.

Le quatrième chapitre propose une première contribution. Nous présenterons une nouvelle logique multi-agents où chaque formule est associée à un ensemble d'agents. Ce chapitre fournit les volets syntaxiques, sémantiques et axiomatiques de cette logique. Nous montrons également les propriétés de ces différents volets.

Dans le cinquième chapitre, nous allons nous intéresser à une autre nouvelle logique multi-agents qui est la logique possibiliste multi-agent. Cette dernière permet de représenter les connaissances incertaines d'un groupe d'agents. En effet, la logique possibiliste offre un cadre naturel et simple pour représenter les connaissances incertaines.

Dans le sixième chapitre, nous présenterons les méthodes de déduction automatique pour la logique multi-agents. Dans ce contexte, nous proposons un algorithme de réfutation par résolution linéaire pour la logique multi-agents. Cet algorithme utilise une stratégie de recherche de type A^* ou des réfutations avec les plus grands ensembles d'agents sont trouvées.

Le septième chapitre, s'étalera sur un nouvel algorithme de réfutation par résolution linéaire pour la logique possibiliste multi-agents. En effet, l'algorithme de réfutation par résolution multi-agents linéaire, développé dans le chapitre précédent, s'étend naturellement, permettant ainsi de prendre en compte des degrés de certitude, en termes de sous-ensembles flous d'agents.

Le huitième chapitre, présente une expérimentation relative à l'implémentation des algorithmes d'inférence. Une première partie présente des tests effectués sur des petits exemples. Et une seconde partie présente une expérimentation utilisant des profils de croyances aléatoires. Ces expérimentations permettent d'illustrer le comportement des algorithmes développés.

Le neuvième chapitre présente une ébauche vers d'éventuelles applications pour les logiques multi-agents développées au cours de ce travail.

A la fin de cette thèse, nous dresserons un bilan général de ce présent travail et nous abordons quelques futures perspectives.

Première partie

État de l'art

Introduction de la partie I

Cette première partie est consacrée à l'état de l'art. Plus précisément, dans le chapitre 1, nous rappelons brièvement les différents concepts de bases de la logique propositionnelle qui sont nécessaires pour la compréhension de la suite de la thèse. Quant au chapitre 2, nous passons en revue certaines logiques multi-agents. Cette présentation permet ainsi de rendre compte des différentes façons avec lesquelles le concept du multi-agents a été approché dans le domaines de la logique. Le chapitre 3, fournit une introduction à la théorie des possibilités. Dans ce chapitre, nous présentons d'abord les notions de bases relatives à la logique possibiliste avant d'aborder la résolution possibiliste.

Chapitre 1

La logique propositionnelle

1.1 Introduction

La logique propositionnelle a depuis longtemps été considérée comme un outil privilégié des chercheurs en Intelligence Artificielle pour la formalisation et l'exploitation des connaissances. En effet, de multiples problèmes pratiques peuvent se représenter de manière simple et se résoudre en logique propositionnelle. Par ailleurs, la logique propositionnelle est la pierre angulaire de nombreuses logiques plus sophistiquées.

Ce chapitre introduit brièvement la logique propositionnelle. Dans la section 2, nous présentons la syntaxe de la logique propositionnelle. La section 3 présente sa sémantique. La section 4 expose l'aspect axiomatique de la logique propositionnelle. Quand à la section 5, elle présente les principales propriétés de la logique propositionnelle telles que : la correction, la complétude et la décidabilité. La section 6 présente l'algorithme de résolution de la logique propositionnelle. Enfin, La section 7 présente un algorithme de satisfiabilité d'une formule propositionnelle, à savoir le DPLL (Davis-Putnam-Logemann-Loveland).

1.2 Aspect syntaxique

Pour étudier la syntaxe d'un langage, il faut donner un alphabet (un ensemble de symboles) et des règles de constructions syntaxiques d'expressions à partir de ces symboles.

L'alphabet de la logique propositionnelle est constitué de l'ensemble des symboles suivant :

- Les éléments de base de la logique propositionnelle sont les énoncés atomiques (Les éléments syntaxiques indivisibles), notés : p, q, r, \dots . Il s'agit des propositions atomiques telles que par exemple : 6 est divisible par 2.
- Les parenthèses (et).
- La proposition toujours vraie : Vrai (\top) et la proposition toujours fausse : Faux (\perp).
- Les connecteurs logiques. Les cinq connecteurs les plus fréquemment utilisés sont :
 - La négation \neg : Un énoncé tel que $\neg p$ est la négation de p . Un littéral est soit une phrase atomique (un littéral positif) soit une phrase atomique négative (un littéral négatif) ;

- La conjonction \wedge : Un énoncé dont le principal connecteur est \wedge , comme $p \wedge q$ est une conjonction ; ses composants sont des termes conjonctifs ;
- La disjonction \vee : Un énoncé utilisant \vee , comme $((p \wedge q) \vee r)$ est une disjonction de termes disjonctifs $(p \wedge q)$ et r ;
- L'implication \rightarrow : Un énoncé tel que $(p \wedge q) \rightarrow \neg r$, est une implication. Sa prémisse (ou antécédent) est $(p \wedge q)$ et sa conclusion (ou conséquence) est $\neg r$;
- L'équivalence \leftrightarrow : L'énoncé $p \leftrightarrow \neg q$ est une double implication.

Le langage \mathcal{L} est constitué de l'ensemble des Formules Bien Formées (aussi appelé expressions bien formées ou énoncés complexes) défini inductivement ainsi :

- Toutes les propositions atomiques, i.e. $p, q, r \dots$ sont des expressions bien formées ;
- La proposition 'Vrai' et la proposition 'Faux' sont des expressions bien formées ;
- Si A est une expression bien formée, alors $\neg A$ est une expression bien formée ;
- Si A et B sont deux expressions bien formées, alors $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$ et $(A \leftrightarrow B)$ sont des expressions bien formées ;
- Toute expression bien formée est obtenue par application des règles ci-dessus c.à.d. qu'il n'y a pas d'autres expressions bien formées que celles construites avec les règles précédentes.

Exemple 1.1 Soient p, q, r et s quatre énoncés atomiques. Alors $(p \vee q) \wedge ((r \rightarrow s) \wedge \neg p)$ est une expression bien formée.

Tout énoncé construit avec des connecteurs binaires doit être encadré par des parenthèses. Les parenthèses permettent de trouver sans ambiguïté la structure syntaxique des formules. En logique propositionnelle, l'ordre de précedence est (par ordre décroissant) : $\neg, \wedge, \vee, \rightarrow$ et \leftrightarrow .

Exemple 1.2 A titre d'exemple l'énoncé : $\neg p \vee q \wedge r \rightarrow s$ correspond à l'énoncé : $((\neg p) \vee (q \wedge r)) \rightarrow s$.

Nous pouvons visualiser la structure des formules par un arbre, où les feuilles contiennent les constantes ou les variables et les nœuds les connecteurs logiques. Le nœud racine est le connecteur à appliquer en dernier.

Exemple 1.3 La structure de la formule $((p \wedge \neg(p \vee q)) \wedge \neg r)$ est mise en évidence par l'arbre donné par la Figure 1.1.

1.3 Aspect sémantique

Après avoir spécifié la syntaxe de la logique propositionnelle, nous abordons dans cette section la sémantique. La sémantique définit les règles de détermination de la vérité d'un énoncé dans le cadre d'une interprétation bien particulière. En logique propositionnelle, une interprétation ne fait que déterminer la valeur de vérité (vrai ou faux) de chaque énoncé. Formellement, une interprétation ω de \mathcal{L} est une fonction qui associe à chaque énoncé atomique p une valeur de vérité appartenant à l'ensemble $\{0, 1\}$ (0 pour faux, 1 pour vrai). Il est à noter que pour une formule à n propositions atomiques il y a 2^n interprétations possibles.

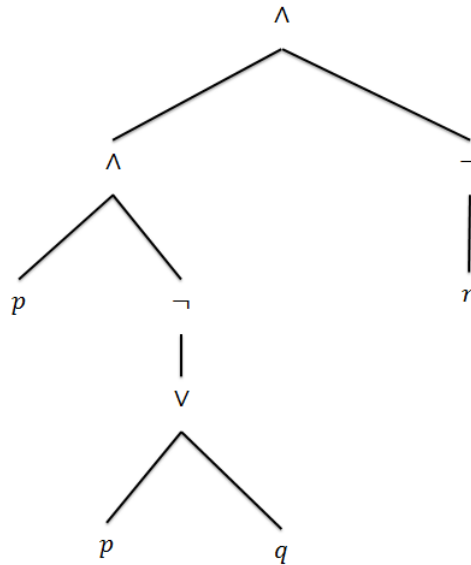


FIGURE 1.1 – Structure en arbre associée à l'exemple 1.3

A	B	$\neg A$	$A \wedge B$	$A \vee B$	$A \rightarrow B$	$A \leftrightarrow B$
v	v	f	v	v	v	v
v	f	f	f	v	f	f
f	v	v	f	v	v	f
f	f	v	f	f	v	v

TABLE 1.1 – Table de vérité du calcul propositionnel

Exemple 1.4 Par exemple, si nous disposons de trois énoncés atomiques p , q et r alors une interprétation possible est : $\omega_1 = \{p = \text{faux}, q = \text{faux}, r = \text{vrai}\}$

Et avec 03 propositions atomiques, il y a $2^3 = 8$ interprétations possibles.

Une valuation, notée $\mathcal{V}(A, \omega)$, associe ensuite à une formule A de \mathcal{L} et à une interprétation ω une des valeurs 0 ou 1, par la définition récursive suivante :

1. si p est un atome propositionnel, alors $\mathcal{V}(p, \omega) = \omega(p)$;
2. $\mathcal{V}(\top, \omega) = 1$;
3. $\mathcal{V}(\perp, \omega) = 0$;
4. si A et B sont deux formules de \mathcal{L} , alors :
 - $\mathcal{V}(\neg A, \omega) = 1$ si et seulement si $\mathcal{V}(A, \omega) = 0$;
 - $\mathcal{V}(A \wedge B, \omega) = \min(\mathcal{V}(A, \omega), \mathcal{V}(B, \omega))$;
 - $\mathcal{V}(A \vee B, \omega) = \max(\mathcal{V}(A, \omega), \mathcal{V}(B, \omega))$;
 - $\mathcal{V}(A \rightarrow B, \omega) = 1$ si et seulement si $\mathcal{V}(\neg A \vee B, \omega) = 1$;
 - $\mathcal{V}(A \leftrightarrow B, \omega) = 1$ si et seulement si $\mathcal{V}((A \rightarrow B) \wedge (B \rightarrow A), \omega) = 1$.

Les tables de vérité des cinq connecteurs logiques peuvent être synthétisées dans la table 1.1.

Les tables de vérité de *et*, *ou* et *non* sont en étroite conformité avec notre intuition quant à la signification de ces mots. Le principal point de confusion possible est que $p \vee q$

est vrai lorsque p est vrai ou que q est vrai ou les deux à la fois. Il existe un autre connecteur appelé : ou exclusif qui aboutit à faux lorsque les deux termes disjonctifs sont vrais.

La table de vérité de \rightarrow peut paraître étonnante à première vue, car elle ne correspond pas tout à fait à ce que tout un chacun pourrait intuitivement comprendre par ' p implique q ' ou 'si p alors q '. Un premier point tient au fait que la logique propositionnelle n'exige pas de relation de causalité ou de pertinence entre p et q . L'énoncé '5 est un nombre impair implique que Alger est la capitale de l'Algérie' est vrai en logique propositionnelle, même s'il est indubitablement étrange. Un autre point de confusion est que toute implication est vraie dès lors que son antécédent est faux. Par exemple, '5 est un nombre pair implique que Nadine est intelligente' est vrai quelle que soit l'intelligence de Nadine. Cela peut sembler étrange, mais on peut le comprendre si l'on considère que $p \rightarrow q$ signifie 'si p est vraie, alors j'affirme que q est vrai ; sinon je n'affirme rien'. La seule possibilité pour que cet énoncé soit faux est que p soit vrai mais q faux.

Exemple 1.5 La table de vérité de la formule $(p \rightarrow q) \leftrightarrow (\neg p \vee q)$ est donnée par la table 1.2.

p	q	$p \rightarrow q$	$\neg p \vee q$	$(p \rightarrow q) \leftrightarrow (\neg p \vee q)$
1	1	1	1	1
1	0	0	0	1
0	1	1	1	1
0	0	1	1	1

TABLE 1.2 – Table de vérité

Deux formules A et B sont logiquement équivalentes ($A \equiv B$) si pour toute valuation elles ont la même valeur. D'autres équivalences sont représentées dans la table 1.3.

Une formule A est valide ; notée $\models A$; si elle est à vraie pour toute valuation. Les formules valides sont également appelées tautologies. Comme la proposition \top est vraie pour toute valuation, toute formule valide est équivalente à \top .

Exemple 1.6 – Nous pouvons écrire $\models p \vee \neg p$, car la formule $p \vee \neg p$ est valide.

$(A \wedge B)$	\equiv	$(B \wedge A)$	commutativité de \wedge
$(A \vee B)$	\equiv	$(B \vee A)$	commutativité de \vee
$((A \wedge B) \wedge C)$	\equiv	$((A \wedge (B \wedge C))$	associativité de \wedge
$((A \vee B) \vee C)$	\equiv	$((A \vee (B \vee C))$	associativité de \vee
$\neg(\neg A)$	\equiv	A	élimination de la double négation
$(A \Rightarrow B)$	\equiv	$(\neg B \Rightarrow \neg A)$	contraposition
$(A \Rightarrow B)$	\equiv	$(\neg A \vee B)$	élimination de l'implication
$(A \Leftrightarrow B)$	\equiv	$((A \Rightarrow B) \wedge (B \Rightarrow A))$	élimination de la double implication
$\neg(A \wedge B)$	\equiv	$(\neg A \vee \neg B)$	De Morgan
$\neg(A \vee B)$	\equiv	$(\neg A \wedge \neg B)$	De Morgan
$((A \wedge (B \vee C))$	\equiv	$((A \wedge B) \vee (A \wedge C))$	distribution de \wedge par rapport à \vee
$((A \vee (B \wedge C))$	\equiv	$((A \vee B) \wedge (A \vee C))$	distribution de \vee par rapport à \wedge

TABLE 1.3 – Les équivalences de la logique propositionnelle

- La formule $(p \rightarrow q) \leftrightarrow (\neg p \vee q)$ est valide.

Le symbole Ω dénote l'ensemble exhaustif des interprétations possibles pour \mathcal{L} . Une interprétation ω satisfait une formule A , noté $\omega \models A$, si et seulement si A est associée à la valeur de vérité vrai pour ω , soit encore $\mathcal{V}(A, \omega) = 1$. L'interprétation ω est alors un modèle de A . L'ensemble des modèles de A est noté : $[A]$.

Exemple 1.7 $[\top] = \Omega$ et $[\perp] = \emptyset$.

Cette notion de modèle s'étend aux ensembles de formules : une interprétation est un modèle d'un ensemble de formules si et seulement si elle est un modèle de chaque formule de l'ensemble.

Exemple 1.8 Un modèle de $\{p \rightarrow q, q \rightarrow r\}$ est : $\omega_1 = \{p = 1, q = 1\}$.

Dans le cadre de la logique des propositions, une formule propositionnelle peut donc être représentée par l'ensemble de ces modèles. Une formule A est alors dite cohérente (satisfaisable) si elle admet au moins un modèle, tandis que A est dite incohérente (insatisfaisable) dans le cas contraire.

Exemple 1.9 La formule $(p \wedge q)$ est cohérente tandis que la formule $(\neg p \wedge p)$ est incohérente.

Une formule B est conséquence logique d'un ensemble Ψ de formules (dite hypothèses), noté $\Psi \models B$, si et seulement si tout modèle de Ψ est également modèle de B , soit formellement $[\Psi] \subseteq [B]$.

Exemple 1.10 Par exemple, nous avons : $p \wedge q \models p$ et $p \models p \vee q$. Par contre, $p \vee q \not\models p$.

Remarque 1.1 La formule A est notée valide par $\models A$, car A est valide si et seulement si A est conséquence logique de l'ensemble vide.

1.4 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, nous définissons 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 même ne nécessitant pas de démonstrations - ainsi que sur une règles d'inférence qui est le modus ponens.

Parmi les ensembles d'axiomes proposés, le système de Hilbert [Imai, Iséki 66] 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) $(\neg A \rightarrow \neg B) \rightarrow ((\neg A \rightarrow B) \rightarrow A)$

Grâce à ces axiomes et à la règle de modus ponens, il est possible de démontrer si de nouvelles formules sont des théorèmes pour le système. Pour pouvoir réaliser des déductions, la règle du modus ponens est définie comme suit :

$$(MP) \quad A, (A \rightarrow B) \vdash B$$

Nous utilisons le symbole \vdash pour noter la relation de déduction ou d'inférence.

Une déduction à partir d'un ensemble d'hypothèses H_1, H_2, \dots, H_m est une suite de formules bien-formées A_1, A_2, \dots, A_p où chaque A_i est soit :

- une hypothèse ;
- un axiome ;
- une formule obtenue à partir de la règle d'inférence : le modus ponens appliqués aux formules placées avant A_i dans la déduction.

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

1.5 Principales propriétés

Les aspects axiomatiques et sémantiques de la logique propositionnelle sont équivalents. Les théorèmes suivants sont issus de [Gödel 86].

Théorème 1.1 (*Théorème de correction*) $\forall A \in \mathcal{L}$, si $\vdash A$ alors $\models A$.

Le théorème de la correction stipule que tout théorème est une tautologie. Autrement dit, toute déduction effectuée axiomatiquement peut également être effectuée sémantiquement.

Théorème 1.2 (*Théorème de complétude*) $\forall A \in \mathcal{L}$, si $\models A$ alors $\vdash A$.

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

Le théorème suivant est central dans la logique propositionnelle. Il stipule que toute formule démontrable en logique propositionnelle peut être démontrée par un algorithme et cette démonstration peut être effectuée en un temps fini. Formellement :

Théorème 1.3 (*Théorème de décidabilité*) $\forall A \in \mathcal{L}$, il existe un programme qui pour toute formule A , calcule en un temps fini si $\vdash A$.

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

Un ensemble de formules est sous forme normale conjonctive *FNC* s'il est une conjonction de clauses. Une clause est une disjonction de littéraux et un littéral est une proposition ou la négation d'une proposition.

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

1.6 Résolution

La règle de résolution a été introduite par Robinson [Robinson 65]. La résolution est une règle d'inférence logique vue comme une généralisation du modus ponens. Elle est utilisée dans les systèmes de preuve automatiques, notamment PROLOG [Colmerauer 90].

La méthode de résolution repose sur l'idée de tester l'incohérence d'un ensemble de clauses. En d'autres termes, il s'agit de tester si la clause vide peut être une conséquence logique de cet ensemble de clauses.

Informellement, soit A , B et X des formules propositionnelles. Suppose qu'il existe une interprétation qui satisfait $A \vee X$ et $B \vee \neg X$. Si cette interprétation satisfait X alors nous déduisons qu'elle satisfait B . Si cette interprétation ne satisfait pas X alors nous déduisons qu'elle satisfait A . Dans les deux cas, cette interprétation satisfait $A \vee B$. Ceci s'écrit plus formellement comme suit :

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

Dans le cas particulier où A et B sont des clauses et X une proposition, le résultat suivant donne le principe de résolution :

Soit \mathcal{C} un ensemble de clauses, soit $c_1, c_2 \in \mathcal{C}$ et soit l un littéral. Si l apparaît dans c_1 et $\neg l$ apparaît dans c_2 tel que : $c_1 = l_{i_1} \vee l_{i_2} \vee \dots \vee l_{i_n} \vee l$ et $c_2 = l_{j_1} \vee l_{j_2} \vee \dots \vee l_{j_n} \vee \neg l$ alors la clause $r = l_{i_1} \vee l_{i_2} \vee \dots \vee l_{i_n} \vee l_{j_1} \vee l_{j_2} \vee \dots \vee l_{j_n}$ est une conséquence logique de \mathcal{C} , appelée : la résolvente.

En conséquence, cette proposition fournit une règle d'inférence, appelé règle de résolution, qui, à partir d'un ensemble de clauses \mathcal{C} , permet de produire de nouvelles clauses et qui préserve la cohérence.

Pour montrer qu'une formule ou un ensemble de formules propositionnelles est incohérent, la mise sous forme conjonctive normale est utilisée qui, selon le théorème de normalisation, produit un ensemble de clauses équivalent. La règle d'inférence ensuite est appliquée jusqu'à ce que : soit la clause vide, notée \perp , soit produite, soit la règle d'inférence ne puisse plus s'appliquer. Si nous pouvons déduire la clause vide \perp de l'ensemble de formules, alors celui-ci est incohérent. Et s'il ne reste plus de formules susceptibles de mener à de nouvelles déductions, alors l'ensemble de formules est cohérent.

L'algorithme de résolution qui, à partir d'un ensemble initial de clauses \mathcal{C} , produit en sortie un ensemble de clauses \mathcal{C}' , est le suivant :

```

Debut

 $\mathcal{C}' \leftarrow \mathcal{C}$ ;
Possible  $\leftarrow$  vrai;
Tant que ( $\perp \notin \mathcal{C}'$  et Possible) faire
    Si (il est possible de produire une nouvelle résolvente) Alors
        choisir  $l, c_1, c_2$  tels que  $l \in c_1$  et  $\neg l \in c_2$ ;
        calculer la résolvente  $r$ ;
         $\mathcal{C}' \leftarrow \mathcal{C}' \cup \{r\}$ .
    Sinon
        possible  $\leftarrow$  faux.
    Fin Si
Fait
Fin

```

Algorithme 1: Résolution (\mathcal{C} : un ensemble de clauses)

Exemple 1.11 Soit l'ensemble de clauses \mathcal{C} suivant :

$$\mathcal{C} = \{p \vee q \vee r, t \vee \neg q, \neg r \vee \neg s, \neg t, s, \neg s \vee \neg p\}$$

Nous pouvons pratiquer une première résolution sur $(p \vee q \vee r)$ et $(\neg r \vee \neg s)$ pour déduire $p \vee q \vee \neg s$. De la même manière, nous obtenons :

- $\{p \vee q \vee \neg s, s\} \models p \vee q$.
- $\{t \vee \neg q, \neg t\} \models \neg q$.
- $\{p \vee q, \neg q\} \models p$.
- $\{s, \neg s \vee \neg p\} \models \neg p$.
- $\{p, \neg p\} \models \perp$.

L'ensemble \mathcal{C} est donc incohérent.

Une propriété importante de la résolution dans la logique propositionnelle est la suivante :

Soit B une formule propositionnelle et Ψ un ensemble de formules propositionnelles, $\Psi \models B$ si et seulement si $\Psi \cup \{\neg B\}$ est incohérent.

1.7 Le problème SAT et la méthode de Davis-Putnam-Logemann-Loveland

Le problème SAT est un problème de décision qui consiste à savoir s'il existe une interprétation qui satisfait un ensemble de formules de la logique propositionnelle. Les formules sont généralement présentées en Forme Normale Conjonctive. Le problème SAT est le premier problème qui a été démontré NP-complet par S.Cook [Cook 71].

Nous présentons dans la suite une méthode classique fondée sur la vérification des modèles permettant de résoudre le problème SAT. Il est souvent appelé l'algorithme de Davis-Puntam du nom des auteurs de l'article fondateur écrit par Martin Davis et

Hilary Putnam en 1960 [Davis, Putnam 60]. Une amélioration sur l'algorithme a été introduite par Davis, Logemann et Loveland en 1962, c'est pourquoi il est appelé DPLL, d'après les initiales des auteurs.

1.7.1 Principe

DPLL prend en entrée un énoncé sous forme normale conjonctive (un ensemble de clauses). Ensuite, une énumération récursive et en profondeur d'abord de tous les modèles possibles est établie. Le principe de DPLL se résume comme suit :

1. **Elagage** : L'algorithme détecte si l'énoncé doit être vrai ou faux, même avec un modèle partiellement complété. Une clause est vraie si l'un des littéraux est vrai, même si les autres littéraux n'ont pas encore eu de valeurs vrai ; il en résulte que l'énoncé dans son ensemble peut être évalué vrai avant que le modèle ne soit complet. Par exemple, l'énoncé $(A \vee \neg B) \wedge (\neg B \vee C)$ est vrai si B est faux quelles que soit les valeurs de A et C. De même, un énoncé est faux si l'une des clauses est fautive. Avec ce principe d'élagage, des branches entières dans l'espace de recherche sont éliminées sans être explorées.
2. **Heuristique des symboles purs** : Un symbole pur est un symbole qui apparaît toujours avec le même signe dans toutes les clauses. Par exemple, dans la formule $(A \vee \neg B) \wedge (\neg B \vee C) \wedge (A \vee \neg C)$, le symbole A est pur car il apparaît uniquement sous forme de littéral positif, et B est pur car il apparaît uniquement sous forme de littéral négatif. L'algorithme DPLL peut facilement instancier les symboles purs tout en éliminant le chemin inverse.
3. **Heuristique de la clause unitaire** : Une clause unitaire est une clause qui ne contient qu'un seul littéral. Dans le contexte de DPLL, elle englobe aussi les clauses pour lesquelles la valeur faux a déjà été affectée à tous les littéraux à l'exception d'un seul. Par exemple, si le modèle contient $B = \text{faux}$ alors $(B \vee \neg C)$ devient une clause unitaire car elle équivaut à $(\text{Faux} \vee \neg C)$, soit juste $\neg C$. De toute évidence, pour que cette clause soit vraie, il faut que C soit défini à faux. L'heuristique de la clause unitaire affecte tous les symboles de ce type avant de traiter le reste. Une conséquence importante de l'heuristique est que toute tentative de prouver un littéral déjà présent dans la base de connaissances sera immédiatement couronnée de succès. Aussi, affecter une clause unitaire peut créer une autre clause unitaire. Par exemple, lorsque C est défini à faux, $(C \vee A)$ devient une clause unitaire ce qui provoque l'affectation de vrai à A. Ce principe s'appelle la propagation unitaire.

L'algorithme suivant présente la structure générale de l'algorithme DPLL.

Début

Si (toute clause de clauses est vraie dans modèle) **Alors**

| retourner vrai.

Fin Si

Si (une clause de clauses est fausse dans modèle) **Alors**

| retourner faux.

| $P, \text{ valeur} \leftarrow \text{Trouver_Symbole_Pure}(\text{symboles}, \text{clauses}, \text{modèle})$

Fin Si

Si (P est non nul) **Alors**

| retourner $\text{DPLL}(\text{clauses}, \text{symboles } -P, \text{modèle} \cup \text{valeur})$

| $\text{propagation_unitaire}(\text{clauses}, \text{modèle})$

| retourner $\text{DPLL}(\text{clauses}, \text{symbole-}P, \text{modèle} \cup P = \text{vrai})$ ou $\text{DPLL}(\text{clauses}, \text{symbole-}P, \text{modèle} \cup P = \text{faux})$.

Fin Si

Fin

Algorithme 2: DPLL

1.7.2 Illustration

Considérons la formule propositionnelle suivante :

$$(p \vee q \vee r) \wedge (\neg p \vee \neg q) \wedge (p \vee r)$$

En utilisant un algorithme de satisfiabilité naïf, nous explorons tout l'arbre de recherche c'est à dire, nous allons instancier tout les littéraux par vrai et par faux, ce qui génère 2^3 cas possibles. Ces derniers, sont représentés par l'arbre décrit par la Figure 1.2.

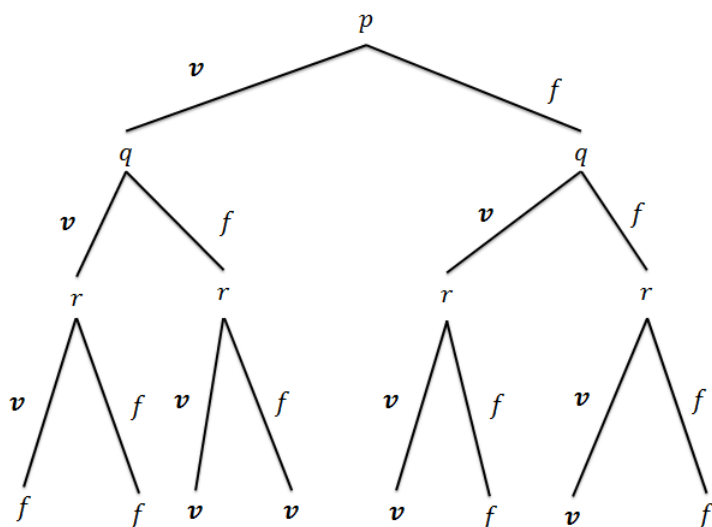


FIGURE 1.2 – Exemple d'arbre de recherche pour le problème SAT

D'après cette figure, nous remarquons que la formule ci-dessus est satisfaite dans les

4 cas suivants :

- (p =vrai, q =faux, r =vrai)
- (p =vrai, q =faux, r =faux)
- (p =faux, q =vrai, r =vrai)
- (p =faux, q =faux, r =vrai)

Comme nous pouvons le remarquer, la complexité de cet algorithme est exponentielle. Ceci génère un temps de calcul considérable tout en augmentant le nombre de propositions. Pour celà, un autre algorithme a été proposé afin de réduire l'espace de recherche. Il s'agit de l'algorithme DPLL. Nous commençons par éliminer les littéraux unitaires, d'après l'exemple, le littéral r toujours se présente positivement, par conséquent, nous instancions r par vrai ce qui donne la proposition suivante :

$$(vrai) \wedge (\neg p \vee \neg q) \wedge (vrai)$$

Nous remarquons qu'il reste une seule clause $C_2 = \{\neg p \vee \neg q\}$. Pour que C_2 soit vraie, il faut instancier p par faux, ce qui donne :

$$(vrai) \wedge (vrai) \wedge (vrai)$$

1.8 Conclusion

Dans ce chapitre nous avons présenté brièvement les éléments de base de la logique propositionnelle qui se trouvent utiles pour la compréhension de ce mémoire. En effet, elle est au cœur des autres logiques qui sont plus expressives. Le chapitre suivant présente les différentes logiques utilisées dans les systèmes multi-agents.

Chapitre 2

Les logiques multi-agents

2.1 Introduction

Le concept d'agent est récemment devenu un concept important en intelligence artificielle. Avec la contribution de chercheurs en intelligence artificielle, en philosophie formelle et en économie, plusieurs approches ont été proposées afin de représenter et raisonner sur les propriétés des agents. Parmi ces dernières, il existe les approches basées sur la logique. Ce chapitre est consacré à donner un bref aperçu sur ces différentes logiques multi-agents. La section 2 présente la logique modale normale qui permet de représenter les croyances, les connaissances, les buts etc. Des agents sont caractérisés par la notion des mondes possibles. La section 3 expose comment la logique modale normale est utilisée pour représenter et raisonner sur les connaissances et les croyances. La section 4 aborde une théorie basée sur la logique modale normale où seulement deux attitudes d'agents sont prises comme primitive, c'est la théorie de l'intention. Quand à la section 5, elle présente une autre théorie qui prend comme primitive trois attitudes, à savoir : les croyances, les désirs et les intentions c'est la théorie de Rao et Georgeff. La section 6 relate les principaux fondements de la logique de coalitions. La section 7 présente la logique temporelle du temps en alternance, une logique qui utilise des quantifications sur les stratégies afin de spécifier les propriétés d'un agent . Enfin, nous abordons la logique individuelle VACQ (veille à ce que) qui permet d'exprimer qu'un agent veille à ce que une proposition soit vraie.

2.2 Logique modale normale

C'est une logique propositionnelle étendue par l'ajout de deux opérateurs : ' \Box ' : nécessairement et ' \Diamond ' : possible [Chellas 80]. Soit $Prop = \{p, q, \dots\}$ l'ensemble des propositions atomiques. Alors la syntaxe de la logique modale normale est définie par les règles suivantes :

- Si $p \in Prop$ alors p est une formule ;
- Si φ, ψ sont des formules, alors $\neg\varphi$ et $\varphi \vee \psi$ sont aussi des formules ;
- Si ψ est une formule alors $\Box\psi$ et $\Diamond\varphi$ sont aussi des formules.

Les opérateurs \neg et \vee sont ceux de la logique propositionnelle classique. La formule $\Box\varphi$ est lue : nécessairement φ et la formule $\Diamond\varphi$ est lue : **possible** φ .

La logique modale normale est basée sur la notion de vérité dans un monde ; les modèles de cette logique contiennent un ensemble de mondes W et une relation binaire

R sur W montrant quel monde est considéré possible par rapport aux autres mondes. Une fonction d'évaluation π est définie montrant les propositions vraies dans chaque monde. Formellement, un modèle est un triplé $\langle W, R, \pi \rangle$, où W est un ensemble non vide de mondes, $R \subseteq W \times W$, et $\pi : W \rightarrow Prop$ est la fonction d'évaluation indiquant, pour chaque monde $w \in W$, les propositions qui sont vraies dans ce monde w .

La sémantique du langage est définie par la relation de satisfaction ' \models ' qui relie entre : les paires de la forme $\langle M, w \rangle$ (où M est un modèle et w est le monde de référence) et les formules du langage.

Les règles sémantiques définissant cette relation sont comme suit :

- $\langle M, w \rangle \models p$ si et seulement si $p \in \pi(w)$
- $\langle M, w \rangle \models \neg\varphi$ si et seulement si $\langle M, w \rangle \not\models \varphi$
- $\langle M, w \rangle \models \varphi \vee \psi$ si et seulement si $\langle M, w \rangle \models \varphi$ ou $\langle M, w \rangle \models \psi$
- $\langle M, w \rangle \models \Box\varphi$ si et seulement si $\forall w' \in W, si(w, w') \in R$ alors $\langle M, w' \rangle \models \varphi$
- $\langle M, w \rangle \models \Diamond\varphi$ si et seulement si $\exists w' \in W, si(w, w') \in R$ alors $\langle M, w' \rangle \models \varphi$

La définition de la satisfaction pour les propositions atomiques est basée sur l'idée de la vérité dans le monde courant (qui apparait sur la partie gauche de \models). Les règles sémantique de \neg et de \vee sont standards. La règle pour \Box capture l'idée de la vérité dans tous les mondes accessibles et la règle pour \Diamond capture l'idée de la vérité dans au moins un monde possible. Il est à noter que les deux opérateurs modaux sont duals :

$$\Box\varphi \Leftrightarrow \neg \Diamond \neg\varphi$$

$$\Diamond\varphi \Leftrightarrow \neg \Box \neg\varphi$$

Afin de comprendre les propriétés de cette logique, les notions de la validité et de la satisfiabilité sont introduites. Une formule est satisfiable si elle satisfait certaines paires modèle/ monde et insatisfiable sinon. Une formule est vraie dans un modèle si elle est satisfaite pour tous monde dans le modèle et valide dans une classe de modèles si elle est vraie dans tous modèles dans la classe. Enfin, une formule est valide (notée $\models \varphi$) si elle est vraie dans la classe de tous les modèles.

Les deux propriétés de base de cette logique sont comme suit :

$$\models \Box(\varphi \Rightarrow \psi) \Rightarrow (\Box\varphi \Rightarrow \Box\psi) \quad (K)$$

$$\text{si } \models \varphi \text{ alors } \models \Box\varphi$$

Etant donné que l'axiome K est valide, c'est un théorème dans n'importe quelle axiomatisation de la logique modale normale. Similairement, la deuxième propriété est la règle d'inférence dans n'importe quelle axiomatisation de la logique modale normale (généralement appelée règle de nécessité).

La plupart des propriétés de la logique modale normale découlent des propriétés de la relation d'accessibilités R dans les modèles. Il existe plusieurs axiomes qui correspondent à certaines propriétés de R : l'étude de comment les propriétés de R correspondent aux axiomes est appelée : la théorie de correspondance. Nous considérons juste quatre axiomes : l'axiome T (qui correspond à la relation d'accessibilité réflexive), l'axiome D (qui correspond à la relation d'accessibilité sérielle), l'axiome 4 (qui correspond à la relation d'accessibilité transitive) et l'axiome 5 (qui correspond à la relation

d'accessibilité euclidienne) :

- (T) $\Box\varphi \Rightarrow \varphi$
- (D) $\Box\varphi \Rightarrow \Diamond\varphi$
- (4) $\Box\varphi \Rightarrow \Box\Box\varphi$
- (5) $\Diamond\varphi \Rightarrow \Box\Diamond\varphi$

Le résultat de la théorie de correspondance permet directement d'obtenir les résultats de complétude pour un rang de la logique modale normale. Un système de logique peut être considéré comme un ensemble de formules valides dans certaines classes de modèles. Un élément de cet ensemble est appelé un théorème de la logique, noté \vdash . La notation $K\Sigma_1\dots\Sigma_n$ est utilisée pour représenter la plus petite logique modale normale contenant les axiomes $\Sigma_1, \dots, \Sigma_n$.

Pour les axiomes T , D , 4 et 5 il existe 16 systèmes distincts de la logique (étant donné que : $2^4 = 16$). Cependant, certains de ces systèmes sont équivalents car ils contiennent les mêmes théorèmes et comme résultats il existe seulement 11 systèmes différents : K , $K4$, $K5$, KD , $KT(= KDT)$, $K45$, $KD5$, $KD4$, $KT4(= KDT4)$, $KD45$ et $KT5(= KT45, KDT5, KDT45)$. Les systèmes modaux suivant sont très utilisés :

- KT connu sous le nom de : T .
- $KD45$ connu sous le nom de : $S5 - faible$.
- $KT4$ connu sous le nom de : $S4$.
- $KT5$ connu sous le nom de : $S5$.

2.3 Logique modale normale pour les connaissances et les croyances

Pour utiliser la logique modale normale comme une logique épistémique la formule $\Box\varphi$ est lue : il est connu que φ . Les mondes dans un modèle sont interprétés comme des alternatives épistémiques. La relation d'accessibilité définit les alternatives pour chaque monde. Cette logique manipule les connaissances d'un seul agent, ainsi pour manipuler les connaissances de plusieurs agents un ensemble d'index de relation d'accessibilité est ajouté à la structure du modèle. Un modèle est alors une structure $\langle W, R_1, \dots, R_n, \pi \rangle$ ou R_i est la relation d'accessibilité de connaissances pour l'agent i . Le langage définie ci-dessus est entendu en remplaçant l'opérateur modale \Box par un ensemble indexé d'opérateurs modaux unaires $\{K_i\}$ ou $i \in \{1, \dots, n\}$. La formule $K_i\varphi$ est lue : i sait que φ . La règle sémantique de \Box est remplacée par la règle suivante :

$\langle M, w \rangle \models K_i\varphi$ si et seulement si $\forall w' \in W$, si $(w, w') \in R_i$ alors $\langle M, w' \rangle \models \varphi$

Par conséquent, chaque opérateur K_i a exactement les mêmes propriétés que \Box . Pour chaque système modal Σ , ci-dessus, un système Σ_n correspondant est défini pour la logique multi-agents. Ainsi K_n est la plus petite logique multi-agents épistémique et $S5_n$ est la plus grande.

La convenance des axiomes D_n , T_n , 4_n et 5_n pour la logique des connaissances et des croyances est comme suit. L'axiome D_n atteste que les connaissances d'un agent sont non contradictoires, il est alors réécrit comme suit :

$$K_i\varphi \Rightarrow \neg K_i\neg\varphi$$

Cet axiome stipule : si l'agent i sait que φ , alors l'agent i ne sait pas $\neg\varphi$. C'est une propriété raisonnable pour les connaissances et les croyances.

L'axiome T_n est souvent appelé l'axiome de la connaissance, étant donné qu'il atteste que ce qui est connue est vrai. Il est accepté comme un axiome qui distingue entre les connaissances et les croyances : les connaissances sont définies comme des croyances vraies : i sait φ si i croit φ et φ est vraie. Définie comme telle, la notion de connaissances satisfait T_n .

L'axiome 4_n est appelé l'axiome d'introspection positive. L'introspection est le processus d'examen de ses propres croyances [Konolige 86]. L'axiome d'introspection positive atteste qu'un agent sait qu'est ce qu'il sait. Similairement, l'axiome 5_n est l'axiome d'introspection négative, il atteste qu'un agent est au courant de ce qu'il ne sait pas. L'appartenance de l'introspection positive et l'introspection négative aux propriétés des connaissances et des croyances est un sujet de débat. Cependant, il est généralement accepté que l'introspection positive est une propriété moins forte que l'introspection négative. Ainsi, le système modale $S5_n$ est souvent choisi comme logique de connaissances et le $S5_n - faible$ est souvent choisi comme logique de croyances.

2.4 Théorie de l'intention : Cohen et Levesque

Le formalisme défini dans [Cohen 90] est utilisé pour développer une théorie de l'intention (comme : j'ai l'intention de ..). Suivant Bratman [Bratman 87] [Bratman 90], Cohen et Levesque identifient sept propriétés qu'une théorie raisonnable de l'intention doit satisfaire :

1. Les intentions posent problèmes pour les agents qui ont besoin de déterminer la façon de les atteindre.
2. Les intentions fournissent un filtre pour adopter d'autres intentions, qui ne doivent pas être conflictuelles.
3. Les agents suivent le succès des leurs intentions, et ils sont portés à réessayer si leur tentative échoue.
4. Les agents croient que leurs intentions sont possibles.
5. Les agents ne croient pas qu'ils ne réaliseront par leurs intentions.
6. Sous certaines circonstances, les agents croient qu'ils réaliseront leurs intentions.
7. Les agents ont besoin de ne pas compter tous les effets secondaires attendus de leurs intentions.

Etant donné ces critères, Cohen et Levesque proposent deux niveaux pour le problème de la formalisation de la théorie des intentions. D'abord, ils construisent une logique d'agent rationnel. En tête de ce formalisme, ils introduisent un nombre de constructeurs dérivés qui constituent : une théorie partielle d'action rationnelle. L'intention est l'une de ces constructeurs. Syntaxiquement, cette logique est multi-sortée, quantifiée, multimodale logique, avec égalité et qui contient 04 modalités primaires :

- $(BEL\ x\ \varphi)$: l'agent x croit φ ,
- $(HAPPENS\ \alpha)$: l'action α va se passer ensuite ,
- $(GOAL\ x\ \varphi)$: l'agent x a pour but φ ,
- $(DONE\ \alpha)$: L'action α vient de se passer.

La sémantique de BEL et $GOAL$ est définie par le biais des mondes possibles au sens usuel : à chaque agent est attribuée une relation d'accessibilité de croyance et une relation d'accessibilité de but. La relation d'accessibilité de croyance est euclidienne, transitive et sérielle donnant une logique de croyance de $KD45$. La relation d'accessibilité de but est sérielle, donnant une logique conative KD . Il est supposé que chaque relation de but d'un agent est un sous ensemble de sa relation de croyance impliquant qu'un agent n'a pas comme but quelque chose qui ne le croit pas. Les mondes de ce formalisme sont une séquence d'évènements discrets qui peuvent être entendu au passé et au futur.

Les deux opérateurs temporels de base $HAPPENS$ et $DONE$ sont augmentés par quelques opérateurs pour décrire la structure des séquences d'évènements. Les deux opérateurs les plus importants sont ; et ? :

- $\alpha; \alpha'$: α est suivie par α' ,
- $\alpha?$: l'action de test α .

Les opérateurs standards du futur de la logique temporelle : \Box pour toujours et \Diamond pour parfois peuvent être définis comme abréviations, avec un opérateur " parfois strict" : $LATER$:

- $\Diamond\alpha =^{df} \exists x(HAPPENS\ x; \alpha?)$
- $\Box\alpha =^{df} \neg \Diamond \neg \alpha$
- $(LATER\ p) =^{df} \neg p \wedge \Diamond p$

L'opérateur de précédence ($BEFORE\ p\ q$) peut aussi être dérivé, il est vrai si p se passe avant q . Une importante supposition est que tous les buts sont finalement passés : $\Diamond \neg (GOAL\ x(LATER\ p))$.

Le premier constructeur major dérivé est : le but persistant :

$$(P - GOAL\ x\ p) =^{df} (GOAL\ x(LATER\ p)) \wedge (BEL\ x\ \neg p) \wedge$$

$$[BEFORE((BEL\ x\ p) \vee (BEL\ x\ \Box\ \neg p)) \neg (GOAL\ x(LATER\ p))]$$

Ainsi un agent a un but persistant p si :

- Il a un but p qui devient finalement vrai et croit que p n'est pas actuellement vraie.
- Avant qu'un agent abandonne un but, les conditions suivantes doivent être satisfaites : l'agent croit que le but est satisfait ou, l'agent croit que le but ne sera jamais satisfait.

La définition du but persistant c.à.d. " vouloir que quelque chose devient vrai " est similaire que la première définition de l'intention : avoir l'intention d'agir mais qui est légèrement différente [Cohen 90] :

$$(INTEND\ x\ \alpha) =^{df} (P - GOAL\ x\ [DONE\ x\ (BEL\ x\ (HAPPENS\ \alpha))]; \alpha)$$

2.5 Croyance, désir et intention : Rao et Georgeff

Cohen et Levesque ont utilisé deux attitudes basiques, à savoir, les croyances et les buts [Rao, Georgeff 95]. Les autres attitudes, comme l'intention, sont définies à partir de ces deux-là. Rao et Georgeff ont développé une architecture logique pour la théorie des agents basée sur trois modalités à savoir, les croyances, les désirs et les intentions.

Les croyances d'un agent représentent les informations qu'il a sur le monde. Les désirs d'un agent sont déterminés suivant les tâches qui lui sont attribuées. Un agent peut ne pas atteindre tous ses désirs, et les désirs peuvent être inconsistants. Les intentions d'un agent représentent les désirs qu'il s'est engagé à les atteindre. L'intuition derrière les intentions est la suivante : un agent en général n'est pas capable d'atteindre tous ses désirs, même si ces désirs sont consistants. Il est obligé de fixer un ensemble disponible de désirs et prendre les ressources nécessaires pour les atteindre. Les désirs choisis sont les intentions.

Dans les systèmes BDI, le système modale $S5 - faible$ (ou KD45) est adopté pour les croyances et les axiomes : D et K pour les désirs et les intentions. En effet, ils doivent être clos sous implication et consistants. La règle de nécessité est aussi adoptée pour les croyances, les désirs et les intentions.

Les systèmes BDI possèdent deux types de propriétés, à savoir : les propriétés statiques et les propriétés dynamiques :

- *Propriétés statiques* : la relation statique entre les croyances, les désirs et les intentions peut être considérée avec deux différentes dimensions : l'une avec l'ensemble des mondes possibles et l'autre avec la structure des mondes possibles. Etant donné deux relations, quatre relations entre elles sont possibles : l'une est un sous ensemble de l'autre et vice versa, et leurs intersections sont nulles ou non nulles. Similairement, chaque monde possible est un arbre de temps. Il existe alors quatre relations structurelles possibles qui peuvent être définies entre n'importe quelle paire de mondes : l'un peut être un sous monde de l'autre ou vice versa, ou les mondes peuvent être identique ou non.

L'ensemble des mondes possibles et la relation structurelle des croyances, des désirs et des intentions peuvent être combinés. Cette combinaison donne naissance à douze différents systèmes BDI. Certaines de ces relations et axiomatisations peuvent être définies les une à partir des autres. Trois de ces relations et axiomatisations sont connus sous les termes de : réalisme [Cohen 90] : si un agent croit une proposition il l'a désirera, réalisme fort [Rao 91 a] : si un agent désire atteindre une proposition il croira que la proposition est une option et faible réalisme [Rao 91 b] : si un agent désire atteindre une proposition il ne croira pas que la négation de la proposition est inévitable.

Le choix du système BDI dépend aussi des propriétés désirées par un agent. Par exemple : d'autres recherches ont proposé une asymétrie entre les croyances et les autres attitudes [Rao 91 b] et un important principe de fermeture [Cohen 90]. Le premier requiert qu'un agent maintient la consistance entre ses croyances, désirs et intentions mais pas la complétude. Le second requiert que les croyances, les désirs et les intentions ne doivent pas être clos sous implication des autres attitudes. Toutes les propriétés ci-dessus sont satisfaites par le système BDI dans lequel les

interactions des croyances, désirs et intentions sont non nulles. D'autres systèmes BDI dans lesquels les intentions sont des sous mondes des désirs qui sont des sous mondes des croyances satisfont certaines, mais pas toutes, de ces propriétés.

- *Propriétés dynamiques* : l'aspect important d'une architecture BDI est la notion d'engagement dans les précédentes décisions.

Un engagement a deux parties : la première est la condition que l'agent est engagé de maintenir, nommée la condition de l'engagement et la deuxième est la condition sous laquelle l'agent abandonne l'engagement, nommée condition de terminaison.

Un agent peut choisir quoi faire avec ses intentions. De ce fait, les conditions de l'engagement sont restreintes aux intentions. Un agent peut s'engager dans une intention basée sur l'objet que l'intention est satisfaite dans un chemin futur ou dans tous les chemins futurs menant à des conditions d'engagement différentes et ainsi des comportements dynamiques différents.

Différentes conditions de terminaison résultent suite aux variations du comportement. Par exemple, l'un peut définir un agent à engagement aveugle qui ne n'importe quelle chance où ses croyances et désirs sont en conflit avec ses engagements, aussi l'autre peut définir un agent à esprit étroit qui refuse de changer ses croyances et supprime ses engagement en conséquence.

Les différentes formes de terminaison et d'engagement peuvent être exprimées comme axiome de la logique BDI. Les contraintes sémantiques peuvent être placées suivant l'évolution dynamique de la relation d'accessibilité. Le choix de la stratégie d'engagement est défini par l'utilisateur suivant l'application voulue.

Le but de cette formalisation est de construire des systèmes formellement vérifiables et pratiques.

2.6 Logique de coalition

La logique de coalition LC [Pauly 01] est une logique modale propositionnelle. Elle manipule une collection indexée d'opérateurs modaux unaires $\langle \mathcal{C} \rangle$ et $[\mathcal{C}]$ où \mathcal{C} est la coalition c.a.d un sous ensemble d'un ensemble donné Ag .

$\langle \mathcal{C} \rangle_\varphi$ est interprétée comme suit : \mathcal{C} peut atteindre φ ou \mathcal{C} est efficace pour \mathcal{C} . Les formules de LC sont définies par la grammaire suivante :

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle \mathcal{C} \rangle_\varphi$$

Telle que $p \in \Phi_0$ (où Φ_0 est un ensemble de variables Booléennes) est une proposition atomique et \mathcal{C} un sous ensemble de Ag . Par convention, le reste des connecteurs de la logique classique sont définies comme des abréviations : $\perp \equiv \neg\top$, $\varphi \rightarrow \psi \equiv (\neg\varphi) \vee \psi$ et $\varphi \leftrightarrow \psi \equiv (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$.

Un modèle \mathcal{M} (sur Φ_0, Ag) est le triplet $\mathcal{M} = \langle S, \xi, \pi \rangle$ où :

- $S = \{s_1, \dots, s_0\}$ est un ensemble fini et non vide d'états ;

- $\xi : 2^{Ag} \times S \rightarrow 2^{2^S}$ est une fonction d'efficacité, ou $T \in \xi(\mathcal{C}, s)$ veut dire : d'un état s , la coalition \mathcal{C} peut opérer pour assurer que l'état prochain sera un membre de T ;
- $\pi : S \rightarrow 2^{\Phi_0}$ est une fonction d'évaluation.

$\mathcal{G}en$ est l'ensemble de tous les modèles généraux. Il est possible de définir un nombre de contraintes sur les fonctions d'efficacité en se basant sur exactement le genre de scénario à modéliser.

Une interprétation, pour la logique de coalition, est une paire \mathcal{M}, s où \mathcal{M} est un modèle et s est un état dans \mathcal{M} .

La relation de satisfiabilité \models_{LC} pour la logique de coalition relie entre les interprétations et les formules de la LC . Une coalition \mathcal{C} peut renforcer φ dans s si pour quelques $T \in \xi(\mathcal{C}, s)$, φ est vraie dans tous $t \in T$. Cela veut dire que \mathcal{C} peut faire un choix tel que, sans tenir compte des choix des autres, φ tiendra. Formellement, la relation de satisfiabilité est définie par la règle d'induction suivante :

$$\mathcal{M}, s \models_{CL} \langle \mathcal{C} \rangle_{\varphi} \text{ si et seulement si } \exists T \in \xi(\mathcal{C}, s) \text{ tel que } \forall t \in T, \mathcal{M}, t \models_{CL} \varphi$$

Les notions de vérités de φ dans un modèle ($\mathcal{M} \models_{LC} \varphi$) et de validité dans une classe de modèles \mathcal{C} ($\mathcal{C} \models_{LC} \varphi$) sont définies comme usuellement. La relation d'inférence \vdash_{LC} et les axiomes de la logique des coalitions sont comme suit :

$$(Prop) \quad \vdash_{LC} \psi$$

$$(Ag\perp) \quad \vdash_{LC} \neg \langle Ag \rangle_{\perp}$$

$$(\top) \quad \vdash_{LC} \neg \langle \emptyset \rangle_{\perp} \rightarrow \langle \mathcal{C} \rangle_{\top}$$

$$(\perp) \quad \vdash_{LC} \langle \mathcal{C} \rangle_{\perp} \rightarrow \langle \mathcal{C}' \rangle_{\perp}$$

$$(Ag) \quad \vdash_{LC} \neg \langle \emptyset \rangle_{\neg \varphi} \rightarrow \langle Ag \rangle_{\varphi}$$

$$(S) \quad \vdash_{LC} (\langle \mathcal{C}_1 \rangle_{\varphi_1} \wedge \langle \mathcal{C}_2 \rangle_{\varphi_2}) \rightarrow \langle \mathcal{C}_1 \cup \mathcal{C}_2 \rangle_{(\varphi_1 \wedge \varphi_2)}$$

$$(MP) \quad \vdash_{LC} \varphi \rightarrow \psi, \vdash_{LC} \varphi \Rightarrow \vdash_{LC} \psi$$

$$(Distr) \quad \vdash_{LC} \varphi \rightarrow \psi \Rightarrow \vdash_{LC} \langle \mathcal{C} \rangle_{\varphi} \rightarrow \langle \mathcal{C} \rangle_{\psi}$$

Il est a noter que dans $(Prop)$, ψ est une tautologie propositionnelle, dans l'axiome (\perp) , $\mathcal{C}' \subseteq \mathcal{C}$ et pour (S) , $\mathcal{C}_1 \cap \mathcal{C}_2 = \emptyset$. Aussi, la relation d'inférence est correcte et complète dans la classe des modèles dite de faible jouabilité [Pauly 01].

2.7 Logique temporelle du temps en alternance

Le premier article sur la logique temporelle du temps en alternance (*LTTA*) est [Alur 97]. Ce travail préliminaire était restreint aux jeux à base de tour de rôle c.à.d. aux jeux où chaque transition est gouvernée par un seul agent. Dans [Alur 99] une structure générale nommée systèmes de transitions en alternance (*STA*) a été définie, où les choix sont exprimés comme des ensembles de résultats possibles. Dans [Alur 02], les auteurs changent le modèle au : structure de jeu simultanée (*SJS*) où les choix sont identifiés avec des étiquettes explicites. Les *STA* les *SJS* ont été prouvés équivalente [Goranko 04]. De ce fait, la sémantique de la *LTTA* est définie en termes de *STA* ou de *SJS*.

Syntaxe : Dans ce qui suit, *Atm* représente un ensemble de propositions atomiques et *Agt* est un ensemble fini de tous les agents. Soit p dans *Atm* et A dans 2^{Agt} , le langage de la *LTTA* est définie par :

$$\varphi, \psi, \dots ::= p \mid \neg\varphi \mid \varphi \wedge \psi \mid \langle\langle A \rangle\rangle X\varphi \mid \langle\langle A \rangle\rangle G\varphi \mid \langle\langle A \rangle\rangle \varphi \mathcal{U} \psi$$

où n est une formule temporelle linéaire. $\langle\langle A \rangle\rangle_\eta$ se lit : le groupe A peut garantir η peu importe ce que font les agents dans $Agt \setminus A$.

Modèles : Les modèles de la *LTTA* sont représentés en terme de systèmes de transition en alternance qui sont des tuples $\mathcal{M} = \langle W, \delta, v \rangle$ où :

- W est un ensemble non vide d'états.
- $\delta : S \times Agt \rightarrow 2^{2^W}$ est une fonction de transition ou pour chaque moment et chaque agent elle retourne une famille d'ensemble non vide de moments successeurs possibles.
- $v : Atm \rightarrow 2^W$ est une fonction de valuation.

Chaque $\mathcal{Q} \in \delta(w, a)$ peut être vu comme un choix d'un agent d'une action particulière dans son répertoire.

Les *STA* synchrones à étape bloquée sont utilisés. Cela signifie que dans chaque état, tous les agents procèdent simultanément. La fonction δ est non bloquante (les actions des agents sont toujours compatible) et le choix simultané de chaque agent dans *Agt* détermine un unique prochain état : soit $Agt = \{a_1, \dots, a_n\}$, pour chaque état $w \in W$ et chaque ensemble $\{\mathcal{Q}_1, \dots, \mathcal{Q}_n\}$ de choix $\mathcal{Q}_i \in \delta(w, a_i)$, l'intersection $\mathcal{Q}_1 \cap \dots \cap \mathcal{Q}_n$ est un singletons.

Une stratégie pour un agent a est une application $f_a : W^+ \rightarrow 2^W$ qui associe à chaque séquence d'états $w_0 \dots w_k$ un élément de $\delta(w_k, a)$. Une stratégie collective pour un ensemble d'agents $A \subseteq Agt$ est un tuple $F_A = \langle f_{a_1}, \dots, f_{a_n} \rangle$ de stratégies, une pour chaque agent dans *Agt*. Le résultat de F_A de w est définie comme :

$$out(w, F_A) = \{ \lambda \mid \lambda = w_0 w_1 w_2 \dots, w_0 = w, \forall i \geq 0 (w_{i+1} \in \bigcap_{a \in A} f_a(w_0 \dots w_i)) \}$$

Un profil de stratégie est une stratégie collective F_{Agt} pour tous les agents de *Agt* et similairement un tuple $\langle \mathcal{Q}_1, \dots, \mathcal{Q}_n \rangle$ (un \mathcal{Q}_i pour chaque $i \in Agt$) est nommé un profil de choix.

Sémantique et axiomatisation : $\lambda[i]$ est la i -ème position dans le chemin λ . Une formule est évaluée en ce qui concerne un *STA* $\mathcal{M} = \langle W, \delta, v \rangle$ et un moment $w \in W$.

$$\mathcal{M}, w \models \langle\langle A \rangle\rangle X\varphi \Leftrightarrow \exists F_A, \forall \lambda \in \text{out}(w, F_A), \mathcal{M}, \lambda[1] \models \varphi$$

$$\mathcal{M}, w \models \langle\langle A \rangle\rangle G\varphi \Leftrightarrow \exists F_A, \forall \lambda \in \text{out}(w, F_A), \mathcal{M}, \lambda[i] \models \varphi, \forall i \geq 0$$

$$\mathcal{M}, w \models \langle\langle A \rangle\rangle_{\varphi} \mathcal{U}_{\psi} \Leftrightarrow \exists F_A, \forall \lambda \in \text{out}(w, F_A), \exists i \geq 0 (\mathcal{M}, \lambda[i] \models \psi, \forall j \in [0, i], \mathcal{M}, \lambda[j] \models \varphi)$$

L'axiomatisation complète suivante de la *LTTA* (comme une extension de n'importe quelle axiomatisation de la logique propositionnelle) est donnée dans [Goranko 06]. $\mathcal{M}, w \models \langle\langle \emptyset \rangle\rangle_{\eta}$ veut dire que : η est vraie sans tenir compte des choix faits par A .

$$(\perp) \quad \neg \langle\langle A \rangle\rangle X \perp$$

$$(\top) \quad \langle\langle A \rangle\rangle X \top$$

$$(N) \quad \neg \langle\langle \emptyset \rangle\rangle X \neg \varphi \rightarrow \langle\langle A_{gt} \rangle\rangle X \varphi$$

$$(S) \quad \langle\langle A_1 \rangle\rangle X \varphi \wedge \langle\langle A_2 \rangle\rangle X \psi \rightarrow \langle\langle A_1 \cup A_2 \rangle\rangle X (\varphi \wedge \psi) \quad \text{si } A_1 \cap A_2 = \emptyset$$

$$(FP_G) \quad \langle\langle A \rangle\rangle G \varphi \equiv \varphi \wedge \langle\langle A \rangle\rangle X \langle\langle A \rangle\rangle G \varphi$$

$$(GFP_G) \quad \langle\langle \emptyset \rangle\rangle G (\theta \rightarrow (\varphi \wedge \langle\langle A \rangle\rangle X \theta)) \rightarrow \langle\langle \emptyset \rangle\rangle G (\theta \rightarrow (\varphi \wedge \langle\langle A \rangle\rangle G \varphi))$$

$$(FP_U) \quad \langle\langle A \rangle\rangle \psi \mathcal{U} \varphi \equiv \varphi \vee (\psi \wedge \langle\langle A \rangle\rangle X \langle\langle A \rangle\rangle \psi \mathcal{U} \varphi)$$

$$(LFP_U) \quad \langle\langle \emptyset \rangle\rangle G ((\varphi \vee (\psi \wedge \langle\langle A \rangle\rangle X \theta)) \rightarrow \theta) \rightarrow \langle\langle \emptyset \rangle\rangle G (\langle\langle A \rangle\rangle \psi \mathcal{U} \varphi \rightarrow \theta)$$

Il est à noter que l'axiome (N) découle du déterminisme du profil du choix (les actions sont construites par les choix simultanés de chaque agent dans le système) : quand chaque agent opte pour un choix, la prochaine étape est complètement déterminée, ainsi si quelque chose n'est pas stable la coalition de tous les agents (A_{gt}) peut toujours travailler ensemble pour mettre sa négation à vraie.

L'axiome (S) dit que deux coalitions peuvent combiner leurs efforts afin d'assurer une conjonction de propriétés si elles sont disjointes. Il est à noter que de (S) découle que $\langle\langle A_1 \rangle\rangle_{\varphi} \wedge \langle\langle A_2 \rangle\rangle_{\neg \varphi}$ n'est pas satisfiable pour A_1 et A_2 disjoints. Donc deux coalitions disjointes ne peuvent pas garantir des propositions inconsistantes.

L'axiome (FP_G) caractérise la modalité globale comme point fixe de la modalité suivante, et l'axiome (GFP_G) indique qu'il est le plus grand point fixe.

L'axiome (FP_U) caractérise l'opérateur 'jusqu'au' comme un point fixe de l'opérateur suivant, et l'axiome (LFP_U) exprime que la sémantique impose de prendre le plus petit point fixe.

2.8 La logique individuelle VACQ

Cette logique [Herzig 08] permet de représenter des informations de la forme : l'agent i veille à ce que ϕ .

Le langage \mathcal{L}_{VACQ_n} de la logique individuelle VACQ est construit à partir d'un ensemble atomique, dénombrable de propositions ATM et un ensemble fini d'agents $AGT = \{1, \dots, n\}$. Il est définie par :

$$\phi ::= p \mid \neg\phi \mid (\phi \wedge \psi) \mid [i]\phi \mid \Box\phi$$

Où p dans ATM et i dans AGT . $[i]\phi$ est lue : l'agent i veille à ce que ϕ (quel que soit ce que font les autres agents), et $\Box\phi$ est lu : ϕ est historiquement nécessaire, ou : ϕ tient quel que soit le choix des autres agents.

Sémantique : la sémantique de (VACQ) est proche de la logique modale standard. Elle a été proposée dans [Kooi 08]. Un modèle de Kripke pour la logique (VACQ) est un tuple $\mathcal{W} = \langle W, R, V \rangle$ où :

- W est un ensemble non vide ;
- R est une application associant à chaque $i \in AGT$ une relation d'équivalence R_i dans W tel que pour tout $(w_1, \dots, w_n) \in W^n$, $\bigcap_{i \in AGT} R_i(w_i) \neq \emptyset$;
- V est une fonction d'évaluation $V : ATM \rightarrow 2^W$.

Une formule est évaluée par un modèle et un point :

$$\begin{aligned} \mathcal{W}, w \models p & \text{ si et seulement si } w \in V(p), \text{ pour } p \in ATM \\ \mathcal{W}, w \models \neg\phi & \text{ si et seulement si } \mathcal{W}, w \not\models \phi \\ \mathcal{W}, w \models \phi \wedge \psi & \text{ si et seulement si } \mathcal{W}, w \models \phi \text{ et } \mathcal{W}, w \models \psi \\ \mathcal{W}, w \models \Box\phi & \text{ si et seulement si } \mathcal{W}, w' \models \phi \text{ pour tout } w' \in W \\ \mathcal{W}, w \models [i]\phi & \text{ si et seulement si } \mathcal{W}, w' \models \phi \text{ pour tout } w' \in R_i(w) \end{aligned}$$

La notion de validité et la notion de satisfiabilité dans les modèles de Kripke restent inchangées.

Axiomatisation : la STT a les axiomes suivants :

$S5(\Box)$: le schéma d'axiome de S_5 pour \Box ;

$S5(i)$: le schéma d'axiome de S_5 pour chaque $[i]$, pour chaque $i \in AGT$;

$(\Box \rightarrow i)$: $\Box\phi \rightarrow [i]\phi$, pour chaque $i \in AGT$;

AIA_n : $(\diamond[1]\phi_1 \wedge \dots \wedge \diamond[n]\phi_n) \rightarrow \diamond([1]\phi_1 \wedge \dots \wedge [n]\phi_n)$.

AIA_n est nommé le schéma d'axiome pour l'indépendance des agents. L'axiomatisation de la VACQ a pour règle d'inférence le modus ponens standard et la nécessitation pour \Box .

Une formule ϕ de \mathcal{L}_{VACQ_n} est valide dans un modèle de Kripke si et seulement si ϕ est prouvable des schémas : $S5(\Box)$, $S5(i)$, $(\Box \rightarrow i)$ et AIA_n par les règles de modus ponens et la nécessitation pour \Box .

2.9 Conclusion

Les différentes logiques modales présentées dans ce chapitre ont divers mérites dans le domaine de la représentation des connaissances et le raisonnement dans les systèmes multi-agents. En effet, chacune de ces logiques contribue à l'analyse logique de certaines propriétés des agents, par exemple : les croyances, les connaissances, les actions etc.

Nous présentons dans le chapitre suivant un état de l'art sur la logique possibiliste, logique de base pour une nouvelle logique multi-agents qui sera développée.

Chapitre 3

La Logique possibiliste

3.1 Introduction

La théorie des possibilités a été définie par Zadeh [Zadeh 78] puis développée par Dubois et Prade [Dubois, Prade 87 a] à partir de la théorie des sous-ensembles flous. C'est une extension de la logique classique qui associe à chaque formule un degré de certitude. Elle est présentée comme un cadre alternatif pour représenter des informations incertaines de nature non probabiliste. Ainsi, D. Dubois et H. Prade introduisent des mesures de possibilités et de nécessités comme des outils de représentation de l'incertain plus général, permettant de représenter l'ignorance et de prendre en compte la pertinence d'une information incertaine [Dubois, Prade 04].

3.2 Langage

Soit \mathcal{L} un langage de la logique propositionnelle. Les formules bien formées de \mathcal{L} , notées $a, b, c, \dots, a_i, \dots$, sont construites d'atomes, notés p, q, r, \dots et les connecteurs usuels \neg, \wedge, \vee (où $a \vee b =_{def} \neg(\neg a \wedge \neg b)$) avec les parenthèses $()$, en utilisant les règles usuelles. \top et \perp désignent la tautologie et la contradiction respectivement.

Définition 3.1 Une formule possibiliste N -valuée est un couple (a, α) où a est une formule bien formée de la logique propositionnelle et α un nombre réel appartenant à l'intervalle $]0, 1]$.

(a, α) exprime que a est certaine à au moins au degré α . En effet, la formule possibiliste (a, α) est la traduction syntaxique de la contrainte sémantique $N(a) \geq \alpha$ où N indique le degré de nécessité de cette formule.

Remarque 3.1 Les formules de la forme $(a, 0)$ ne sont pas prises en compte étant donné que $\forall a, N(a) > 0$.

Définition 3.2 Une base de connaissances possibilistes (ou un ensemble de formules possibilistes) est une conjonction finie de formules possibilistes. Un ensemble de formules possibilistes sera noté par Φ :

$$\Phi = (a_1, \alpha_1) \wedge \dots \wedge (a_n, \alpha_n) = \{(a_1, \alpha_1), \dots, (a_n, \alpha_n)\}$$

Exemple 3.1 $\Phi = \{(\neg a \vee b, 0.1), (a \vee b \vee c, 0.3), (\neg b \vee d, 0.5), (\neg d, 0.7)\}$ est une base de connaissances possibilistes.

Pour toute formule possibiliste (a, α) , la fonction Val est définie par $Val((a, \alpha)) = \alpha$.

La projection classique d'un ensemble de formules possibilistes est l'ensemble de formules classiques, noté Φ^* , obtenu à partir de Φ en ignorant les valuations, c'est-à-dire que si $\Phi = \{(a_i, \alpha_i), i = 1, \dots, n\}$ alors $\Phi^* = \{a_i, i = 1, \dots, n\}$.

Le langage ainsi défini est restrictif par rapport à deux niveaux :

- Au premier niveau, seules les conjonctions de formules possibilistes sont considérées. De ce fait, les disjonctions, les négations, les quantifications existentielles et universelles de formules possibilistes ne sont pas prises en compte. Par exemple $(a, \alpha) \vee (b, \beta)$, ou $\exists x(a, \alpha)$ n'appartiennent pas au langage.
- Au second niveau, le langage ne permet pas l'encapsulation des valuations telle que $((a, \alpha), \beta)$.

Les notations suivantes sont adoptées dans le document :

Notations 3.1 - $M(a)$ est l'ensemble des modèles de a .

- ω désigne une interprétation pour \mathcal{L} .
- Ω désigne l'ensemble des interprétations pour \mathcal{L} .
- $\omega \models a$ signifie $\omega \in M(a)$.
- x, y, \dots désignent des variables.

3.3 Le treillis des valuations

Dans ce qui suit, nous allons définir un ordre total sur les valuations α . De ce fait, nous allons utiliser un treillis : $\mathcal{V} = [0, 1]$ muni des opérateurs : minimum (min), maximum (max) (qui définissent une structure d'ordre) et la complémentation ($1 - (\cdot)$) (qui définit une structure d'ordre inverse).

Cet ordre total sur les valuations permet de considérer une base de connaissances possibiliste comme un ensemble flou de formules classiques, dont la fonction d'appartenance a ses valeurs dans \mathcal{V} .

Naturellement, les α -coupes (Φ_α) (où $\alpha \in \mathcal{V}$) et α -coupes strictes $\Phi_{\bar{\alpha}}$ d'une base de connaissances possibilistes (Φ) sont définies par :

$$\Phi_\alpha = \{(a_i, \alpha_i) \in \Phi \mid \alpha_i \geq \alpha\}$$

$$\Phi_{\bar{\alpha}} = \{(a_i, \alpha_i) \in \Phi \mid \alpha_i > \alpha\}$$

3.4 Sémantique de la logique possibiliste

3.4.1 Les distributions de possibilité normalisées comme sémantique de la logique possibiliste

3.4.1.1 Distribution de possibilités

Soit \mathcal{L} un langage de la logique classique. Soit Ω l'ensemble des interprétations pour \mathcal{L} . Alors une formule a de \mathcal{L} peut être vue comme une restriction de l'ensemble des interprétations possibles. En effet, l'ensemble des modèles d'un ensemble vide de formules est Ω , et rajouter une nouvelle formule a à un ensemble de formules Ψ revient à

ajouter une nouvelle contrainte restreignant ainsi l'ensemble des modèles de Ψ . Ce dernier devient alors l'ensemble des modèles de Ψ et de a (qui est inclus dans le précédent).

La logique possibiliste a un comportement similaire : les contraintes devenant graduelles. En effet, une formule possibiliste est une contrainte souple restreignant (plus ou moins) l'ensemble des interprétations possibles ; plus la formule possibiliste considérée a une valuation élevée, moins la contrainte est souple.

En logique possibiliste, la satisfaction d'une formule est définie en terme de distribution de possibilités dans le monde Ω . Une distribution de possibilité [Zadeh 78], notée π , est une fonction de Ω vers $[0, 1]$. $\pi(\omega)$ représente le degré de compatibilité de l'interprétation ω avec les croyances disponibles.

Par convention, $\pi(\omega) = 1$ signifie que ω est totalement compatible avec les croyances disponibles, alors que $\pi(\omega) = 0$ signifie que ω n'est certainement pas compatible avec les croyances disponibles. $\pi(\omega) > \pi(\omega')$ signifie que ω est plus compatible (plausible) que ω' .

Dans le cas où il existe un élément ω de Ω tel que $\pi(\omega) = 1$ alors la distribution de possibilité π est dite normalisée. La propriété de normalisation reflète l'existence d'au moins un élément de Ω qui soit totalement compatible avec les connaissances disponibles (en d'autre terme, elle reflète la consistance de l'ensemble des connaissances disponibles).

3.4.1.2 Les mesures de possibilités et de nécessité

La distribution de possibilité π donne naissance à la notion de la mesure de possibilité [Zadeh 78] notée Π . C'est une fonction qui associe à chaque formule a un nombre réel compris entre 0 et 1. Elle est définie par :

$$\begin{aligned} \Pi : \mathcal{L} &\rightarrow [0, 1] \\ \Pi(a) &= \max\{\pi(\omega) \mid \omega \models a\} \end{aligned}$$

La mesure de possibilité $\Pi(a)$ estime à quel point a est possible, compte tenu de l'information disponible exprimée par π . Par conséquent :

- $\Pi(a) = 0$ signifie que a n'est pas du tout possible i.e. complètement impossible donc a est certainement fausse.
- $\Pi(a) = 1$ signifie que a est totalement possible, ce qui est beaucoup *moins fort* que a est certainement vraie.
- $\Pi(a) = \Pi(\neg a) = 1$ représente le cas de l'ignorance (le cas de l'absence totale de l'information). En effet, a et $\neg a$ sont toutes les deux totalement possibles. De ce fait, l'une des deux seulement est vraie ; d'où : $\max(\Pi(a), \Pi(\neg a)) = 1$. Cela signifie alors que de deux énoncés contraires, l'un (au moins) doit être complètement possible.

La mesure de possibilité satisfait les axiomes suivants :

- $\Pi(\perp) = 0$
- $\Pi(\top) = 1$
- $\forall a, \forall b \quad \Pi(a \vee b) = \max(\Pi(a), \Pi(b))$

Si le degré de possibilité d'une disjonction peut être directement calculé à partir des degrés de possibilités des sous formules, il n'en est pas de même pour les conjonctions. Pour ces dernières seules l'inequation suivante est vérifiée :

$$\Pi(a \wedge b) \leq \min(\Pi(a), \Pi(b))$$

Remarque 3.2 *En général, l'égalité n'est pas vérifiée. A titre d'exemple, dans le cas de l'ignorance totale, $\min(\Pi(a), \Pi(\neg a)) = \min(1, 1) = 1$ alors que $\Pi(a \wedge \neg a) = \Pi(\perp) = 0$.*

L'inégalité n'est pas aussi vérifiée pour la négation. En effet, si $\Pi(a) = 1$, alors $\Pi(\neg a)$ peut prendre n'importe quelle valeur de $[0, 1]$.

De façons duale, la mesure de nécessité [Dubois, Prade 80] induite par π , notée N est définie par :

$$\begin{aligned} N : \mathcal{L} &\rightarrow [0, 1] \\ N(a) &= \min\{1 - \pi(\omega) \mid \omega \models \neg a\} \\ &= 1 - \Pi(\neg a) \end{aligned}$$

Le degré de nécessité (ou de certitude) de a évalue à quel point a est certaine (i.e. évalue dans quelle mesure a peut être déduite à partir des croyances disponibles). Ainsi :

- $N(a) = 1$ signifie que ' a est totalement certaine', donc a est certainement vraie.
- $N(a) = 0$ signifie *seulement* que ' a n'est pas du tout certaine', ce qui est beaucoup *moins fort* que ' a est fausse'.
- $N(a) = N(\neg a) = 0$ représente l'état de l'ignorance totale et pourtant l'une seulement des deux est fausse. Plus généralement : $\min(N(a), N(\neg a)) = 0$ ce qui signifie que deux propositions qui s'opposent ne peuvent pas être en même temps considérées comme quelque peu certaines .

La mesure de nécessité satisfait les axiomes suivants :

- $N(\perp) = 0$
- $N(\top) = 1$
- $\forall a, \forall b \ N(a \wedge b) = \min(\Pi(a), \Pi(b))$

Le degré de nécessité de la disjonction est donnée par l'inégalité suivante :

$$N(a \vee b) \geq \max(N(a), N(b))$$

Remarque 3.3 *Cette inégalité n'est en générale pas une égalité. Il est de même pour la négation.*

De plus, les propriétés suivantes sont vérifiées :

- $\forall a, N(a) \leq \Pi(a)$.
- $\forall a, N(a) > 0 \Rightarrow \Pi(a) = 1$. Ou d'une façon équivalente : $\forall a, \Pi(a) < 1 \Rightarrow N(a) = 0$.

Remarque 3.4 - *Si la proposition a est prouvée vraie, alors $N(a) = 1$ (ou $\Pi(\neg a) = 0$); cela implique $\Pi(a) = 1$. Cependant, $\Pi(a) = 1$ n'est pas une condition suffisante pour affirmer que a est vrai.*

- *Quand la proposition a est prouvée être fausse, $\neg a$ est vraie. Ceci indique que $N(\neg a) = 1$ (ou d'une façon équivalente $\Pi(a) = 0$).*

3.4.1.3 Principe du minimum de spécificité

Nous introduisons ici une notion souvent utilisée en représentation des connaissances qui est le principe du minimum de spécificité.

Soit x une variable définie sur un référentiel Ω fini. Soit π_x une distribution de possibilité. Par convention, $\pi_x(\omega) = 0$ signifie qu'il est tout à fait impossible que x ait la valeur ω . $\pi_x(\omega) > 0$ signifie qu'il est quelque peu possible que x ait la valeur ω , et $\pi_x(\omega) = 1$ signifie qu'il est tout à fait possible que x ait la valeur ω .

Définition 3.3 Une distribution de possibilité π_x est dite **plus spécifique que** π'_x [Dubois, Prade 83] [Dubois, Prade 86], notée $\pi_x \leq \pi'_x$, si et seulement si $\forall \omega \in \Omega, \pi_x(\omega) \leq \pi'_x(\omega)$.

Définition 3.4 π_x est dite **strictement plus spécifique que** π'_x , notée $\pi_x < \pi'_x$, si et seulement si $\forall \omega \in \Omega, \pi_x(\omega) \leq \pi'_x(\omega)$ et il existe au moins un ω tel que pas $\pi'_x(\omega) < \pi_x(\omega)$.

Définition 3.5 Une distribution de possibilité π_x est dite **de spécificité minimale** parmi un ensemble de distributions de possibilités si et seulement s'il n'existe pas de distribution de possibilité π'_x de cet ensemble telle que $\pi_x < \pi'_x$.

Définition 3.6 Une distribution de possibilité π_x est dite **la moins spécifique** si et seulement si elle est de spécificité minimale et de plus elle est unique.

L'idée du principe de minimum de spécificité est que : étant donné un ensemble de contraintes \mathcal{C} restreignant les valeurs possibles de la variable x , alors la distribution de possibilité donnée par le principe de minimum de spécificité est celle qui alloue à chaque singleton ω de Ω le plus grand degré de possibilité compatible avec \mathcal{C} .

3.4.1.4 Satisfaction d'une base de connaissance possibiliste par la distribution de possibilité

Dans cette section, nous définissons la relation de satisfiabilité entre :

- une distribution de possibilité et une formule possibiliste ;
- une distribution de possibilité et un ensemble de formules possibilistes.

Soit π une distribution de possibilité sur \mathcal{L} et N la mesure de nécessité induite par π ; on dit que :

$$\pi \text{ satisfait } (a, \alpha) \text{ si et seulement si } N(a) \geq \alpha.$$

De la même façon, si ϕ_1, \dots, ϕ_n sont des formules possibilistes,

$$\pi \text{ satisfait } \phi_1 \wedge \dots \wedge \phi_n \text{ si et seulement si } \forall i = 1..n, \pi \text{ satisfait } \phi_i.$$

Notation 3.1 Souvent, et quand il n'y aura pas d'ambiguïté, nous noterons $\pi \models \Phi$ pour " π satisfait Φ ".

Remarque 3.5 Une même notation est utilisée pour deux concepts différents : $\omega \models a$ signifie que l'interprétation ω satisfait la formule classique a , alors que $\pi \models \Phi$ signifie que la distribution de possibilité π satisfait l'ensemble des formules possibiliste Φ .

Exemple 3.2 Soit $\Phi = \{(a, 0.7), (\neg a \vee b, 0.8)\}$

$$\begin{aligned} \pi \models \Phi &\Leftrightarrow N(a) \geq 0.7 \text{ et } N(\neg a \vee b) \geq 0.8 \\ &\Leftrightarrow \min\{1 - \pi(\omega) \mid \omega \models \neg a\} \geq 0.7 \text{ et } \min\{1 - \pi(\omega) \mid \omega \models a \wedge \neg b\} \geq 0.8 \end{aligned}$$

Soient : $[ab], [a\neg b], [\neg ab], [\neg a\neg b]$ les quatre interprétations du langage propositionnel engendrées par a, b (où $[ab]$ assoie à a et à b la valeur Vrai, etc.). Nous avons alors :

$$\pi \models \Phi \Leftrightarrow \pi([\neg ab]) \leq 0.3, \pi([\neg a\neg b]) \leq 0.3, \pi([a\neg b]) \leq 0.2, \pi([ab]) \leq 1$$

$\mathcal{PO}(\Phi)$ est l'ensemble des distributions de possibilité satisfaisant Φ :

$$\mathcal{PO}(\Phi) = \{\pi, \pi \text{ satisfait } \Phi\}$$

Elle est appelée l'ensemble des modèles possibilistes de Φ .

Φ est consistante si $\mathcal{PO}(\Phi) \neq \emptyset$, inconsistante si $\mathcal{PO}(\Phi) = \emptyset$, valide si toute distribution de possibilité sur Ω satisfait Φ (on notera $\models \Phi$).

Le couple formé de l'ensemble des distributions **de possibilité normalisées** sur Ω et de la relation de satisfiabilité constitue **la sémantique normalisée** pour la logique possibiliste N-valuée.

3.4.1.5 Conséquence logique

(b, β) est une conséquence logique de Φ (noté $\Phi \models (b, \beta)$) si et seulement si toute distribution de possibilité satisfaisant Φ satisfait aussi (b, β) . Formellement :

$$\Phi \models (b, \beta) \text{ si et seulement si } \forall \pi, (\pi \models \Phi) \Rightarrow (\pi \models (b, \beta))$$

Σ est sémantiquement équivalente à Φ , noté $\Sigma \equiv \Phi$, si et seulement si $\mathcal{PO}(\Sigma) = \mathcal{PO}(\Phi)$.

Exemple 3.3 De l'exemple 3.2, nous vérifions immédiatement que $\Phi \models (b, 0.7)$. En effet, une distribution de possibilité π qui satisfait Φ est tel que $\pi([\neg a\neg b]) \leq 0.3$ et $\pi([\neg a\neg b]) \leq 0.3$, et vérifie $N(b) = \min\{1 - \pi([\neg a\neg b]), 1 - \pi([\neg a\neg b])\} \geq 0.7$ donc $\Phi \models (b, 0.7)$.

Les propriétés suivantes sont vérifiées :

- $(a, \alpha) \models (a, \alpha') \quad \forall \alpha' \leq \alpha$.
- $\forall \alpha > 0, \models (a, \alpha)$ si et seulement si a est une tautologie.
- $(a, \alpha) \wedge (a \rightarrow b, \beta) \models (b, \min(\alpha, \beta))$. Cette forme du modus ponens possibiliste est un cas particuliers de la résolution possibiliste.

3.4.1.6 Le problème de la déduction

Soit Φ un ensemble de formules possibilistes, les formules de Φ étant considérées comme des hypothèses. Soit a une formule quelconque ; nous nous intéressons à l'information déductible sur la vérité ou la fausseté de a . En d'autres termes, nous cherchons la meilleure borne inférieure du degré de nécessité de a déductible de la connaissance

Φ (i.e. (a, α) est une conséquence logique de Φ). Cette meilleure borne inférieure est définie par :

$$Val(a, \Phi) = \max\{\alpha \in (0, 1], \Phi \models (a, \alpha)\}$$

Proposition 3.1 $Val(a, \Phi) = \min\{N(a), \pi \in \mathcal{PO}(\Phi)\}$

Exemple 3.4 De l'exemple 3.2, nous ne trouvons pas : $\alpha > 0.7$ tel que $\Phi \models (b, \alpha)$ de ce fait $Val(b, \Phi) = 0.7$.

3.4.1.7 Le modèle possibiliste le moins spécifique

Dans les sections précédentes, l'ensemble des modèles possibilistes d'une base de connaissances Φ est défini. Cependant, [Dubois et al. 94 b] ont montré qu'en pratique il n'est pas nécessaire de déterminer tous l'ensemble des modèles de Φ mais seulement un seul modèle possibiliste. Ce modèle, noté π_Φ , est obtenu en choisissant la distribution de possibilité la moins spécifique parmi l'ensemble $\{\pi, \pi \in \mathcal{PO}(\Phi)\}$. La distribution de possibilité la moins spécifique parmi celles satisfaisant Φ est définie alors comme suit :

$$\pi_\Phi(\omega) = \begin{cases} 1 & \text{si } \forall (a_i, \alpha_i) \in \Phi, \omega \models \alpha_i. \\ \{1 - \alpha_i : (a_i, \alpha_i) \in \Phi, \omega \models \neg \alpha_i\} & \text{sinon.} \end{cases}$$

Proposition 3.2 Pour toute distribution de possibilité π , π satisfait Φ si et seulement si $\pi \leq \pi_\Phi$ i.e. $\forall \omega, \pi(\omega) \leq \pi_\Phi(\omega)$.

Ainsi, (b, β) est une conséquence logique de Φ si et seulement si $\pi_\Phi(\omega)$ est inclus dans $\pi_{(b, \beta)}(\omega)$. Formellement :

$$\Phi \models (b, \beta) \text{ si et seulement si } \forall \omega, \pi_\Phi(\omega) \subseteq \pi_{(b, \beta)}(\omega)$$

Proposition 3.3

$$\Phi \models (a, \alpha) \text{ si et seulement si } \pi_\Phi \models (a, \alpha)$$

En d'autres termes, $Val(a, \Phi) = N_\Phi(a)$, où $N_\Phi(a)$ est la mesure de nécessité induite par π_Φ .

Proposition 3.4 Soient Φ, Σ deux ensembles de formules possibilistes et π_Φ la distribution de possibilité la moins spécifique parmi celles satisfaisant Φ . Alors :

$$\Phi \models \Sigma \text{ si et seulement si } \pi_\Phi \text{ satisfait } \Sigma$$

Proposition 3.5 Soit Φ un ensemble consistant de formules de la logique possibiliste, et b une formule quelconque. Alors $\Phi \models (b, \alpha)$ si et seulement si $\Phi_\alpha \models (b, \alpha)$.

Ce résultat exprime que pour prouver qu'une formule est (au moins) α – certaine, seules les formules de la base de connaissances qui sont elles-mêmes au moins α – certaines sont utiles.

3.4.2 Inconsistances partielles en logique possibiliste

3.4.2.1 Introduction

Dans les sections précédentes, la logique possibiliste est présentée comme une simple extension de la logique classique pour la manipulation des informations incertaine.

Or il serait dommage d'avoir défini une logique plus souple au niveau de la représentation de connaissances que la logique classique sans pouvoir relativiser la notion d'inconsistance.

L'objectif est de pouvoir encore utiliser les formules possibilistes qui ne sont pas affectées par l'inconsistance de la base possibiliste.

3.4.2.2 La sémantique sous normalisée de la logique possibiliste

Dans cette sous-section, une deuxième sémantique pour la logique possibiliste est proposée. Cette sémantique se vaut être plus souple et permet de traiter les inconsistances partielles.

L'impossibilité de trouver une distribution de possibilité satisfaisant Φ est dûe à la contrainte de normalisation : $\exists \omega, \pi(\omega) = 1$.

Soit Φ une conjonction de formules possibilistes N-valuées. $\mathcal{PO}_{SN}(\Phi)$ est défini comme l'ensemble des distributions de possibilité, normalisées ou non, satisfaisant Φ .

L'ensemble des distributions **de possibilité normalisées ou non**, munie de la relation de satisfaisabilité définie comme dans le cas normalisé, est appelée **sémantique sous-normalisée** pour la logique possibiliste.

Le nombre $SN(\pi) = 1 - \min\{\pi(\omega), \omega \in \Omega\}$ est appelé degré de sous normalisation de π , et le nombre $\min\{SN(\pi), \pi \in \mathcal{PO}_{SN}(\Phi)\}$ est appelé degré d'inconsistance de Φ [Dubois et al. 88] [Dubois et al. 94 b]. Il est noté $Incons(\Phi)$.

Par définition, $SN(\pi) = 1 - \min\{\pi(\omega), \omega \in \Omega\} = \min\{1 - \pi(\omega), \omega \in \Omega\} = \min\{1 - \pi(\omega), \omega \neq \perp\} = N(\perp)$ et par conséquent $Incons(\Phi) = \min\{N(\perp), \pi \in \mathcal{PO}_{SN}(\Phi)\}$, ce qui justifie l'appellation "degré d'inconsistance", par abus de langage pour "degré de certitude de la contradiction".

Dans cette sémantique, certaines des propriétés de la sémantique normalisée restent vérifiées.

Propriété 3.1 1. $\forall a \in \mathcal{L}, \forall b \in \mathcal{L} \quad \Pi(a \vee b) = \max[\Pi(a), \Pi(b)]$

2. $\forall a \in \mathcal{L} \quad N(a) = 1 - \Pi(-a)$

3. $\forall a \in \mathcal{L}, \forall b \in \mathcal{L} \quad N(a \wedge b) = \min[N(a), N(b)]$

4. $\forall a, b \in \mathcal{L} \quad \text{si } a \models b \text{ alors } \Pi(b) \geq \Pi(a) \text{ et } N(b) \geq N(a)$

5. $\Pi(\perp) = 0$

Par contre $N(\perp) = 0$ n'est pas nécessairement vrai.

6. $N(\top) = 1$
Par contre $\Pi(\top) = 1$ n'est pas nécessairement vrai.
7. $\forall a \in \mathcal{L} \quad N(a) \geq N(\perp)$
8. $\forall a \in \mathcal{L} \quad \min[N(a), N(\neg a)] = N(\perp) (> 0 \text{ en général})$
9. $\forall a \in \mathcal{L} \quad \max[\Pi(a), \Pi(\neg a)] = \Pi(\top) (< 1 \text{ en général})$
10. Nous n'avons plus nécessairement $N(a) \leq \Pi(a)$, ni $N(a) > 0 \Rightarrow \Pi(a) = 1$

3.4.2.3 Conséquence logique en logique possibiliste sous normalisée

(a, α) est une conséquence logique de Φ dans la sémantique sous normalisée (notée $(a, \alpha) \models_{SN} \Phi$) si et seulement si $\mathcal{PO}_{SN}(\Phi)$ est inclus dans $\mathcal{PO}_{SN}(a, \alpha)$.

Exemple 3.5 Soit $\Phi = \{(a, 0.7), (\neg a \vee b, 0.8), (c, 0.9), (\neg c, 0.1)\}$. Ainsi,
 $\Phi \models_{SN} (b, 0.7)$ mais $\Phi \not\models_{SN} (b, 0.9)$.
 $\Phi \models_{SN} (\perp, 0.1)$ mais $\Phi \not\models_{SN} (\perp, 0.2)$.
 $\Phi \models_{SN} (\phi, 0.1)$ pour n'importe quelle formule ϕ .

3.4.2.4 La distribution de possibilité normalisée ou non, la moins spécifique parmi les distributions les moins sous-normalisées

La proposition 3.2 est généralisée à la sémantique sous normalisée :

Proposition 3.6 Pour toute distribution de possibilité normalisée ou non π_{SN} , π_{SN} satisfait Φ si et seulement si $\pi_{SN} \leq \pi_{SN\Phi}$ i.e. $\forall \omega, \pi_{SN}(\omega) \leq \pi_{SN\Phi}(\omega)$.

Proposition 3.7

$$\Phi \models (a, \alpha) \text{ si et seulement si } \pi_{SN\Phi} \models (a, \alpha)$$

En d'autre termes, $Val(a, \Phi) = N_{\Phi}(a)$, où $N_{\Phi}(a)$ est la mesure de nécessité induite par π_{Φ} .

L'expression " distribution de possibilité la moins spécifique parmi celles satisfaisant Φ " n'est pas réutilisée. Pour cause, l'inégalité $\pi \leq \pi'$ n'exprime réellement que π est plus spécifique que π' qu'à un niveau de normalisation égal (c'est-à-dire $\max_{\omega \in \Omega} \pi(\omega) = \max_{\omega \in \Omega} \pi'(\omega)$); sinon $\pi \leq \pi'$ exprime avant tout que π est plus inconsistante (ou la plus sous-normalisée) que π' . La distribution maximale π est donc en fait la distribution de possibilité normalisée ou non la moins spécifique parmi les distributions les moins sous-normalisées satisfaisant Φ .

Proposition 3.8 $Incons(\Phi) = SN(\pi_{\Phi}) = N_{\Phi}(\perp)$

Le degré d'inconsistance de Φ est égal au degré de sous normalisation de π_{Φ} . En particulier, si π_{Φ} est normalisée, alors $Incons(\Phi) = 0$, ce qui correspond à la consistance dans la sémantique normalisée (puisque'il existe au moins une distribution normalisée, à savoir π_{Φ} , satisfaisant Φ).

Proposition 3.9 *Soit Φ une base de connaissance partiellement inconsistante ($Incons(\Phi) > 0$). Alors :*

- $\forall b \in \mathcal{L}, N_{\Phi}(b) \geq Incons(\Phi) > 0.$
- $\forall b \in \mathcal{L}, \min[N_{\Phi}(b), N_{\Phi}(\neg b)] = Incons(\Phi) > 0.$

Théorème 3.1 [Dubois, Prade 87 a] *Soit Φ une conjonction de formules possibilistes. Alors $Incons(\Phi) = 0$ si et seulement si Φ^* est consistante.*

3.4.2.5 Déduction sous inconsistance partielle

$\pi_{\Phi} \subset \pi_{SN_{\Phi}}$, puisqu'une distribution de possibilité normalisée est un cas particulier d'une distribution de possibilité " normalisée ou non ". D'où $(\Sigma \models_{SN} \Phi) \Rightarrow (\Sigma \models \Phi)$ (la réciproque n'étant éventuellement pas vraie). La sémantique sous-normalisée étend donc naturellement la sémantique normalisée, en ce sens qu'elle est moins tolérante pour la déduction.

Notation 3.2 - \models est étulisé à la place de \models_{SN} , lorsqu'il n'y aura pas d'ambiguïté.
- La logique possibiliste se réfère par défaut à la sémantique sous-normalisée.

Les propositions 3.4 et 3.5 tiennent encore dans la sémantique sous-normalisée, sous la même formulation puisqu'elles n'utilisent pas la normalisation :

Proposition 3.10 *Soient Φ, Σ deux ensembles de formules possibilistes et π_{Φ} la distribution de possibilité la moins spécifique parmi celles satisfaisant Φ . Alors*

$$\Phi \models \Sigma \text{ si et seulement si } \pi_{\Phi} \text{ satisfait } \Sigma$$

Proposition 3.11 *Soit Φ un ensemble de formules de la logique possibiliste N -valuée, consistant, et b une formule quelconque. Alors $\Phi \models (b, \alpha)$ si et seulement si $\Phi_{\alpha} \models (b, \alpha)$.*

3.4.2.6 Déduction triviale vs déduction non-triviale

Quelle que soit la formule b , b est déductible de Φ avec un degré supérieur ou égal à $Incons(\Phi)$. Les déductions de (b, α) avec $\alpha = Incons(\Phi)$ ne doivent rien à la nature de b , mais sont provoquées par l'inconsistance partielle, qui généralement n'a rien à voir avec b ; elles seront appelées déductions triviales. Les déductions de (b, α) avec $\alpha > Incons(\Phi)$ seront appelées déductions non-triviales.

Le degré d'inconsistance de Φ se comporte comme un seuil au-dessous duquel rien n'est significatif (comme dans le cas d'une inconsistance classique) et au-dessus duquel toutes les déductions sont significatives (comme s'il n'y avait pas d'inconsistance). Les résultats suivants sont énoncés :

Proposition 3.12 *Soit $Incons(\Phi) = \alpha$. Alors*

$$\Phi \equiv \Phi_{\alpha} \equiv (\perp, \alpha) \wedge \Phi_{\bar{\alpha}}$$

Proposition 3.13 *Il existe (au moins) une interprétation ω^* telle que $\max\{\pi(\omega), \omega \in \Omega\} = \pi(\omega^*)$.*

Théorème 3.2 *Soit $Incons(\Phi) = \alpha$. Alors $\Phi_{\bar{\alpha}}$ est consistante.*

Corollaire 3.1 (*inconsistance partielle et α – coupes*) :

$Incon(\Phi) = \max\{\alpha, \Phi_\alpha^* \text{ inconsistent}\} = \min\{\alpha, \Phi_\alpha^* \text{ consistant}\}$ (et ces deux bornes sont atteintes).

La logique possibiliste N-valuée se ramène donc à la logique classique. En effet, une base de connaissances possibiliste N-valuée est équivalente à une famille finie de bases de connaissances classiques emboîtées. En particulier, Φ est consistante si et seulement si Φ_0^* est consistante, et Φ est totalement inconsistante si et seulement si Φ_1^* est inconsistante.

Théorème 3.3 *Toutes les déductions non-triviales qui peuvent être effectuées à partir de Φ peuvent être déduites de $\Phi_{\bar{\alpha}}$. Plus formellement :*

$$[\Phi \models (b, \beta) \text{ non trivialement}] \Rightarrow [\Phi_{\bar{\alpha}} \models (b, \beta)]$$

Proposition 3.14 *Soit Φ une conjonction de formules possibilistes N-valuées et b une formule quelconque. Alors $\exists \alpha > 0$ tel que $\Phi \models (b, \alpha)$ si et seulement si $\Phi^* \models b$.*

Lemme 3.1 $\omega \models \Phi^* \Leftrightarrow \pi(\omega) = 1$

3.5 Axiomatique de la logique possibiliste

Nous nous intéressons dans cette section à la recherche d'un système formel pour la logique possibiliste N-valuée munie de la sémantique sous-normalisée. D'abord, il est à noter que les tautologies de la logique possibiliste N-valuée étant toutes les formules de la forme (τ, α) où τ est une tautologie de la logique classique et $\alpha \in]0, 1]$. Ainsi, nous pouvons construire de façon triviale un système formel produisant toutes les tautologies de la logique possibiliste N-valuée.

Nous nous intéressons plutôt au problème suivant : existe-t-il un système formel (un ensemble d'axiomes et des règles d'inférence) tel que pour toute ensemble Φ de formules possibilistes N-valuées et pour toute formule possibiliste (b, α) conséquence logique de Φ , (b, α) peut être dérivée à partir des axiomes, de Φ , et des règles d'inférences ?

Le système formel correct et complet suivant est proposé pour la logique possibiliste N-valuée, munie de la sémantique sous normalisée :

Schémas d'axiomes :

$$(A1) \ (a \rightarrow (b \rightarrow a), 1).$$

$$(A2) \ ((a \rightarrow (b \rightarrow c)) \rightarrow ((a \rightarrow b) \rightarrow (a \rightarrow c)), 1).$$

$$(A3) \ ((\neg a \rightarrow \neg b) \rightarrow ((\neg a \rightarrow b) \rightarrow a), 1).$$

Règles d'inférences :

$$(MPG) \ (a, \alpha)(a \rightarrow b, \beta) \vdash (b, \min(\alpha, \beta)).$$

$$(S) \ (a, \alpha) \vdash (a, \beta) \text{ pour tout } \beta \leq \alpha.$$

Les axiomes sont ceux de la logique classique valuée par 1. La règle (MPG) [Dubois, Prade 85] [Froidevaux, Grossetête 90] qui est le cœur du système formel, est le modus ponens graduel. (S) est la règle de subsomption.

Théorème 3.4 *Le système formel proposé est correct et complet [Dubois et al. 94 b] relativement à la sémantique sous-normalisée de la logique possibiliste N -valuée, c'est-à-dire que pour tout ensemble de formules possibilistes Φ , nous avons :*

$$\Phi \models_{SN} (b, \alpha) \text{ si et seulement si } \Phi \vdash (b, \alpha)$$

Où $\Phi \vdash (b, \alpha)$ signifie : " il existe une dérivation de (b, α) dans le système formel constitué de Φ , des schémas d'axiomes et des règles d'inférence du système.

Corollaire 3.2 *la logique possibiliste est axiomatisable.*

3.6 Exemple illustratif

Soit à illustrer les notions présentées ci-dessus sur la base de connaissances possibiliste Φ suivante :

$$\Phi = \{(\neg p \vee q, 0.8), (\neg p \vee r, 0.9), (\neg p \vee \neg r, 0.1), (\neg q \vee r, 0.6), (p, 0.3), (q, 0.7), (\neg q, 0.2), (r, 0.8)\}.$$

Nous pouvons vérifier que $Incons(\Phi) = 0.2$. Par conséquent, la sous base $\Phi_{0.3} = \{(\neg p \vee q, 0.8), (\neg p \vee r, 0.9), (\neg q \vee r, 0.6), (p, 0.3), (q, 0.7), (r, 0.8)\}$ est à l'abri de l'inconsistance, et sa fermeture déductive est consistante c.à.d. $\nexists a, \nexists \alpha > 0, \nexists \beta > 0$ telle que $\Phi_{0.3} \vdash (a, \alpha)$ et $\Phi_{0.3} \vdash (\neg a, \beta)$. Par contre, $\Phi_{0.1} \vdash (\neg r, 0.1)$ et $\Phi_{0.1} \vdash (r, 0.8)$.

Notons aussi que, bien que $(\neg p \vee r, 0.9), (p, 0.3) \vdash (r, 0.3)$, nous avons $\Phi \vdash (r, 0.8)$ aussi. Ceci illustre le fait que la logique possibiliste s'intéresse en pratique aux preuves menant au plus grand niveau de certitude. En plus, dans le cas où Φ contient $(r, 0.2)$ plutôt que $(r, 0.8)$, alors $(r, 0.2)$ ne serait d'aucune utilité, étant donné qu'elle est subsumée par $(r, 0.3)$. En effet, nous pouvons vérifier que $\Phi \setminus \{(r, 0.8)\}$ et $(\Phi \setminus \{(r, 0.8)\}) \cup \{(r, 0.2)\}$ sont associées à la même distribution de possibilité.

ω	$\pi_{\{(\neg p \vee q, .8)\}}$	$\pi_{\{(\neg p \vee r, .9)\}}$	$\pi_{\{(\neg p \vee \neg r, .1)\}}$	$\pi_{\{(\neg q \vee r, .6)\}}$	$\pi_{\{(p, .3)\}}$	$\pi_{\{(q, .7)\}}$	$\pi_{\{(\neg q, .2)\}}$	$\pi_{\{(r, .8)\}}$	π_{Φ}
pqr	1	1	0.9	1	1	1	0.8	1	0.8
$pq\neg r$	1	0.1	1	0.4	1	1	0.8	0.2	0.1
$p\neg qr$	0.2	1	0.9	1	1	0.3	1	1	0.2
$p\neg q\neg r$	0.2	0.1	1	1	1	0.3	1	0.2	0.1
$\neg pqr$	1	1	1	1	0.7	1	0.8	1	0.7
$\neg pq\neg r$	1	1	1	0.4	0.7	1	0.8	0.2	0.2
$\neg p\neg qr$	1	1	1	1	0.7	0.3	1	1	0.3
$\neg p\neg q\neg r$	1	1	1	1	0.7	0.3	1	0.2	0.2

TABLE 3.1 – Calcul détaillé de la distribution de possibilité de l'exemple illustratif 3.6

La distribution de possibilité associée a Φ (dont le calcul est détaillé dans la table 3.1) est donnée par :

$$\pi_{\Phi}(pqr) = 0.8; \pi_{\Phi}(\neg pqr) = 0.7; \pi_{\Phi}(p\neg qr) = 0.3; \pi_{\Phi}(pq\neg r) = \pi_{\Phi}(\neg pq\neg r) = \pi_{\Phi}(\neg p\neg q\neg r) = 0.2; \pi_{\Phi}(p\neg q\neg r) = \pi_{\Phi}(p\neg q\neg r) = 0.1.$$

Ainsi, $Cons(\Phi) = \max_{\omega \in \Omega} \pi_{\Phi}(\omega) = 0.8$ et $Incons(\Phi) = 1 - 0.8 = 0.2$.

D'une manière similaire, $Incons(\Phi \setminus \{(\neg q, 0.2)\}) = 0.1$ et $Incons(\Phi \setminus \{(\neg q, 0.2), (\neg p \vee \neg r, 0.1)\}) = 0$.

3.7 La résolution possibiliste

3.7.1 Motivations et définitions

La mise en œuvre de la méthode de démonstration automatique passe par la transformation de chaque formule initiale en une conjonction de clauses (c'est-à-dire la forme clausale) dont l'inconsistance lui est équivalente. De ce fait, dans cette section la forme clausale possibiliste est étudiée.

Définition 3.7 Une clause possibiliste est un couple (a, α) , où a est une clause de la logique classique et $\alpha \in]0, 1]$.

Définition 3.8 Une formule clausale possibiliste est une conjonction finie de clauses possibilistes.

3.8 Mise sous forme clausale dans la logique possibiliste

Nous nous intéressons maintenant à la mise sous forme clausale d'une formule possibiliste quelconque [Lang 00] plus précisément à la déduction automatique par réfutation. Il suffit de chercher pour toute formule possibiliste une formule clausale ayant le même degré d'inconsistance. L'algorithme suivant transforme toute formule possibiliste en une formule sous forme clausale ayant le même degré d'inconsistance :

Début

Φ conjonction de formules possibiliste : $\Phi = \wedge\{(a_i, \alpha_i), i = 1..n\}$

Pour $i = 1..n$ **faire**

mettre a_i sous forme clausale : $a_i = \wedge\{c_{ij}, j = 1..m_i\}$

Fin Pour

$\mathcal{C}(\Phi) = \wedge\{\wedge\{c_{ij}, j = 1..m_i\}, i = 1..n\}$ est une forme clausale de Φ .

Fin

Algorithme 3: Algorithme de mise sous forme clausale

Théorème 3.5 Soit Φ un ensemble de formules possibilistes et $\mathcal{C}(\Phi)$ sa forme clausale déterminée par l'algorithme précédent. Alors Φ et $\mathcal{C}(\Phi)$ ont le même degré d'inconsistance.

Lemme 3.2 – $\mathcal{C}(\Phi)_\alpha$ est une forme clausale de Φ_α

- $\mathcal{C}(\Phi)_{\bar{\alpha}}$ est une forme clausale de $\Phi_{\bar{\alpha}}$
- $\mathcal{C}(\Phi)^*$ est une forme clausale de Φ^* .

3.8.1 Les règles de résolution

Soient (c, A) et (c', B) deux clauses possibilistes, et soit c'' une résolvente quelconque de c et c' . La règle suivante [Dubois, Prade 87 a] [Dubois, Prade 90] est appelée la règle

de résolution possibiliste :

$$(c, \alpha), (c', \beta) \vdash (c'', \min(\alpha, \beta)).$$

La règle suivante est appelée règle de subsomption :

$$\text{Si } \beta \leq \alpha \text{ alors } (c, \alpha) \vdash (c, \beta).$$

3.8.2 Correction des règles de la logique possibiliste

Les résultats suivants justifient la règle de résolution possibiliste, en montrant qu'elle est bien en accord avec la logique possibiliste :

Théorème 3.6 (*correction de la règle : résolution possibiliste*) : soient c et c' deux clauses possibilistes et c'' une clause obtenue par l'application de la (résolution possibiliste) à partir de c et c' . Alors nous avons : $c, c' \models c''$.

Théorème 3.7 (*correction de la règle de subsomption*) : soit $C = (c, \alpha)$ une clause possibiliste et $C' = (c, \beta)$ avec $\beta \leq \alpha$ une clause obtenue à partir de (c, α) par la règle de subsomption. Alors $C \models C'$.

Notation 3.3 Soit Φ un ensemble de clauses possibilistes et C une clause possibiliste ; nous noterons : $\Phi \vdash_{Res} C$ si et seulement si C peut être obtenue par un nombre fini d'application des règles (résolution possibiliste) et (subsomption) à partir des clauses de Φ et de leurs résolvantes.

Corollaire 3.3 (*correction de l'application en chaîne de la résolution*) : $[\Phi \vdash_{Res} C] \Rightarrow [\Phi \models C]$.

Remarque 3.6

La résolution possibiliste recouvre la résolution classique. En effet, si $\alpha = \beta = 1$ alors l'inférence de la clause certaine c'' est calculée à partir des clauses certaines c et c' .

3.8.3 Réfutation par résolution

L'introduction de la règle de résolution possibiliste permet d'effectuer des inférences qui sont correctes par rapport aux sémantiques décrites par la section 3.4.

Il faudrait avoir une méthode de démonstration automatique qui soit complète pour cette sémantique, garantissant ainsi que toute conséquence logique d'un ensemble de formules possibilistes pourra être inférée par la méthode.

Ceci n'est pas le cas avec la résolution possibiliste si nous l'utilisons pour la déduction (la résolution classique n'étant elle-même pas complète pour la déduction). C'est pour cela que la réfutation par résolution est généralisée.

En logique classique, démontrer par réfutation que b est une conséquence logique de a revient, par définition à montrer que $a \wedge \neg b$ est inconsistante. En effet, une réfutation par résolution est une déduction de la clause vide \perp par résolution.

3.8.3.1 La réfutation par résolution dans la logique possibiliste

Dans cette section, la preuve par réfutation est généralisée à la logique possibiliste.

Le principe de la réfutation est d'ajouter à l'ensemble de formules possibilistes Φ la négation de la formule b à prouver avec la valuation 1. Ainsi, b est supposée certainement fausse.

Le degré d'inconsistance de l'ensemble de formules possibilistes Φ' résultant sera la valuation maximale avec laquelle b peut être déduit de Φ . Ce qui s'écrit plus formellement comme suit :

Théorème 3.8 (*Réfutation*) :

$$\Phi \models (b, \beta) \Leftrightarrow \text{Incons}(\Phi \wedge (\neg b, 1)) \geq \beta$$

Ce qui peut être écrit de manière équivalente $\text{Incons}(\Phi \wedge (\neg b, 1)) = \max\{\beta, \Phi \models (b, \beta)\}$.

La méthode de réfutation par résolution en logique possibiliste est décrite par l'algorithme suivant [Dubois, Prade 87 a] :

Début

Soit Φ une base de connaissances possibiliste et b une formule.

Mettre Φ sous forme clausale (Soit \mathcal{C} la forme clausale obtenue) ainsi que $\neg b$ (Soit c_1, \dots, c_p les clauses obtenues) ;

$$\mathcal{C}' \leftarrow \mathcal{C} \cup \{(c_1, 1), \dots, (c_p, 1)\};$$

Chercher une déduction de (\perp, w^*) par résolution à partir de \mathcal{C}' , avec w^* maximal ;

Si ($w^* = \alpha$) **Alors**

$$\left| \begin{array}{l} \Phi \models (b, \alpha) \\ \{C' \text{ est-à-dire } \text{Val}(b, \Phi) = w^*\} \end{array} \right.$$

Fin Si

Fin

Algorithme 4: Algorithme de réfutation par résolution

Chaque déduction de (\perp, w) sera appelée w -réfutation ; toute réfutation avec une valuation maximale sera appelée réfutation optimale.

3.8.3.2 Correction et complétude de la réfutation par résolution

Le théorème suivant exprime la correction de la réfutation par résolution dans la logique possibiliste.

Théorème 3.9 *Pour toute base de connaissances possibiliste Φ , pour toute formule b , pour toute forme clausale \mathcal{C}' de $\Phi \wedge (\neg b, 1)$ et $\alpha \in]0, 1]$, nous avons :*

$$[\mathcal{C}' \vdash_{Res} (\perp, \alpha)] \Rightarrow [\Phi \models (b, \alpha)]$$

Les théorèmes suivants expriment la complétude de la résolution pour la réfutation dans la logique clausale possibiliste et la logique possibiliste :

Théorème 3.10 *Pour tout ensemble de clauses possibilistes \mathcal{C}' , si $\text{Incons}(\mathcal{C}') = \alpha$ avec $\alpha > 0$ alors $\mathcal{C}' \vdash_{Res} (\perp, \alpha)$.*

Théorème 3.11 *Pour toute base de connaissances possibiliste Φ , pour toute formule b , pour toute forme clausale \mathcal{C}' de $\Phi \wedge (\neg b, 1)$ et $\forall \alpha > 0$ nous avons :*

$$[\Phi \models (b, \alpha)] \Rightarrow [\mathcal{C}' \vdash_{Res} (\perp, \alpha)]$$

Les résultats suivants seront utiles pour la section suivante :

Proposition 3.15 *Toute réfutation significative a pour valuation la plus petite des valuations des clauses, utilisées dans la réfutation.*

Proposition 3.16 *Soit \mathcal{C} un ensemble inconsistant de clauses possibilistes. Alors $Incons(\mathcal{C})$ est égal à l'une des valuations des clauses de \mathcal{C} .*

3.8.4 Stratégie de résolution

Nous remarquons que la procédure de réfutation par résolution contient l'étape de recherche d'une réfutation optimale, qui consiste à chercher une déduction de (\perp, α) par résolution possibiliste, avec α maximal.

Nous nous intéressons dans cette section à la recherche d'algorithmes permettant de trouver le plus rapidement possible une réfutation optimale.

Comme en logique classique [Farreny, Ghallab 87], une stratégie de résolution restreint le choix des clauses ou littéraux sur laquelle sera tentée la résolution, selon des critères bien déterminés.

Définition 3.9 *Une stratégie de résolution possibiliste est un **algorithme** destiné à trouver une réfutation optimale d'un ensemble de clauses de la logique clausale possibiliste.*

Définition 3.10 *Une stratégie de résolution \mathcal{S} pour la logique clausale possibiliste est **complète** si et seulement si pour tout ensemble inconsistant \mathcal{C} de clauses de la logique clausale possibiliste, il existe une réfutation optimale par résolution à partir de \mathcal{C} compatible avec les critères de \mathcal{S} .*

Définition 3.11 *Une stratégie de résolution \mathcal{S} pour la logique clausale possibiliste est **directement complète** si et seulement si pour tout ensemble inconsistant \mathcal{C} de clauses de la logique clausale possibiliste, l'application de \mathcal{S} produit une réfutation optimale de \mathcal{C} en un temps fini.*

Définition 3.12 *Une stratégie de résolution \mathcal{S} pour la logique clausale possibiliste est dite **réellement directement complète** si et seulement si son application à \mathcal{C} inconsistante produit une réfutation optimale de \mathcal{C} en la reconnaissant comme optimale en un temps fini.*

Définition 3.13 *Une stratégie de résolution \mathcal{S} pour la logique clausale possibiliste est **décidable** si et seulement si elle est réellement directement complète et si de plus, son application à un ensemble quelconque de clauses de la logique possibiliste s'arrête en un temps fini.*

Définition 3.14 *Une stratégie de résolution possibiliste sera dite **informée** si le choix des clauses pour l'étape de résolution à venir est guidé par les valuations. Sinon, la stratégie sera dite **aveugle**.*

3.8.5 Stratégie par α -coupes dichotomique

La valeur de $Val(a, \Phi)$ peut être calculée par dichotomie, où Φ est une base de connaissances possibiliste et a est une formule propositionnelle à déduire et ceci en utilisant n'importe quel prouveur pour le problème de satisfiabilité propositionnelle *SAT*.

Soit $\alpha_0 = 0$ et soient $\alpha_1, \dots, \alpha_n$ les valuations distinctes qui apparaissent dans Φ , ordonnées d'une manière croissante : $0 < \alpha_1, \dots, \alpha_n < 1$. La procédure de calcul de $Val(a, \Phi)$ peut être résumée comme suit :

```

Début

 $l := 0$ ;
 $u := n$ ;
Tant que ( $l < u$ ) faire
    |  $r := \lfloor (l + u)/2 \rfloor$ 
    | Si ( $(\Phi_{\alpha_r}^* \wedge \neg a)$  est consistante) Alors
    | |  $u := r - 1$ ;
    | Sinon
    | |  $l := r$ ;
    | Fin Si
Fait
 $Val(a, \Phi) := \alpha_r$ .

Fin
    
```

Algorithme 5: Calcul de $Val(a, \Phi)$

Cet algorithme contient exactement $\lceil \log_2 n \rceil$ appels à *SAT* pour le test de satisfiabilité, où n représente le nombre des différents strates qui existent dans Φ [Lang 00].

Le problème de déduction en logique possibiliste standard revient donc à calculer un degré d'incohérence. L'algorithme proposé par Lang utilise la stratification des bases de connaissances. Les formules des strates les moins prioritaires étant les moins certaines sont rejetées plus facilement en cas d'incohérence.

3.8.6 Stratégies linéaires de type *pseudo* – A^* en logique possibiliste

Dans cette section, nous nous intéressons à une classe de stratégies informées, qui est la stratégie linéaire de type *pseudo* – A^* .

3.8.6.1 Réfutations par résolution possibilistes linéaires

Les stratégies de résolution possibilistes linéaires sont définies exactement de la même façon qu'en logique classique : soit \mathcal{C} un ensemble inconsistant de clauses de la logique possibiliste, dont une clause C_0 (généralement choisie telle que $\mathcal{C} - C_0$ soit consistant, afin que les stratégies linéaires soient complètes) est appelée clause centrale initiale. Les stratégies linéaires n'autorisent que les résolutions entre une clause CC descendante de C_0 (appelée clause centrale) et une clause qui est soit parmi les clauses de \mathcal{C} (dites clauses d'entrée) soit parmi les résolvantes ancêtres de CC .

Définition 3.15 Une *réfutation linéaire* est une réfutation par résolution obtenue avec une stratégie linéaire.

Proposition 3.17 Soit \mathcal{C} un ensemble inconsistant de clauses possibiliste. Parmi les réfutations optimales de \mathcal{C} il en existe au moins une qui soit linéaire.

Or, cela ne veut pas dire que si une clause quelconque C_0 de \mathcal{C} est prise pour clause centrale initiale, alors une réfutation linéaire optimale utilisant C_0 pour clause centrale initiale sera une réfutation optimale de \mathcal{C} . Pour cela, il faut s'assurer d'abord que $\mathcal{C} - C_0$ est consistante (c'est-à-dire $Incons(\mathcal{C} - \{C_0\}) = 0$).

En effet, l'hypothèse de la consistance de $\mathcal{C} - C_0$ est vérifiée dans le cas où partant d'une base de connaissance Φ de la logique causale possibiliste, consistante, nous cherchons une borne inférieure de $N(b)$ pour une formule donnée b .

Soit \mathcal{C} l'ensemble des clauses équivalent à $\Phi \wedge \{(-b, 1)\}$. Si la négation de b ne donne qu'une seule clause (c'est-à-dire si b est une formule atomique ou une conjonction de formules atomiques) alors cette clause est prise comme clause centrale initiale et l'ensemble de clauses restant est équivalent à Φ , donc consistant. Sinon, nous pouvons prendre pour clause centrale initiale successivement chacune des clauses issue de $-b$ (comme en logique classique [Farreny, Ghallab 87]) et retenir le maximum des valuations des réfutations optimales obtenues.

Le degré d'inconsistance de \mathcal{C} peut donc être déterminé en utilisant une stratégie linéaire.

3.8.6.2 Stratégies linéaires informées

La recherche d'une réfutation par résolution par une stratégie linéaire peut s'exprimer en terme de recherche arborescente dans un espace d'états [Dubois et al. 87].

Un état est défini par une clause centrale et l'ensemble des clauses centrales ancêtres de celle-ci.

A chaque état de l'arbre de recherche, est associée la valuation de la dernière clause centrale produite, c.-à-d. qu'à $(C_0C_1\dots C_i)$, est associée la valuation $Val(C_i)$.

L'objectif est de chercher les états associés à une clause vide et avec une valuation optimale.

Nous remarquons alors une analogie avec la recherche dans les espaces d'états où :

- L'état initial S_0 est défini par la clause centrale initiale C_0 et a pour coût $Val(S_0)$;
- Le coût associé à l'arc $(C_0C_1\dots C_i) \rightarrow (C_0C_1, \dots, C_iC_{i+1})$ est la valuation associée à C_{i+1} ;
- Le coût global du chemin $C_0 \rightarrow C_1 \rightarrow \dots \rightarrow C_i$ est l'intersection des coût de ses arcs élémentaires ;
- Les états objectifs sont des états $(C_0C_1\dots C_i)$ tels que $C_i = (\perp, \alpha_i)$ avec $\alpha_i > 0$;
- L'état $(C_0C_1\dots C_n)$ est développé en produisant toutes les résolvantes de C_n autorisées par la stratégie linéaire.

La recherche d'une réfutation optimale est alors équivalente à la recherche d'un chemin de coût maximal de l'état initial vers des états objectifs. Cependant, plusieurs différences existent :

- Le coût ici est à maximiser et non pas à minimiser.
- Les coûts ne sont pas additifs, mais ce sont des valuations combinées par l'opérateur minimum.

Comme pour la recherche heuristique dans les espaces d'états [Farreny, Ghallab 87], la recherche ordonnée est guidée par une fonction d'évaluation f qui est calculée ainsi : pour chaque état S de l'arbre de recherche, $f(S) = \min[g(S), h(S)]$, où $g(S)$ est le coût du chemin de l'état initial vers S , et $h(S)$ une estimation du coût de S vers un état objectif optimal. L'algorithme suivant a été proposé dans [Yager 86] :

```

Début

Open ← {S0};
Closed ← {∅};
Tant que (Open ≠ ∅) faire
    Choisir un état Sn dans Open maximisant f
    Si (Sn est un état objectif) Alors
        | envoyer succès.
    Sinon
        Développer le nœud Sn en créant l'ensemble E'n des états produits.
        En ← E'n \ Closed
        Open ← (Open - {Sn}) ∪ En
        Closed ← Closed ∪ {Sn}
        Calculer f pour chaque nouvel état de Open
    Fin Si
Fait
Si (Open = ∅) Alors
    | échec
Fin Si
Fin

```

Algorithme 6: Algorithme de réfutation par résolution possibiliste linéaire

Théorème 3.12 *Soit $h^*(S)$ le véritable coût du chemin optimal de l'état S vers un état but. Le système de production possibiliste ne contient qu'un nombre fini d'états (c'est-à-dire que le nombre de clauses dérivables à partir des clauses initiales est fini, ce qui est le cas avec des clauses toutes propositionnelles). Alors l'exécution de l'algorithme ci-dessus utilisant comme fonction d'évaluation $f(S) = \min[g(S), h(S)]$ telle que la fonction heuristique h soit optimiste, c'est-à-dire $\forall S \ h(S) \geq h^*(S)$, se termine dans tous les cas et en donnant une réfutation optimale s'il en existe une (c'est-à-dire qu'il constitue une stratégie décidable).*

Si la condition $\forall S \ h(S) \geq h^*(S)$ est vérifiée, la fonction heuristique h est dite **admissible**.

3.8.6.3 Une famille de fonctions d'évaluation admissibles

Yager [Yager 86] donne de plus le résultat suivant : si h_1 et h_2 sont deux fonctions heuristiques, toutes les deux admissibles, telles que $\forall S \ h_2(S) \leq h_1(S) \leq (h^*(S))$, alors h_1 est plus informée que h_2 . Cela veut dire que toute clause produite par l'application de l'algorithme utilisant h_1 sera aussi produite par celui utilisant h_2 .

La fonction constante $h(S) = 1$ est donc la moins informée des fonctions heuristiques. Une famille de fonctions heuristiques admissibles est proposée dans la suite de

cette section.

Soit \mathcal{C} un ensemble de clauses possibilistes, $R(\mathcal{C})$ désigne l'ensemble des clauses productibles en une application du principe de résolution à partir de \mathcal{C} , et $R^\infty(\mathcal{C}) = \cup_i R^i(\mathcal{C})$ désigne l'ensemble de toute clause productible en un nombre fini d'étapes à partir de \mathcal{C} .

Par convention, pour toute clause classique $c = [\neg]L_1 \vee [\neg]L_2 \vee \dots \vee [\neg]L_n$, $c = l_1 \vee l_2 \vee \dots \vee l_n$, ou $l_i = [\neg]L_i$; nous écrirons aussi, $\neg l = [L_i \text{ si } l = \neg L_i]$, $\neg l = [\neg L_i \text{ si } l = L_i]$.

Soit $H_1(l) = \max\{\alpha, (c, \alpha) \in R^\infty(\mathcal{C}), \neg l \in c\}$. Si nous voulons obtenir la clause vide à partir de l'ensemble initial de clauses \mathcal{C} , alors pour toute réfutation utilisant la clause c , pour tout littéral l de c , il faudra, pour dériver \perp , utiliser à un moment ou un autre une clause c' contenant un littéral $\neg l$. D'où : pour tout littéral l' de c , une réfutation utilisant c aura une valuation inférieure ou égale à $H_1(l')$.

Ainsi, en notant (par abus de notation) $h(C)$ pour $h(S_C)$, où S_C est l'état dont la dernière clause est C , nous avons :

$$\forall C = (c, \alpha), \forall l' \in c, h^*(C) \leq H_1(l')$$

Telle qu'elle est définie la fonction H_1 n'est pas directement calculable à partir de la donnée de \mathcal{C} . Or, H_1 est en fait une fonction statique étant donné que :

Proposition 3.18

$$H_1(l) = \max\{\alpha, (c, \alpha) \in \mathcal{C}, \neg l \in c\}$$

Par conséquent, H_1 est une fonction statique qui est calculée une fois pour toutes avant toute production de résolvantes.

Posons maintenant pour toute clause $C = (c, \alpha)$:

$$h_1(C) = \min\{H_1(l), l \in c\} (= 1 \text{ si } c = \perp)$$

Soit $C = (c, \alpha)$, de $\forall l \in c, H_1(l) \geq h^*(C)$. $h_1(C) \geq h^*(C)$, c'est-à-dire que toute réfutation utilisant C a une valuation inférieure ou égale à $h_1(C)$. La fonction h_1 est par conséquent admissible.

Une suite de fonctions d'évaluation $(f_p)_{p \geq 0}$ est définie alors comme suit :

$$f_p : R^\infty(\mathcal{C}) \rightarrow [0, 1]$$

$$h_0(C) = 1$$

$$f_p(C) = \min[Val(C), h_p(C)]$$

$$h_{p+1}(C) = \min_{l \in c} \max_{C' \in \mathcal{C}, \neg l \in cl(C')} f_p(C')$$

Proposition 3.19 Les fonctions h_p vérifient : $\forall p \geq 0, \forall C \in R^\infty(\mathcal{C})$:

$$h^*(C) \leq h_p(C)$$

$$h_{p+1}(C) \leq h_p(C)$$

3.9 Exemple illustratif

Soit la base possibiliste Φ composée des clauses suivantes :

- $C_1 : (\neg p \vee q, 0, 6)$
- $C_2 : (\neg p \vee r, 0, 9)$
- $C_3 : (\neg q \vee r, 0, 6)$
- $C_4 : (p, 0, 3)$
- $C_5 : (q, 0, 7)$

Il s'agit de trouver la valuation optimale de la formule r . Soit alors Φ' l'ensemble des clauses équivalentes à $\Phi' = \Phi \cup \{(\neg r, 1)\}$.

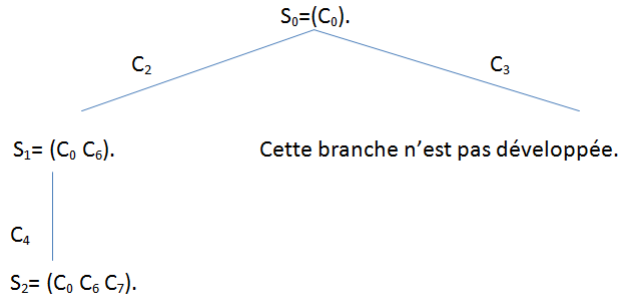


FIGURE 3.1 – Arbre de recherche de l'exemple illustratif 3.9

Soit $C_0 = (\neg r, 1)$ car $\Phi' - \{C_0\}$ est cohérente.

Les deux clauses C_2 et C_3 contiennent le littéral r . Il faudrait alors calculer leurs fonctions d'évaluation.

La plus grande valuation qui maximise la fonction d'évaluation est 0.9, car $\max[\min[Val(C_0), Val(C_2)], \min[Val(C_0), Val(C_3)]] = 0.9$. Effectivement vu cet ordre total, la branche correspondant à la clause C_3 ne sera pas explorée.

L'état suivant est donc : $S_1 = (C_0 C_6)$ et a comme coût : $\min[Val(C_0), Val(C_2)] = Val(C_2) = Val(C_6) = 0.9$.

Il existe un seul chemin possible ensuite, avec la clause C_4 donc $S_2 = (C_0 C_6 C_7)$ et a comme coût : $\min[Val(C_6), Val(C_4)] = Val(C_7) = Val(C_4) = 0.3$. La clause C_7 est une contradiction. L'état objectif est atteint et est optimal. Ainsi, $\Phi \models (r, 0.3)$.

3.10 Conclusion

Ce chapitre a été consacré à présenter la logique possibiliste qui permet une représentation naturelle de la notion d'incertitude. Cette dernière est évaluée par deux mesures duales, à savoir : le degré de nécessité et le degré de possibilités. Ensuite, nous nous sommes intéressés aux mécanismes d'inférences en logique possibiliste. De ce fait, l'extension de la réfutation par résolution aux clauses pondérées par la mesure de nécessité est présentée.

En intelligence artificielle, la logique a été considérée, depuis longtemps, à la base de la représentation et du raisonnement sur les connaissances. Plus particulièrement, nous nous intéressons dans ce travail à la représentation des connaissances des ensembles

d'agents. Le chapitre suivant présente notre première contribution qui consiste au développement d'une extension multi-agents de la logique possibiliste.

Deuxième partie
Contributions

Introduction de la partie II

La deuxième partie de la thèse sera consacrée à présenter les principales investigations développées dans cette étude. Le premier chapitre de cette deuxième partie (Chapitre 4) présente une nouvelle logique multi-agents. Une logique où les propositions sont attachées à des sous ensemble d'agents. Le chapitre 5 porte sur une extension de notre nouvelle logique qui est la logique possibiliste multi-agents, une logique où les propositions sont attachées à des sous-ensembles flous d'agents. Dans les chapitres 6 et 7, nous montrons comment déduire automatiquement des connaissances à partir des bases de connaissances multi-agents et des bases de connaissances possibilistes multi-agents. Quant au chapitre 8, il présente des expérimentations relatives à l'implémentation des algorithmes d'inférence. Enfin, le chapitre 9 indique quelques usages possibles des formalismes proposés au cours de ce travail.

Chapitre 4

La logique multi-agents

4.1 Introduction

Ce chapitre présente une nouvelle approche pour représenter des connaissances détenue par des ensembles d'agents en s'inspirant de la théorie des possibilités [Dubois, Prade 07]. La logique modale est souvent utilisée pour représenter de telles informations. Cependant, la logique multi-agents [Belhadi et al. 13] est plus proche de la logique classique et elle possède une sémantique en terme d'ensembles d'interprétations étiquetées.

Ce chapitre est organisé comme suit : dans la section suivante nous proposons un langage formel pour la logique multi-agents ainsi que quelques concepts fondamentaux. La section 3 présente le treillis des sous-ensembles. Puis nous définissons dans la section 4 deux sémantiques pour la logique multi-agents fondées sur la notion de distribution de possibilités de la logique possibiliste. Nous présentons deux sémantiques dont l'une tenant compte des inconsistances individuelles entre groupes d'agents. La section 5 présente l'ensemble des interprétation d'un agent. Quant à la section 6, elle présente le système formel de la logique multi-agents où le système formel de la logique possibiliste est étendu pour prendre en compte des sous-ensembles d'agents. Nous terminerons par un exemple où nous illustrons les différentes notions proposées dans ce chapitre.

4.2 Langage

Soit \mathcal{L} un langage de la logique propositionnelle. Soit All un ensemble fini contenant tous les agents. Un sous ensemble d'agents est noté par les lettres majuscules A, B, C ou par des lettres indexées A_i pour $i = 1, \dots, m$. Une formule en logique multi-agents peut être construite à partir d'une formule classique en y rajoutant explicitement un ensemble d'agents qui indique le sous ensemble d'agents maximal qui croit en cette formule.

Définition 4.1 *Une formule multi-agents est un couple (a, A) où a est une formule bien formée de la logique propositionnelle et A un sous ensemble non vide d'agents dans All c.à.d. $A \subseteq All$.*

(a, A) exprime qu'au moins tous les agents dans A croient a . En effet, la formule multi-agents (a, A) est la traduction syntaxique de la contrainte sémantique $\mathbf{N}(a) \supseteq A$ où \mathbf{N} est la mesure de nécessité multi-agents.

Il est à noter que les formules de la forme (a, \emptyset) ne sont pas considérées étant donné que $\forall a, \mathbf{N}(a) \supseteq \emptyset$.

Une base de connaissances multi-agents (nous dirons aussi un ensemble de formules multi-agents) est une conjonction finie de formules multi-agents. Généralement un ensemble de formules multi-agents sera noté par Φ :

$$\Phi = (a_1, A_1) \wedge \dots \wedge (a_n, A_n) = \{(a_1, A_1), \dots, (a_n, A_n)\}$$

Exemple 4.1 *Supposons qu'un hôpital dispose de :*

- 03 médecins dermatologues : Hayet, Omar et Reda ;
- 02 médecins pédiatres : Amine et Aida ;
- 02 médecins gastrologues : Youcef et Asma.

Les croyances des différents médecins spécialistes sur les aliments sont les suivantes :

- *Les dermatologues pensent que manger de la viande grasse est mauvais pour la peau contrairement au poisson qui lui est bénéfique.*
- *Les dermatologues croient qu'un patient ne peut pas manger du poisson et des produits laitiers aux même temps.*
- *Les pédiatres croient que les boissons gazeuses ne sont pas adaptées à l'alimentation des enfants.*
- *Les pédiatres croient que les enfants peuvent manger : de la viande grasse, les légumes secs, les viennoiseries ainsi que les produits laitiers.*
- *Les médecins gastrologues croient qu'un patient ne peut pas manger des légumes secs et les boissons gazeuse au même temps.*
- *Les médecins gastrologues croient que manger du poisson est bien pour le patient.*

Ce problème peut être modélisé par la formulation suivante :

Soient A, B et C les trois groupes des spécialistes. Soit les six variables binaires, relatives à :

- *La viande grasse (a) ;*
- *Les légumes secs (b) ;*
- *Les viennoiseries (c) ;*
- *Les boissons gazeuses (d) ;*
- *Le poisson (e) ;*
- *Les produits laitiers (f).*

La base de connaissance multi-agents associée à ce problème est donc (Voir la Figure 4.1) :

$\Phi = \{(\neg a \vee e, A), (e \vee \neg f, A), (\neg e \vee f, A), (\neg d, B), (a \vee b \vee c \vee f, B), (\neg b \vee d, C), (b \vee \neg d, C), (e, C)\}$.

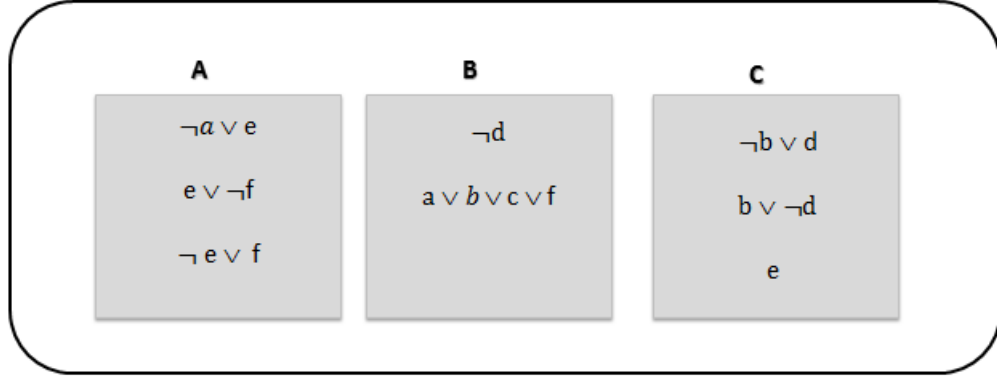


FIGURE 4.1 – Les bases de connaissances associées à chaque groupe des médecins spécialistes

Pour toute formule multi-agents (a, A) , la fonction *Ens* est définie par : $Ens((a, A)) = A$.

La projection classique d'un ensemble de formules multi-agents est l'ensemble des formules classiques, noté Φ^* , obtenu à partir de Φ en ignorant les sous-ensembles, c'est-à-dire que si $\Phi = \{(a_i, A_i), i = 1, \dots, n\}$ alors $\Phi^* = \{a_i, i = 1, \dots, n\}$.

Exemple 4.2 Si $\Phi = \{(\neg a \vee b, A), (a \vee b \vee c, B), (\neg b \vee d, B), (\neg d, C)\}$ alors $\Phi^* = \{(\neg a \vee b), (a \vee b \vee c), (\neg b \vee d), (\neg d)\}$.

Le langage ainsi défini est restrictif par rapport à deux niveaux : au premier niveau, seules les conjonctions de formules multi-agents sont considérées, ni les disjonctions, ni les négations, ni les quantifications existentielles et ni les quantifications universelles de formules multi-agents. Par exemple $(a, A) \vee (b, B)$, ou $\exists x(a, A)$ n'appartiennent pas à notre langage. Au second niveau, le langage ne permet pas l'encapsulation des sous-ensembles telle que $((a, A), B)$.

4.3 Le treillis des sous-ensembles

Dans ce qui suit, nous allons définir un ordre partiel sur le sous-ensemble A . De ce fait, nous allons utiliser un treillis Boolean :

$$\mathcal{T} = 2^{All}$$

où All est l'ensemble de tous les agents. Ce treillis est équipé de l'intersection ensembliste \cap , de l'union ensembliste \cup et de la complémentation $\bar{}$. A la différence de la logique possibiliste, seul un ordre partiel existe entre les ensembles d'agents associés aux formules.

Etant donné que \mathcal{T} n'est pas totalement ordonné comme \mathcal{V} alors nous ne pouvons pas stratifier Φ en couches. Cependant, nous pouvons définir une restriction de la base de connaissances multi-agents Φ en un ensemble d'agents $A \subseteq All$ comme suit :

$$\Phi_A = \{(a_i, A_i \cap A) \mid A_i \cap A \neq \emptyset \text{ et } (a_i, A_i) \in \Phi\}$$

4.4 Sémantique possibiliste de la logique multi-agents

4.4.1 Les distributions de possibilité multi-agents multi-agents-normalisées comme sémantique de la logique multi-agents

4.4.1.1 Définitions de base

Soit \mathcal{L} le langage classique associé à l'ensemble des formules classique Φ^* et soit Ω l'ensemble des interprétations classiques de \mathcal{L} . La sémantique d'un ensemble classique de formules Φ^* est définie en terme de l'ensemble des interprétations de Φ^* qui satisfait toute formule de Φ^* . De telles interprétations sont alors appelées un modèle.

En logique multi-agents, une distribution de possibilités multi-agents sur Ω représente l'ensemble des modèles de Φ .

Une distribution de possibilité multi-agents, notée π , est une fonction de Ω vers 2^{All} . $\pi(\omega)$ représente les sous ensemble d'agents dans All qui trouvent ω possible.

Par convention, $\pi(\omega) = All$ signifie que tous les agents trouvent que ω est possible, alors que $\pi(\omega) = \emptyset$ signifie que tous les agents trouvent que ω est impossible. Dans le cas où il existe un élément ω de Ω tel que $\pi(\omega) = All$ alors la distribution de possibilité multi-agents π est dite multi-agents-normalisée. La propriété de la multi-agents normalisation reflète l'existence d'au moins un élément de Ω tel que tous les agents le croient possible. En d'autres termes, elle reflète la consistance collective.

Exemple 4.3 *Considérons le problème des opinions sur la détection du cancer des poumons. Les paramètres à considérer dans ce problème sont :*

- La présence d'une tumeur aux poumons ;
- Le sexe du patient ;
- L'exposition aux produits toxiques ;
- L'âge du patient.

Nous disposons de trois groupes d'agents, à savoir :

- Le groupe des jeunes qui se compose de trois jeunes : Riad, Adel et Salim ;
- Le groupe des femmes qui se compose de : Nadine et Faiza ;
- Le groupe des médecins qui se compose d'un seul médecin : Mohamed.

Nous pouvons modéliser ce problème comme suit, soient :

- A : le groupe des jeunes ;
- B : le groupe des femmes ;
- C : le groupe des médecins ;
- a : variable binaire relative à la présence d'une tumeur aux poumons ;
- b : variable binaire relative au sexe du patient ;
- c : variable binaire relative à l'exposition aux produits toxiques ;

– d : variable binaire relative à l'âge du patient.

Etant donnée que nous disposons de quatre variables propositionnelles, nous avons 2^4 interprétations possibles. La Table 4.2 donne les distributions de possibilités multi-agents pour chaque interprétation ω .

A titre d'exemple, l'interprétation ω_1 correspond à l'état du monde suivant : le patient a une tumeur aux poumons et est un homme et s'est exposé aux produits toxiques et c'est un jeune. L'ensemble d'agents qui croient que le patient risque d'attraper un cancer des poumons est l'ensemble des femmes et des médecins car les jeunes pensent que seulement les vieux qui risquent d'attraper un cancer des poumons.

ω_i	ω	$\pi(\omega)$
ω_1	$a \wedge b \wedge c \wedge d$	$B \cup C$
ω_2	$a \wedge b \wedge c \wedge \neg d$	All
ω_3	$a \wedge b \wedge \neg c \wedge d$	B
ω_4	$a \wedge b \wedge \neg c \wedge \neg d$	$A \cup B$
ω_5	$a \wedge \neg b \wedge c \wedge d$	C
ω_6	$a \wedge \neg b \wedge c \wedge \neg d$	$A \cup C$
ω_7	$a \wedge \neg b \wedge \neg c \wedge d$	\emptyset
ω_8	$a \wedge \neg b \wedge \neg c \wedge \neg d$	A
ω_9	$\neg a \wedge b \wedge c \wedge d$	B
ω_{10}	$\neg a \wedge b \wedge c \wedge \neg d$	$A \cup B$
ω_{11}	$\neg a \wedge b \wedge \neg c \wedge d$	B
ω_{12}	$\neg a \wedge b \wedge \neg c \wedge \neg d$	$A \cup B$
ω_{13}	$\neg a \wedge \neg b \wedge c \wedge d$	\emptyset
ω_{14}	$\neg a \wedge \neg b \wedge c \wedge \neg d$	A
ω_{15}	$\neg a \wedge \neg b \wedge \neg c \wedge d$	\emptyset
ω_{16}	$\neg a \wedge \neg b \wedge \neg c \wedge \neg d$	A

FIGURE 4.2 – Distribution de possibilités multi-agents associées à l'exemple 4.3

La distribution de possibilité multi-agents π donne naissance à la notion de la mesure de possibilité multi-agents notée $\mathbf{\Pi}$. C'est une fonction qui associe à chaque formule a un sous ensemble d'agents. Elle est définie par :

$$\mathbf{\Pi} : \mathcal{L} \rightarrow 2^{All}$$

$$\mathbf{\Pi}(a) = \bigcup_{\omega \in \Omega} \{\pi(\omega) \mid \omega \models a\}$$

La mesure de possibilité $\mathbf{\Pi}(a)$ est l'ensemble d'agents qui trouvent que a est possible. Par conséquent :

- $\mathbf{\Pi}(a) = \emptyset$ signifie que tous les agents trouvent que a est complètement impossible i.e. aucun ensemble d'agents ne trouve a possible.
- $\mathbf{\Pi}(a) = All$ signifie que tous les agents trouvent que a est totalement possible.
- $\mathbf{\Pi}(a) = \mathbf{\Pi}(\neg a) = A$ représente le cas de l'ignorance **individuelle** (le cas de l'absence totale de l'information). En effet, l'ensemble d'agents A trouve que a et $\neg a$ sont toutes les deux totalement possibles.
- $\mathbf{\Pi}(a) = \mathbf{\Pi}(\neg a) = All$ représente le cas de l'ignorance **globale** (le cas de l'absence totale de l'information pour tous les agents). En effet, tous les agents trouvent que a et $\neg a$ sont toutes les deux totalement possibles.
- $\mathbf{\Pi}(a) \cup \mathbf{\Pi}(\neg a) = All$ signifie que les ensembles d'agents doivent trouver que de deux énoncés contraires, l'un au moins est possible.

Exemple 4.4 *Considérons a nouveau l'exemple 4.3. Les agents qui trouvent que la tumeur aux poumons est une cause possible du cancer des poumons sont calculés par :*
 $\mathbf{\Pi}(a) = \bigcup \{B \cup C, All, B, A \cup B, C, A \cup C, \emptyset, A\}$
donc $\mathbf{\Pi}(a) = All$.

En effet, tous les agents trouvent que la tumeur aux poumons est une cause possible du cancer des poumons.

La mesure de possibilité multi-agents satisfait les axiomes suivants :

- $\mathbf{\Pi}(\perp) = \emptyset$
- $\mathbf{\Pi}(\top) = All$
- $\forall a \in \mathcal{L}, \forall b \in \mathcal{L}, \mathbf{\Pi}(a \vee b) = \mathbf{\Pi}(a) \cup \mathbf{\Pi}(b)$

Si la mesure de possibilité multi-agents d'une disjonction peut être directement calculée à partir des mesure de possibilités multi-agents des sous formules, il n'en est pas de même pour les conjonctions. Pour ces dernières, nous avons seulement :

$$\mathbf{\Pi}(a \wedge b) \subseteq \mathbf{\Pi}(a) \cup \mathbf{\Pi}(b)$$

Dans le cas général, l'égalité n'est pas vérifiée. Si nous prenons par exemple le cas de l'ignorance totale, nous avons $\mathbf{\Pi}(a) \cap \mathbf{\Pi}(\neg a) = All$ alors que $\mathbf{\Pi}(a \wedge \neg a) = \mathbf{\Pi}(\perp) = \emptyset$. L'inégalité n'est pas aussi vérifiée pour la négation. En effet, si $\mathbf{\Pi}(a) = All$, alors $\mathbf{\Pi}(\neg a)$ peut prendre n'importe quel sous ensemble de All .

De façons duale, la mesure de nécessité multi-agents induite par π , noté \mathbf{N} est définie par :

$$\begin{aligned} \mathbf{N} : \mathcal{L} &\rightarrow 2^{All} \\ \mathbf{N}(a) &= \bigcap_{\omega \in \Omega} \{\overline{\pi(\omega)} \mid \omega \models \neg a\} \\ &= \overline{\mathbf{\Pi}(\neg a)} \end{aligned}$$

La mesure de nécessité (ou de certitude) multi-agents définie l'ensemble des agents qui sont certains que a est vraie. Ainsi :

- $\mathbf{N}(a) = All$ signifie que tous les agents sont certains que a est vraie.
- $\mathbf{N}(a) = \emptyset$ signifie qu'aucun ensemble d'agents n'est certain que a est vraie.
- $\mathbf{N}(a) = \mathbf{N}(\neg a) = \emptyset$ représente l'état de l'ignorance totale.
- $\mathbf{N}(a) \cap \mathbf{N}(\neg a) = \emptyset$ signifie que les ensembles d'agents doivent être certains qu'au moins l'une des deux propositions a et $\neg a$ est vraie.

Exemple 4.5 Soit l'exemple 4.3. Les agents qui sont certains que la tumeur aux poumons provoque le cancer des poumons sont :

$$\mathbf{N}(a) = \bigcap \{\overline{B}, \overline{A \cup B}, \overline{A}, All\}$$

$$\text{donc } \mathbf{N}(a) = \overline{A} \cap \overline{B}.$$

Ainsi, c'est l'ensemble des médecins qui sont certains que la tumeur aux poumons provoque le cancer des poumons.

La mesure de nécessité multi-agents satisfait les axiomes suivants :

- $\mathbf{N}(\perp) = \emptyset$
- $\mathbf{N}(\top) = All$
- $\forall a \in \mathcal{L}, \forall b \in \mathcal{L}, \mathbf{N}(a \wedge b) = \mathbf{N}(a) \cap \mathbf{N}(b)$

La mesure de nécessité multi-agents de la disjonction est donnée par l'inclusion suivante :

$$\mathbf{N}(a \vee b) \supseteq \mathbf{N}(a) \cap \mathbf{N}(b)$$

Il est à noter que cette inclusion n'est en générale pas une égalité. Il est de même pour la négation.

De plus, les propriétés suivantes sont vérifiées :

- $\forall a, \mathbf{N}(a) \subseteq \mathbf{\Pi}(a)$.
- $\forall a, \mathbf{\Pi}(a) \neq \emptyset \Rightarrow \mathbf{N}(a) = A$: un énoncé doit être totalement possible pour un ensemble d'agents avant d'être quelque peu certain. D'une façon équivalente nous avons : $\forall a, \mathbf{N}(a) \neq \emptyset \Rightarrow \exists A \subseteq All, \mathbf{\Pi}(a) = A$.

4.4.1.2 Satisfaction d'une base de connaissances possibiliste par la distribution de possibilité

Nous définissons dans cette section la relation de satisfiabilité entre une distribution de possibilité multi-agents et une formule multi-agents et ensuite entre une distribution de possibilité multi-agents et un ensemble de formules multi-agents.

Soit π une distribution de possibilité multi-agents sur \mathcal{L} et \mathbf{N} la mesure de nécessité multi-agents induite par π .

$$\pi \text{ satisfait } (a, A) \text{ si et seulement si } \mathbf{N}(a) \supseteq A$$

De la même façon, si ϕ_1, \dots, ϕ_n sont des formules multi-agents,

$$\pi \text{ satisfait } \phi_1 \wedge \dots \wedge \phi_n \text{ si et seulement si } \forall i = 1 \dots n, \pi \text{ satisfait } \phi_i$$

Notation 4.1 *Souvent, et quand il n'y aura pas d'ambiguïté, $\pi \models \Phi$ désigne " π satisfait Φ ".*

Remarque 4.1 *Il est évident que nous utilisons alors une même notation pour deux concepts différents : $\omega \models a$ signifie que l'interprétation ω satisfait la formule classique a , alors que $\pi \models \Phi$ signifie que la distribution de possibilité π satisfait l'ensemble des formules multi-agents Φ .*

Exemple 4.6 *Soit $\Phi = \{(a, A), (\neg a \vee b, B)\}$*

$$\begin{aligned} \pi \models \Phi &\Leftrightarrow \mathbf{N}(a) \supseteq A \text{ et } \mathbf{N}(\neg a \vee b) \supseteq B \\ &\Leftrightarrow \cap \{\overline{\pi(\omega)} \mid \omega \models \neg a\} \supseteq A \text{ et } \cap \{\overline{\pi(\omega)} \mid \omega \models a \wedge \neg b\} \supseteq B \end{aligned}$$

Soient : $ab, a\neg b, \neg ab, \neg a\neg b$ les quatre interprétations du langage propositionnel engendrées par a, b (où ab associe à a et à b la valeur Vrai, etc.). Nous avons alors :

$$\pi \models \Phi \Leftrightarrow \pi(\neg ab) \subseteq \bar{A}, \pi(\neg a\neg b) \subseteq \bar{A}, \pi(a\neg b) \subseteq \bar{B}, \pi(ab) \subseteq All$$

Si nous cherchons une distribution de possibilité multi-agents normalisée satisfaisant Φ alors nous poserons $\pi(ab) = All$.

$\mathcal{P}(\Phi)$ est l'ensemble des distributions de possibilité multi-agents satisfaisant Φ :

$$\mathcal{P}(\Phi) = \{\pi, \pi \text{ satisfait } \Phi\}$$

est appelé l'ensemble des modèles multi-agents de Φ .

Φ est multi-agents-consistante si $\mathcal{P}(\Phi) \neq \emptyset$.

Φ est multi-agents-inconsistante si $\mathcal{P}(\Phi) = \emptyset$.

Elle est valide si toute distribution de possibilité multi-agents sur Ω satisfait Φ (notée $\models \Phi$).

Le couple formé de l'ensemble des distributions multi-agents-normalisées sur Ω et de la relation de satisfiabilité constitue la sémantique multi-agents normalisée pour la logique multi-agents.

4.4.1.3 Conséquence logique

(b, B) est une conséquence logique de Φ (notée $\Phi \models (b, B)$) si et seulement si toute distribution de possibilité multi-agents satisfaisant Φ satisfait aussi (b, B) . Formellement :

$$\Phi \models (b, B) \text{ si et seulement si } \forall \pi, (\pi \models \Phi) \Rightarrow (\pi \models (b, B))$$

Σ est sémantiquement équivalente à Φ (noté $\Sigma \equiv \Phi$) si et seulement si $\mathcal{P}(\Phi) = \mathcal{P}(\Sigma)$.

Exemple 4.7 *De l'exemple précédent, nous vérifions immédiatement que $\Phi \models (b, A \cap B)$. En effet, une distribution de possibilité multi-agents π qui satisfait Φ est telle que $\pi(a\neg b) \subseteq \bar{B}$ et $\pi(\neg a\neg b) \subseteq \bar{A}$, et vérifie $\mathbf{N}(b) = \cap \{\overline{\pi(a\neg b)}, \overline{\pi(\neg a\neg b)}\} \supseteq A \cap B$ donc $\Phi \models (b, A \cap B)$.*

Les propriétés immédiates suivantes sont vérifiées :

- $(a, A) \models (a, B) \quad \forall B \subseteq A$ (preuve immédiate)
- $\models (a, All)$ si et seulement si a est une tautologie.

Preuve 4.1 (\Leftarrow) :

Si a est une tautologie alors nous avons toujours : $\mathbf{N}(a) = All$. Donc, toute distribution de possibilité satisfait (a, All) .

(\Rightarrow) :

Si a n'est pas une tautologie alors il existe une interprétation ω qui falsifie a . Donc, si π est la distribution de possibilité définie par $\pi(\omega) = All$ et $\forall \omega^* \neq \omega \quad \pi(\omega^*) = \emptyset$ nous avons $\mathbf{N}(a) = \emptyset$ et π ne satisfait donc pas (a, All) . La formule multi-agents (a, All) n'est donc pas valide.

- $(a, A) \wedge (a \rightarrow b, B) \models (b, A \cap B)$. Cette forme du modus ponens ensembliste est un cas particulier de la résolution multi-agents.

4.4.1.4 Le problème de déduction

Soit Φ un ensemble de formules multi-agents. Les formules de Φ sont considérées comme des hypothèses. Soit b une formule quelconque ; il s'agit de chercher le plus grand ensemble d'agents qui croient b déductible de la connaissance Φ (i.e. (b, B) est une conséquence logique de Φ). Ce plus grand ensemble d'agents est défini par :

$$Ens(b, \Phi) = \cup \{B \in 2^{All}, \Phi \models (b, B)\}$$

Proposition 4.1

$$Ens(b, \Phi) = \cap \{\mathbf{N}(b), \pi \in \mathcal{P}(\Phi)\}$$

Preuve 4.2

$$\begin{aligned} Ens(b, \Phi) &= \cup \{B \in 2^{All}, \Phi \models (b, B)\} \\ &= \cup \{B \in 2^{All} \mid \forall \pi \in \mathcal{P}(\Phi), \mathbf{N}(b) \supseteq B\} \\ &= \cup \{B \in 2^{All} \mid \cap \{\mathbf{N}(b), \pi \in \mathcal{P}(\Phi)\} \supseteq B\} \\ &= \cap \{\mathbf{N}(b), \pi \in \mathcal{P}(\Phi)\} \end{aligned}$$

Exemple 4.8 Dans l'exemple précédent, il n'existe pas un sous ensemble \mathcal{E} tel que : $\mathcal{E} \supset A \cap B$ tel que $\Phi \models (b, \mathcal{E})$ de ce fait $Ens(b, \Phi) = A \cap B$.

4.4.1.5 Le modèle multi-agents le moins spécifique

La formule multi-agents (a, A) représente l'information qu'au moins tous les agents qui sont dans A croient a . En d'autres termes, les agents dans A trouvent les interprétations de $\neg a$ impossible. Cela veut dire que l'ensemble maximal d'agents qui pensent que $\neg a$ est possible est \bar{A} . Ainsi, le fait que tous les agents qui sont dans A croient a n'empêche pas tous les agents qui sont dans All de trouver que a est possible. Cela nous mène à la représentation sémantique de la formule (a, A) par la distribution de possibilité multi-agents $\pi_{(a,A)}$ suivante :

$$\forall \omega \in \Omega, \pi_{\{(a,A)\}}(\omega) = \begin{cases} All & \text{si } \omega \models a \\ \bar{A} & \text{si } \omega \models \neg a \end{cases}$$

où Ω est l'ensemble des interprétations associées à \mathcal{L} .

Plus généralement, la distribution de possibilité multi-agents π_Φ sémantiquement associée à l'ensemble des formules multi-agents Φ est définie par :

$$\pi_\Phi(\omega) = \begin{cases} All & \text{si } \forall (a_i, A_i) \in \Phi, \omega \models a_i \\ \bigcap \{\overline{A_i} : (a_i, A_i) \in \Phi, \omega \models \neg a_i\} & \text{sinon.} \end{cases}$$

En effet, la valeur de la distribution de possibilité multi-agents $\pi_\Phi(\omega)$ pour ω est obtenue comme l'intersection des différents sous-ensembles d'agents $\overline{A_i}$ qui trouvent ω possible selon les différentes formules (a_i, A_i) violées par cette interprétation : plus grand est l'ensemble d'agents qui trouvent l'interprétation impossible, plus petit est l'ensemble maximal de ceux qui la trouvent peut être possible. Nous avons alors :

$$N_\Phi(a_i) \supseteq A_i \text{ pour } i = 1, \dots, n.$$

Proposition 4.2 : *Pour toute distribution de possibilité multi-agents π , π satisfait Φ (noté $\pi \models \Phi$) si et seulement si $\pi \subseteq \pi_\Phi$ i.e. $\forall \omega, \pi(\omega) \subseteq \pi_\Phi(\omega)$.*

Preuve 4.3

$$\begin{aligned} \pi \models \Phi &\Leftrightarrow \forall i = 1, \dots, n \ \pi \models (a_i, A_i) \\ &\Leftrightarrow \forall i = 1, \dots, n \ N(a_i) \supseteq A_i \\ &\Leftrightarrow \forall i = 1, \dots, n \ \bigcap \{\overline{\pi(\omega)} \mid \omega \models \neg a_i\} \supseteq A_i \\ &\Leftrightarrow \forall i = 1, \dots, n \ \forall \omega \models \neg a_i \ \pi(\omega) \subseteq \overline{A_i} \\ &\Leftrightarrow \pi(\omega) \subseteq \bigcap \{\overline{A_i} \mid \omega \models \neg a_i, i = 1, \dots, n\} \\ &\Leftrightarrow \pi(\omega) \subseteq \pi_\Phi(\omega) \end{aligned}$$

Ainsi, (b, B) est une conséquence logique de Φ si et seulement si $\pi_\Phi(\omega)$ est inclus dans $\pi_{(b,B)}(\omega)$, formellement :

$$\Phi \models (b, B) \text{ si et seulement si } \forall \omega, \pi_\Phi(\omega) \subseteq \pi_{(b,B)}(\omega)$$

4.4.1.6 Déduction en logique multi-agents

Corollaire 4.1

$$\Phi \models (a, A) \text{ si et seulement si } \pi_\Phi \models (a, A)$$

En d'autres termes, $Ens(a, \Phi) = N_\Phi(a)$, ou $N_\Phi(a)$ est la mesure de nécessité multi-agents induite par π_Φ .

Proposition 4.3 *Soient Φ, Σ deux ensembles de formules multi-agents et π_Φ la distribution de possibilité multi-agents la moins spécifique parmi celles satisfaisant Φ . Alors*

$$\Phi \models \Sigma \text{ si et seulement si } \pi_\Phi \text{ satisfait } \Sigma$$

Preuve 4.4

$$\begin{aligned} \Phi \models \Sigma &\Leftrightarrow \mathcal{P}(\Phi) \subseteq \mathcal{P}(\Sigma) \\ &\Leftrightarrow \forall \pi (\pi \subseteq \pi_\Phi \Rightarrow \pi \subseteq \pi_\Sigma) \\ &\Leftrightarrow \pi_\Phi \subseteq \pi_\Sigma \\ &\Leftrightarrow \pi_\Phi \text{ satisfait } \Sigma. \end{aligned}$$

Proposition 4.4 *Soit Φ un ensemble de formules de la logique multi-agents, multi-agents-consistant, et b une formule quelconque. Alors $\Phi \models (b, B)$ si et seulement si $\Phi_B \models (b, B)$.*

Preuve 4.5 (\Leftarrow) : est immédiate car Φ_B est inclus dans Φ .

(\Rightarrow) :

Soit $\Phi = \{(a_i, A_i), i = 1, \dots, n\}$.

$$\begin{aligned} \Phi \models (b, B) &\Rightarrow \forall \omega \models \neg b, \cap \{\overline{A_i}, \omega \models \neg a_i\} \subseteq \overline{B} \\ &\Rightarrow \forall \omega \models \neg b, \cap [\cap \{\overline{A_i}, \omega \models \neg a_i, A_i \supseteq B\}, \cap \{\overline{A_i}, \omega \models \neg a_i, A_i \subset B\}] \subseteq \overline{B} \\ &\Rightarrow \forall \omega \models \neg b, \cap \{\overline{A_i}, \omega \models \neg a_i, A_i \supseteq B\} \subseteq \overline{B} (\text{car } \cap \{\overline{A_i}, \omega \models \neg a_i, A_i \subset B\} \supset \overline{B}) \\ &\Rightarrow \forall \omega \models \neg b, \pi_{\Phi_B}(\omega) \subseteq \overline{B} \\ &\Rightarrow \Phi_B \models (b, B). \end{aligned}$$

Ce résultat exprime que pour prouver qu'une formule est (au moins) crue par un ensemble d'agents, seules les formules de la base de connaissances qui sont elles-mêmes au moins crues par cet ensemble d'agents sont utiles.

4.4.2 Les distributions de possibilité multi-agents individuellement-normalisées comme sémantique de la logique multi-agents

4.4.2.1 La sémantique individuellement normalisée de la logique multi-agents

Dans cette sous-section, nous proposons une deuxième sémantique naturelle pour la logique multi-agents. Elle est plus souple et elle permet de traiter les inconsistances collectives.

L'impossibilité de trouver une distribution de possibilité multi-agents satisfaisant Φ est due en fait à la contrainte de la multi-agents-normalisation : $\exists \omega, \pi(\omega) = All$.

Considérons maintenant \mathcal{P}_{IN} l'ensemble des distributions de possibilité multi-agents multi-agents-normalisé ou non sur Ω . Une distribution de possibilité multi-agents (multi-agents-normalisé ou non) est simplement une application de Ω dans 2^{All} . Nous définissons pour l'instant les mesures de possibilité et de nécessité multi-agents comme dans le cas du multi-agents-normalisé.

Soit Φ une conjonction de formules multi-agents. Nous définissons $\mathcal{P}_{IN}(\Phi)$ comme l'ensemble des distributions de possibilité multi-agents, multi-agents normalisées ou non, satisfaisant Φ .

L'ensemble des distributions de possibilité multi-agents multi-agents normalisées ou non, muni de la relation de satisfaisabilité définie comme dans le cas normalisé, est appelé sémantique individuellement-normalisée pour la logique multi-agents.

Exemple 4.9 *Reprenons l'exemple 4.3 sur les opinions sur la détection du cancer des poumons. Rajoutons maintenant un autre groupe qui est le groupe des hommes. Ce dernier se compose de 03 hommes à savoir : Djamel, Rachid et Nadir.*

La Table 4.3 donne les distributions de possibilités multi-agents pour chaque interprétation ω . A titre d'exemple, l'interprétation ω_1 correspond à l'état du monde suivant : le patient a une tumeur aux poumons et est un homme et s'est exposé aux produits toxiques

et c'est un jeune. L'ensemble d'agents qui croient que le patient risque d'attraper un cancer des poumons est l'ensemble des femmes et des médecins car les jeunes pensent que seulement les vieux qui risquent d'attraper un cancer des poumons et le groupe des hommes pensent que seulement les femmes qui sont concernées par le cancer des poumons.

Nous remarquons que les distributions de possibilités multi-agents données par cette table ne sont pas normalisées. En effet : $\sum \omega, \pi(\omega) = All$.

ω_i	ω	$\pi(\omega)$
ω_1	$a \wedge b \wedge c \wedge d$	$B \cup C$
ω_2	$a \wedge b \wedge c \wedge \neg d$	$A \cup B \cup C$
ω_3	$a \wedge b \wedge \neg c \wedge d$	B
ω_4	$a \wedge b \wedge \neg c \wedge \neg d$	$A \cup B$
ω_5	$a \wedge \neg b \wedge c \wedge d$	$C \cup D$
ω_6	$a \wedge \neg b \wedge c \wedge \neg d$	$A \cup C \cup D$
ω_7	$a \wedge \neg b \wedge \neg c \wedge d$	D
ω_8	$a \wedge \neg b \wedge \neg c \wedge \neg d$	$A \cup D$
ω_9	$\neg a \wedge b \wedge c \wedge d$	B
ω_{10}	$\neg a \wedge b \wedge c \wedge \neg d$	$A \cup B$
ω_{11}	$\neg a \wedge b \wedge \neg c \wedge d$	B
ω_{12}	$\neg a \wedge b \wedge \neg c \wedge \neg d$	$A \cup B$
ω_{13}	$\neg a \wedge \neg b \wedge c \wedge d$	D
ω_{14}	$\neg a \wedge \neg b \wedge c \wedge \neg d$	$A \cup D$
ω_{15}	$\neg a \wedge \neg b \wedge \neg c \wedge d$	D
ω_{16}	$\neg a \wedge \neg b \wedge \neg c \wedge \neg d$	$A \cup D$

FIGURE 4.3 – Distributions de possibilités multi-agents non normalisée associées à l'exemple 4.9

Le nombre $IN(\pi) = \cap \{\overline{\pi(\omega)}, \omega \in \Omega\}$ est appelé l'ensemble de l'individuelle normalisation de π , et le nombre $\cap \{IN(\pi), \pi \in \mathcal{P}_{IN}(\Phi)\}$ est appelé l'individuelle inconsistance de Φ . Il est noté $i - inc(\Phi)$.

Nous avons : $IN(\pi) = \cap \{\overline{\pi(\omega)}, \omega \in \Omega\} = \cap \{\overline{\pi(\omega)}, \omega \neq \perp\} = \mathbf{N}(\perp)$ et par conséquent $i - inc(\Phi) = \cap \{\mathbf{N}(\perp), \pi \in \mathcal{P}_{IN}(\Phi)\}$, ce qui justifie l'appellation "l'individuelle inconsistance", par abus de langage pour "la nécessité de la contradiction".

Dans cette sémantique, certaines des propriétés de la sémantique multi-agents-normalisée restent valables et d'autres ne le sont plus.

Propriété 4.1 1. $\forall a \in \mathcal{L}, \forall b \in \mathcal{L}, \Pi(a \vee b) = \Pi(a) \cup \Pi(b)$

2. $\forall a \in \mathcal{L}, \mathbf{N}(a) = \overline{\Pi(\neg a)}$

$$3. \forall a \in \mathcal{L}, \forall b \in \mathcal{L}, \quad \mathbf{N}(a \wedge b) = \mathbf{N}(a) \cap \mathbf{N}(b)$$

$$4. \forall a, b \text{ si } a \models b \text{ alors } \mathbf{\Pi}(b) \supseteq \mathbf{\Pi}(a) \text{ et } \mathbf{N}(b) \supseteq \mathbf{N}(a)$$

$$5. \mathbf{\Pi}(\perp) = \emptyset$$

Par contre, $\mathbf{N}(\perp) = \emptyset$ n'est pas nécessairement vérifiée

$$6. \mathbf{N}(\top) = All$$

Par contre, $\mathbf{\Pi}(\top) = All$ n'est pas nécessairement vérifiée

$$7. \forall a \in \mathcal{L}, \quad \mathbf{N}(a) \supseteq \mathbf{N}(\perp)$$

$$8. \forall a \in \mathcal{L}, \quad \mathbf{N}(a) \cap \mathbf{N}(\neg a) = \mathbf{N}(\perp) (\neq \emptyset \text{ en général})$$

$$9. \forall a \in \mathcal{L}, \quad \mathbf{\Pi}(a) \cup \mathbf{\Pi}(\neg a) = \mathbf{\Pi}(\top) (\neq All \text{ en général})$$

$$10. \mathbf{N}(a) \subseteq \mathbf{\Pi}(a) \text{ n'est pas nécessairement vérifiée .}$$

$$11. \mathbf{N}(a) \neq \emptyset \Rightarrow \mathbf{\Pi}(a) = A \text{ n'est pas nécessairement vérifiée.}$$

4.4.2.2 Conséquence logique en logique multi-agents individuellement normalisée

(b, B) est une conséquence logique de Φ dans la sémantique individuellement normalisée (notée $(b, B) \models_{IN} \Phi$) si et seulement si $\mathcal{P}_{IN}(\Phi)$ est inclus dans $\mathcal{P}_{IN}((b, B))$.

Exemple 4.10 $\Phi = \{(a, A), (\neg a \vee b, B), (c, A), (\neg c, C)\}$

On a $\Phi \models_{IN} (b, A \cap B)$ mais pas $\Phi \models_{IN} (b, A)$.

On a $\Phi \models_{IN} (\perp, A \cap C)$ mais pas $\Phi \models_{IN} (\perp, A)$.

On a $\Phi \models_{IN} (\phi, A \cap C)$ pour n'importe quelle formule ϕ .

4.4.2.3 La distribution de possibilité multi-agents, multi-agents normalisée ou non, la moins spécifique parmi les distributions les moins individuellement-normalisées

La proposition 4.2 est généralisée à la sémantique individuellement normalisée.

Proposition 4.5 : Pour toute distribution de possibilité multi-agents, multi-agents normalisée ou non π_{IN} , π_{IN} satisfait Φ si et seulement si $\pi_{IN} \subseteq \pi_{IN\Phi}$ i.e. $\forall \omega, \pi_{IN}(\omega) \subseteq \pi_{IN\Phi}(\omega)$.

Preuve 4.6

$$\begin{aligned}
\pi_{IN} \models \Phi &\Leftrightarrow \forall i = 1, \dots, n \ \pi_{IN} \models (a_i, A_i) \\
&\Leftrightarrow \forall i = 1, \dots, n \ \mathbf{N}(a_i) \supseteq A_i \\
&\Leftrightarrow \forall i = 1, \dots, n \ \cap \{ \overline{\pi(\omega)} \mid \omega \models \neg a_i \} \supseteq a_i \\
&\Leftrightarrow \forall i = 1, \dots, n \ \forall \omega \models \neg A_i \ \pi(\omega) \subseteq \overline{A_i} \\
&\Leftrightarrow \pi_{IN}(\omega) \subseteq \cap \{ \overline{A_i} \mid \omega \models \neg a_i, i = 1, \dots, n \} \\
&\Leftrightarrow \pi_{IN}(\omega) \subseteq \pi_{IN_\Phi}(\omega)
\end{aligned}$$

Ainsi, (b, B) est une conséquence logique de Φ si et seulement si $\pi_{IN_\Phi}(\omega)$ est inclus dans $\pi_{IN(b,B)}(\omega)$, formellement :

$$\Phi \models (b, B) \text{ si et seulement si } \forall \omega, \pi_{IN_\Phi}(\omega) \subseteq \pi_{IN(b,B)}(\omega)$$

Corollaire 4.2

$$\Phi \models (a, A) \text{ si et seulement si } \pi_{IN_\Phi} \models (a, A)$$

En d'autres termes, $Ens(a, \Phi) = \mathbf{N}_\Phi(a)$, ou $\mathbf{N}_\Phi(a)$ est la mesure de nécessité multi-agents induite par π_Φ .

L'expression " distribution de possibilité multi-agents la moins spécifique parmi celles satisfaisant Φ " n'a pas été réutilisée volontairement. En effet, l'inclusion $\pi \subseteq \pi'$ n'exprime réellement que π est plus spécifique que π' qu'au niveau de la multi-agents normalisation égale (c'est-à-dire $\cup \omega \in \Omega \pi(\omega) = \cup \omega \in \Omega \pi'(\omega)$); **sinon** $\pi \subseteq \pi'$ exprime avant tout que π est plus individuellement-normalisée que π' . La distribution maximale π_Φ est donc en fait la distribution de possibilité multi-agents, multi-agents normalisée ou non la moins spécifique parmi les distributions de possibilité multi-agents les moins individuellement-normalisées satisfaisant Φ .

Proposition 4.6 :

$$i - inc(\Phi) = IN(\pi_\Phi) = \mathbf{N}_\Phi(\perp).$$

L'individuelle inconsistance de Φ est égale à l'ensemble de l'individuelle normalisation de π_Φ . En particulier, si π_Φ est multi-agents normalisée, alors $i - inc(\Phi) = \emptyset$, ce qui correspond à la consistance dans la sémantique multi-agents normalisée (puisqu'il existe au moins une distribution multi-agents normalisée, à savoir π_Φ , satisfaisant Φ).

Preuve 4.7

$$\begin{aligned}
i - inc(\Phi) &= \cap \{ IN(\pi), \pi \in \mathcal{P}_{IN}(\Phi) \} \\
&= \cap \{ IN(\pi), \pi \supseteq \pi_\Phi \} \\
&= \cap \{ \cap \{ \overline{\pi(\omega)}, \omega \in \Omega, \pi \supseteq \pi_\Phi \} \} \\
&= \cap \{ \overline{\pi_\Phi(\omega)}, \omega \in \Omega \} \\
&= IN(\pi_\Phi).
\end{aligned}$$

$$\begin{aligned}
\mathbf{N}_\Phi(\perp) &= \cap \{ \overline{\pi_\Phi(\omega)}, \omega \models \perp \} \\
&= \cap \{ \overline{\pi_\Phi(\omega)}, \omega \in \Omega \} \\
&= \cap \{ \overline{\pi(\omega)}, \omega \in \Omega \} \\
&= IN(\pi_\Phi) \\
&= i - inc(\Phi).
\end{aligned}$$

Proposition 4.7 : Soit Φ une base de connaissances multi-agents individuellement inconsistante ($i - inc(\Phi) \neq \emptyset$). Alors

1. $\forall b \in \mathcal{L}, \mathbf{N}_\Phi(b) \supseteq i - inc(\Phi)$
2. $\forall b \in \mathcal{L}, \mathbf{N}_\Phi(b) \cap \mathbf{N}_\Phi(\neg b) = i - inc(\Phi)$

Preuve 4.8 1. $\mathbf{N}_\Phi(b) \supseteq \mathbf{N}_\Phi(\perp) = i - inc(\Phi)$.

2. La preuve est plus général que celle définie précédemment. Elle découle des propriétés 4.1.

Théorème 4.1 Soit Φ une conjonction de formules multi-agents. Alors $i - inc(\Phi) = \emptyset$ si et seulement si Φ^* est consistante.

Preuve 4.9 Soit $\Phi = \{(a_i, A_i), i = 1, \dots, n\}$.

(\Leftarrow) :

Si Φ^* est consistante alors Φ^* admet un modèle ω^* . Alors $\pi_\Phi(\omega^*) = \cap \{\overline{A_i} \mid \omega^* \models \neg a_i\} = All$ étant donné que $\omega^* \models a_i$. Donc π_Φ est normalisée et $i - inc = \emptyset$.

(\Rightarrow) :

Par Définition : $i - inc(\Phi) = \emptyset$ si et seulement si π_Φ est normalisée, c'est à dire si et seulement si $\exists \omega^* \in \Omega$ tel que $\pi_\Phi(\omega^*) = All$. Cela implique : $\omega^* \models a_i, \forall i$. D'où Φ^* est consistante.

4.4.2.4 Déduction sous individuelle inconsistance

L'inclusion : $\pi_\Phi \subset \pi_{IN_\Phi}$ est immédiate, puisqu'une distribution de possibilité multi-agents multi-agents normalisée est un cas particulier d'une distribution de possibilité " multi-agents multi-agents normalisées ou non ". D'où : $(\phi \models_{IN} \Phi) \Rightarrow (\phi \models \Phi)$ (la réciproque n'étant éventuellement pas vraie). La sémantique individuellement-normalisée étend donc naturellement la sémantique multi-agents normalisée, en ce sens qu'elle est moins tolérante pour la déduction.

Notation 4.2 Dorénavant nous noterons \models à la place de \models_{IN} , lorsqu'il n'y aura pas d'ambiguïté, et quand nous parlerons de la logique multi-agents par défaut à la sémantique individuellement-normalisée.

Les propositions 4.3 et 4.4 sont valables dans la sémantique individuellement-normalisée, sous la même formulation et avec les mêmes preuves (car elles n'utilisent pas la normalisation) :

Proposition 4.8 : Soient Φ, Σ deux ensembles de formules multi-agents et π_Φ la distribution de possibilité multi-agents la moins spécifique parmi celles satisfaisant Φ . Alors

$$\Phi \models \Sigma \text{ si et seulement si } \pi_\Phi \text{ satisfait } \Sigma$$

Preuve 4.10 Elle est identique à la preuve de la proposition 4.3.

Proposition 4.9 : Soit Φ un ensemble de formules de la logique multi-agents, multi-agents consistant, et b une formule quelconque. Alors $\Phi \models (b, A)$ si et seulement si $\Phi_A \models (b, A)$.

Preuve 4.11 Elle est identique à la preuve de la proposition 4.4.

4.4.2.5 Dédution multi-agents triviale vs déduction multi-agents non-triviale

Quelle que soit la formule b , b est déductible de Φ avec un ensemble plus grand ou égal à $i - inc(\Phi)$.

Les déductions de (b, A) avec $A = i - inc(\Phi)$ ne doivent rien à la nature de b , mais sont provoquées par l'inconsistance individuelle. Elles seront appelées déductions multi-agents triviales. Les déductions de (b, A) avec $A \supset i - inc(\Phi)$ seront appelées déductions multi agents non-triviales.

L'inconsistance individuelle de Φ se comporte comme un ensemble en dehors duquel rien n'est significatif (comme dans le cas d'une inconsistance classique) et dans lequel toutes les déductions sont significatives (comme s'il n'y avait pas de l'inconsistance au sens classique).

Les résultats suivants sont alors obtenus :

Proposition 4.10 *Posons $i - inc(\Phi) = i - inc$. Alors*

$$\Phi \equiv \Phi_{i-inc} \equiv (\perp, i - inc) \wedge \overline{\Phi_{i-inc}}$$

Preuve 4.12 *Posons $\Phi = \{(a_i, A_i), i = 1..n\}$*

1. $\Phi \models \Phi_{i-inc}$ est immédiate car Φ contient Φ_{i-inc} .

2. $\Phi_{i-inc} \models (\perp, i - inc) \wedge \overline{\Phi_{i-inc}}$

$\Phi_{i-inc} \models \overline{\Phi_{i-inc}}$: est immédiate car Φ_{i-inc} contient $\overline{\Phi_{i-inc}}$.

Montrons maintenant que $\Phi_{i-inc} \models (\perp, i - inc)$, C'est-à-dire que $i - inc(\Phi_{i-inc}) \supseteq i - inc$.

Soit $\pi_{\Phi_{i-inc}}(\omega)$ la distribution de possibilités la moins spécifique qui satisfait la base de connaissances Φ_{i-inc} nous avons alors :

$$\forall \omega \quad \pi_{\Phi_{i-inc}}(\omega) = \cap \{ \overline{A_i} | \omega \models \neg a_i | A_i \supseteq i - inc \}$$

$$\begin{aligned} i - inc(\Phi) = i - inc &\Rightarrow \forall \omega, \quad \cap \{ \overline{A_i} | \omega \models \neg a_i \} \subseteq \overline{i - inc} \\ &\Rightarrow \forall \omega, \quad \cap [\cap \{ \overline{A_i} | \omega \models \neg a_i | A_i \supseteq i - inc \}, \cap \{ \overline{A_i} | \omega \models \neg a_i | A_i \subseteq i - inc \}] \subseteq \overline{i - inc} \\ &\Rightarrow \forall \omega, \quad \cap \{ \overline{A_i} | \omega \models \neg a_i | A_i \supseteq i - inc \} \subseteq \overline{i - inc} \\ &\quad (\text{car } \{ \overline{A_i} | \omega \models \neg a_i | A_i \subseteq i - inc \} \supseteq \overline{i - inc}) \\ &\Rightarrow \forall \omega, \quad \pi_{\Phi_{i-inc}}(\omega) \subseteq \overline{i - inc} \\ &\Rightarrow i - inc(\Phi_{i-inc}) \supseteq i - inc \end{aligned}$$

Nous avons alors : $\Phi_{i-inc} \models (\perp, i - inc) \wedge \overline{\Phi_{i-inc}}$

3. Soit $\pi \models (\perp, i - inc) \wedge \overline{\Phi_{i-inc}}$, c'est-à-dire : $\forall \omega, \quad \pi(\omega) \subseteq \overline{i - inc}$ et $\pi(\omega) \subseteq \cap \{ \overline{A_i} | \omega \models \neg a_i, A_i \supseteq i - inc \}$

Alors $\forall (a_i, A_i) \in \Phi$:

Si $A_i \supseteq i - inc$ alors $\pi \models (a_i, A_i)$.

Et si $A_i \subseteq i - inc$ alors $N(a_i) \supseteq N(\perp) \supseteq i - inc \supseteq A_i$ donc $\pi \models (a_i, A_i)$. D'où $\pi \models \Phi$.

Proposition 4.11 : Il existe (au moins) une interprétation ω^* telle que $\cup\{\pi_\Phi(\omega), \omega \in \Omega\} = \pi_\Phi(\omega^*)$.

Preuve 4.13 Elle est immédiate car Ω est fini.

Théorème 4.2 Posons $i - inc(\Phi) = i - inc$. Alors :

$\overline{\Phi_{i-inc}}$ est consistante.

Preuve 4.14 Posons $\Phi = \{(a_i, A_i), i = 1 \dots n\}$

Soit $\pi_{\overline{\Phi_{i-inc}}}(\omega)$ la distribution de possibilités qui satisfait $\overline{\Phi_{i-inc}}$.

La proposition 4.11 assure l'existence d'une interprétation ω^* telle que $\pi_\Phi(\omega^*) = \overline{i - inc} = \cap\{\overline{A_i} | \omega \models \neg a_i\}$.

$$\begin{aligned} \overline{i - inc} = \cap\{\overline{A_i} | \omega \models \neg a_i\} &\Rightarrow \forall i, (\omega^* \models \neg a_i \Rightarrow A_i \subseteq i - inc) \\ &\Rightarrow \omega^* \text{ satisfait } \wedge \{a_i | A_i \supset i - inc\} \\ &\Rightarrow \omega^* \models \overline{\Phi_{i-inc}^*} \\ &\Rightarrow \overline{\Phi_{i-inc}^*} \text{ est consistante} \\ &\Rightarrow \overline{\Phi_{i-inc}} \text{ est consistante (thorme 4.1)}. \end{aligned}$$

Corollaire 4.3

$$i - inc(\Phi) = \cup\{A, \Phi_A^* \text{ inconsistent}\} = \cap\{A, \Phi_A^* \text{ consistant}\}.$$

Preuve 4.15 Elle est immédiate à partir des propositions 4.1, 4.10 et du résultat précédent.

Théorème 4.3 Toutes les déductions non-triviales qui peuvent être effectuées à partir de Φ peuvent être déduites de $\overline{\Phi_{i-inc}}$, c'est-à-dire plus formellement :

$$[\Phi \models (b, B) \text{ non trivialement}] \Rightarrow [\overline{\Phi_{i-inc}} \models (b, B)]$$

Preuve 4.16 Posons $\Phi = \{(a_i, A_i), i = 1 \dots n\}$

$$\begin{aligned} \Phi \models (b, B) \text{ non trivialement} &\Rightarrow \Phi \models (b, B) \text{ et } B \supset i - inc \\ &\Rightarrow \forall \omega \models \neg b, \cap\{\overline{A_i} | \omega \models \neg a_i\} \subseteq \overline{B} \subset i - inc \quad (1) \\ &\Rightarrow (\forall \omega \models \neg b, \omega \models \neg a_i \Rightarrow A_i \supset i - inc) \\ &\Rightarrow \forall \omega \models \neg b, \cap\{\overline{A_i} | \omega \models \neg a_i | A_i \supset i - inc\} = \cap\{\overline{A_i}, \omega \models \neg a_i\} \quad (2) \\ &\Rightarrow \forall \omega \models \neg b, \cap\{\overline{A_i} | \omega \models \neg a_i | A_i \supset i - inc\} \subseteq \overline{B} \quad (De(1) \text{ et}(2)) \\ &\Rightarrow \overline{\Phi_{i-inc}} \models (b, B) \end{aligned}$$

Proposition 4.12 Soit Φ une conjonction de formules multi-agents et b une formule quelconque. Alors $\exists B \neq \emptyset$ tel que $\Phi \models (b, B)$ si et seulement si $\Phi^* \models b$.

Preuve 4.17 Posons $\Phi = \{(a_i, A_i), i = 1 \dots n\}$

Lemme 4.1 $\omega \models \Phi^* \Leftrightarrow \pi_\Phi(\omega) = All$.

Preuve 4.18 $\omega \models \Phi^* \Leftrightarrow (\forall i = 1, \dots, n) \omega \models a_i \Leftrightarrow \pi_\Phi(\omega) = \cap\{\overline{A_i} | \omega \models \neg a_i\} = ALL$.

(\Rightarrow) :

Supposons que $\exists B \neq \emptyset$ tel que $\Phi \models (b, B)$.

1. $i - inc(\Phi) \supset \emptyset$:

Dans ce cas, Φ^* est inconsistante (théorème 4.1) d'où $\Phi^* \models b$.

2. $i - inc(\Phi) = \emptyset$:

Soit alors $\omega \models \Phi^*$, montrons que $\omega \models b$.

Supposons que $\omega \models \neg b$, alors $\pi_\Phi(\omega) \subseteq \overline{B} \subset All$ (car $\Phi \models (b, B)$). Ceci est impossible d'après le lemme 4.1. Donc $\omega \models b$. Tout modèle de Φ^* étant donc un modèle de b , nous avons alors $\Phi^* \models b$.

(\Leftarrow) :

Supposons que $\Phi^* \models b$.

Soit $\omega \models \neg b$ alors $\omega \models \neg \Phi^*$ (car $\Phi^* \models b$) C'est-à-dire : $\exists i(\omega)$ tel que $\omega \models \neg a_{i(\omega)}$, nous avons donc $\pi_\Phi(\omega) = \cap \{\overline{B}_i \mid \omega \models \neg a_i\} \subseteq \overline{B_{i(\omega)}} \subset All$. Ceci étant valable $\forall \omega \models \neg b$, nous avons ainsi $N_\Phi(b) = \cap \{\overline{\pi_\Phi(\omega)} \mid \omega \models \neg b\} \supseteq \cap \{B_{i(\omega)} \mid \omega \models \neg b\} \supset \emptyset$.

D'où : $\exists B \supset \emptyset$ tel que $\Phi \models (b, B)$.

4.5 L'ensemble d'interprétations individuelles

A lieu de mettre l'accent sur l'ensemble $\pi(\omega)$ d'agents qui considère chaque interprétation possible, nous pouvons représenter la même information en mettant l'accent sur l'ensemble des interprétations E_k considérées possibles par chaque agents $k \in All$ tel que $E_k = \{\omega, k \in \pi(\omega)\}$.

Par dualité, nous avons $\pi(\omega) = \{k \in All, \omega \in E_k\}$. Donc naturellement :

$$\mathbf{\Pi}(a) = \{k, E_k \cap [a] \neq \emptyset\}$$

Comme suggéré dans [Smets 88] pour les fonctions de croyances, nous pouvons définir l'ensemble d'agents qui croient a par :

$$Bel(a) = \{k : \emptyset \neq E_k \subseteq [a]\}$$

Cet ensemble rassemble les sous ensembles consistants d'agents qui sont sûres que a est vraie. Nous avons évidemment la désirable inclusion : $\forall a, Bel(a) \subseteq \mathbf{\Pi}(a)$. A savoir, les agents consistants qui sont sûres que a est vraie sont parmi ceux qui croient que a est possiblement vraie. Nous avons :

$$Bel(a) = \mathbf{\Pi}(\top) \setminus \mathbf{\Pi}(\neg a) = \mathbf{\Pi}(\top) \cap \mathbf{N}(a)$$

et $Bel(a) \cap Bel(\neg a) = \emptyset$. Inversement, $\mathbf{N}(a) = Bel(a) \cup I = \{k : E_k \subseteq [a]\}$. En d'autres termes, $\mathbf{N}(a)$ correspond à l'idée de l'implicabilité logique et à cause du fait que dans la logique classique tout vient de la contradiction ; alors que $Bel(a)$ correspond plus à l'idée intuitive de la croyance non triviale.

Les propriétés suivantes sont alors vérifiées :

- Si $a \models b$ alors $Bel(a) \subseteq Bel(b)$.
- $Bel(a \wedge b) = Bel(a) \cap Bel(b)$.

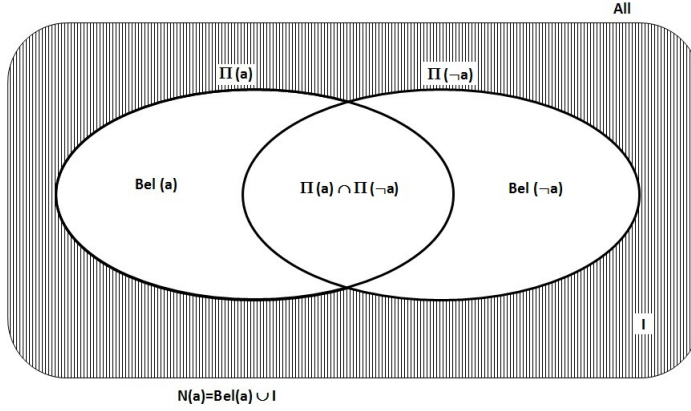


FIGURE 4.4 – Les différents sous-ensembles d'agents selon leurs comportements envers a .

La figure 4.4 représente la relation entre l'ensemble $\Pi(a)$ d'agents qui pensent que a est possible, l'ensemble $\Pi(\neg a)$ d'agents qui pensent que $\neg a$ est possible, l'ensemble I d'agents qui sont individuellement consistants, l'ensemble $Bel(a)$ des agents consistants qui sont certains que a et l'ensemble $Bel(\neg a)$ consistant d'agents qui sont certains que $\neg a$.

Exemple 4.11 Reprenons l'exemple 4.9 et calculons l'ensemble des interprétations E_k considérées possibles pour chaque agent $k \in All$:

- $E_{Riade} = \{\omega_1, \omega_6, \omega_8, \omega_{10}, \omega_{12}, \omega_{14}, \omega_{16}\}$;
- $E_{Adel} = \{\omega_1, \omega_6, \omega_8, \omega_{10}, \omega_{12}, \omega_{14}, \omega_{16}\}$;
- $E_{Salim} = \{\omega_1, \omega_6, \omega_8, \omega_{10}, \omega_{12}, \omega_{14}, \omega_{16}\}$;
- $E_{Nadine} = \{\omega_1, \omega_2, \omega_3, \omega_4, \omega_9, \omega_{10}, \omega_{11}, \omega_{12}\}$;
- $E_{Faiza} = \{\omega_1, \omega_2, \omega_3, \omega_4, \omega_9, \omega_{10}, \omega_{11}, \omega_{12}\}$;
- $E_{Mohamed} = \{\omega_1, \omega_2, \omega_5, \omega_6\}$;
- $E_{Djamel} = \{\omega_5, \omega_6, \omega_7, \omega_8, \omega_{13}, \omega_{14}, \omega_{15}, \omega_{16}\}$;
- $E_{Rachid} = \{\omega_5, \omega_6, \omega_7, \omega_8, \omega_{13}, \omega_{14}, \omega_{15}, \omega_{16}\}$;
- $E_{Nadir} = \{\omega_5, \omega_6, \omega_7, \omega_8, \omega_{13}, \omega_{14}, \omega_{15}, \omega_{16}\}$;

De plus, nous avons $\Pi(a) = \{Riad, Adel, Salim, Nadine, Faiza, Mohamed, Djamel, Rachid, Nadir\}$.

Aussi, $Bel(a) = \{Mohamed\}$ car $E_{Mohamed} \subseteq [a]$.

4.6 Axiomatique de la logique multi-agents

Nous nous intéressons dans cette partie à la recherche d'un système formel pour la logique multi-agents munie de la sémantique individuellement normalisée. D'abord, il est à noter que les tautologies de la logique multi-agents étant toutes les formules de la forme (τ, A) où τ est une tautologie de la logique classique et $A \in 2^{All}$, nous pouvons construire de façon triviale un système formel produisant toutes les tautologies de la logique multi-agents.

Nous nous intéressons plutôt au problème suivant : existe-t-il un système formel, c'est-à-dire un ensemble d'axiomes et des règles d'inférence, tel que pour tout ensemble Φ de formules multi-agents et pour toute formule multi-agents (b, B) conséquence logique de Φ , nous puissions dériver (b, B) à partir des axiomes, de Φ , et des règles d'inférences ?.

Nous proposons le système formel correct et complet suivant pour la logique multi-agents, munie de la sémantique individuellement normalisée :

Schémas d'axiomes :

$$(A1) \quad (a \rightarrow (b \rightarrow a), All)$$

$$(A2) \quad ((a \rightarrow (b \rightarrow c)) \rightarrow ((a \rightarrow b) \rightarrow (a \rightarrow c)), All)$$

$$(A3) \quad ((\neg a \rightarrow \neg b) \rightarrow ((\neg a \rightarrow b) \rightarrow a), All)$$

Règles d'inférences :

$$(Modus\ ponens\ ensembliste) \quad (a, A), (a \rightarrow b, B) \vdash (b, A \cap B)$$

$$(Subsomption\ ensembliste) \quad (a, A) \vdash (a, B) \text{ pour tout } B \subseteq A$$

Les axiomes sont ceux de la logique classique associés à l'ensemble de tous les agents All . La règle (Modus ponens ensembliste) représente le cœur du système formel.

Théorème 4.4 *Le système formel proposé est correct et complet relativement à la sémantique individuellement normalisée de la logique multi-agents, c'est-à-dire que pour tout ensemble de formules multi-agents Φ , nous avons :*

$$\Phi \models (a, A) \text{ si et seulement si } \Phi \vdash (a, A)$$

où $\Phi \vdash (a, A)$ signifie : " il existe une dérivation de (a, A) dans le système formel constitué de Φ , des schémas d'axiomes et des règles d'inférence du système.

Preuve 4.19 *Pour prouver la correction de la règle, nous aurons à vérifier, si $A_1 \cap A_2 \neq \emptyset$,*

$$\forall \omega, \pi(\omega)_{\{(c_1, A_1), (c_2, A_2)\}} \subseteq \pi(\omega)_{\{(c_3, A_1 \cap A_2)\}}$$

où (c_1, A_1) et (c_2, A_2) sont des clause multi-agents, et c_3 une résolvante classique de c_1 et c_2 ; alors $(c_3, A_1 \cap A_2)$ est la résolvante de (c_1, A_1) et (c_2, A_2) par la A-B-résolution $(c_1, A_1), (c_2, A_2) \vdash (c_3, A_1 \cap A_2)$. On a

$$\pi_{\{(c_1, A_1), (c_2, A_2)\}}(\omega) = \begin{cases} All & \text{si } \omega \models c_1 \wedge c_2 \\ \overline{A_1} \cap \overline{A_2} & \text{si } \omega \models \neg(c_1 \wedge c_2) \end{cases}$$

et,

$$\pi_{\{(c_3, A_1 \cap A_2)\}}(\omega) = \begin{cases} All & \text{si } \omega \models c_3 \\ \overline{A_1} \cap \overline{A_2} & \text{si } \omega \models \neg c_3 \end{cases}$$

L'inclusion est obtenue étant donné que la situation $\pi_{\{(c_1, A_1), (c_2, A_2)\}}(\omega) = All$ et $\pi_{\{(c_3, A_1 \cap A_2)\}}(\omega) = \overline{A_1} \cap \overline{A_2}$ ne peut pas se présenter car par hypothèse $\omega \models c_1 \wedge c_2$. Pour les autres cas, l'inclusion est maintenue donc la A-B-résolution est correcte.

Pour la complétude, supposons $\Phi \not\models (a, A)$. Alors $\nexists \Delta \subseteq \Phi, \Delta \vdash (a, A)$ en utilisant la A-B résolution, avec toutes les formules de Δ . D'où, pour toute Δ tel que $\Delta \subseteq \Phi$, soit $\Delta^* \not\models a$ ou $\Delta^* \vdash a$, mais $A \not\subseteq \bigcap_{(a_i, A_i) \in \Delta} A_i$.

Pour le premier cas, $\Delta^* \not\models a$ (complétude de la logique possibiliste). D'où $\Delta \not\models (a, A)$ aussi.

Pour le deuxième cas, $\exists (i, k) : k \in A, k \notin A_i$. Soit $\Delta_k^* = \{a_j : k \in A_j, (a_j, A_j) \in \Delta\}$, clairement $\Delta_k^* \not\models a$ car elle ne contient pas a_i et a_i est utilisé pour dériver a . D'où, $\Delta_k^* \not\models a$ (complétude de la logique possibiliste). Mais alors $\Delta_k \not\models (a, \{k\})$ aussi, et $\Delta \not\models (a, A)$, comme c'est le cas pour tous les sous-ensembles de Φ , $\Phi \not\models (a, A)$.

Remarque 4.2 Une autre preuve peut être faite par équivalence :

$\Phi \vdash (a, A) \iff \Phi \vdash (a, \{k\}), \forall k \in A \iff \Phi_{\{k\}} \vdash (a, \{k\}), \forall k \in A \iff \Phi_{\{k\}}^* \vdash a, \forall k \in A \iff \Phi_{\{k\}}^* \models a, \forall k \in A$ (complétude de la logique possibiliste). Ce dernier reste valable pour $E_k = [\Phi_{\{k\}}^*] \subseteq [a]$.

Alors encore une fois, $\Phi_{\{k\}}^* \models a \iff \Phi_{\{k\}} \models (a, \{k\})$ ($\{(\omega, k) : \omega \in E_k\} \subseteq \{(\omega, k) : \omega \in [a]\}$); alors $\Phi_{\{k\}} \models (a, \{k\}) \iff \Phi \models (a, \{k\})$ (en rajoutant les agents inconsistants a $\Phi_{\{k\}}$). Et étant donné que toutes ces dérivations tiennent pour All $k \in A$, ce qui est équivalent à : $\Phi \models (a, A)$.

Corollaire 4.4 La logique multi-agents est axiomatisable.

4.7 Exemple pour la logique multi-agents

Considérons la base de connaissances multi agents suivante :

$$\Phi = \{(\neg p \vee q, A), (\neg p \vee q, C), (\neg p \vee r, A), (\neg p \vee \neg r, B), (\neg q \vee r, All), (p, All), (q, A), (\neg q, D), (r, C)\}.$$

Notons que la base de connaissances classique Φ^* coïncide avec Φ^* de l'exemple du chapitre précédent. La distribution de possibilité multi-agents associée à Φ est :

$$\begin{aligned} \pi_{\Phi}(pqr) &= \overline{B} \cap \overline{D}; \\ \pi_{\Phi}(pq\neg r) &= \pi_{\Phi}(\neg pqr) = \pi_{\Phi}(\neg p q \neg r) = \pi_{\Phi}(\neg p \neg q r) = \pi_{\Phi}(\neg p \neg q \neg r) = \emptyset; \\ \pi_{\Phi}(p\neg qr) &= \overline{A} \cap \overline{B} \cap \overline{C}; \\ \pi_{\Phi}(p\neg q\neg r) &= \overline{A} \cap \overline{C}. \end{aligned}$$

Le calcul de la distribution de possibilité multi-agents détaillé est donné dans la table 4.1.

L'ensemble d'agents qui sont individuellement consistants est alors :

$$cons-s(\Phi) = \mathbf{\Pi}_{\Phi}(\top) = (\overline{A} \cap \overline{C}) \cup (\overline{A} \cap \overline{B} \cap \overline{C}) \cup (\overline{B} \cap \overline{D}) = (\overline{A} \cap \overline{C}) \cup (\overline{B} \cap \overline{D})$$

ω	$\pi(\neg p \vee q, A)$	$\pi(\neg p \vee q, C)$	$\pi(\neg p \vee r, A)$	$\pi(\neg p \vee \neg r, B)$	$\pi(\neg q \vee r, All)$	$\pi(p, All)$	$\pi(q, A)$	$\pi(\neg q, D)$	$\pi(r, C)$	π_{Φ}
pqr	<i>All</i>	<i>All</i>	<i>All</i>	\bar{B}	<i>All</i>	<i>All</i>	<i>All</i>	\bar{D}	<i>All</i>	$\bar{B} \cap \bar{D}$
$pq\neg r$	<i>All</i>	<i>All</i>	\bar{A}	<i>All</i>	\emptyset	<i>All</i>	<i>All</i>	\bar{D}	\bar{C}	\emptyset
$p\neg qr$	\bar{A}	\bar{C}	<i>All</i>	\bar{B}	<i>All</i>	<i>All</i>	\bar{A}	<i>All</i>	<i>All</i>	$\bar{A} \cap \bar{B} \cap \bar{C}$
$p\neg q\neg r$	\bar{A}	\bar{C}	\bar{A}	<i>All</i>	<i>All</i>	<i>All</i>	\bar{A}	<i>All</i>	\bar{C}	$\bar{A} \cap \bar{C}$
$\neg pqr$	<i>All</i>	<i>All</i>	<i>All</i>	<i>All</i>	<i>All</i>	\emptyset	<i>All</i>	\bar{D}	<i>All</i>	\emptyset
$\neg pq\neg r$	<i>All</i>	<i>All</i>	<i>All</i>	<i>All</i>	\emptyset	\emptyset	<i>All</i>	\bar{D}	\bar{C}	\emptyset
$\neg p\neg qr$	<i>All</i>	<i>All</i>	<i>All</i>	<i>All</i>	<i>All</i>	\emptyset	\bar{A}	<i>All</i>	<i>All</i>	\emptyset
$\neg p\neg q\neg r$	<i>All</i>	<i>All</i>	<i>All</i>	<i>All</i>	<i>All</i>	\emptyset	\bar{A}	<i>All</i>	\bar{C}	\emptyset

TABLE 4.1 – Calcul détaillé de la distribution de possibilité multi-agents de l'exemple de la section 4.7

et l'ensemble d'agents qui sont individuellement inconsistant est :

$$I = i - inc(\Phi) = (A \cup C) \cap (B \cup D)$$

Si nous cherchons à savoir qui croit r , nous trouvons que c'est $\mathbf{N}(r) = A \cup C \cup D$ étant donné que $\Phi \vdash (r, A)$, $\Phi \vdash (r, C)$, $\Phi \vdash (r, D)$ et par conséquent $\Phi \vdash (r, A \cup C \cup D)$.

Si nous voulons maintenant savoir qui croit r et est consistant, nous aurons alors à calculer $\mathbf{Bel}(r)$:

$$(A \cup C \cup D) \setminus i - inc(\Phi) = (A \cup C \cup D) \setminus ((A \cup C) \cap (B \cup D)) = (\bar{B} \cap C \cap \bar{D}) \cup (A \cap \bar{B} \cap \bar{C} \cap \bar{D})$$

De plus, nous notons que dans Φ , nous avons quatre symboles différents par rapport aux sous-ensembles d'agents, à savoir A, B, C, D . Cela induit une partition d'ensembles d'agents en $2^4 = 16$ sous-ensembles indiscernables d'agents qui partage la même opinion (qui correspond aux interprétations du langage induit par ces symboles). Pour chaque sous ensemble S de cette partition, nous pouvons calculer la partie propositionnelle $(\Phi_S)^*$ de la restriction Φ_S de Φ à S . Le résultat est donné ci-dessous. La base de connaissances propositionnelle marquée avec $*$ est consistante. Nous pouvons vérifier que $\Phi \vdash (r, A \cup C \cup D)$ et que $i - inc(\Phi) = (A \cup C) \cap (B \cup D)$ en utilisant ces bases propositionnelles.

$$\Phi_{A \cap B \cap C \cap D}^{\circ} = \{\neg p \vee q, \neg p \vee r, \neg q \vee r, p, q, \neg p \vee \neg r, r, \neg q\}$$

$$\Phi_{A \cap B \cap C \cap \bar{D}}^{\circ} = \{\neg p \vee q, \neg p \vee r, \neg q \vee r, p, q, \neg p \vee \neg r, r\}$$

$$\Phi_{A \cap B \cap \bar{C} \cap D}^{\circ} = \{\neg p \vee q, \neg p \vee r, \neg q \vee r, p, q, \neg p \vee \neg r, \neg q\}$$

$$\Phi_{A \cap B \cap \bar{C} \cap \bar{D}}^{\circ} = \{\neg p \vee q, \neg p \vee r, \neg q \vee r, p, q, \neg p \vee \neg r\}$$

$$\Phi_{A \cap \bar{B} \cap C \cap D}^{\circ} = \{\neg p \vee q, \neg p \vee r, \neg q \vee r, p, q, r, \neg q\}$$

$$\Phi_{A \cap \bar{B} \cap C \cap \bar{D}}^{\circ} = \{\neg p \vee q, \neg p \vee r, \neg q \vee r, p, q, r\} *$$

$$\Phi_{A \cap \bar{B} \cap \bar{C} \cap D}^{\circ} = \{\neg p \vee q, \neg p \vee r, \neg q \vee r, p, q, \neg q\}$$

$$\Phi_{A\bar{B}\bar{C}\bar{D}}^{\circ} = \{\neg p \vee q, \neg p \vee r, \neg q \vee r, p, q\} *$$

$$\Phi_{A\bar{B}B\bar{C}\bar{D}}^{\circ} = \{\neg p \vee q, \neg q \vee r, p, \neg p \vee \neg r, r, \neg q\}$$

$$\Phi_{A\bar{B}B\bar{C}D}^{\circ} = \{\neg p \vee q, \neg q \vee r, p, \neg p \vee \neg r, r\}$$

$$\Phi_{A\bar{B}B\bar{C}D}^{\circ} = \{\neg q \vee r, p, \neg p \vee \neg r, \neg q\} *$$

$$\Phi_{A\bar{B}B\bar{C}\bar{D}}^{\circ} = \{\neg q \vee r, p, \neg p \vee \neg r\} *$$

$$\Phi_{A\bar{B}\bar{B}C\bar{D}}^{\circ} = \{\neg p \vee q, \neg q \vee r, p, r, \neg q\}$$

$$\Phi_{A\bar{B}\bar{B}C\bar{D}}^{\circ} = \{\neg p \vee q, \neg q \vee r, p, r\} *$$

$$\Phi_{A\bar{B}\bar{B}C\bar{D}}^{\circ} = \{\neg q \vee r, p, \neg q\} *$$

$$\Phi_{A\bar{B}\bar{B}C\bar{D}}^{\circ} = \{\neg q \vee r, p\} *$$

4.8 Conclusion

Dans ce chapitre, nous nous sommes intéressés à présenter les différents fondements de base de la logique multi-agents. Nous avons vu qu'une formule multi-agents est une paire (a, A) où a est une formule propositionnelle et A un sous ensemble d'agents. Sémantiquement, elle permet d'exprimer qu'au moins tous les agents qui sont dans A croient que a est vraie.

Cette logique s'étend naturellement à la prise en compte de degrés d'incertitude avec lesquels les sous-ensembles d'agents tiennent pour vraies des propositions. Cette extension nommée : logique possibiliste multi-agents fera l'objet du chapitre suivant.

Chapitre 5

La logique possibiliste multi-agents

5.1 Introduction

Nous avons vu dans le chapitre précédent que la logique multi-agents permet de modéliser les croyances d'un ensemble d'agents dans un formalisme logique. Néanmoins, cette dernière présente des limites en ce qui concerne la représentation d'une information incertaine. Ainsi, nous proposons dans ce chapitre une extension de la logique multi-agents nommée logique possibiliste multi-agents.

La logique multi-agents peut naturellement être étendue à la logique possibiliste multi-agents où les propositions sont associées à la fois aux ensembles d'agents et aux degrés de certitudes. Cette extension va donc nous permettre de manipuler des formules exprimant qu'au moins tous les agents qui sont dans l'ensemble A croient que la formule a est vraie à au moins un degré α .

La logique possibiliste multi-agents inclut la logique possibiliste comme un cas particulier lorsque les croyances sont rapportées à un seul agent. Cependant, les sous-ensembles d'agents, éventuellement pondérés par des degrés de certitudes forment une structure partiellement ordonnée, contrairement à la logique possibiliste où les degrés de certitude sont totalement ordonnés.

Nous allons aborder dans ce chapitre les différentes notions inhérentes à la logique possibiliste multi-agents en commençant d'abord par donner sa syntaxe. Ensuite, la section 3 exhibe le treillis des sous ensembles flous. La section 4 présente le volet sémantique de la logique possibiliste multi-agents, tout en mentionnant la différence qui existe entre la sémantique possibiliste multi-agents normalisée et la sémantique possibiliste multi-agents partiellement individuellement normalisée. La section 5 expose l'axiomatique de la logique possibiliste multi-agents. Enfin, la section 6 est dédiée à présenter un exemple où les principaux concepts de la logique possibiliste multi-agents sont éclaircis.

5.2 Langage

Soit \mathcal{L} un langage de la logique propositionnelle. Soit All un ensemble fini contenant tous les agents. Un sous ensemble d'agents est noté par les lettres majuscules A, B, C ou par des lettres indexées A_i pour $i = 1, \dots, m$. Une formule en logique possibiliste multi-agents peut être construite à partir d'une formule classique en y rajoutant explicitement un ensemble flou d'agents.

Définition 5.1 Une formule possibiliste multi-agents est un couple (a, F) où a est une formule bien formée de la logique propositionnelle et F un ensemble flou d'agents appartenant à All .

Dans ce qui suit, nous utilisons l'ensemble flou $F = (\alpha/A)$ défini comme suit : $\mu_{\alpha/A}(k) = \alpha$ si $k \in A$, et $\mu_{\alpha/A}(k) = 0$ si $k \in \bar{A}$. De ce fait, nous allons restreindre le langage aux formules de la forme (α/A) .

(α/A) représente l'information : au moins tous les agents dans A croient a au moins à un degré α . En effet, la formule possibiliste multi-agents $(a, \alpha/A)$ est la traduction syntaxique de la contrainte sémantique $\mathbf{N}(a) \supseteq \alpha/A$ où \mathbf{N} est la mesure de nécessité multi-agents graduée.

Notons que nous ne considérons pas les formules de la forme $(a, 0/\emptyset)$ étant donné que $\forall a, \mathbf{N}(a) \supseteq 0/\emptyset$.

Une base de connaissances possibiliste multi-agents (ou aussi un ensemble de formules possibilistes multi-agents) est une conjonction finie de formules possibiliste multi-agents. Généralement, un ensemble de formules possibilistes multi-agents sera noté par Φ :

$$\Phi = (a_1, \alpha_1/A_1) \wedge \dots \wedge (a_n, \alpha_n/A_n) = \{(a_1, \alpha_1/A_1), \dots, (a_n, \alpha_n/A_n)\}$$

Exemple 5.1 Si nous reprenons l'exemple 4.1 du chapitre précédent en tenant compte cette fois ci que les différents spécialistes sont en première année de spécialité et ils doivent effectuer un stage pratique dans cet hôpital. Cela signifie qu'ils sont encore débutants dans leurs domaines et qu'ils ne sont pas très certains de plusieurs faits.

Remplaçons les anciens médecins spécialistes par les nouveaux :

- 03 médecins dermatologues : Nadir, Redouane et Nesrine ;
- 02 médecins pédiatres : Bouchra et Aya ;
- 02 médecins gastrologues : Héla et Chakibe.

Les croyances des différents nouveaux médecins spécialistes sur ces aliments sont comme suit :

- Les dermatologues sont tout à fait certains que manger de la viande grasse est mauvais pour la peau contrairement au poisson qui lui est bénéfique.
- Les dermatologues sont moyennement certains qu'un patient ne peut pas manger du poisson et des produits laitiers en même temps.
- Les pédiatres sont certains que les boissons gazeuses ne sont pas adaptées à l'alimentation des enfants.
- Les pédiatres pensent qu'il est relativement certain que les enfants peuvent manger : de la viande grasse, les légumes secs, les viennoiseries ainsi que les produits laitiers.
- Les médecins gastrologues croient à un très fort degré qu'un patient ne peut pas manger des légumes secs et les boissons gazeuses au même temps.

- Les médecins gastrologues sont certains que manger du poisson est bénéfique pour le patient.

En utilisant la même modélisation que celle du chapitre précédent :

Soient A , B et C les trois groupes des nouveaux spécialistes et soient les six variables Booléennes suivantes, relatives à :

- La viande grasse (a) ;
- Les légumes secs (b) ;
- Les viennoiseries (c) ;
- Les boissons gazeuses (d) ;
- Le poisson (e) ;
- Les produits laitiers (f).

La base de connaissance multi-agents associée à ce problème est donnée par la Figure 5.1 :

$$\Phi = \{(\neg a \vee e, 1/A), (e \vee \neg f, 0.5/A), (\neg e \vee f, 0.5/A), (\neg d, 1/B), (a \vee b \vee c \vee f, 0.7/B), (\neg b \vee d, 0.9/C), (b \vee \neg d, 0.9/C), (e, 1/C)\}.$$

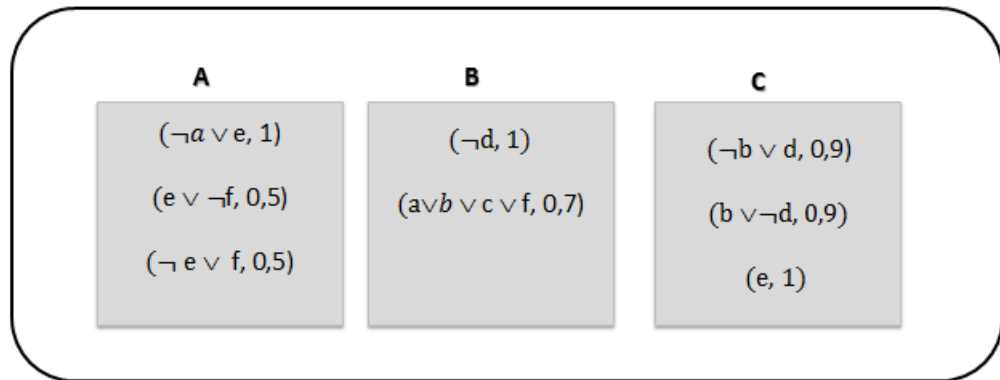


FIGURE 5.1 – La base de connaissances associée aux différents groupes des nouveaux médecins spécialistes

Pour toute formule possibiliste multi-agents $(a, \alpha/A)$, la fonction $EnsF$ est définie par $EnsF((a, \alpha/A)) = \alpha/A$.

La projection classique d'un ensemble de formules possibilistes multi-agents est l'ensemble des formules classiques, noté Φ^* , obtenu à partir de Φ en ignorant les sous-ensembles flous, c'est-à-dire que si $\Phi = \{(a_i, \alpha/A_i), i = 1, \dots, n\}$ alors $\Phi^* = \{a_i, i = 1, \dots, n\}$.

Exemple 5.2 Si $\Phi = \{(\neg a \vee b, 0.1/A), (a \vee b \vee c, 0.3/B), (\neg b \vee d, 0.5/B), (\neg d, 0.7/C)\}$ alors $\Phi^* = \{\neg a \vee b, a \vee b \vee c, \neg b \vee d, \neg d\}$.

La projection possibiliste d'un ensemble de formules possibilistes multi agents Φ^* est l'ensemble des formules possibilistes obtenu à partir de Φ en ignorant les sous ensembles d'agents c'est-à-dire que si $\Phi = \{(a_i, \alpha_i/A_i), i = 1, \dots, n\}$ alors $\Phi^* = \{(a_i, \alpha_i), i = 1, \dots, n\}$.

Exemple 5.3 Si $\Phi = \{(-a \vee b, 0.1/A), (a \vee b \vee c, 0.3/B), (-b \vee d, 0.5/B), (-d, 0.7/C)\}$ alors $\Phi^* = \{(-a \vee b, 0.1), (a \vee b \vee c, 0.3), (-b \vee d, 0.5), (-d, 0.7)\}$.

Le langage que nous venons de définir est restrictif par rapport à deux niveaux. Au premier niveau, seules les conjonctions de formules possibilistes multi-agents sont considérées, ni les disjonctions, ni les négations, ni les quantifications existentielles et universelles de formules possibiliste multi-agents ne sont prises en compte. Par exemple $(a, \alpha/A) \vee (b, \beta/B)$, ou $\exists x(a, \alpha/A)$ n'appartiennent pas à notre langage.

Au second niveau, le langage ne permet pas l'encapsulation des sous-ensembles flous tel que $((a, \alpha/A), \beta/B)$.

5.3 Le treillis des sous-ensembles flous

D'un point de vue théorique, une simple généralisation de la logique multi-agents est de faire une projection des distributions de possibilité et de nécessité non pas dans 2^{All} mais plus généralement dans $[0, 1]^{All}$.

Dans ce qui suit, nous considérons le treillis $\mathcal{TF} = [0, 1]^{All}$ où All est l'ensemble de tous les agents. \mathcal{TF} est équipé de l'intersection ensembliste floue \cap , de l'union ensembliste floue \cup et de la complémentation floue $\bar{\cdot}$ définies au moyen des operateurs : minimum, maximum et $1 - (\cdot)$ respectivement. L'ordre devient alors l'inclusion ensembliste floue définie par l'inégalité entre les fonctions d'appartenance.

Etant donné que \mathcal{TF} est partiellement ordonné comme \mathcal{T} , nous pouvons définir une restriction de la base de connaissance possibiliste multi-agents Φ en l'ensemble d'agents flou α/A comme suit :

$$\Phi_{\alpha/A} = \{(a_i, \alpha_i/A_i) | A_i \cap A \neq \emptyset \text{ et } \alpha_i \geq \alpha \text{ et } (a_i, \alpha_i/A_i) \in \Phi\}$$

5.4 Sémantique de la logique possibiliste multi-agents

5.4.1 Les distributions de possibilité multi-agents multi-agents-normalisées comme sémantique de la logique multi-agents

5.4.1.1 Définitions de base

En logique possibiliste multi-agents, la satisfaction d'une formule est définie en termes de distributions de possibilités multi-agents graduées dans le monde Ω .

Une distribution de possibilité multi-agents graduée, notée π , est une fonction de Ω vers $[0, 1]^{All}$. $\pi(\omega)$ représente les sous ensemble flous d'agents dans All qui trouvent ω possible à un certain degré.

Par convention, $\pi(\omega) = 1/All$ signifie que tous les agents trouvent que ω est *totalment* possible, alors que $\pi(\omega) = 0/All$ signifie que tous les agents trouvent que ω est

impossible. Dans le cas où il existe un élément ω de Ω tel que $\pi(\omega) = 1/All$ alors la distribution de possibilité multi-agents graduée π est dite multi-agents-normalisée. La propriété de la multi-agents normalisation reflète l'existence d'au moins un élément de Ω tel que tous les agents croient *totalemment* possible (en d'autre terme : elle reflète la consistance collective).

Exemple 5.4 *Considérons le même problème des opinions sur le cancer des poumons de l'exemple 4.3 du chapitre 4. Nous reprenons le problème avec les mêmes paramètres, les mêmes groupes d'agents et la même modélisation. Cependant, à la différence de l'exemple 4.3, la distribution de possibilités multi-agents est graduée.*

La Table 5.2 suivante donne les distributions de possibilités multi-agents graduée pour chaque interprétation ω . A titre d'exemple, l'interprétation ω_1 correspond à l'état du monde suivant : le patient a une tumeur aux poumons et est un homme et s'est exposé aux produits toxiques et c'est un jeune. L'ensemble flou d'agents qui croient que le patient risque d'attraper un cancer des poumons est l'ensemble des femmes avec un degré égal à 1 et l'ensemble des médecins avec un degré égal à 1 car les jeunes pensent que seuls les vieux risquent d'attraper un cancer des poumons.

ω_i	ω	$\pi(\omega)$
ω_1	$a \wedge b \wedge c \wedge d$	$1/B \cup 1/C$
ω_2	$a \wedge b \wedge c \wedge \neg d$	$1/All$
ω_3	$a \wedge b \wedge \neg c \wedge d$	$1/B$
ω_4	$a \wedge b \wedge \neg c \wedge \neg d$	$1/A \cup 1/B$
ω_5	$a \wedge \neg b \wedge c \wedge d$	$1/C$
ω_6	$a \wedge \neg b \wedge c \wedge \neg d$	$1/A \cup 1/C$
ω_7	$a \wedge \neg b \wedge \neg c \wedge d$	\emptyset
ω_8	$a \wedge \neg b \wedge \neg c \wedge \neg d$	$1/A$
ω_9	$\neg a \wedge b \wedge c \wedge d$	$1/B$
ω_{10}	$\neg a \wedge b \wedge c \wedge \neg d$	$1/A \cup 1/B$
ω_{11}	$\neg a \wedge b \wedge \neg c \wedge d$	$1/B$
ω_{12}	$\neg a \wedge b \wedge \neg c \wedge \neg d$	$1/A \cup 1/B$
ω_{13}	$\neg a \wedge \neg b \wedge c \wedge d$	\emptyset
ω_{14}	$\neg a \wedge \neg b \wedge c \wedge \neg d$	$1/A$
ω_{15}	$\neg a \wedge \neg b \wedge \neg c \wedge d$	\emptyset
ω_{16}	$\neg a \wedge \neg b \wedge \neg c \wedge \neg d$	$1/A$

FIGURE 5.2 – Distribution de possibilités multi-agents associée à l'exemple 5.4

La distribution de possibilité multi-agents graduée π donne naissance à la notion de la mesure de possibilité multi-agents graduée notée Π . C'est une fonction qui associe à chaque formule a un sous ensemble flou d'agents. Elle est définie par :

$$\Pi : \mathcal{L} \rightarrow [0, 1]^{All}$$

$$\mathbf{\Pi}(\omega) = \bigcup_{\omega \models a} \pi(\omega)$$

où l'union \bigcup est l'union d'ensembles flous, définie classiquement à l'aide du maximum. La mesure de possibilité graduée $\mathbf{\Pi}(a)$ est l'ensemble flou d'agents qui trouvent que a est possible à un certain degré. Le degré d'appartenance à $\mathbf{\Pi}(\omega)$ d'un agent reflète ce qu'on sait de son niveau de certitude dans sa croyance en a . Par conséquent :

- $\mathbf{\Pi}(a) = 0/All$ signifie que tous les agents trouvent que a est *complètement* impossible i.e. aucun ensemble d'agents ne trouve a possible.
- $\mathbf{\Pi}(a) = 1/All$ signifie que tous les agents trouvent que a est *totalememt* possible.
- $\mathbf{\Pi}(a) = \mathbf{\Pi}(\neg a) = 1/A$ représente le cas de l'ignorance **individuelle** (le cas de l'absence totale de l'information). En effet, l'ensemble d'agents A trouve que a et $\neg a$ sont toutes les deux *totalememt* possibles.
- $\mathbf{\Pi}(a) = \mathbf{\Pi}(\neg a) = 1/All$ représente le cas de l'ignorance **globale** (le cas de l'absence totale de l'information pour tous les agents). En effet, tous les agents trouvent que a et $\neg a$ sont toutes les deux *totalememt* possibles.
- $\mathbf{\Pi}(a) \cup \mathbf{\Pi}(\neg a) = 1/All$ signifie que les ensembles d'agents doivent trouver que de deux énoncés contraires l'un au moins est *complètement* possible.

Exemple 5.5 *Considérons à nouveau l'exemple 5.4.*

L'ensemble flou d'agents qui trouvent que la tumeur aux poumons est une cause possible du cancer des poumons est donné par :

$$\mathbf{\Pi}(a) = \bigcup \{1/B \cup 1/C, 1/All, 1/B, 1/A \cup 1/B, 1/C, 1/A \cup 1/C, \emptyset, 1/A\}$$

donc $\mathbf{\Pi}(a) = 1/All$,

Ce qui signifie que tous les agents trouvent que la tumeur aux poumons est une cause possible du cancer des poumons avec un le degré maximal égale à 1.

La mesure de possibilité multi-agents graduée satisfait les axiomes suivants :

- $\mathbf{\Pi}(\perp) = 0/All$
- $\mathbf{\Pi}(\top) = 1/All$
- $\forall a, \forall b \quad \mathbf{\Pi}(a \vee b) = \mathbf{\Pi}(a) \cup \mathbf{\Pi}(b)$

Si la mesure de possibilité multi-agents graduée d'une disjonction peut être directement calculée à partir des mesure de possibilités multi-agents graduée des sous formules, il n'en est pas de même pour les conjonctions. Pour ces dernières nous avons seulement :

$$\mathbf{\Pi}(a \wedge b) \subseteq \mathbf{\Pi}(a) \cap \mathbf{\Pi}(b)$$

Remarque 5.1 1. *Dans le cas général, l'égalité n'est pas vérifiée. Si nous considérons par exemple le cas de l'ignorance globale totale, $\mathbf{\Pi}(a) \cap \mathbf{\Pi}(\neg a) = 1/All \cap 1/All = 1/All$ alors que $\mathbf{\Pi}(a \wedge \neg a) = \mathbf{\Pi}(\perp) = 0/All$.*

2. *L'inégalité n'est pas aussi vérifiée pour la négation. En effet, si $\mathbf{\Pi}(a) = 1/All$, $\mathbf{\Pi}(\neg a)$ peut prendre n'importe quelle valeur de $[0, 1]^{All}$.*

De façons duale, la mesure de nécessité multi-agents graduée induite par π , notée \mathbf{N} est définie par :

$$\mathbf{N} : \mathcal{L} \rightarrow [0, 1]^{All}$$

$$\begin{aligned} \mathbf{N}(a) &= \bigcap_{\omega \models \neg a} \{\overline{\pi(\omega)}\} \\ &= \overline{\mathbf{\Pi}(\neg a)} \end{aligned}$$

où la complémentarité floue est définie par $\overline{F}(k) = 1 - F(k)$ et où l'intersection floue basée sur le minimum est utilisée. La mesure de nécessité (ou de certitude) multi-agents graduée définit l'ensemble flou des agents qui sont certains, à un certain degré non nul que a est vraie. Ainsi :

- $\mathbf{N}(a) = 1/All$ signifie que tous les agents sont *complètement* certains que a est vraie.
- $\mathbf{N}(a) = 0/All$ signifie qu'aucun ensemble d'agents n'est *complètement* certain que a est vraie.
- $\mathbf{N}(a) = \mathbf{N}(\neg a) = 0/All$ représente l'état de l'ignorance totale.
- Nous avons toujours $\mathbf{N}(a) \cap \mathbf{N}(\neg a) = 0/All$ signifie qu'aucun ensemble d'agent ne peut croire que les deux propositions a et $\neg a$ soient vraies en même temps.

La mesure de nécessité multi-agents graduée satisfait les axiomes suivants :

- $\mathbf{N}(\perp) = 0/All$
- $\mathbf{N}(\top) = 1/All$
- $\forall a, \forall b \quad \mathbf{N}(a \wedge b) = \mathbf{\Pi}(a) \cap \mathbf{\Pi}(b)$

La mesure de nécessité multi-agents graduée de la disjonction est donnée par l'inclusion suivante :

$$\mathbf{N}(a \vee b) \supseteq \mathbf{N}(a) \cup \mathbf{N}(b)$$

Il est à noter que cette inclusion n'est en générale pas une égalité. Il est de même pour la négation.

De plus, les propriétés suivantes sont vérifiées :

- $\forall a, \mathbf{N}(a) \subseteq \mathbf{\Pi}(a)$.
- $\forall a, \mathbf{N}(a) \neq 0/All \Rightarrow \mathbf{\Pi}(a) = 1/A$ (ou bien d'une façon équivalente : $\forall a, \mathbf{\Pi}(a) \neq 0/All \Rightarrow \mathbf{N}(a) = 1/A$).

5.4.1.2 Satisfaction d'une base de connaissance multi-agents possibiliste par la distribution de possibilité multi-agents graduée

Nous définissons dans cette sous section la relation de satisfiabilité entre une distribution de possibilité multi-agents graduée et une formule possibiliste multi-agents et ensuite entre une distribution de possibilité multi-agents graduée et un ensemble de formules possibilistes multi-agents . Soit π une distribution de possibilité multi-agents graduée sur \mathcal{L} et \mathbf{N} la mesure de nécessité multi-agents graduée induite par π .

$$\pi \text{ satisfait } (a, \alpha/A) \text{ si et seulement si } \mathbf{N}(a) \supseteq \alpha/A$$

De la même façon, si ϕ_1, \dots, ϕ_n sont des formules possibilistes multi-agents,

$$\pi \text{ satisfait } \phi_1 \wedge \dots \wedge \phi_n \text{ si et seulement si } \forall i = 1..n, \pi \text{ satisfait } \phi_i$$

Notation 5.1 *Souvent, et quand il n'y aura pas d'ambiguïté, nous noterons $\pi \models \Phi$ pour " π satisfait Φ ".*

Remarque 5.2 *Il est évident que nous utilisons alors une même notation pour deux concepts différents : $\omega \models a$ signifie que l'interprétation ω satisfait la formule classique a , alors que $\pi \models \Phi$ signifie que la distribution de possibilité graduée π satisfait l'ensemble des formules possibiliste multi-agents Φ .*

Exemple 5.6 *Soit $\Phi = \{(a, 0.9/A), (\neg a \vee b, 0.3/B)\}$*

$$\begin{aligned} \pi \models \Phi &\Leftrightarrow \mathbf{N}(a) \supseteq 0.9/A \text{ et } \mathbf{N}(\neg a \vee b) \supseteq 0.3/B \\ &\Leftrightarrow \cap \{\overline{\pi(\omega)} \mid \omega \models \neg a\} \supseteq 0.9/A \text{ et } \cap \{\overline{\pi(\omega)} \mid \omega \models a \wedge \neg b\} \supseteq 0.3/B \end{aligned}$$

Soient $ab, a\neg b, \neg ab, \neg a\neg b$ les quatre interprétations du langage propositionnel engendrées par $\{a, b\}$. Nous aurons alors :

$$\begin{aligned} \pi \models \Phi &\Leftrightarrow \pi(\neg ab) \subseteq (0.1/A \cup 1/\overline{A}), \pi(\neg a\neg b) \subseteq (0.1/A \cup 1/\overline{A}), \\ &\pi(a\neg b) \subseteq (0.7/B \cup 1/\overline{B}), \pi(ab) \subseteq 1/All \end{aligned}$$

Si nous cherchons une distribution de possibilité multi-agents normalisée satisfaisant Φ alors $\pi(ab) = 1/All$.

$\mathcal{PG}(\Phi)$ est l'ensemble des distributions de possibilité multi-agents graduées satisfaisant Φ :

$$\mathcal{PG}(\Phi) = \{\pi, \pi \text{ satisfait } \Phi\}$$

Il constitue l'ensemble des modèles possibiliste multi-agents de Φ .

Φ est multi-agents-consistante si $\mathcal{PG}(\Phi) = 1/All$.

Φ est multi-agents-inconsistante si $\mathcal{PG}(\Phi) = 0/\emptyset$.

Φ est valide si toute distribution de possibilité multi-agents graduée sur Ω satisfait Φ .

Le couple formé de l'ensemble des distributions multi-agents-normalisées graduée sur Ω et de la relation de satisfiabilité constitue la sémantique possibiliste multi-agents normalisée pour la logique possibiliste multi-agents.

5.4.1.3 Conséquence logique

$(b, \beta/B)$ est une conséquence logique de Φ (notée $\Phi \models (b, \beta/B)$) si et seulement si toute distribution de possibilité multi-agents graduée satisfaisant Φ satisfait aussi $(b, \beta/B)$. Formellement :

$$\Phi \models (b, \beta/B) \text{ si et seulement si } \forall \pi, (\pi \models \Phi) \Rightarrow (\pi \models (b, \beta/B))$$

Σ est sémantiquement équivalente à Φ (noté $\Sigma \equiv \Phi$) si et seulement si $\mathcal{PG}(\Phi) = \mathcal{PG}(\Sigma)$.

Exemple 5.7 *De l'exemple précédent, nous vérifions immédiatement que $\Phi \models (b, 0.3/A \cap B)$. En effet, une distribution de possibilité multi-agents graduée π qui satisfait Φ est telle que $\pi(a\neg b) \subseteq (0.7/B \cup 1/\overline{B})$ et $\pi(\neg a\neg b) \subseteq (0.1/A \cup 1/\overline{A})$, et vérifie $\mathbf{N}(b) = \cap \{\overline{\pi(a\neg b)}, \overline{\pi(\neg a\neg b)}\} \supseteq (0.3/\overline{B} \cap 0.9/\overline{A})$ donc $\Phi \models (b, 0.3/A \cap B)$ car $(0.3/\overline{B} \cap 0.9/\overline{A}) \equiv 0.3/A \cap B$.*

Les propriétés immédiates suivantes sont vérifiées :

- $(a, \alpha/A) \models (a, \beta/B) \quad \forall \beta/B \subseteq \alpha/A$ (preuve immédiate).
- $\models (a, 1/All)$ si et seulement si a est une tautologie.

Preuve 5.1 (\Leftarrow) :

Si a est une tautologie alors : $\mathbf{N}(a) = 1/All$. Donc toute distribution de possibilité satisfait $(a, 1/All)$.

(\Rightarrow) :

Si a n'est pas une tautologie alors il existe une interprétation ω qui falsifie a , donc si π est la distribution de possibilité définie par $\pi(\omega) = 1/All$ et $\forall \omega^ \neq \omega$ $\pi(\omega^*) = 0/\emptyset$ on a $\mathbf{N}(a) = 0/\emptyset$ et π ne satisfait donc pas $(a, 1/All)$, $(a, 1/All)$ n'est donc pas valide.*

- $(a, \alpha/A) \wedge (a \rightarrow b, \beta/B) \models (b, \alpha/A \cap \beta/B)$. Cette forme du modus ponens ensembliste graduel est un cas particulier de la résolution possibiliste multi-agents.

5.4.1.4 Le problème de déduction

Soit Φ un ensemble de formules possibilistes multi-agents. Les formules de Φ étant considérées comme des hypothèses. Soit b une formule quelconque ; il s'agit de chercher le plus grand ensemble d'agents qui croient a et avec le plus haut degré déductible de la connaissance Φ (i.e. $(b, \beta/B)$ est une conséquence logique de Φ). Ce plus grand ensemble d'agents est défini par :

$$EnsF(b, \Phi) = \cup \{ \beta/B \in [0, 1]^{All}, \Phi \models (b, \beta/B) \}$$

Proposition 5.1

$$EnsF(b, \Phi) = \cap \{ \mathbf{N}(b), \pi \in \mathcal{PG}(\Phi) \}$$

Preuve 5.2

$$\begin{aligned} EnsF(b, \Phi) &= \cup \{ B \in [0, 1]^{All}, \Phi \models (b, \beta/B) \} \\ &= \cup \{ B \in [0, 1]^{All} \mid \forall \pi \in \mathcal{PG}(\Phi), \mathbf{N}(b) \supseteq \beta/B \} \\ &= \cup \{ B \in [0, 1]^{All} \mid \cap \{ \mathbf{N}(b), \pi \in \mathcal{PG}(\Phi) \} \supseteq \beta/B \} \\ &= \cap \{ \mathbf{N}(b), \pi \in \mathcal{PG}(\Phi) \} \end{aligned}$$

Exemple 5.8 *Dans l'exemple précédent, il n'existe pas un sous ensemble flou \mathcal{EF} tel que : $\mathcal{EF} \supset 0.3/A \cap B$ tel que $\Phi \models (b, \mathcal{EF})$. De ce fait, $EnsF(b, \Phi) = 0.3/A \cap B$.*

5.4.1.5 Le modèle possibiliste multi-agents le moins spécifique

La formule $(a, \alpha/A)$ représente l'information 'a est certaine au moins au degré α pour au moins tous les agents qui sont dans A '. Donc l'ensemble d'agents dans A trouvent n'importe quelle interprétation de a complètement possible. En outre, les autres agents dans \bar{A} sont libres de trouver l'interprétation de a complètement possible. Ainsi, l'ensemble maximal d'agents qui trouvent n'importe quelle interprétation de a complètement possible est $A \cup \bar{A} = All$. En plus, l'ensemble maximal d'agents qui trouvent n'importe quelle interprétation de $\neg a$ possible sont les agents dans A à au moins au degré $1 - \alpha$, et les agents dans \bar{A} à au moins au degré 1. Cela nous mène alors à la représentation sémantique de la formule $(a, \alpha/A)$ suivante :

$$\pi_{\{(a,\alpha/A)\}}(\omega) = \begin{cases} 1/All & \text{si } \omega \models a \\ \{(1-\alpha)/A \cup 1/\bar{A}\} & \text{si } \omega \models \neg a. \end{cases}$$

Plus généralement, la distribution multi-agents graduée sémantiquement associée à l'ensemble des formules possibilistes multi-agents Φ est l'application π de Ω vers \mathcal{L} suivante :

$$\pi_{\Phi}(\omega) = \begin{cases} 1/All & \text{si } \forall (a_i, \alpha_i/A_i) \in \Phi, \omega \models a_i \\ \bigcap \{(1-\alpha_i)/A_i \cup 1/\bar{A}_i \mid (a_i, \alpha_i/A_i) \in \Phi, \omega \models \neg a_i\} & \text{sinon.} \end{cases}$$

Par conséquent, la sémantique de la base Φ est maintenant exprimée en terme d'ensembles flous d'agents, pour laquelle nous aurons :

$$\mathbf{N}_{\Phi}(a_i) \supseteq \alpha_i/A_i \text{ pour } i = 1, \dots, n.$$

Proposition 5.2 *Pour toute distribution de possibilité multi-agents graduée π , π satisfait Φ (noté $\pi \models \Phi$) si et seulement si $\pi \subseteq \pi_{\Phi}$ i.e. $\forall \omega, \pi(\omega) \subseteq \pi_{\Phi}(\omega)$.*

Preuve 5.3

$$\begin{aligned} \pi \models \Phi &\Leftrightarrow \forall i = 1, \dots, n, \pi \models (a_i, \alpha_i/A_i) \\ &\Leftrightarrow \forall i = 1, \dots, n, \mathbf{N}(a_i) \supseteq \alpha_i/A_i \\ &\Leftrightarrow \forall i = 1, \dots, n, \bigcap \{\pi(\omega) \mid \omega \models \neg a_i\} \supseteq \alpha_i/A_i \\ &\Leftrightarrow \forall i = 1, \dots, n, \forall \omega \models \neg a_i, \pi(\omega) \subseteq (1-\alpha_i)/A_i \cup 1/\bar{A}_i \\ &\Leftrightarrow \pi(\omega) \subseteq \bigcap \{(1-\alpha_i)/A_i \cup 1/\bar{A}_i, \omega \models \neg a_i, i = 1, \dots, n\} \\ &\Leftrightarrow \pi(\omega) \subseteq \pi_{\Phi}(\omega) \end{aligned}$$

Ainsi, $(b, \beta/B)$ est une conséquence logique de Φ si et seulement si $\pi_{\Phi}(\omega)$ est inclus dans $\pi_{(b,\beta/B)}(\omega)$. Formellement :

$$\Phi \models (b, \beta/B) \text{ si et seulement si } \forall \omega, \pi_{\Phi}(\omega) \subseteq \pi_{(b,\beta/B)}(\omega)$$

5.4.1.6 Déduction en logique possibiliste multi-agents

Corollaire 5.1

$$\Phi \models (b, \beta/B) \text{ si et seulement si } \pi_{\Phi} \models (b, \beta/B)$$

En d'autres termes, $\text{Ens}F(b, \Phi) = \mathbf{N}_{\Phi}(b)$, où $\mathbf{N}_{\Phi}(b)$ est la mesure de nécessité multi-agents graduée induite par π_{Φ} .

Proposition 5.3 *Soient Φ, Σ deux ensembles de formules possibilistes multi-agents et π_{Φ} la distribution de possibilité multi-agents graduée la moins spécifique parmi celles satisfaisant Φ . Alors*

$$\Phi \models \Sigma \text{ si et seulement si } \pi_{\Phi} \text{ satisfait } \Sigma$$

Preuve 5.4

$$\begin{aligned} \Phi \models \Sigma &\Leftrightarrow \mathcal{PG}(\Phi) \subseteq \mathcal{PG}(\Sigma) \\ &\Leftrightarrow \forall \pi (\pi \subseteq \pi_{\Phi} \Rightarrow \pi \subseteq \pi_{\Sigma}) \\ &\Leftrightarrow \pi_{\Phi} \subseteq \pi_{\Sigma} \\ &\Leftrightarrow \pi_{\Phi} \text{ satisfait } \Sigma. \end{aligned}$$

Proposition 5.4 *Soit Φ un ensemble de formules de la logique possibiliste multi-agents, multi-agents-consistant, et b une formule quelconque. Alors $\Phi \models (b, \beta/B)$ si et seulement si $\Phi_{\beta/B} \models (b, \beta/B)$.*

Preuve 5.5 (\Leftarrow) : est immédiate car $\Phi_{\beta/B}$ est inclus dans Φ .

(\Rightarrow) :

Soit $\Phi = \{(a_i, \alpha_i/A_i), i = 1, \dots, n\}$.

Soit π_Φ la distribution de possibilités multi-agents graduée la moins spécifique parmi celles satisfaisant Φ . Alors : $\forall \omega, \pi_\Phi = \bigcap \{(1 - \alpha_i)/A_i \cup 1/\overline{A_i} \mid \omega \models \neg a_i\}$.

Soit $\pi_{\Phi_{\beta/B}}$ la distribution de possibilités multi-agents graduée la moins spécifique parmi celles satisfaisant $\Phi_{\beta/B}$. Alors : $\forall \omega, \pi_{\Phi_{\beta/B}} = \bigcap \{(1 - \alpha_i)/A_i \cup 1/\overline{A_i} \mid \omega \models \neg a_i, \alpha_i/A_i \supseteq \beta/B\}$.

$$\begin{aligned}
\Phi \models (b, \beta/B) &\Rightarrow \forall \omega \models \neg b, \bigcap \{(1 - \alpha_i)/A_i \cup 1/\overline{A_i} \mid \omega \models \neg a_i\} \subseteq (1 - \beta)/B \cup 1/\overline{B} \\
&\Rightarrow \forall \omega \models \neg b, \bigcap [\bigcap \{(1 - \alpha_i)/A_i \cup 1/\overline{A_i} \mid \omega \models \neg a_i, \alpha_i/A_i \supseteq \beta/B\}, \\
&\quad \bigcap \{(1 - \alpha_i)/A_i \cup 1/\overline{A_i} \mid \omega \models \neg a_i, \alpha_i/A_i \subset \beta/B\}] \subseteq (1 - \beta)/B \cup 1/\overline{B} \\
&\Rightarrow \forall \omega \models \neg b, \bigcap \{(1 - \alpha_i)/A_i \cup 1/\overline{A_i} \mid \omega \models \neg a_i, \alpha_i/A_i \supseteq \beta/B\} \subseteq (1 - \beta)/B \cup 1/\overline{B} \\
&\quad (\text{car } \bigcap \{(1 - \alpha_i)/A_i \cup 1/\overline{A_i} \mid \omega \models \neg a_i, \alpha_i/A_i \subset \beta/B\} \supset (1 - \beta)/B \cup 1/\overline{B}) \\
&\Rightarrow \forall \omega \models \neg b, \pi_{\Phi_{\beta/B}}(\omega) \subseteq (1 - \beta)/B \cup 1/\overline{B} \\
&\Rightarrow \omega \models \neg b, \pi_{\Phi_{\beta/B}}(\omega) \subseteq \pi_{(b, \beta/B)}(\omega) \\
&\Rightarrow \Phi_{\beta/B} \models (b, \beta/B).
\end{aligned}$$

Ce résultat exprime que pour prouver qu'une formule est (au moins) crue par un ensemble d'agents B à un degré au moins égal à β , seules les formules de la base de connaissances qui sont elles-mêmes au moins crues par cette ensemble d'agents avec un degré au moins égal à β sont utiles.

5.4.2 Les distributions de possibilité multi-agents graduée partiellement individuellement normalisées comme sémantique de la logique possibiliste multi-agents

5.4.2.1 La sémantique partiellement individuellement normalisée de la logique possibiliste multi-agents

Dans cette sous-section, nous proposons **une deuxième** sémantique naturelle pour la logique possibiliste multi-agents ; plus souple et qui permet de traiter les inconsistances partielles collectives.

L'impossibilité de trouver une distribution de possibilité multi-agents graduée satisfaisant Φ est due en fait à la contrainte de la multi-agents-normalisation : $\exists \omega, \pi(\omega) = 1/All$.

Considérons maintenant \mathcal{PG}_{GIN} l'ensemble des distributions de possibilité multi-agents graduée multi-agents-normalisé ou non sur Ω . Une distribution de possibilité multi-agents graduée (multi-agents-normalisé ou non) est simplement une application de Ω dans $[0, 1]^{All}$. Les mesures de possibilité et de nécessité multi-agents graduées sont

définies comme dans le cas du possibiliste multi-agents-normalisé.

Soit Φ une conjonction de formules possibiliste multi-agents. \mathcal{PG}_{GIN_Φ} est défini comme l'ensemble des distributions de possibilité multi-agents graduées, possibilistes multi-agents normalisées ou non, satisfaisant Φ .

L'ensemble des distributions de possibilité multi-agents graduée possibilistes multi-agents normalisées ou non, muni de la relation de satisfaisabilité définie comme dans le cas normalisé, est appelé sémantique partiellement-individuellement-normalisée pour la logique possibiliste multi-agents.

Exemple 5.9 Reprenons l'exemple 5.4 sur les opinions sur la détection du cancer des poumons. Rajoutons maintenant un autre groupe qui est le groupe des hommes. Ce dernier se compose de 03 hommes à savoir : Djamel, Rachid et Nadir.

La Table 5.3 donne les distributions de possibilités multi-agents graduées pour chaque interprétation ω . A titre d'exemple, l'interprétation ω_1 qui correspond à l'état du monde suivant : le patient a une tumeur aux poumons et est un homme et s'est exposé aux produits toxiques et c'est un jeune. L'ensemble d'agents qui croient que le patient risque d'attraper un cancer des poumons est : L'ensemble des jeunes avec un degré de 0.3, l'ensemble des femmes avec un degré de 0.9 et des médecins avec un degré de 0.8.

Les distributions de possibilités multi-agents graduées données par cette table ne sont pas normalisées. En effet : $\# \omega, \pi(\omega) = 1/All$.

Le nombre $GIN(\pi) = \cap \{\overline{\pi(\omega)}, \omega \in \Omega\}$ est appelé l'ensemble de la graduelle individuelle normalisation de π , et le nombre $\cap \{GIN(\pi), \pi \in \mathcal{PG}_{GIN_\Phi}\}$ est appelé la graduelle individuelle inconsistance de Φ . Il est noté $g - i - inc(\Phi)$.

$GIN(\pi) = \cap \{\overline{\pi(\omega)}, \omega \in \Omega\} = \cap \{\overline{\pi(\omega)}, \omega \neq \perp\} = \mathbf{N}(\perp)$ et par conséquent $g - i - inc(\Phi) = \cap \{\mathbf{N}(\perp), \pi \in \mathcal{PG}_{GIN_\Phi}\}$, ce qui justifie l'appellation "la graduelle individuelle inconsistance", par abus de langage pour "la nécessité de la contradiction".

Dans cette sémantique, certaines des propriétés de la sémantique possibiliste multi-agents-normalisée restent vérifiées, d'autre non.

Propriété 5.1 1. $\forall a \in \mathcal{L}, \forall b \in \mathcal{L}, \mathbf{\Pi}(a \vee b) = \mathbf{\Pi}(a) \cup \mathbf{\Pi}(b)$ ou \cup est l'union floue basée maximum.

2. $\forall a \in \mathcal{L}, \mathbf{N}(a) = \overline{\mathbf{\Pi}(\neg a)}$ où la complémentation est définie classiquement par $\overline{F}(k) = 1 - F(k)$.

3. $\forall a \in \mathcal{L}, \forall b \in \mathcal{L}, \mathbf{N}(a \wedge b) = \mathbf{N}(a) \cap \mathbf{N}(b)$ ou \cap est l'intersection floue basée minimum.

4. $\forall a, b \in \mathcal{L},$ si $a \models b$ alors $\mathbf{\Pi}(b) \supseteq \mathbf{\Pi}(a)$ et $\mathbf{N}(b) \supseteq \mathbf{N}(a)$.

5. $\mathbf{\Pi}(\perp) = 0/\emptyset$.

Par contre, $\mathbf{N}(\perp) = 0/\emptyset$ n'est pas nécessairement vérifiée.

ω_i	ω	$\pi(\omega)$
ω_1	$a \wedge b \wedge c \wedge d$	$0,3/A \cup 0,9/B \cup 0,8/C$
ω_2	$a \wedge b \wedge c \wedge \neg d$	$0,7/A \cup 0,9/B \cup 0,8/C$
ω_3	$a \wedge b \wedge \neg c \wedge d$	$0,3/A \cup 0,9/B \cup 0,2/C$
ω_4	$a \wedge b \wedge \neg c \wedge \neg d$	$0,7/A \cup 0,9/B \cup 0,2/C$
ω_5	$a \wedge \neg b \wedge c \wedge d$	$0,3/A \cup 0,1/B \cup 0,8/C \cup 1/D$
ω_6	$a \wedge \neg b \wedge c \wedge \neg d$	$0,7/A \cup 0,1/B \cup 0,8/C \cup 1/D$
ω_7	$a \wedge \neg b \wedge \neg c \wedge d$	$0,3/A \cup 0,1/B \cup 0,2/C \cup 1/D$
ω_8	$a \wedge \neg b \wedge \neg c \wedge \neg d$	$0,7/A \cup 0,1/B \cup 0,2/C \cup 1/D$
ω_9	$\neg a \wedge b \wedge c \wedge d$	$0,3/A \cup 0,9/B \cup 0,2/C$
ω_{10}	$\neg a \wedge b \wedge c \wedge \neg d$	$0,7/A \cup 0,9/B \cup 0,2/C$
ω_{11}	$\neg a \wedge b \wedge \neg c \wedge d$	$0,3/A \cup 0,9/B \cup 0,2/C$
ω_{12}	$\neg a \wedge b \wedge \neg c \wedge \neg d$	$0,7/A \cup 0,9/B \cup 0,2/C$
ω_{13}	$\neg a \wedge \neg b \wedge c \wedge d$	$0,3/A \cup 0,1/B \cup 0,2/C \cup 1/D$
ω_{14}	$\neg a \wedge \neg b \wedge c \wedge \neg d$	$0,7/A \cup 0,1/B \cup 0,2/C \cup 1/D$
ω_{15}	$\neg a \wedge \neg b \wedge \neg c \wedge d$	$0,3/A \cup 0,1/B \cup 0,2/C \cup 1/D$
ω_{16}	$\neg a \wedge \neg b \wedge \neg c \wedge \neg d$	$0,7/A \cup 0,1/B \cup 0,2/C \cup 1/D$

FIGURE 5.3 – Distributions de possibilités multi-agents graduée non normalisées de l'exemple 5.9

6. $N(\top) = 1/All$.

Par contre, $\Pi(\top) = 1/All$ n'est pas nécessairement vérifiée.

7. $\forall a \in \mathcal{L}, N(a) \supseteq N(\perp)$.

8. $\forall a \in \mathcal{L}, N(a) \cap N(\neg a) = N(\perp)$.

9. $\forall a \in \mathcal{L}, \Pi(a) \cup \Pi(\neg a) = \Pi(\top)$.

10. $N(a) \subseteq \Pi(a)$, ni $N(a) \supset 0/\emptyset \Rightarrow \Pi(a) = 1/All$ n'est pas nécessairement vérifiée.

5.4.2.2 Conséquence logique en logique possibiliste multi-agents partiellement-individuellement normalisée

$(b, \beta/B)$ est une conséquence logique de Φ dans la sémantique partiellement individuellement normalisée (notée $(b, \beta/B) \models_{GIN} \Phi$) si et seulement si \mathcal{PG}_{GIN_Φ} est inclus dans $\mathcal{PG}_{GIN(b, \beta/B)}$.

Exemple 5.10 $\Phi = \{(a, 0.7/A), (\neg a \vee b, 0.5/B), (c, 0.3/A), (\neg c, 0.2/C)\}$

$\Phi \models_{IN} (b, 0.5/A \cap B)$ mais pas $\Phi \models_{IN} (b, 0.8/A)$.

$\Phi \models_{IN} (\perp, 0.2/A \cap C)$ mais pas $\Phi \models_{IN} (\perp, 0.3/A)$.

$\Phi \models_{IN} (\phi, 0.2/A \cap C)$ pour n'importe quelle formule ϕ .

5.4.2.3 La distribution de possibilité multi-agents graduée, multi-agents normalisée ou non, la moins spécifique parmi les distributions les moins partiellement individuellement-normalisées

La proposition 5.2 est généralisée à la sémantique partiellement individuellement normalisée :

Proposition 5.5 *Pour toute distribution de possibilité multi-agents graduée, multi-agents normalisée ou non π_{GIN} , π_{GIN} satisfait Φ si et seulement si $\pi_{GIN} \subseteq \pi_{GIN\Phi}$ i.e. $\forall \omega, \pi_{GIN}(\omega) \subseteq \pi_{GIN\Phi}(\omega)$.*

Preuve 5.6

$$\begin{aligned}
\pi_{GIN} \models \Phi &\Leftrightarrow \forall i = 1, \dots, n \quad \pi_{GIN} \models (a_i, \alpha_i/A_i) \\
&\Leftrightarrow \forall i = 1, \dots, n \quad \mathbf{N}(a_i) \supseteq \alpha_i/A_i \\
&\Leftrightarrow \forall i = 1, \dots, n \quad \bigcap \{ \overline{\pi(\omega)} \mid \omega \models \neg a_i \} \supseteq \alpha_i/A_i \\
&\Leftrightarrow \forall i = 1, \dots, n \quad \forall \omega \models \neg a_i \quad \pi(\omega) \subseteq (1 - \alpha_i)/A_i \cup 1/\overline{A_i} \\
&\Leftrightarrow \pi_{GIN}(\omega) \subseteq \bigcap \{ (1 - \alpha_i)/A_i \cup 1/\overline{A_i}, \omega \models \neg a_i, i = 1, \dots, n \} \\
&\Leftrightarrow \pi_{GIN}(\omega) \subseteq \pi_{GIN\Phi}(\omega)
\end{aligned}$$

Ainsi, $(b, \beta/B)$ est une conséquence logique de Φ si et seulement si $\pi_{GIN\Phi}(\omega)$ est inclus dans $\pi_{GIN(b, \beta/B)}(\omega)$, formellement :

$$\Phi \models (b, \beta/B) \text{ si et seulement si } \forall \omega, \pi_{GIN\Phi}(\omega) \subseteq \pi_{GIN(b, \beta/B)}(\omega)$$

Corollaire 5.2

$$\Phi \models (b, \beta/B) \text{ si et seulement si } \pi_{GIN\Phi} \models (b, \beta/B)$$

En d'autres termes, $EnsF(b, \Phi) = \mathbf{N}_\Phi(b)$, ou $\mathbf{N}_\Phi(b)$ est la mesure de nécessité multi-agents graduée induite par π_Φ .

Remarque 5.3 *L'expression " distribution de possibilité multi-agents graduée la moins spécifique parmi celles satisfaisant Φ " n'a pas été réutilisée.*

En effet, l'inclusion floue $\pi \subseteq \pi'$ n'exprime réellement que π est plus spécifique que π' qu'à niveau de normalisation floue égal (c'est-à-dire $\cup \omega \in \Omega \pi(\omega) = \cup \omega \in \Omega \pi'(\omega)$). Sinon $\pi \subseteq \pi'$ exprime avant tout que π est plus partiellement individuellement normalisée que π' .

La distribution maximale π est donc en fait la distribution de possibilité multi-agents graduée, multi-agents normalisée ou non la moins spécifique parmi les distributions de possibilité multi-agents graduées les moins partiellement individuellement normalisées satisfaisant Φ .

Proposition 5.6 $g - i - inc(\Phi) = GIN(\pi_\Phi) = \mathbf{N}_\Phi(\perp)$.

C'est-à-dire que la graduelle individuelle inconsistance de Φ est égal à l'ensemble de la graduelle individuelle normalisation de π_Φ .

En particulier, si π_Φ est multi-agents normalisée, alors $g - i - inc(\Phi) = \emptyset$, ce qui correspond à la consistance dans la sémantique possibiliste multi-agents normalisée (puisque'il existe au moins une distribution multi-agents graduée normalisée, à savoir π_Φ , satisfaisant Φ).

$$\begin{aligned}
g - i - inc(\Phi) &= \cap\{GIN(\pi), \pi \in \mathcal{PG}_{GIN}(\Phi)\} \\
&= \cap\{IN(\pi), \pi \supseteq \pi_\Phi\} \\
&= \cap\{\cap\{\pi(\omega), \omega \in \Omega, \pi \supseteq \pi_\Phi\}\} \\
&= \cap\{\pi_\Phi(\omega), \omega \in \Omega\} \\
&= GIN(\pi_\Phi).
\end{aligned}$$

Preuve 5.7

$$\begin{aligned}
N_\Phi(\perp) &= \cap\{\overline{\pi_\Phi(\omega)}, \omega \models \perp\} \\
&= \cap\{\overline{\pi_\Phi(\omega)}, \omega \in \Omega\} \\
&= \cap\{\pi(\omega), \omega \in \Omega\} \\
&= IN(\pi_\Phi) \\
&= g - i - inc(\Phi).
\end{aligned}$$

Proposition 5.7 *Soit Φ une base de connaissances possibiliste multi-agents graduellement individuellement inconsistante ($g - i - inc(\Phi) \neq 0/\emptyset$). Alors*

1. $\forall b \ N_\Phi(b) \supseteq g - i - inc(\Phi)$.
2. $\forall b \ N_\Phi(b) \cap N_\Phi(\neg b) = g - i - inc(\Phi)$.

Preuve 5.8 1. $N_\Phi(b) \supseteq N_\Phi(\perp) = g - i - inc(\Phi)$.

2. *La preuve est plus générale que celle définie précédemment. Elle découle des propriétés 5.1.*

Théorème 5.1 *Soit Φ une conjonction de formules possibilistes multi-agents. D'où $g - i - inc(\Phi) = 0/\emptyset$ si et seulement si Φ^* est consistante.*

Preuve 5.9 *Soit $\Phi = \{(a_i, A_i), i = 1, \dots, n\}$.*

(\Leftarrow) :

Si Φ^ est consistante alors Φ^* admet un modèle ω^* . Alors $\pi_\Phi(\omega^*) = \cap\{\overline{A_i} | \omega^* \models \neg a_i\} = 1/All$ étant donné que $\omega^* \models a_i$. Donc π_Φ est normalisée et $g - i - inc = 0/\emptyset$.*

(\Rightarrow) :

Par Définition : $g - i - inc(\Phi) = 0/\emptyset$ si et seulement si π_Φ est normalisée : si et seulement si $\exists \omega^ \in \Omega$ tel que $\pi_\Phi(\omega^*) = 1/All$. Cela implique : $\omega^* \models a_i, \forall i$. D'ici Φ^* est consistante.*

5.4.2.4 Déduction sous la graduelle individuelle inconsistance

$\pi_\Phi \subset \pi_{GIN_\Phi}$, puisqu'une distribution de possibilité multi-agents graduée possibiliste multi-agents normalisée est un cas particulier d'une distribution de possibilité " multi-agents graduée possibiliste multi-agents normalisée ou non ".

D'où $(\phi \models_{GIN} \Phi) \Rightarrow (\phi \models \Phi)$ (la réciproque n'étant éventuellement pas vraie). La sémantique partiellement-individuellement-normalisée étend donc naturellement la sémantique multi-agents normalisée, en ce sens qu'elle est moins tolérante pour la déduction.

Notation 5.2 *Dorénavant, nous utiliserons la notation \models à la place de \models_{GIN} , lorsqu'il n'y aura pas d'ambiguïté, et quand nous évoquerons la logique possibiliste multi-agents par défaut à la sémantique possibiliste partiellement-individuellement-normalisée.*

Les propositions 5.3 et 5.4 restent encore valables dans la sémantique partiellement-individuellement-normalisée, sous la même formulation et avec les mêmes preuves (puisqu'elles n'utilisent pas la normalisation) :

Proposition 5.8 *Soient Φ , Σ deux ensembles de formules possibilistes multi-agents et π_Φ la distribution de possibilité multi-agents graduée la moins spécifique parmi celles satisfaisant Φ . Alors*

$$\Phi \models \Sigma \text{ si et seulement si } \pi_\Phi \text{ satisfait } \Sigma$$

Preuve 5.10 *Elle est identique à celle de la proposition 5.3.*

Proposition 5.9 *Soit Φ un ensemble de formules de la logique possibiliste multi-agents, multi-agents consistant, et b une formule quelconque. Alors $\Phi \models (b, \beta/B)$ si et seulement si $\Phi_{\beta/B} \models (b, \beta/B)$.*

Preuve 5.11 *Elle est identique à celle de la proposition 5.4.*

5.4.2.5 Déduction possibiliste multi-agents triviale vs déduction possibiliste multi-agents non-triviale

Quelle que soit la formule b , b est déductible de Φ avec un ensemble flou plus grand ou égal à $g - i - inc(\Phi)$.

Les déductions de $(b, \beta/B)$ avec $\beta/B = g - i - inc(\Phi)$ ne doivent rien à la nature de b , mais elles sont provoquées par l'inconsistance individuelle graduelle, qui généralement n'a rien à voir avec b . Elles seront appelées déductions possibilistes multi-agents triviales ; les déductions de $(b, \beta/B)$ avec $\beta/B \supset g - i - inc(\Phi)$ seront appelées déductions possibilistes multi-agents non-triviales.

Les résultats suivants sont obtenus :

Proposition 5.10 *Posons $g - i - inc(\Phi) = g - i - inc$. Alors*

$$\Phi \equiv \Phi_{g-i-inc} \equiv (\perp, g - i - inc) \wedge \Phi_{\overline{g-i-inc}}$$

Preuve 5.12 *Posons $\Phi = \{(a_i, \alpha_i/A_i), i = 1..n\}$*

1. $\Phi \models \Phi_{g-i-inc}$ est immédiate car Φ contient $\Phi_{g-i-inc}$.
2. $\Phi_{g-i-inc} \models (\perp, g - i - inc) \wedge \Phi_{\overline{g-i-inc}}$
 $\Phi_{g-i-inc} \models \Phi_{\overline{g-i-inc}}$: est immédiate car $\Phi_{g-i-inc}$ contient $\Phi_{\overline{g-i-inc}}$.

Montrons maintenant que $\Phi_{g-i-inc} \models (\perp, g - i - inc)$, c'est-à-dire que $g - i - inc(\Phi_{g-i-inc}) \supseteq g - i - inc$.

Soit $\pi_{\Phi_{g-i-inc}}(\omega)$ la distribution de possibilités la moins spécifique qui satisfait la base de connaissance $\Phi_{g-i-inc}$ d'où :

$$\forall \omega \quad \pi_{\Phi_{g-i-inc}}(\omega) = \bigcap \{(1 - \alpha_i)/A_i \cup 1/\overline{A_i} \mid \omega \models \neg a_i, \alpha_i/A_i \subseteq g - i - inc\}$$

$$\begin{aligned} g - i - inc(\Phi) = g - i - inc &\Rightarrow \forall \omega, \quad \bigcap \{(1 - \alpha_i)/A_i \cup 1/\overline{A_i} \mid \omega \models \neg a_i\} \subseteq \overline{g - i - inc} \\ &\Rightarrow \forall \omega, \quad \bigcap [\bigcap \{(1 - \alpha_i)/A_i \cup 1/\overline{A_i} \mid \omega \models \neg a_i \mid \alpha_i/A_i \supseteq g - i - inc\}, \\ &\quad \bigcap \{(1 - \alpha_i)/A_i \cup 1/\overline{A_i} \mid \omega \models \neg a_i \mid \alpha_i/A_i \subseteq g - i - inc\}] \subseteq \overline{g - i - inc} \\ &\Rightarrow \forall \omega, \quad \bigcap \{(1 - \alpha_i)/A_i \cup 1/\overline{A_i} \mid \omega \models \neg a_i \mid \alpha_i/A_i \supseteq g - i - inc\} \\ &\quad \subseteq \overline{g - i - inc} \\ &\quad (\text{car } \{(1 - \alpha_i)/\alpha_i/A_i \cup 1/\overline{A_i} \mid \omega \models \neg a_i \mid A_i \subseteq g - i - inc\} \\ &\quad \supseteq \overline{g - i - inc}) \\ &\Rightarrow \forall \omega, \quad \pi_{\Phi_{g-i-inc}}(\omega) \subseteq \overline{g - i - inc} \\ &\Rightarrow g - i - inc(\Phi_{g-i-inc}) \supseteq g - i - inc \end{aligned}$$

$$d'où : \Phi_{g-i-inc} \models (\perp, g - i - inc) \wedge \overline{\Phi_{g-i-inc}}$$

3. Soit $\pi \models (\perp, g - i - inc) \wedge \overline{\Phi_{g-i-inc}}$, c'est-à-dire : $\forall \omega, \quad \pi(\omega) \subseteq \overline{g - i - inc}$ et $\pi(\omega) \subseteq \bigcap \{(1 - \alpha_i)/A_i \cup 1/\overline{A_i} \mid \omega \models \neg a_i, \alpha_i/A_i \supseteq g - i - inc\}$

Alors $\forall (a_i, \alpha_i/A_i) \in \Phi :$

Si $\alpha_i/A_i \supseteq g - i - inc$ alors $\pi \models (a_i, \alpha_i/A_i)$.

Et si $\alpha_i/A_i \subseteq g - i - inc$ alors $N(a_i) \supseteq N(\perp) \supseteq g - i - inc \supseteq \alpha_i/A_i$ donc $\pi \models (a_i, \alpha_i/A_i)$.

Nous obtenons ainsi : $\pi \models \Phi$.

Proposition 5.11 :

Il existe (au moins) une interprétation ω^* telle que $\bigcup \{\pi_{\Phi}(\omega), \omega \in \Omega\} = \pi_{\Phi}(\omega^*)$.

Preuve 5.13 Elle est immédiate car Ω est fini.

Théorème 5.2 Soit $g - i - inc(\Phi) = g - i - inc :$

$\overline{\Phi_{g-i-inc}}$ est alors consistante.

Preuve 5.14 Posons $\Phi = \{(a_i, \alpha_i/A_i), i = 1 \dots n\}$

Soit $\pi_{\overline{\Phi_{g-i-inc}}}(\omega)$ la distribution de possibilités graduée qui satisfait $\overline{\Phi_{g-i-inc}}$.

La proposition 5.11 assure l'existence d'une interprétation ω^* telle que $\pi_{\Phi}(\omega^*) = \overline{g - i - inc} = \bigcap \{(1 - \alpha_i)/A_i \cup 1/\overline{A_i} \mid \omega \models \neg a_i\}$.

$$\begin{aligned} \overline{g - i - inc} = \bigcap \{(1 - \alpha_i)/A_i \cup 1/\overline{A_i} \mid \omega \models \neg a_i\} &\Rightarrow \forall i, (\omega^* \models \neg a_i \Rightarrow \alpha_i/A_i \subseteq g - i - inc) \\ &\Rightarrow \omega^* \text{ satisfait } \wedge \{a_i \mid \alpha_i/A_i \supseteq g - i - inc\} \\ &\Rightarrow \omega^* \models \overline{\Phi_{g-i-inc}^*} \\ &\Rightarrow \overline{\Phi_{g-i-inc}^*} \text{ est consistante} \\ &\Rightarrow \overline{\Phi_{g-i-inc}} \text{ est consistante (thorme 5.1)}. \end{aligned}$$

Corollaire 5.3

$$g - i - inc(\Phi) = \bigcup \{ \alpha/A, \Phi_{\alpha/A}^* \text{ inconsistent} \} = \bigcap \{ \alpha/A, \Phi_{\alpha/A}^* \text{ consistant} \}.$$

Preuve 5.15 Elle est immédiate à partir des propositions 5.1, 5.10 et du résultat précédent.

Théorème 5.3 Toutes les déductions possibilistes multi-agents non-triviales que nous pouvons effectuer à partir de Φ peuvent être déduites de $\Phi_{g-i-inc}$, c'est-à-dire plus formellement :

$$[\Phi \models (b, \beta/B) \text{ non trivialement}] \Rightarrow [\Phi_{g-i-inc} \models (b, \beta/B)]$$

Preuve 5.16 Posons $\Phi = \{(a_i, \alpha_i/A_i), i = 1..n\}$

$$\begin{aligned} \Phi \models (b, \beta/B) \text{ non trivialement} &\Rightarrow \Phi \models (b, \beta/B) \text{ et } \beta/B \supset g - i - inc \\ &\Rightarrow \forall \omega \models \neg b, \bigcap \{ (1 - \alpha_i)/A_i \cup 1/\overline{A_i} \mid \omega \models \neg a_i \} \\ &\quad \subseteq \overline{\beta/B} \subset g - i - inc \quad (1) \\ &\Rightarrow (\forall \omega \models \neg b, \omega \models \neg a_i \Rightarrow \alpha_i/A_i \supset g - i - inc) \\ &\Rightarrow \forall \omega \models \neg b, \bigcap \{ (1 - \alpha_i)/A_i \cup 1/\overline{A_i} \mid \omega \models \neg a_i \mid \\ &\quad \alpha_i/A_i \supset g - i - inc \} = \bigcap \{ (1 - \alpha_i)/A_i \cup 1/\overline{A_i}, \omega \models \neg a_i \} \quad (2) \\ &\Rightarrow \forall \omega \models \neg b, \bigcap \{ (1 - \alpha_i)/A_i \cup 1/\overline{A_i} \mid \omega \models \neg a_i \mid \\ &\quad \alpha_i/A_i \supset g - i - inc \} \subseteq \overline{\beta/B} \quad (De(1) \text{ et } (2)) \\ &\Rightarrow \Phi_{g-i-inc} \models (b, \beta/B) \end{aligned}$$

Proposition 5.12 Soit Φ une conjonction de formules possibilistes multi-agents et b une formule quelconque. Alors $\exists \beta/B \neq 0/\emptyset$ tel que $\Phi \models (b, \beta/B)$ si et seulement si $\Phi^* \models b$.

Preuve 5.17 Posons $\Phi = \{(a_i, \alpha_i/A_i), i = 1..n\}$.

La preuve nécessite l'établissement du lemme suivant :

Lemme 5.1 $\omega \models \Phi^* \Leftrightarrow \pi_{\Phi}(\omega) = 1/All$.

Preuve 5.18 $\omega \models \Phi^* \Leftrightarrow (\forall i = 1, \dots, n) \omega \models a_i \Leftrightarrow \pi_{\Phi}(\omega) = \bigcap \{ (1 - \alpha_i)/A_i \cup 1/\overline{A_i} \mid \omega \models \neg a_i \} = 1/All$.

(\Rightarrow) :

Supposons que $\exists \beta/B \neq 0/\emptyset$ tel que $\Phi \models (b, \beta/B)$.

1. $g - i - inc(\Phi) \supset 0/\emptyset$:

Dans ce cas, Φ^* est inconsistante (théorème 5.1) d'où $\Phi^* \models b$.

2. $g - i - inc(\Phi) = 0/\emptyset$:

Soit alors $\omega \models \Phi^*$, montrons que $\omega \models b$.

Supposons que $\omega \models \neg b$, alors $\pi_{\Phi}(\omega) \subseteq \overline{\beta/B} \subset 1/All$ (car $\Phi \models (b, \beta/B)$). Ceci est impossible d'après le lemme 5.1. Donc $\omega \models b$. Tout modèle de Φ^* étant donc un modèle de b . Nous avons alors $\Phi^* \models b$.

(\Leftarrow) :

Supposons que $\Phi^* \models b$.

Soit $\omega \models \neg b$ alors $\omega \models \neg \Phi^*$ (car $\Phi^* \models b$) C'est-à-dire : $\exists i(\omega)$ tel que $\omega \models \neg a_{i(\omega)}$.
Nous avons donc $\pi_\Phi(\omega) = \cap \{\overline{B_i} \mid \omega \models \neg a_i\} \subseteq \overline{B_{i(\omega)}} \subset All$.

Ceci étant valable $\forall \omega \models \neg b$, on a $N_\Phi(b) = \cap \{\overline{\pi_\Phi(\omega)} \mid \omega \models \neg b\} \supseteq \cap \{\overline{B_{i(\omega)}} \mid \omega \models \neg b\} \supset 0/\emptyset$.

D'où : $\exists \beta/B \supset 0/\emptyset$ tel que $\Phi \models (b, \beta/B)$.

5.5 Axiomatique de la logique possibiliste multi-agents

Nous nous intéressons, dans cette partie, à la recherche d'un système formel pour la logique possibiliste multi-agents munie de la sémantique partiellement-individuellement-normalisée. D'abord, il est à noter que les tautologies de la logique possibiliste multi-agents étant toutes les formules de la forme $(\tau, \alpha/A)$ où τ est une tautologie de la logique classique, $\alpha \in [0, 1]$ et $A \in 2^{All}$, il est possible de construire, de façon triviale, un système formel produisant toutes les tautologies de la logique possibiliste multi-agents.

Nous nous intéressons plutôt au problème suivant : existe-t-il un système formel, c'est-à-dire un ensemble d'axiomes et de règles d'inférence, tel que pour toute ensemble Φ de formules possibiliste multi-agents et pour toute formule possibiliste multi-agents $(b, \beta/B)$ conséquence logique de Φ , nous puissions dériver $(b, \beta/B)$ à partir des axiomes, de Φ , et des règles d'inférences ?.

Nous proposons le système formel correct et complet suivant pour la logique possibiliste multi-agents, munie de la sémantique partiellement-individuellement normalisée :

Schémas d'axiomes :

$$(A1) \quad (a \rightarrow (b \rightarrow a), 1/All)$$

$$(A2) \quad ((a \rightarrow (b \rightarrow c)) \rightarrow ((a \rightarrow b) \rightarrow (a \rightarrow c)), 1/All)$$

$$(A3) \quad ((\neg a \rightarrow \neg b) \rightarrow ((\neg a \rightarrow b) \rightarrow a), 1/All)$$

Règles d'inférences :

(Modus ponens ensembliste graduel) $(a, \alpha/A), (a \rightarrow b, \beta/B) \vdash (b, \alpha/A \cap \beta/B)$.

(Subsumption ensembliste graduelle) $(a, \alpha/A) \vdash (a, \beta/B)$ pour tout $\beta/B \subseteq \alpha/A$.

Théorème 5.4 *Le système formel proposé est correct et complet relativement à la sémantique partiellement-individuellement-normalisée de la logique possibiliste multi-agents.*

C'est-à-dire que pour tout ensemble de formules possibiliste multi-agents Φ nous avons :

$$\Phi \models (b, \beta/B) \text{ si et seulement si } \Phi \vdash (b, \beta/B)$$

où $\Phi \vdash (b, \beta/B)$ signifie : " il existe une dérivation de $(b, \beta/B)$ dans le système formel constitué de Φ , des schémas d'axiomes et des règles d'inférence du système".

Preuve 5.19 *Pour prouver la correction de la règle il faudra vérifier que si $A_1 \cap A_2 \neq \emptyset$, alors :*

$$\forall \omega, \pi(\omega)_{\{(c_1, \alpha_1/A_1), (c_2, \alpha_2/A_2)\}} \subseteq \pi(\omega)_{\{(c_3, \min(\alpha_1, \alpha_2)/A_1 \cap A_2)\}}$$

où $(c_1, \alpha_1/A_1)$ et $(c_2, \alpha_2/A_2)$ sont des clauses possibilistes multi-agents, et c_3 une résolvante classique de c_1 et c_2 ; alors $(c_3, \min(\alpha_1, \alpha_2)/A_1 \cap A_2)$ est la résolvante de $(c_1, \alpha_1/A_1)$ et $(c_2, \alpha_2/A_2)$ par la A-B-résolution graduelle $(c_1, \alpha_1/A_1), (c_2, \alpha_2/A_2) \vdash (c_3, \min(\alpha_1, \alpha_2)/A_1 \cap A_2)$. Nous avons :

$$\pi_{\{(c_1, \alpha_1/A_1), (c_2, \alpha_2/A_2)\}}(\omega) = \begin{cases} 1/All & \text{si } \omega \models c_1 \wedge c_2 \\ [(1 - \alpha_1)/A_1 \cup 1/\overline{A_1}] \cap [(1 - \alpha_2)/A_2 \cup 1/\overline{A_2}] & \text{si } \omega \models \neg(c_1 \wedge c_2) \end{cases}$$

et,

$$\pi_{\{(c_3, \min(\alpha_1, \alpha_2)/A_1 \cap A_2)\}}(\omega) = \begin{cases} 1/All & \text{si } \omega \models c_3 \\ [(1 - \min(\alpha_1, \alpha_2))/A_1 \cap A_2] \cup [1/\overline{A_1 \cap A_2}] & \text{si } \omega \models \neg c_3 \end{cases}.$$

on a :

$$\begin{aligned} [(1 - \alpha_1)/A_1 \cup 1/\overline{A_1}] \cap [(1 - \alpha_2)/A_2 \cup 1/\overline{A_2}] &= [(1 - \alpha_1)/A_1 \cap (1 - \alpha_2)/A_2] \cup \\ & [(1 - \alpha_1)/A_1 \cap 1/\overline{A_2}] \cup [1/\overline{A_1} \cap (1 - \alpha_2)/A_2] \cup \\ & [1/\overline{A_1} \cap 1/\overline{A_2}] \\ &= [\min((1 - \alpha_1), (1 - \alpha_2))/A_1 \cap A_2] \cup \\ & [(1 - \alpha_1)/A_1 \cap \overline{A_2}] \cup [(1 - \alpha_2)/\overline{A_1} \cap A_2] \\ & \cup [1/\overline{A_1} \cap \overline{A_2}] \\ &= [\min((1 - \alpha_1), (1 - \alpha_2))/A_1 \cap A_2] \cup \\ & [\max((1 - \alpha_1), (1 - \alpha_2))/(A_1 \cap \overline{A_2}) \cup (\overline{A_1} \cap A_2)] \\ & \cup [1/\overline{A_1} \cap \overline{A_2}]. \end{aligned}$$

Etant donné que :

1. $[\min((1 - \alpha_1), (1 - \alpha_2))/A_1 \cap A_2] \subseteq [(1 - \min(\alpha_1, \alpha_2))/A_1 \cap A_2]$.
2. $[1/\overline{A_1} \cap \overline{A_2}] \subseteq [1/\overline{A_1 \cap A_2}]$
3. $[\max((1 - \alpha_1), (1 - \alpha_2))/(A_1 \cap \overline{A_2}) \cup (\overline{A_1} \cap A_2)] \subseteq [1/\overline{A_1 \cap A_2}]$.

et que la situation $\pi_{\{(c_1, \alpha_1/A_1), (c_2, \alpha_2/A_2)\}}(\omega) = 1/All$ et $\pi_{\{(c_3, \min(\alpha_1, \alpha_2)/A_1 \cap A_2)\}}(\omega) = [(1 - \min(\alpha_1, \alpha_2))/A_1 \cap A_2] \cup [1/\overline{A_1 \cap A_2}]$ ne peut pas se présenter car par hypothèse $\omega \models c_1 \wedge c_2 \Rightarrow \omega \models c_3$

Ainsi, la A-B-résolution graduelle est correcte.

Pour la complétude, supposons $\Phi \not\vdash (a, \alpha/A)$. Alors $\exists \Delta \subseteq \Phi, \Delta \vdash (a, \alpha/A)$ en utilisant la A-B-résolution graduelle, avec toutes les formules de Δ . D'où, pour tous les groupes d'agents tels que $\Delta \subseteq \Phi$, soit $\Delta^* \not\vdash (a, \alpha)$ ou $\Delta^* \vdash (a, \alpha)$ mais $A \not\subseteq \bigcap_{(a_i, \alpha_i/A_i) \in \Delta} A_i$.

Pour le premier cas $\Delta^* \not\vdash (a, \alpha)$ (complétude de la logique possibiliste), d'où $\Delta \not\vdash (a, \alpha/A)$ aussi.

Pour le deuxième cas, $\exists (i, k) : k \in A, k \notin A_i$. Soit $\Delta_k^* = \{(a_j, \alpha_j) : k \in A_j, (a_j, \alpha_j/A_j) \in \Delta\}$, clairement $\Delta_k^* \not\vdash (a, \alpha)$ car elle ne contient pas (a_i, α_i) et (a_i, α_i) est utilisé pour dériver (a, α) . D'où, $\Delta_k^* \not\vdash (a, \alpha)$ (complétude de la logique possibiliste). Mais alors

$\Delta_k \not\models (a, \alpha/\{k\})$ aussi, et $\Delta \not\models (a, \alpha/A)$. Comme c'est le cas pour tous les sous-ensembles de Φ , $\Phi \not\models (a, \alpha/A)$.

Nous pouvons aussi procéder par équivalence comme suit :

$$\begin{aligned} \Phi \vdash (a, \alpha/A) &\Leftrightarrow \forall k \in A, \Phi_k \vdash (a, \alpha) \text{ (par définition)} \\ &\quad \forall k \in A, \Phi_k \models (a, \alpha) \text{ (complétude de la logique possibiliste)} \\ &\quad \Phi^{\alpha/A} \models (a, \alpha/A) \text{ (par définition, en ne gardons que les formules de} \\ &\quad \Phi \text{ qui peuvent jouer un rôle dans l'inférence de } (a, \alpha/A)) \\ &\quad \Phi \models (a, \alpha/A) \text{ (par monotonie de l'inférence)} \end{aligned}$$

où $\Phi_k = \{(a_i, \alpha_i) \mid (a_i, \alpha_i/A_i) \in \Phi \text{ et } k \in A_i\}$ et $\Phi^{\alpha/A} = \{(a_i, \alpha_i/A_i) \mid A_i \cap A \neq \emptyset \text{ et } \alpha_i \geq \alpha \text{ et } (a_i, \alpha_i/A_i) \in \Phi\}$.

Corollaire 5.4 *la logique possibiliste multi-agents est axiomatisable.*

5.6 Exemple pour la logique possibiliste multi-agents

Soit la base de connaissances possibiliste multi-agents suivante :

$$\Phi = \{(\neg p \vee q, 0.6/A), (\neg p \vee q, 0.8/C), (\neg p \vee r, 0.9/A), (\neg p \vee \neg r, 0.1/B), (\neg q \vee r, 0.6/All), (p, 0.3/All), (q, 0.7/A), (r, 0.8/C), (\neg q, 0.2/D)\}.$$

Elle est obtenue en combinant le degré de certitude ainsi que les sous-ensembles d'agents des deux exemples précédents pour chaque formule propositionnelle.

La distribution de possibilité multi-agents graduée associée à Φ est :

$$\begin{aligned} \pi_{\Phi}(pqr) &= (0.9/B \cup 1/\bar{B}) \cap (0.8/D \cup 1/\bar{D}) \\ \pi_{\Phi}(pq\neg r) &= (0.1/A \cup 0.4/\bar{A}) \cap (0.2/C \cup 0.4/\bar{C}) \cap (0.8/D \cup 0.4/\bar{D}) \\ \pi_{\Phi}(p\neg qr) &= (0.3/A \cup 1/\bar{A}) \cap (0.9/B \cup 1/\bar{B}) \cap (0.2/C \cup 1/\bar{C}) \\ \pi_{\Phi}(p\neg q\neg r) &= (0.1/A \cup 1/\bar{A}) \cap (0.2/C \cup 1/\bar{C}) \\ \pi_{\Phi}(\neg pqr) &= (0.7/ALL) \\ \pi_{\Phi}(\neg pq\neg r) &= (0.2/C \cup 0.4/\bar{C}) \cap (0.4/D \cup 0.4/\bar{D}) \\ \pi_{\Phi}(\neg p\neg qr) &= (0.3/A \cup 0.7/\bar{A}) \\ \pi_{\Phi}(\neg p\neg q\neg r) &= (0.3/A \cup 0.7/\bar{A}) \cap (0.2/C \cup 0.7/\bar{C}) \end{aligned}$$

Le calcul de la distribution de possibilité multi-agents graduée détaillé est donné dans les tables 5.1 et 5.2.

Comme dans l'exemple précédent, nous pouvons calculer la partie logique possibiliste $(\Phi_S)^*$ pour chaque sous ensemble de la partition de l'ensemble des agents, et calculer son degré d'inconsistance. Les cas correspondant à des degrés d'inconsistance nul sont marqués par des * . Nous trouvons exactement les sept cas de l'exemple précédent ; comme attendu (étant donné que cet exemple coïncide avec le deuxième exemple lorsque les degrés de certitudes sont ignorés). Le degré d'inconsistance correspond ainsi au degré d'inconsistance des différents sous-groupes d'agents.

Le degré d'inconsistance global de Φ lorsque les sous-ensembles d'agents sont ignorés c.à.d. $inc-l((\Phi)^*)$ est égal à 0.2 comme dans le premier exemple. Ainsi, la 0.3-coupe de

ω	$\pi_{(\neg p \vee q, .6/A)}$	$\pi_{(\neg p \vee q, .8/C)}$	$\pi_{(\neg p \vee r, .9/A)}$	$\pi_{(\neg p \vee \neg r, .1/B)}$	$\pi_{(\neg q \vee r, .6/All)}$	$\pi_{(q, .7/A)}$
pqr	1/All	1/All	1/All	$(0.9/B \cup 1/\bar{B})$	1/All	1/All
$pq\neg r$	1/All	1/All	$(0.1/A \cup 1/\bar{A})$	1/All	0.4/All	1/All
$p\neg qr$	$(0.4/A \cup 1/\bar{A})$	$(0.2/C \cup 1/\bar{C})$	1/All	$(0.9/B \cup 1/\bar{B})$	1/All	$(0.3/A \cup 1/\bar{A})$
$p\neg q\neg r$	$(0.4/A \cup 1/\bar{A})$	$(0.2/C \cup 1/\bar{C})$	$(0.1/A \cup 1/\bar{A})$	1/All	1/All	$(0.3/A \cup 1/\bar{A})$
$\neg pqr$	1/All	1/All	1/All	1/All	1/All	1/All
$\neg pq\neg r$	1/All	1/All	1/All	1/All	0.4/All	1/All
$\neg p\neg qr$	1/All	1/All	1/All	1/All	1/All	$(0.3/A \cup 1/\bar{A})$
$\neg p\neg q\neg r$	1/All	1/All	1/All	1/All	1/All	$(0.3/A \cup 1/\bar{A})$

TABLE 5.1 – Calcul détaillé de la distribution de possibilité multi-agents graduée - 1 de l'exemple de la section 5.6

ω	$\pi_{(p, .3/All)}$	$\pi_{(\neg q, .2/D)}$	$\pi_{(r, .8/C)}$	π_{Φ}
pqr	1/All	$(0.8/D \cup 1/\bar{D})$	1/All	$(0.9/B \cup 1/\bar{B}) \cap (0.8/D \cup 1/\bar{D})$
$pq\neg r$	1/All	$(.8/D \cup 1/\bar{D})$	$(.2/C \cup 1/\bar{C})$	$(.1/A \cup 1/\bar{A}) \cap (.4/All) \cap (.2/C \cup 1/\bar{C}) \cap (.8/D \cup 1/\bar{D})$
$p\neg qr$	1/All	1/All	1/All	$(0.3/A \cup 1/\bar{A}) \cap (0.9/B \cup 1/\bar{B}) \cap (0.2/C \cup 1/\bar{C})$
$p\neg q\neg r$	1/All	1/All	$(0.2/C \cup 1/\bar{C})$	$(0.1/A \cup 1/\bar{A}) \cap (0.2/C \cup 1/\bar{C})$
$\neg pqr$	0.7/All	$(0.8/D \cup 1/\bar{D})$	1/All	$(0.7/All) \cap (0.8/D \cup 1/\bar{D})$
$\neg pq\neg r$	0.7/All	$(0.8/D \cup 1/\bar{D})$	$(0.2/C \cup 1/\bar{C})$	$(0.4/All) \cap (0.2/C \cup 1/\bar{C}) \cap (0.8/D \cup 1/\bar{D})$
$\neg p\neg qr$	0.7/All	1/All	1/All	$(0.7/All) \cap (0.3/A \cup 1/\bar{A})$
$\neg p\neg q\neg r$	0.7/All	1/All	$(0.2/C \cup 1/\bar{C})$	$(0.7/All) \cap (0.3/A \cup 1/\bar{A}) \cap (0.2/C \cup 1/\bar{C})$

TABLE 5.2 – Calcul détaillé de la distribution de possibilité multi-agents graduée - 2 de l'exemple de la section 5.6

Φ , à savoir l'ensemble des formules multi-agents : $\{(a_i, A_i) \mid (a_i, \alpha_i/A_i) \in \Phi \text{ et } \alpha_i \geq 0.3\}$ est collectivement consistante. De plus, étant donné que $cons-s(\Phi) = (\bar{A} \cap \bar{C}) \cup (\bar{B} \cap \bar{D})$ dans le deuxième exemple, on aura alors, par exemple, $inc-l(\Phi_{\bar{B} \cap \bar{D}}) = 0$, c.à.d. les agents dans $\bar{B} \cap \bar{D}$ détiennent individuellement des bases de connaissances possibiliste consistantes.

Notons que dans Φ , nous avons quatre symboles différents par rapport aux sous-ensembles d'agents, à savoir A, B, C, D . Cela induit une partition d'ensembles d'agents en $2^4 = 16$ sous-ensembles indiscernables d'agents. Pour chaque sous ensemble S de cette partition, nous pouvons calculer la partie possibiliste $(\Phi_S)^*$ de la restriction Φ_S de Φ à S . Le résultat est donné ci-dessous.

$$\begin{aligned}
\Phi_{A \cap B \cap C \cap D}^* &= \{(\neg p \vee q, 0.6), (\neg p \vee q, 0.8), (\neg p \vee r, 0.9), (\neg p \vee \neg r, 0.1), (\neg q \vee r, 0.6), (p, 0.3), \\
&\quad (q, 0.7), (r, 0.8), (\neg q, 0.2)\} \quad inc(\Phi_{A \cap B \cap C \cap D}) = 0.2 \\
\Phi_{A \cap B \cap C \cap \bar{D}}^* &= \{(\neg p \vee q, 0.6), (\neg p \vee q, 0.8), (\neg p \vee r, 0.9), (\neg p \vee \neg r, 0.1), (\neg q \vee r, 0.6), (p, 0.3), \\
&\quad (q, 0.7), (r, 0.8)\} \quad inc(\Phi_{A \cap B \cap C \cap \bar{D}}) = 0.1 \\
\Phi_{A \cap B \cap \bar{C} \cap D}^* &= \{(\neg p \vee q, 0.6), (\neg p \vee r, 0.9), (\neg p \vee \neg r, 0.1), (\neg q \vee r, 0.6), (p, 0.3), (q, 0.7), \\
&\quad (\neg q, 0.2)\} \quad inc(\Phi_{A \cap B \cap \bar{C} \cap D}) = 0.2
\end{aligned}$$

$$\begin{aligned}
\Phi_{A \cap B \cap \bar{C} \cap \bar{D}}^* &= \{(\neg p \vee q, 0.6), (\neg p \vee r, 0.9), (\neg p \vee \neg r, 0.1), (\neg q \vee r, 0.6), (p, 0.3), (q, 0.7)\} \\
&\quad inc(\Phi_{A \cap B \cap \bar{C} \cap \bar{D}}) = 0.1 \\
\Phi_{A \cap \bar{B} \cap C \cap D}^* &= \{(\neg p \vee q, 0.6), (\neg p \vee q, 0.8), (\neg p \vee r, 0.9), (\neg q \vee r, 0.6), (p, 0.3), (q, 0.7), \\
&\quad (r, 0.8), (\neg q, 0.2)\} \quad inc(\Phi_{A \cap \bar{B} \cap C \cap D}) = 0.2 \\
\Phi_{A \cap \bar{B} \cap C \cap \bar{D}}^* &= \{(\neg p \vee q, 0.6), (\neg p \vee q, 0.8), (\neg p \vee r, 0.9), (\neg q \vee r, 0.6), (p, 0.3), (q, 0.7), \\
&\quad (r, 0.8)\} * \\
\Phi_{A \cap \bar{B} \cap \bar{C} \cap D}^* &= \{(\neg p \vee q, 0.6), (\neg p \vee r, 0.9), (\neg q \vee r, 0.6), (p, 0.3), (q, 0.7), (\neg q, 0.2)\} \\
&\quad inc(\Phi_{A \cap \bar{B} \cap \bar{C} \cap D}) = 0.2 \\
\Phi_{A \cap \bar{B} \cap \bar{C} \cap \bar{D}}^* &= \{(\neg p \vee q, 0.6), (\neg p \vee r, 0.9), (\neg q \vee r, 0.6), (p, 0.3), (q, 0.7)\} * \\
\Phi_{\bar{A} \cap B \cap C \cap D}^* &= \{(\neg p \vee q, 0.8), (\neg p \vee \neg r, 0.1), (\neg q \vee r, 0.6), (p, 0.3), (r, 0.8), (\neg q, 0.2)\} \\
&\quad inc(\Phi_{\bar{A} \cap B \cap C \cap D}) = 0.2 \\
\Phi_{\bar{A} \cap B \cap C \cap \bar{D}}^* &= \{(\neg p \vee q, 0.8), (\neg p \vee \neg r, 0.1), (\neg q \vee r, 0.6), (p, 0.3), (r, 0.8)\} \\
&\quad inc(\Phi_{\bar{A} \cap B \cap C \cap \bar{D}}) = 0.1 \\
\Phi_{\bar{A} \cap B \cap \bar{C} \cap D}^* &= \{(\neg p \vee \neg r, 0.1), (\neg q \vee r, 0.6), (p, 0.3), (\neg q, 0.2)\} * \\
\Phi_{\bar{A} \cap B \cap \bar{C} \cap \bar{D}}^* &= \{(\neg p \vee \neg r, 0.1), (\neg q \vee r, 0.6), (p, 0.3)\} * \\
\Phi_{\bar{A} \cap \bar{B} \cap C \cap D}^* &= \{(\neg p \vee q, 0.8), (\neg q \vee r, 0.6), (p, 0.3), (r, 0.8), (\neg q, 0.2)\} \\
&\quad inc(\Phi_{\bar{A} \cap \bar{B} \cap C \cap D}) = 0.2 \\
\Phi_{\bar{A} \cap \bar{B} \cap C \cap \bar{D}}^* &= \{(\neg p \vee \neg r, 0.1), (\neg q \vee r, 0.6), (p, 0.3), (r, 0.8)\} * \\
\Phi_{\bar{A} \cap \bar{B} \cap \bar{C} \cap D}^* &= \{(\neg q \vee r, 0.6), (p, 0.3), (\neg q, 0.2)\} * \\
\Phi_{\bar{A} \cap \bar{B} \cap \bar{C} \cap \bar{D}}^* &= \{(\neg q \vee r, 0.6), (p, 0.3)\} *
\end{aligned}$$

Etant donné que cette base de connaissances sans les degrés de certitudes coïncide avec la base de connaissances multi-agents du deuxième exemple, $inc-s(\Phi) = (A \cup C) \cap (B \cup D)$ reste inchangé, mais cette fois-ci la base de connaissances est partitionnée en sous-ensembles d'agents qui sont inconsistants au degré 0, 2, à savoir $(A \cap D) \cup (\bar{A} \cap C \cap D)$ et en sous-ensembles d'agents qui sont inconsistants au degré 0, 1 seulement, à savoir $(A \cap B \cap \bar{D}) \cup (\bar{A} \cap B \cap C \cap \bar{D})$.

5.7 Conclusion

Ce chapitre a été consacré pour présenter une nouvelle logique qui étend naturellement non seulement la logique multi-agents du chapitre précédent mais aussi la logique possibiliste standard du chapitre 3. L'incertitude associée à une proposition détenue par un ensemble d'agents est évaluée par deux mesures duales : la mesure de nécessité graduelle et la mesure de possibilités graduelle. En effet, le traitement de l'incertitude pris en compte par la logique possibiliste est étendu.

Dans les chapitres 4 et 5, nous nous sommes intéressés au volet de la représentation des connaissances où en premier lieu, à une formule est associée un ensemble d'agents et en second lieu, à une formule est associée un ensemble flou d'agents. Dans le chapitre suivant, nous allons nous pencher sur le volet du raisonnement.

Chapitre 6

La résolution multi-agents

6.1 Introduction

Après avoir présenté les premières contributions de notre étude qui touchent l'aspect de la représentation des connaissances attachées à un ensemble d'agents, nous nous intéressons dans ce chapitre à l'aspect du raisonnement à partir des connaissances en logique multi-agents.

Nous présentons ainsi l'algorithme de résolution par réfutation développé dans [Belhadi et All. 14] pour la logique multi-agents. Cet algorithme est une adaptations de l'algorithmes développé dans le cadre de la logique possibiliste [Yager 86]. Il contribue à répondre à la question suivante : soit Φ un base de connaissance multi-agents, avec quel ensemble d'agents nous pouvons déduire la formule b ?

Ce chapitre est organisé comme suit : dans la section 2 nous définissons la forme clausale d'une base de connaissances multi-agents. La section suivante donne l'algorithme de mise sous forme clausale pour la logique multi-agents. Puis, nous introduisons dans la section 4 les règles de résolution. La section 5 aborde la méthode de réfutation par résolution tout en invoquant des résultats de la complétude et de la correction. Ce chapitre s'achève avec la définition de stratégie de résolution multi-agents efficace.

6.2 Motivations et définitions

Dans les méthodes de démonstration automatiques basées sur la résolution, la première étape consiste à transformer les formules données en un ensemble de clauses qui ont la même consistance que les formules initiales : c'est la mise sous forme clausale. Nous étendons, dans cette section les notions essentielles à cette transformation à la logique multi-agents.

Définition 6.1 *Une clause multi-agents est un couple (a, A) , où a est une clause de la logique classique et A un ensemble non vide d'agents dans All .*

Définition 6.2 *Une formule clausale multi-agents est une conjonction finie de clauses multi-agents.*

Exemple 6.1 $[(\neg a \vee b, All) \wedge (\neg b \vee c, A) \wedge (a \vee \neg b \vee c, B) \wedge (a \vee \neg b \vee \neg c, D)]$ est une forme clausale multi-agents

6.3 Mise sous forme clausale dans la logique multi-agents

La mise en place du principe de résolution dépend de la normalisation des formules : pour pouvoir comparer plus facilement si deux formules sont identiques ou pas, un standard d'écriture est sélectionné et qui correspond à la forme clausale afin de faciliter le calcul.

Dans cette section, nous établissons un algorithme qui permet de transformer une formule multi-agents quelconque en une formule mise sous forme clausale multi-agents qui possède la même individuelle inconsistance.

Début

Φ conjonction de formules multi-agents : $\Phi = \wedge\{(a_i, A_i), i = 1..n\}$

Pour $i = 1..n$ **faire**

 | mettre a_i sous forme clausale : $a_i = \wedge\{c_{ij}, j = 1..m_i\}$

Fin Pour

$\mathcal{C}(\Phi) = \wedge\{\wedge\{c_{ij}, j = 1..m_i\}, i = 1..n\}$ est une forme clausale de Φ .

Fin

Algorithme 7: Algorithme de mise sous forme clausale pour la logique multi-agents

Théorème 6.1 *Soit Φ un ensemble de formules multi-agents et $\mathcal{C}(\Phi)$ sa forme clausale déterminée par l'algorithme précédent. Alors Φ et $\mathcal{C}(\Phi)$ ont la même individuelle inconsistance.*

Preuve 6.1 *La preuve nécessite l'établissement du lemme suivant :*

Lemme 6.1 – $\mathcal{C}(\Phi_A)$ (Respectivement $\mathcal{C}(\Phi_{\bar{A}})$) est une forme clausale de Φ_A (Respectivement $\Phi_{\bar{A}}$).

– $\mathcal{C}(\Phi)^*$ est une forme clausale de Φ^* .

Preuve 6.2 *Elle est immédiate.*

$i - inc(\Phi) = A \Leftrightarrow \Phi_A^*$ est individuellement inconsistante et $\Phi_{\bar{A}}^*$ est multi-agents consistante.
 $\Leftrightarrow \mathcal{C}(\Phi_A^*)$ est individuellement inconsistante et $\mathcal{C}(\Phi_{\bar{A}}^*)$ est multi-agents consistante ou
 $\mathcal{C}(\Phi_A^*)$ et $\mathcal{C}(\Phi_{\bar{A}}^*)$ sont des formes clausales respectivement de Φ_A^* et $\Phi_{\bar{A}}^*$
(d'après l'équivalence de l'inconsistance d'une formule et de celle de sa forme clausale en logique classique).
 $\Leftrightarrow \mathcal{C}(\Phi_A)^*$ est individuellement inconsistante et $\mathcal{C}(\Phi_{\bar{A}})^*$ est multi-agents consistante (lemme 6.1).
 $\Leftrightarrow \mathcal{C}(\Phi)_A^*$ est individuellement inconsistante et $\mathcal{C}(\Phi)_{\bar{A}}^*$ multi-agents consistante (lemme 6.1).
 $\Leftrightarrow i - inc(\mathcal{C}(\Phi)) = A$ (corollaire 4.3).

6.4 Les règles de résolution

6.4.1 Les règles de la logique multi-agents

Soient (c, A) et (c', B) deux clauses multi-agents, et soit c'' une résolvante quelconque de c et c' . Les règles suivantes sont appelées règles de résolution multi-agents.

$\forall A \in 2^{All} \setminus \emptyset, (c, A), (c', A) \vdash (c'', A)$ (résolution sous ensembliste)

Si $A \cap B \neq \emptyset$ alors $(c, A), (c', B) \vdash (c'', A \cap B)$ (A-B-résolution)

Nous introduisons la règle suivante, appelée règle de l'affaiblissement sous ensembliste :

Si $B \subseteq A$ alors $(c, A) \vdash (c, B)$ (affaiblissement sous ensembliste)

6.4.2 Correction des règles de la logique multi-agents

Les résultats suivants justifient les règles (résolution sous ensembliste), (A-B-résolution) et (affaiblissement sous ensembliste) en montrant qu'elles sont bien en accord avec la logique multi-agents :

Théorème 6.2 (correction de la règle : résolution sous ensembliste)

Soient c et c' deux clauses multi-agents et c'' une clause obtenue par l'application de la résolution sous ensembliste à partir de c et c' . Alors nous obtenons : $c, c' \models c''$.

Preuve 6.3 Elle est immédiate.

Théorème 6.3 (correction de la règle : A-B-résolution)

Soient c et c' deux clauses multi-agents et c'' une clause obtenue par l'application de la (A-B-résolution) à partir de c et c' . Alors : $c, c' \models c''$.

Preuve 6.4 Soient $C = (c, A)$, $C' = (c', B)$ et $C'' = (c'', A \cap B)$ où C'' est une résolvante de c et c' .

Nous avons donc $\forall \pi$ satisfait $C \wedge C'$ on a $N(c) \supseteq A$ et $N(c') \supseteq B$ d'où $N(c \wedge c') = N(c) \cap N(c') \supseteq A \cap B$ et $N(c'') \supseteq N(c \wedge c') \supseteq A \cap B$.

La A-B-résolution est donc correcte.

Théorème 6.4 (correction de la règle : affaiblissement sous ensembliste) :

Soit $C = (c, A)$ une clause multi-agents et $C' = (c, B)$ avec $B \subseteq A$ une clause obtenue à partir de (c, A) par la règle (affaiblissement sous ensembliste). Alors $C \models C'$.

Preuve 6.5 Toute distribution de possibilité multi-agents sur Ω satisfaisant (c, A) satisfait (c, B) (immédiate).

Ceci montre alors la correction de la règle (affaiblissement sous ensembliste).

Notation : Soient Φ un ensemble de clauses multi-agents et C une clause multi-agents. $\Phi \vdash_{Res} C$ si et seulement si C peut être obtenue par un nombre fini d'applications des règles (résolution sous ensembliste), (A-B-résolution) et (affaiblissement sous ensembliste) à partir des clauses de Φ et de leurs résolvantes.

Corollaire 6.1 (*correction de l'application en chaîne de la résolution*) : $[\Phi \vdash_{Res} C] \Rightarrow [\Phi \models C]$.

Preuve 6.6 *Elle est immédiate par induction sur la dérivation de C en utilisant le théorème précédent.*

Remarque 6.1 *La résolution sous ensembliste et la A - B -résolution recouvrent la résolution classique. En effet, si $A = B = k$ et par conséquent l'ensemble d'agents sera réduit à un seul agent nous retrouvons l'inférence de la clause c'' à partir des clauses c et c' du même agent.*

6.5 Réfutation par résolution

Etant donné un ensemble Ψ de formules du calcul propositionnel, nous cherchons à savoir s'il est satisfaisable ou non. La procédure la plus naturelle consiste à faire une table de vérité qui assure d'obtenir une réponse en un temps fini. Cependant, le temps de réponse de cette méthode augmente exponentiellement avec le nombre d'atomes. En effet, il est de l'ordre de 2^n , où n est le nombre d'atomes intervenant dans les formules.

La méthode de résolution de Robinson [Robinson 65] est plus simple. Il s'agit d'un algorithme simple de démonstration automatique.

A la base de la méthode de résolution, se trouve le principe de la réfutation. Le principe de la réfutation est le suivant : démontrer que b est une conséquence logique de a est par définition revient à montrer que $a \wedge \neg b$ mène à une contradiction.

Le mécanisme de résolution par réfutation en logique propositionnelle consiste à utiliser la résolution comme seule règle de déduction pour déduire la \perp à partir d'un ensemble de clauses.

Dans ce qui suit, nous présentons une méthode de démonstration automatique qui est complète pour la sémantique de la logique multi-agents.

6.5.1 La réfutation par résolution dans la logique multi-agents

Nous généralisons dans cette section la preuve par réfutation à la logique multi-agents. Le principe de la réfutation pour la logique multi-agents consiste à ajouter à l'ensemble de formules multi-agents Φ la négation de la formule b à prouver avec l'ensemble de tous les agents All (c'est-à-dire que nous supposons que b est crue par tous les agents). L'individuelle inconsistance de l'ensemble de formules multi-agents Φ' résultant sera le plus grand ensemble d'agents avec lequel b est déduite de Φ .

Ceci s'exprime formellement comme suit :

Théorème 6.5 (*Réfutation*) :

$$\Phi \models (b, B) \Leftrightarrow i - inc(\Phi \wedge (\neg b, All)) \supseteq B$$

Ce qui peut être écrit de manière équivalente $i - inc(\Phi \wedge (\neg b, All)) = \cup\{B, \Phi \models (b, B)\}$.

Preuve 6.7 (\Rightarrow):

Supposons que $\Phi \models (b, B)$, c'est-à-dire $\forall \pi$ satisfaisant Φ , $N(b) \supseteq B$ ce qui s'écrit encore : $\forall \omega$ tel que $\omega \models \neg b, \pi(\omega) \subseteq \overline{B}$.

Soit π satisfait $\Phi \wedge (\neg b, All)$, c'est-à-dire $\forall \omega$ tel que $\omega \models b, \pi(\omega) = \emptyset$.

Or π satisfait aussi Φ et $\forall \omega$ tel que $\omega \models \neg b, \pi(\omega) \subseteq \overline{B}$, d'où $N(\perp) = \cap \{\overline{\pi(\omega)}, \omega \models b \text{ ou } \omega \models \neg b\} \supseteq B$.

Nous déduisons alors que $i - inc(\Phi \wedge (\neg b, All)) \supseteq B$.

(\Leftarrow):

$i - inc(\Phi \wedge (\neg b, All)) \supseteq B$ alors :

$N(\perp) = \cap \{\overline{\pi(\omega)}, \omega \models b \text{ ou } \omega \models \neg b\} \supseteq B$ donc

$\pi(\omega) \subseteq \overline{B}$, donc π satisfait Φ et on a donc : $\forall \omega$ tel que $\omega \models \neg b$

Or, π satisfait $\Phi \wedge (\neg b, All)$ donc $\forall \omega$ tel que $\omega \models b, \pi(\omega) = \emptyset$. On a alors $\Phi \models (b, B)$.

La méthode de réfutation par résolution en logique multi-agents est donnée par l'algorithme suivant :

Début

Soit Φ une base de connaissances multi-agents et b une formule.

Mettre Φ sous forme clausale (Soit \mathcal{C} la forme clausale obtenue) ainsi que $\neg b$ (Soit c_1, \dots, c_p les clauses obtenues) ;

$\mathcal{C}' \leftarrow \mathcal{C} \cup \{(c_1, All), \dots, (c_p, All)\}$;

Chercher une déduction de (\perp, \mathcal{E}^*) par résolution à partir de \mathcal{C}' , avec \mathcal{E}^* le plus grand ;

Si ($\mathcal{E}^* = A$) **Alors**

$\Phi \models (b, A)$

 {C'est-à-dire $Ens(b, \Phi) = \mathcal{E}^*$ }.

Fin Si

Fin

Algorithme 8: Algorithme de réfutation par résolution

Chaque déduction de (\perp, \mathcal{E}) sera appelée \mathcal{E} -réfutation ; toute réfutation avec un ensemble d'agents maximal sera appelée réfutation optimale.

Exemple 6.2 Soit \mathcal{C} la base de connaissance multi-agents composée des quatre clauses multi-agents suivantes :

$C_1 : (p \vee q, All)$;

$C_2 : (\neg q, All)$;

$$C_3 : (p \vee r, A);$$

$$C_4 : (\neg r, B).$$

Initialement, il n'existe aucune réfutation à partir de \mathcal{C} . Soit maintenant à trouver le plus grand ensemble d'agents qui croit la formule p . Soit alors \mathcal{C}' l'ensemble des clauses possibilistes équivalent à $\mathcal{C}' = \mathcal{C} \cup \{(\neg p, All)\}$. Il existe, dans ce cas, deux réfutations possibles à partir de \mathcal{C}' , données par la figure 6.1 :

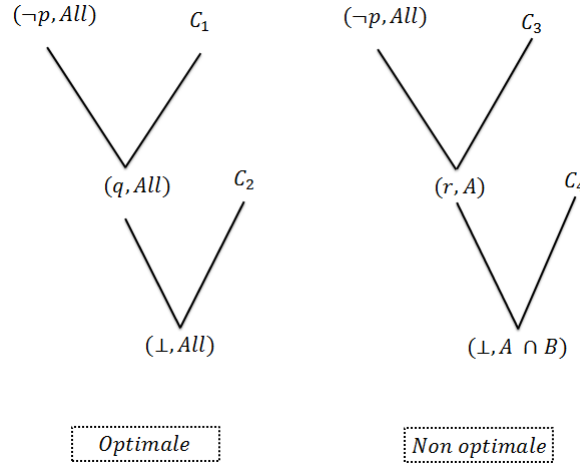


FIGURE 6.1 – Les deux réfutations possibles.

Nous concluons donc que : $\mathcal{C} \models (p, 0.9/A)$.

6.5.2 Correction et complétude de la réfutation par résolution

Le théorème suivant exprime la correction de la réfutation par résolution dans la logique multi-agents.

Théorème 6.6 *Pour toute base de connaissances multi-agents Φ , pour toute formule b , pour toute forme clausale \mathcal{C}' de $\Phi \wedge (\neg b, All)$ et pour tout ensemble $B \subseteq All$, nous avons :*

$$[\mathcal{C}' \vdash_{Res} (\perp, B)] \Rightarrow [\Phi \models (b, B)]$$

Preuve 6.8

$$\begin{aligned}
 \Phi \vdash_{Res} (b, B) \text{ par réfutation} &\Rightarrow \mathcal{C}' \vdash_{Res} (\perp, B) \\
 &\Rightarrow \mathcal{C}' \models (\perp, B) \text{ (thorèmes 6.4 et 6.3)} \\
 &\Rightarrow i-inc(\mathcal{C}') \supseteq B
 \end{aligned}$$

Or \mathcal{C}' est une forme clausale de $\Phi \wedge (\neg b, All)$ donc $i-inc(\Phi \wedge (\neg b, All)) \supseteq i-inc(\mathcal{C}') \supseteq B$ et donc par le (théorème 6.5) : $[\Phi \models (b, B)]$.

Les théorèmes suivants expriment la complétude de la résolution pour la réfutation dans la logique clausale multi-agents et la logique multi-agents :

Théorème 6.7 *Pour tout ensemble de clauses multi-agents \mathcal{C}' , si $i-inc(\mathcal{C}') = A$ avec $A \supset \emptyset$ alors $\mathcal{C}' \vdash_{Res} (\perp, A)$.*

Preuve 6.9 $i - inc(\mathcal{C}') = A$

$\Rightarrow \mathcal{C}' \equiv \mathcal{C}'_A$ (proposition 4.10) et donc $i - inc(\mathcal{C}'_A) \supseteq A$.

$\Rightarrow \mathcal{C}'_A^*$ est inconsistante (théorème 4.1).

\Rightarrow Il existe une réfutation par résolution \mathcal{R}^* à partir de \mathcal{C}'_A^* (complétude de la résolution pour la réfutation en logique classique).

Construisons alors la réfutation \mathcal{R} à partir de \mathcal{R}^* en rétablissant les sous-ensembles : toute étape de résolution $c_p, c_q \vdash c_r$ est remplacé par $(c_p, A_p), (c_q, A_q) \vdash (c_r, A_r)$ où les sous-ensembles sont calculés comme suit :

- Si c_i ($i = p$ ou $i = q$) est une clause initialement contenue dans \mathcal{C}'_A^* alors A_i est son sous-ensemble dans \mathcal{C}'_A ;
- sinon, A_i est le sous-ensemble associé à c_i dans l'étape de résolution qui a précédemment donnée c_i ;
- enfin, A_r est calculée par $A_r = A_p \cap A_q$.

La réfutation \mathcal{R} ainsi obtenue est une suite d'étapes de résolution appliquant la règle (A-B résolution). C'est donc une réfutation par résolution multi-agents.

Par ailleurs, toutes les clauses initialement contenues dans \mathcal{C}'_A ont un sous-ensemble plus grand ou égal à A . Par induction sur la réfutation (en remarquant que la seule règle d'inférence utilisée, appliquée à deux clauses dont le sous-ensemble est $\supseteq A$, produit une clause dont le sous ensemble est toujours $\supseteq A$), toutes les clauses dérivées au cours de la réfutation ont un même sous ensemble $\supseteq A$. Ceci est vraie en particulier pour la clause produite à la dernière étape, c'est-à-dire \perp .

Nous avons donc dérivé, par résolution multi-agents, (\perp, B) avec $B \supseteq A$ à partir de \mathcal{C}'_A , donc à partir de \mathcal{C}' c'est-à-dire que $\mathcal{C}' \vdash_{Res} (\perp, B)$ avec $B \supseteq A$.

Par la règle de (affaiblissement sous ensembliste), nous pouvons enfin inférer (\perp, B) d'où $\mathcal{C}' \vdash_{Res} (\perp, A)$.

Théorème 6.8 Pour toute base de connaissances multi-agents Φ , pour toute formule b , pour toute forme clausale \mathcal{C}' de $\Phi \wedge (\neg b, All)$ et $\forall B \supset \emptyset$, nous avons :

$$[\Phi \models (b, B)] \Rightarrow [\mathcal{C}' \vdash_{Res} (\perp, B)]$$

Preuve 6.10 $[\Phi \models (b, B)]$

$\Rightarrow i - inc(\Phi \wedge (\neg b, All)) \supseteq B$ (théorème 6.5)

$\Rightarrow i - inc(\mathcal{C}') = i - inc(\Phi \wedge (\neg b, All))$ (théorème 6.1) $\supseteq B$.

De $\mathcal{C}' \vdash_{Res} (\perp, i - inc(\mathcal{C}'))$ (théorème 6.7) et $i - inc(\mathcal{C}') \supseteq B$, nous déduisons $\mathcal{C}' \vdash_{Res} (\perp, B)$.

Définition 6.3 Une réfutation significative est une réfutation d'un ensemble non vide.

Les résultats suivants nous seront utiles pour la section suivante :

Proposition 6.1 *Toute réfutation significative a pour sous-ensemble d'agents le plus petit des sous-ensembles d'agents des clauses, utilisées dans la réfutation.*

Preuve 6.11 *Elle est immédiate d'après les règles de résolution.*

Proposition 6.2 *Soit \mathcal{C} un ensemble individuellement inconsistant de clauses multi-agents. Alors $i - inc(\mathcal{C})$ est égal à l'un des sous-ensembles des clauses de \mathcal{C} .*

Preuve 6.12 *Elle est immédiate d'après la proposition précédente et la complétude de la résolution multi-agents pour la réfutation.*

6.6 Stratégie de résolution

La procédure générale de réfutation par résolution multi-agents peut être raffinée au niveau de l'étape de recherche d'une déduction de (\perp, A) par résolution multi-agents, avec A le plus grand. Selon la manière dont les clauses à résoudre sont choisies, diverses stratégies de réfutation par résolution peuvent être définies. Une telle stratégie restreint ou ordonne les choix de clauses et littéraux sur lesquels sera tentée la résolution selon des critères qui sont propres à la stratégie.

Nous nous intéressons dans cette section à la recherche de stratégies permettant de trouver le plus rapidement possible une réfutation optimale.

Définition 6.4 *Une stratégie de résolution multi-agents est **un algorithme** destiné à trouver une réfutation optimale d'un ensemble de clauses de la logique clause multi-agents.*

Définition 6.5 *Une stratégie de résolution \mathcal{S} pour la logique clause multi-agents est **complète** si et seulement si pour tout ensemble individuellement inconsistant \mathcal{C} de clauses de la logique clause multi-agents, il existe une réfutation optimale par résolution à partir de \mathcal{C} compatible avec les critères de \mathcal{S} .*

Définition 6.6 *Une stratégie de résolution \mathcal{S} pour la logique clause multi-agents est **directement complète** si et seulement si pour tous ensemble individuellement inconsistant \mathcal{C} de clauses de la logique clause multi-agents, l'application de \mathcal{S} produit une réfutation optimale de \mathcal{C} en un temps fini.*

Définition 6.7 *Une stratégie de résolution \mathcal{S} pour la logique clause multi-agents **réellement directement complète** si non seulement son application à \mathcal{C} individuellement inconsistant produit une réfutation optimale de \mathcal{C} en la reconnaissant comme optimale en un temps fini.*

Définition 6.8 *Une stratégie de résolution \mathcal{S} pour la logique clause multi-agents est **décidable** si et seulement si elle est réellement directement complète et si, de plus, son application à un ensemble quelconque de clauses de la logique multi-agents s'arrête en un temps fini.*

Définition 6.9 *Une stratégie de résolution multi-agents sera dite **informée** si le choix des clauses pour l'étape de résolution à venir est guidé par les sous-ensembles. Sinon, la stratégie sera dite **aveugle**.*

6.6.1 Stratégies linéaires de type *pseudo* – A^* en logique multi-agents

Nous allons maintenant définir une classe de stratégies informées, à savoir les stratégies linéaires de type *pseudo* – A^* .

6.6.1.1 Réfutations par résolution multi-agents linéaires

La stratégie de résolution multi-agents linéaire procède de la façon suivante :

1. Le principe de résolution est appliqué entre :
 - une clause C_0 (clause centrale initiale) appartenant à l'ensemble \mathcal{C} (clauses d'entrée), individuellement inconsistant de clauses multi-agents. La clause C_0 est sélectionnée de telle façon à ce que $\mathcal{C} - C_0$ soit multi-agents consistant pour que la stratégie linéaire soit complète.
 - et une clause de \mathcal{C} .

La résolvante entre C_0 et \mathcal{C} est appelée clause centrale et est notée CC .
2. A chaque étape la clause centrale CC_{i+1} est obtenue en appliquant le principe de résolution entre CC_i (au départ seule C_0 est centrale) et une clause, dite : clause de bord, notée CB qui doit être choisie :
 - soit parmi les clauses de \mathcal{C} ;
 - soit parmi les résolvantes (donc les clauses centrales) ancêtre CC_j ($j \leq i$) de CC_{i+1} .

Notons qu'une clause centrale ne peut être résolue avec une autre clause centrale que si l'une est ancêtre de l'autre.

La construction de l'arbre de réfutation prend alors la forme donnée par la Figure 6.2 :

La Figure 6.3 illustre le cas où les noms des clauses de bord sont placés sur les arcs joignant les clauses centrales.

C'est à cause de l'allure du deuxième type de graphe que le terme de stratégies linéaires est utilisé. La linéarité provient bien du fait que de deux clauses centrales ne peuvent se résoudre ensemble que si l'une est ancêtre de l'autre.

Définition 6.10 *Une réfutation linéaire est une réfutation par résolution obtenue avec une stratégie linéaire.*

Proposition 6.3 *Soit \mathcal{C} un ensemble individuellement inconsistant de clauses multi-agents. Parmi les réfutations optimales de \mathcal{C} , il en existe au moins une qui soit linéaire.*

Preuve 6.13 *Soit : $i - inc(\mathcal{C}) = A$ avec $A \supset \emptyset$.*

$i - inc(\mathcal{C}_A) = i - inc(\mathcal{C}) = A$ (proposition 4.10).

Soit \mathcal{C}' un sous-ensemble minimal individuellement inconsistant de \mathcal{C}_A (c'est-à-dire que si l'une des clauses de \mathcal{C}' est supprimée, \mathcal{C}' il devient consistant).

Comme \mathcal{C}' ne contient que des clauses avec des sous-ensembles $\supseteq A$ (puisque'il est un sous-ensemble de \mathcal{C}_A), toute réfutation de \mathcal{C}' (et il en existe, puisque \mathcal{C}' est individuellement inconsistant) aura un sous-ensemble $\supseteq A$, d'où $i - inc(\mathcal{C}') \supseteq A$; comme il

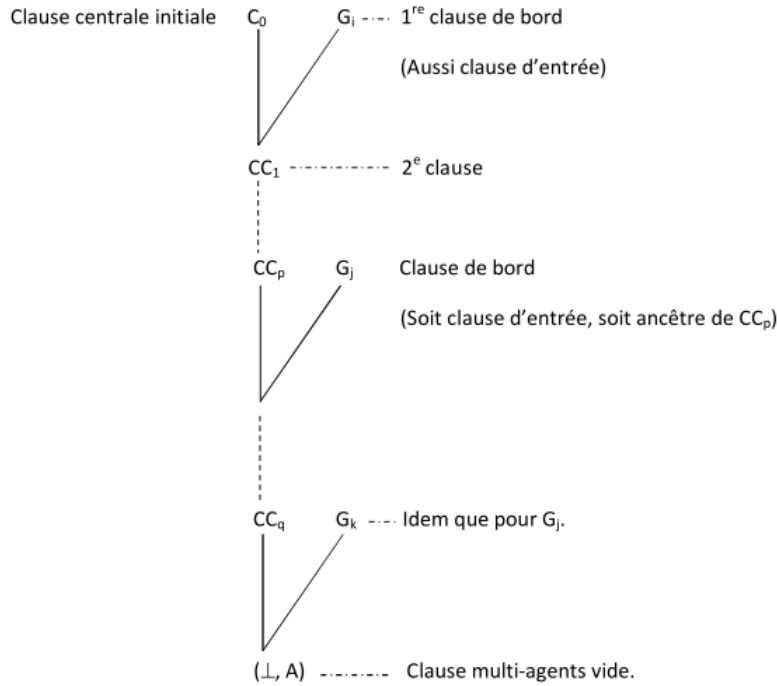


FIGURE 6.2 – Illustration de la construction de l'arbre de réfutation

n'est pas possible d'avoir $i - inc(\mathcal{C}') \supset A$ puisque \mathcal{C}' est un sous-ensemble de \mathcal{C} , alors $i - inc(\mathcal{C}') = A$.

Soit alors C_0 une clause quelconque de \mathcal{C}' . Puisque \mathcal{C}' est minimal individuellement inconsistant : $\mathcal{C}' - \{C_0\}$ est multi-agents consistant et sa projection classique $(\mathcal{C}' - \{C_0\})^$ l'est aussi.*

La complétude de la stratégie linéaire en logique classique entraîne alors qu'il existe une réfutation linéaire de \mathcal{C}'^ avec C_0^* une clause centrale initiale. Donc qu'il existe une réfutation linéaire de \mathcal{C}'^* avec C_0^* pour clause centrale initiale. Le sous ensemble de cette réfutation est $\supseteq A$ car \mathcal{C}' est un sous-ensemble de \mathcal{C}_A , et $\subseteq A$ car la résolution multi-agents est correcte et $i - inc(\mathcal{C}') \subseteq i - inc(\mathcal{C}_A) = A$. Cette réfutation de \mathcal{C}' , qui est donc aussi une réfutation de \mathcal{C} puisque \mathcal{C}' est un sous-ensemble de \mathcal{C} , a donc pour sous-ensemble $A = i - inc(\mathcal{C})$. Elle est donc optimale.*

Remarque 6.2 *Pour appliquer la stratégie de résolution il faut tout d'abord choisir une clause centrale de départ C_0 de façon que $\mathcal{C} - C_0$ soit multi-agents consistante. En effet, si n'importe quelle clause de \mathcal{C} est choisie comme clause centrale initiale, cela ne veut pas dire forcément que la réfutation utilisant C_0 trouvée est optimale.*

Exemple 6.3 *Soit la base multi agents Φ composée des clauses suivantes :*

- $C_1 : (a, A)$
- $C_2 : (\neg a, B)$
- $C_3 : (b, C)$
- $C_4 : (\neg b, D)$

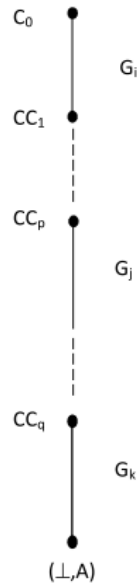


FIGURE 6.3 – Illustration de l'application d'une stratégie linéaire

Si C_1 est sélectionnée comme clause centrale initiale, nous aurons $(\perp, A \cap B)$. Cependant, cette réfutation n'est pas optimale car si C_3 est sélectionnée comme clause centrale initiale, nous aurons aussi $(\perp, C \cap D)$.

Dans le cas de la résolution par réfutation multi-agents l'ensemble multi-agents de clauses est consistant. La clause centrale initiale est la négation de la conclusion recherchée. Dans ce cas nous pouvons présumer que $\mathcal{C} - C_0$ est multi-agents consistant.

Soit \mathcal{C} l'ensemble des clauses équivalent à $\Phi \wedge \{(\neg b, All)\}$. Si la négation de b ne donne qu'une seule clause (c'est-à-dire si b est une formule atomique ou une conjonction de formules atomiques) alors cette clause est choisie comme clause centrale initiale et l'ensemble de clauses restant est équivalent à Φ , donc multi-agents consistant. Sinon, nous pouvons prendre pour clause centrale initiale successivement chacune des clauses issue de $\neg b$, comme en logique classique, et retenir le plus grand sous-ensemble des réfutations optimales obtenues.

L'individuelle inconsistance de \mathcal{C} peut donc être déterminée en utilisant une stratégie linéaire.

6.6.1.2 Stratégies linéaires informées

En intelligence artificielle, la notion de résolution de problème est étroitement associée aux techniques de recherche heuristiques dans les graphes. Ainsi, la recherche d'une réfutation par résolution par une stratégie linéaire peut s'exprimer en terme de recherche arborescente dans un espace d'états.

Un état est une description de la situation initiale dans laquelle le problème est posé, ou d'une situation intermédiaire quelconque rencontrée lors de sa résolution. L'ensemble des états constitue l'espace de représentation du problème. Dans notre cas, un état est défini par une clause centrale et l'ensemble des clauses centrales ancêtres de celle-ci.

A chaque état de l'arbre de recherche, nous associons le sous-ensemble d'agents de la dernière clause centrale produite, c.-à-d. qu'à $(C_0C_1\dots C_i)$, est associée le sous-ensemble d'agents $Ens(C_i)$. L'objectif est de chercher les états associés à une clause vide et avec le plus grand sous-ensemble d'agents.

Il existe alors une analogie avec la recherche dans les espaces d'états avec un coût. Elle est établie comme suit :

- L'état initial S_0 est défini par la clause centrale initiale C_0 et a pour coût $Ens(S_0)$;
- Le coût associé à l'arc $(C_0C_1\dots C_i) \rightarrow (C_0C_1, \dots, C_iC_{i+1})$ est l'ensemble associé à C_{i+1} ;
- Le coût global du chemin $C_0 \rightarrow C_1 \rightarrow \dots \rightarrow C_i$ est l'intersection des coût de ses arcs élémentaires ;
- Les états objectifs sont des états $(C_0C_1\dots C_i)$ tels que $C_i = (\perp, A_i)$ avec $A_i \neq \emptyset$;
- L'état $(C_0C_1\dots C_n)$ est développé en produisant toutes les résolvantes de C_n autorisées par la stratégie linéaire.

La recherche d'une réfutation avec le plus grand ensemble d'agents est alors équivalente à la recherche d'un chemin de coût maximal de l'état initial vers des états objectifs. Cependant, plusieurs différences existent :

- Le coût ici est à maximiser et non pas à minimiser. En effet, le but est de trouver le plus grand ensemble d'agents qui croient une formule.
- Les coûts ne sont pas additifs, mais ce sont des ensembles combinés par l'opérateur d'intersection.
- Etant donné que seul un ordre partiel existe entre les sous-ensembles, plusieurs états objectifs existent. Ces derniers seront ensuite combinés avec l'opérateur d'union.
- Si un ordre d'inclusion existe entre les sous-ensembles, le plus grand sous-ensemble est pris et l'autre chemin n'est jamais exploré, contrairement à la recherche dans les espaces d'états.
- Même si les fonctions d'évaluation des fils d'un noeud produisent les mêmes ensembles d'agents, tous les fils doivent être parcourus car différents résultats pourraient être obtenus.

La difficulté essentielle d'un algorithme de recherche porte sur l'étape de choix parmi l'ensemble des états connus et susceptibles d'être développés, celui qui le sera effectivement. Ce choix est déterminé en premier lieu par la stratégie d'organisation de la recherche qu'utilise l'algorithme.

Le choix de l'état qui sera développé est fait par rapport à l'ordre dans lequel les alternatives sont rencontrées, ou bien si un critère de coût est considéré, par rapport à une évaluation du coût de la solution potentielle associée à chaque état rencontré.

L'ensemble des alternatives est ordonné sous forme d'une pile et l'état choisi sera celui pour lequel la fonction d'évaluation a le coût maximal.

En effet, la recherche partiellement ordonnée est guidée par une fonction d'évaluation f qui est calculée ainsi : pour chaque état S de l'arbre de recherche, $f(S) = g(S) \cap h(S)$, où $g(S)$ est le coût du chemin de l'état initial vers S , et $h(S)$ une estimation du coût de

S vers un état objectif (qui a le plus grand ensemble d'agents). Une heuristique $h(S)$ est un moyen quelconque (un critère, une procédure ...) destinée à réduire les alternatives et guider les choix non-déterministes doivent développer faire un algorithme de recherche. Le plus souvent une heuristique exploite une information spécifique au problème posé. Elle n'est pertinente et efficace que pour ce problème-là, et ne peut pas être immédiatement généralisée à d'autres problèmes. Ainsi, le même algorithme de recherche devra utiliser, suivant les problèmes, différentes heuristiques.

L'algorithme suivant est proposé dans [Belhadi et All. 14]. C'est une adaptation de l'algorithme de [Yager 86].

```

Début

 $Open \leftarrow \{S_0\};$ 
 $Closed \leftarrow \{S_0\};$ 
 $bEns = \emptyset$ 
Tant que ( $Open \neq \emptyset$ ) faire
    Choisir un état  $S_n$  dans Open maximisant f
    Si ( $S_n$  est un état objectif) Alors
         $bEns = bEns \cup S_n$ 
    Sinon
        Développer le nœud  $S_n$  en créant l'ensemble  $E'_n$  des états produits.
        Si (Dans l'ensemble  $E'_n$  il existe des sous-ensembles incluent dans
        d'autres) Alors
            Les supprimer de  $E'_n$ 
        Fin Si
         $E_n \leftarrow E'_n \setminus Closed$ 
         $Open \leftarrow (Open - \{S_n\}) \cup E_n$ 
         $Closed \leftarrow Closed \cup \{S_n\}$ 
        Calculer f pour chaque nouvel état de Open
    Fin Si
Fait
Si ( $Open = \emptyset$ ) Alors
    | échec
Sinon
    | Afficher bEns
Fin Si

Fin

```

Algorithme 9: Algorithme de réfutation par résolution multi-agents linéaire

Théorème 6.9 *Supposons que le système de production multi-agents ne contient qu'un nombre fini d'états (c'est-à-dire que le nombre de clauses dérivable à partir des clauses initiales est fini, ce qui est le cas avec des clauses toutes propositionnelles). Appelons $h^*(S)$ le véritable coût du chemin optimal de l'état S vers un état but. Alors l'exécution de l'algorithme ci-dessus utilisant comme fonction d'évaluation $f(S) = g(S) \cap h(S)$ telle que la fonction heuristique h soit optimiste, c'est-à-dire $\forall S \ h(S) \supseteq h^*(S)$, se termine dans tous les cas et en donnant une réfutation optimale s'il en existe une (c'est-à-dire*

qu'il constitue une stratégie décidable).

Preuve 6.14 Voir [Yager 86]. Le théorème donné par Yager nous assure que sous les hypothèses précédentes, l'algorithme trouve une réfutation linéaire optimale à partir de la clause centrale initiale C_0 .

Lorsque nous cherchons une solution particulière satisfaisant un critère de coût, la qualité souhaitable pour un algorithme de recherche sera d'être admissible, i.e. de fournir la meilleure solution relativement à ce critère, s'il existe des solutions. De ce fait, si la condition $\forall S \ h(S) \supseteq h^*(S)$ est vérifiée, la fonction heuristique h est dite admissible.

6.6.1.3 Une famille de fonctions d'évaluation admissibles

Yager [Yager 86] donne également le résultat suivant : si h_1 et h_2 sont deux fonctions heuristiques, toutes les deux admissibles, telles que $\forall S \ h_2(S) \supseteq h_1(S) \supseteq (h^*(S))$, alors h_1 est plus informée que h_2 , en ce sens que toute clause produite par l'application de l'algorithme utilisant h_1 sera aussi produite par celui utilisant h_2 . La fonction constante $h(S) = All$ est donc la moins informée des fonctions heuristiques. Nous proposons maintenant une famille de fonctions heuristiques admissibles.

Soit \mathcal{C} un ensemble de clauses multi-agents, $R(\mathcal{C})$ désigne l'ensemble des clauses productibles en une application du principe de résolution à partir de \mathcal{C} , et $R^\infty(\mathcal{C}) = \cup_i R^i(\mathcal{C})$ désigne l'ensemble de toute clause productible en un nombre fini d'étapes à partir de \mathcal{C} ; par convention, pour toute clause classique $c = [\neg]L_1 \vee [\neg]L_2 \vee \dots \vee [\neg]L_n$, on écrira $c = l_1 \vee l_2 \vee \dots \vee l_n$, ou $l_i = [\neg]L_i$; on écrira aussi, $\neg l = [L_i \text{ si } l = \neg L_i]$, $\neg l = [\neg L_i \text{ si } l = L_i]$.

Posons $H_1(l) = \cup\{A, (c, A) \in R^\infty(\mathcal{C}), \neg l \in c\}$. Il s'agit d'obtenir la clause vide à partir de l'ensemble initial de clauses \mathcal{C} . Alors pour toute réfutation utilisant la clause c , pour tout littéral l de c , il faudra, pour dériver \perp , utiliser à un moment ou un autre une clause c' contenant un littéral $\neg l$, d'où : pour tout littéral l' de c une réfutation utilisant c aura un sous ensemble inclus ou égal à $H_1(l')$, c'est-à-dire en notant (par abus de notation) $h(C)$ pour $h(S_C)$, où S_C est l'état dont la dernière clause est C :

$$\forall C = (c, A), \forall l' \in c, h^*(C) \subseteq H_1(l')$$

Telle qu'elle est définie la fonction H_1 n'est pas directement calculable à partir de la donnée de \mathcal{C} . Cependant, H_1 est en fait une fonction statique étant donné que :

Proposition 6.4

$$H_1(l) = \cup\{A, (c, A) \in \mathcal{C}, \neg l \in c\}$$

Preuve 6.15 $H_1(l)$ est égal à un sous-ensemble de l'une des clauses de \mathcal{C} , soit $C_i = (c_i, A_i)$. Deux cas se présentent :

1. Soit $C_i \in \mathcal{C}$ et dans ce cas $\cup\{A, (c, A) \in \mathcal{C}, \neg l \in c\} \supseteq Ens(C_i) \supseteq H_1(l)$, l'égalité provenant ensuite de ce que $R^\infty(\mathcal{C})$ contient \mathcal{C} ;
2. Soit $C_i \notin \mathcal{C}$ et dans ce cas soient $C_i^{(1)}, C_i^{(2)}, \dots, C_i^{(q)}$ les ancêtres de C_i appartenant à \mathcal{C} .

Comme C_i contient $\neg l$, au moins l'une des ancêtres de C_i (sans perte de généralité, $C_i^{(1)}$) contient $\neg l$.

De plus, $Ens(C_i) = \cap_{j=1,\dots,q} Ens(C_i^{(j)}) \subseteq Ens(C_i^{(1)})$. D'autre part, $Ens(C_i) = \cup A, (c, A) \in R^\infty(\mathcal{C}), \neg l \in c \supseteq Ens(C_i^1)$ car $C_i^1 \in R^\infty$. D'où $Ens(C_i) = Ens(C_i^{(1)})$, et $\cup\{A, (c, A) \in \mathcal{C}, \neg l \in c\} \supseteq \cup\{A, (c, A) \in R^\infty(\mathcal{C}), \neg l \in c\} (= Ens(C_i)), R^\infty(\mathcal{C})$ contenant \mathcal{C} .

Par conséquent, H_1 est une fonction statique qui est calculée une fois pour toutes avant toute production de résolvantes.

Pour toute clause $C = (c, A)$, la fonction h_1 est définie comme suit :

$$h_1(C) = \cap\{H_1(l), l \in c\} (= All \text{ si } c = \perp)$$

Soit $C = (c, A)$, de $\forall l \in c, H_1(l) \supseteq h^*(C)$. $h_1(C) \supseteq h^*(C)$, c'est-à-dire que toute réfutation utilisant C avec un ensemble inclus ou égal à $h_1(C)$. La fonction h_1 est par conséquent admissible.

La suite de fonctions d'évaluation $(f_p)_{p \geq 0}$ est définie comme suit :

$$\begin{aligned} f_p &: R^\infty(\mathcal{C}) \rightarrow 2^{All} \\ h_0(C) &= All \\ f_p(C) &= \cap[Ens(C), h_p(C)] \\ h_{p+1}(C) &= \bigcap_{l \in c} \bigcup_{C' \in \mathcal{C}, \neg l \in c'} f_p(C') \end{aligned}$$

6.6.1.4 Exemple

Soit la base multi-agents Φ composée des clauses suivantes :

$$\begin{aligned} C_1 &: (\neg p \vee q, A) \\ C_2 &: (\neg p \vee q, B) \\ C_3 &: (\neg p \vee r, A) \\ C_4 &: (\neg q \vee r, All) \\ C_5 &: (p, All) \\ C_6 &: (q, A) \end{aligned}$$

Soit maintenant à trouver le plus grand ensemble d'agents qui croient à la formule r . Soit alors Φ' l'ensemble des clauses équivalentes à $\Phi' = \Phi \cup \{(\neg r, All)\}$.

Nous choisissons $C_0 = (\neg r, All)$ car $\Phi' - \{C_0\}$ est cohérente.

Les deux clauses C_3, C_4 contiennent le littéral r . Il faudrait alors calculer leurs fonctions d'évaluation.

Le plus grand ensemble qui maximise la fonction est All , car $A \subset All$. Effectivement, vu cet ordre d'inclusion, la branche correspondante ne sera pas explorée (voir Figure 6.4).

L'état suivant est $S_1 = (C_0 C_7)$. Il a comme coût : $Ens(C_4) \cap Ens(C_0) = Ens(C_4) = Ens(C_7) = All$.

Plusieurs chemins possibles existent à partir de cet état 6.5. Ces chemins seront alors explorés car leurs fonctions d'évaluation sont égales. Cela est dû à l'ordre partiel des

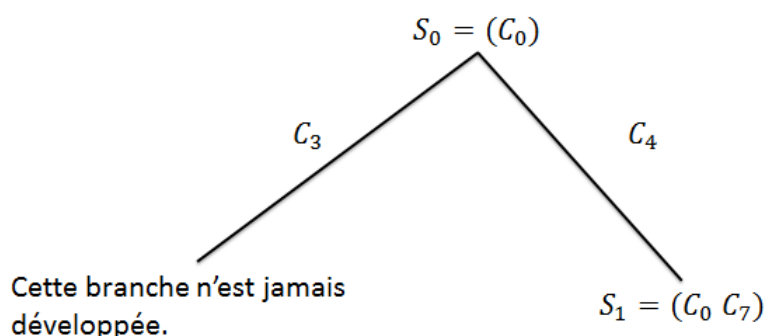


FIGURE 6.4 – Arbre de recherche -1-

sous-ensembles.

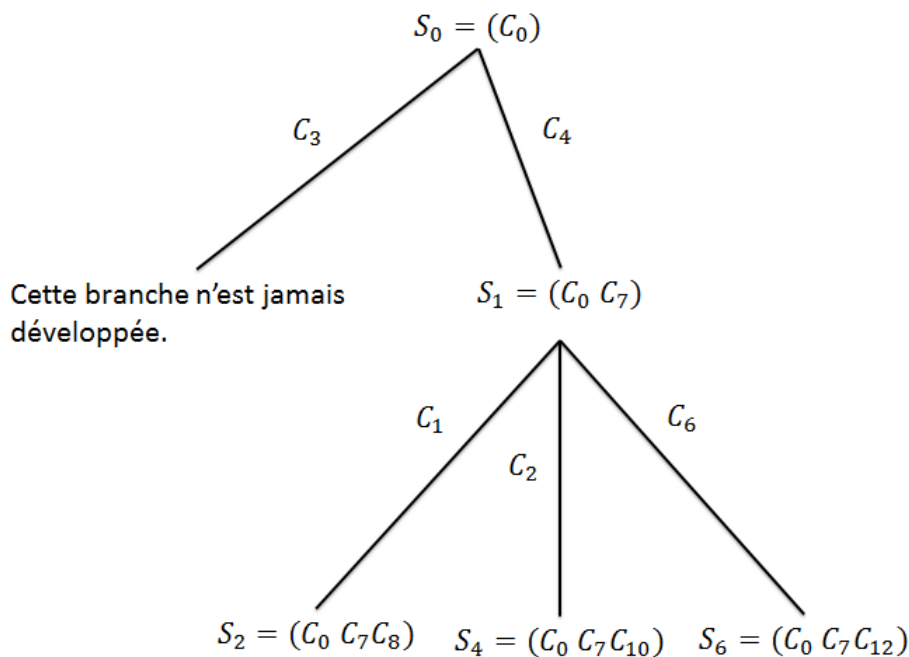


FIGURE 6.5 – Arbre de recherche -2-

Soit l'état suivant (voir Figure 6.6) : $S_2 = (C_0 C_7 C_8)$, il a comme coût : $Ens(C_7) \cap Ens(C_1) = Ens(C_1) = Ens(C_8) = A$.

Il existe un seul chemin ensuite avec la clause C_5 donc $S_3 = (C_0 C_7 C_8 C_9)$ et a comme coût : $Ens(C_8) \cap Ens(C_5) = Ens(C_8) = Ens(C_9) = A$. La clause C_9 est une contradiction, le premier état objectif est atteint.

l'état suivant (voir Figure 6.7) : $S_4 = (C_0 C_7 C_{10})$, il a comme coût : $Ens(C_7) \cap Ens(C_2) = Ens(C_2) = Ens(C_{10}) = B$.

Il existe un seul chemin ensuite avec la clause C_5 donc $S_5 = (C_0 C_7 C_{10} C_{11})$ et a comme coût : $Ens(C_{10}) \cap Ens(C_5) = Ens(C_{10}) = Ens(C_{11}) = B$. La clause C_{11} est

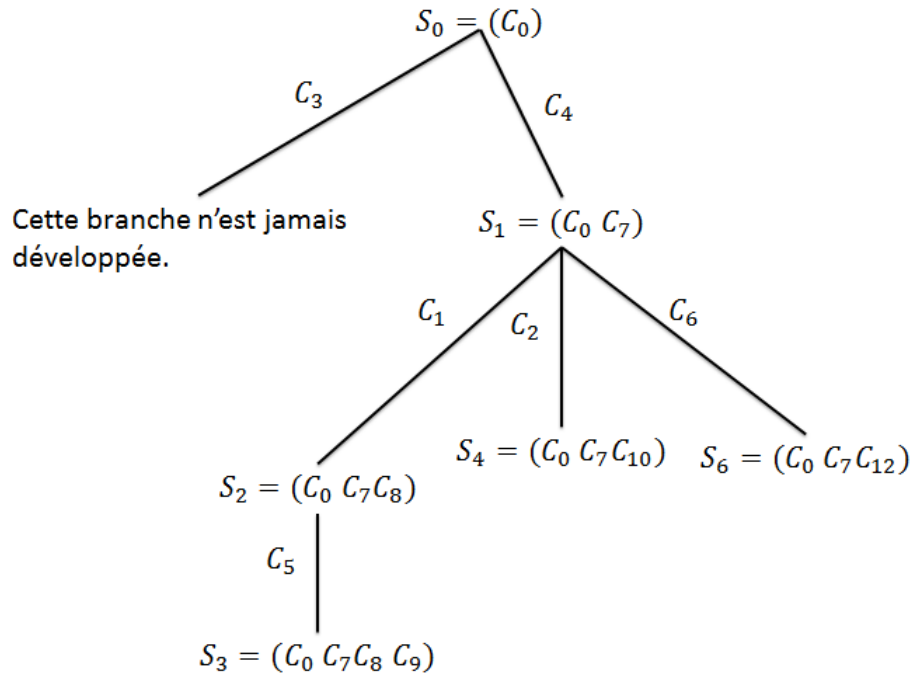


FIGURE 6.6 – Arbre de recherche -3-

une contradiction. De cela, le deuxième état objectif est atteint.

L'état suivant (voir Figure 7.6) : $S_6 = (C_0 C_7 C_{12})$, il a comme coût : $Ens(C_7) \cap Ens(C_6) = Ens(C_6) = Ens(C_{12}) = A$. Cependant, La clause C_{12} est une contradiction. Le dernier état objectif est atteint.

Ainsi, tous les états objectifs sont collectés. D'où $\Phi \models (r, A \cup B)$

6.7 Conclusion

L'algorithme de réfutation par résolution présenté dans ce chapitre est une adaptation directe de l'algorithme réfutation par résolution développé pour la logique possibiliste. En effet, une méthode de réfutation basée sur un principe de résolution généralisée utilisant une stratégie linéaire pour la logique multi-agents est proposée.

Après avoir abordé les modes de raisonnement pour la logique multi-agents, nous nous intéressons dans le chapitre suivant, aux modes de raisonnement pour la logique possibiliste multi-agents.

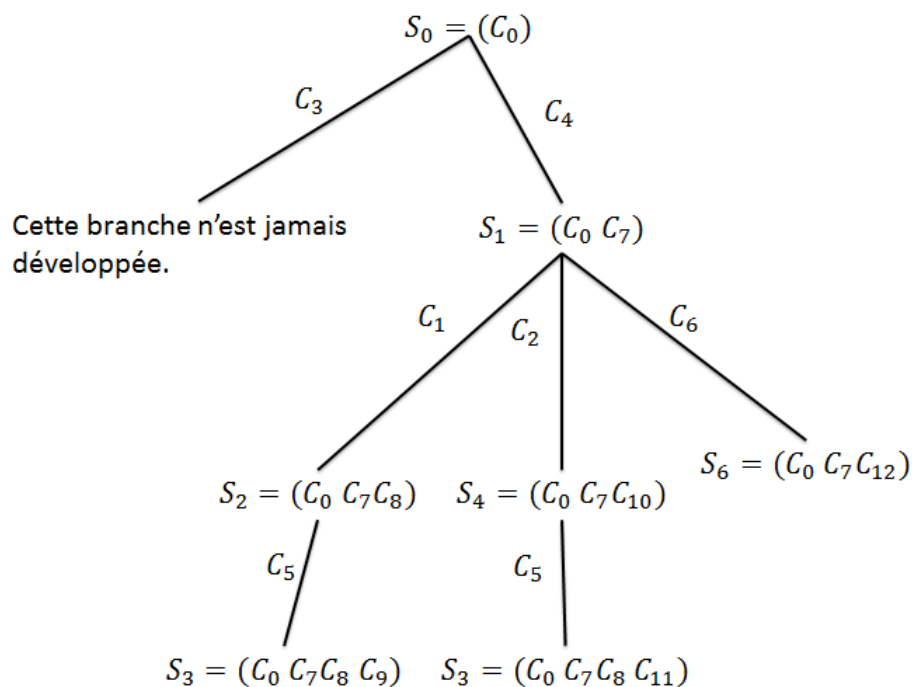


FIGURE 6.7 – Arbre de recherche -4-

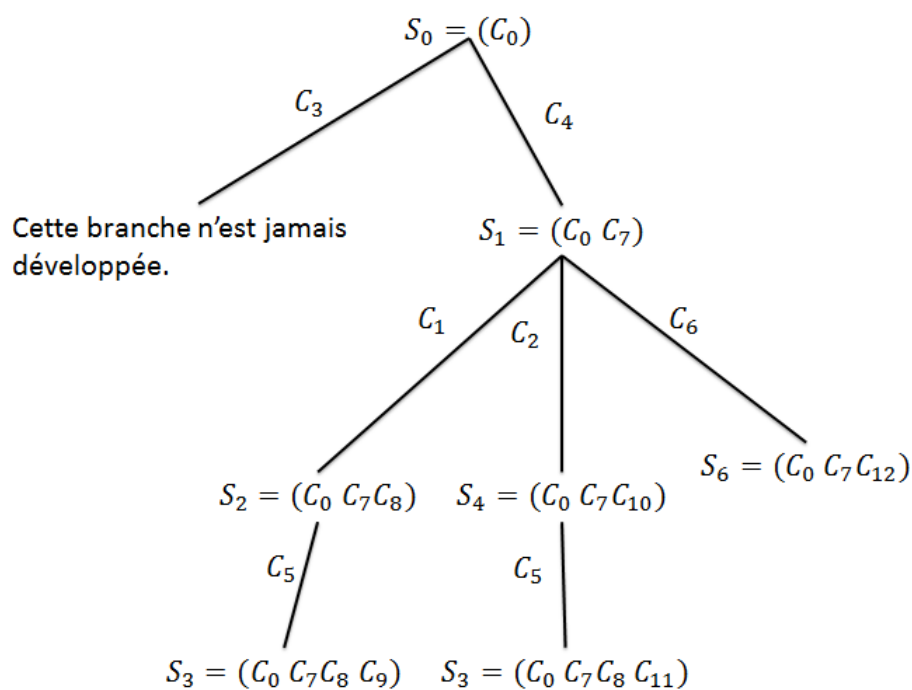


FIGURE 6.8 – Arbre de recherche -5-

Chapitre 7

La résolution possibiliste multi-agents

7.1 Introduction

Dans ce chapitre, nous étudions l'extension possibiliste multi-agents de la méthode de la réfutation par résolution. Pour cela, nous proposons une adaptation de l'algorithme de recherche A^* pour prendre en compte les sous-ensembles flous d'agents.

Ce chapitre est organisé comme suit. La section 2 fournit des définitions sur la forme clausale possibiliste multi-agents, nécessaire au processus de résolution. Afin d'étendre la résolution à la logique possibiliste multi-agents, la section 3 présente une méthode pour la transformation d'une base de connaissance possibiliste multi-agents en une forme clausale. Ainsi, la section 4 s'intéresse aux règles de résolution possibiliste multi-agents. Par la suite, nous aborderons dans la section 5 les fondements de base de la réfutation par résolution en logique possibiliste multi-agents. La section 6 présente les définitions de base utilisées dans la stratégie de résolution ainsi que l'algorithme de réfutation par résolution adéquats. Enfin, la section 7 donne un exemple pour illustrer la réfutation par résolution possibiliste multi-agents.

7.2 Motivations et définitions

L'objectif de la démonstration automatique est de développer des méthodes de calcul permettant de prouver qu'une ou plusieurs formules sont des conséquences logiques d'un ensemble de formules. La méthode que nous utilisons, dans le cadre de ce travail, est la résolution.

Pour pouvoir appliquer la résolution, il convient nécessairement de transformer les formules données en un type particulier de formules appelées clauses.

Définition 7.1 Une clause possibiliste multi-agents est un couple $(a, \alpha/A)$, où a est une clause de la logique classique, $\alpha \in]0, 1]$ et $A \subseteq All$.

Définition 7.2 Une formule clausale possibiliste multi-agents est une conjonction finie de clauses possibilistes multi-agents.

Exemple 7.1 $(\neg a \vee b, 1/All) \wedge (\neg b \vee c, 0.2/A) \wedge (a \vee \neg b \vee c, 0.5/B) \wedge (a \vee \neg b \vee \neg c, 0.9/D)$ est une forme clausale possibiliste multi-agents

7.3 Mise sous forme clausale dans la logique possibiliste multi-agents

Il est possible de transformer, d'une manière totalement automatique, n'importe quelle formule en logique possibiliste multi-agents Φ en un ensemble (ou conjonction) de clauses possibilistes multi-agents \mathcal{C} telle que \mathcal{C} est graduellement individuellement inconsistante si et seulement si Φ l'est aussi.

Début

Φ conjonction de formules possibiliste multi-agents : $\Phi = \wedge\{(a_i, \alpha_i/A_i), i = 1..n\}$

Pour $i = 1..n$ **faire**

 | mettre a_i sous forme clausale : $a_i = \wedge\{c_{ij}, j = 1..m_i\}$

Fin Pour

$\mathcal{C}(\Phi) = \wedge\{\wedge\{c_{ij}, j = 1..m_i\}, i = 1..n\}$ est une forme clausale de Φ .

Fin

Algorithme 10: Algorithme de mise sous forme clausale pour la logique possibiliste multi-agents

Exemple 7.2 $(\neg(a \wedge b), 0.9/A), (a \wedge (b \vee c), 0.4/B)$ est transformée en $(\neg a \vee \neg b, 0.9/A) \wedge (a \vee b, 0.4/B) \wedge (a \vee c, 0.4/B)$

Théorème 7.1 Soit Φ un ensemble de formules possibilistes multi-agents et $\mathcal{C}(\Phi)$ sa forme clausale déterminée par l'algorithme précédent. Alors Φ et $\mathcal{C}(\Phi)$ ont la même graduelle individuelle inconsistante.

Lemme 7.1 – $\mathcal{C}(\Phi_{\alpha/A})$ (Respectivement $\mathcal{C}(\Phi_{\alpha/A}^*)$) est une forme clausale de $\Phi_{\alpha/A}$ (Respectivement $\Phi_{\alpha/A}^*$).

– $\mathcal{C}(\Phi)^*$ est une forme clausale de Φ^* .

Preuve 7.1 Elle est immédiate.

Preuve 7.2 La preuve du théorème 7.1 est :

$$\begin{aligned}
g - i - inc(\Phi) = \alpha/A &\Leftrightarrow \Phi_{\alpha/A}^* \text{ est graduellement individuellement inconsistante et } \\
&\quad \Phi_{\alpha/A}^* \text{ est multi-agents consistante.} \\
&\Leftrightarrow \mathcal{C}(\Phi_{\alpha/A}^*) \text{ est graduellement individuellement inconsistante et } \\
&\quad \mathcal{C}(\Phi_{\alpha/A}^*) \text{ est multi-agents consistante ou} \\
&\quad \mathcal{C}(\Phi_{\alpha/A}^*) \text{ et } \mathcal{C}(\Phi_{\alpha/A}^*) \text{ sont des formes clausales respectivement de } \Phi_{\alpha/A}^* \text{ et } \Phi_{\alpha/A}^* \\
&\quad \text{(d'après l'équivalence de l'inconsistance d'une formule et de celle de sa} \\
&\quad \text{forme clausale en logique classique).} \\
&\Leftrightarrow \mathcal{C}(\Phi_{\alpha/A})^* \text{ est graduellement individuellement inconsistante et } \\
&\quad \mathcal{C}(\Phi_{\alpha/A})^* \text{ est multi-agents consistante (lemme 7.1).} \\
&\Leftrightarrow \mathcal{C}(\Phi)_{\alpha/A}^* \text{ est graduellement individuellement inconsistante et } \\
&\quad \mathcal{C}(\Phi)_{\alpha/A}^* \text{ multi-agents consistante (lemme 7.1).} \\
&\Leftrightarrow g - i - inc(\mathcal{C}(\Phi)) = \alpha/A \text{ (corollaire 5.3).}
\end{aligned}$$

7.4 Les règles de résolution

7.4.1 Les règles de la logique possibiliste multi-agents

Soient $(c, \alpha/A)$ et $(c', \beta/B)$ deux clauses possibilistes multi-agents, et soit c'' une résolvente quelconque de c et c' . Les règles suivantes sont appelées règles de résolution possibiliste multi-agents.

$(c, \alpha/A), (c', \alpha/A) \vdash (c'', \alpha/A)$ (résolution sous ensembliste graduelle)

$(c, \alpha/A), (c', \beta/B) \vdash (c'', \min(\alpha, \beta)/(A \cap B))$ (graduella A-B-résolution)

Nous introduisons la règle suivante, appelée règle de l'affaiblissement sous ensembliste graduel :

Si $\beta/B \subseteq \alpha/A$ alors $(c, \alpha/A) \vdash (c, \beta/B)$ (affaiblissement sous ensembliste graduel)

7.4.2 Correction des règles de la logique possibiliste multi-agents

Les résultats suivants justifient les règles (résolution sous ensembliste graduelle) et (graduella A-B-résolution) en montrant qu'elles sont bien en accord avec la logique possibiliste multi-agents :

Théorème 7.2 (correction de la règle : résolution sous ensembliste graduelle)

Soient c et c' deux clauses possibilistes multi-agents et c'' une clause obtenue par l'application de la (résolution sous ensembliste graduelle) à partir de c et c' . Alors : $c, c' \models c''$.

Preuve 7.3 Elle est immédiate.

Théorème 7.3 (correction de la règle : A-B-résolution graduelle) :

Soient c et c' deux clauses possibilistes multi-agents et c'' une clause obtenue par l'application de la A-B-résolution graduelle à partir de c et c' . Alors : $c, c' \models c''$.

Preuve 7.4 Soient $C = (c, \alpha/A)$, $C' = (c', \beta/B)$ et $C'' = (c'', \min(\alpha, \beta)/A \cap B)$ où C'' est une résolvente de c et c' .

Nous avons $\forall \pi$ satisfait $C \wedge C'$. Comme $N(c) \supseteq \alpha/A$ et $N(c') \supseteq \beta/B$ d'où $N(c \wedge c') = N(c) \cap N(c') \supseteq \alpha/A \cap \beta/B$ et $N(c'') \supseteq N(c \wedge c') \supseteq \min(\alpha, \beta)/A \cap B$.

La A-B-résolution graduelle est donc correcte.

Théorème 7.4 (correction de la règle : affaiblissement sous ensembliste graduel) :

soit $C = (c, \alpha/A)$ une clause possibiliste multi-agents et $C' = (c, \beta/B)$ avec $\beta/B \subseteq \alpha/A$ une clause obtenue à partir de $(c, \alpha/A)$ par la règle (affaiblissement sous ensembliste graduel). Alors $C \models C'$.

Preuve 7.5 Toute distribution de possibilité multi-agents graduée sur Ω satisfaisant $(c, \alpha/A)$ satisfait $(c, \beta/B)$ (immédiate).

Ceci montre alors la correction de la règle : (affaiblissement sous ensembliste graduel).

Notation 7.1 Soit Φ un ensemble de clauses possibilistes multi-agents et C une clause possibiliste multi-agents ; nous noterons : $\Phi \vdash_{Res} C$ si et seulement si C peut être obtenue par nombre fini d'applications des règles (résolution sous ensembliste graduelle), (graduelle A-B-résolution) et (affaiblissement sous ensembliste graduel) à partir des clauses de Φ et de leurs résolvantes.

Corollaire 7.1 (correction de l'application en chaîne de la résolution) : $[\Phi \vdash_{Res} C] \Rightarrow [\Phi \models C]$.

Preuve 7.6 Elle est immédiate par induction sur la dérivation de C en utilisant le théorème précédent.

7.5 Réfutation par résolution

7.5.1 La réfutation par résolution dans la logique possibiliste multi-agents

Le principe de la réfutation est d'ajouter à l'ensemble des formules possibilistes multi-agents Φ la négation de la formule b à prouver avec le degré 1 et l'ensemble de tous les agents All , c'est-à-dire que tous les agents supposent que b est certainement fausse.

La graduelle-individuelle inconsistance de l'ensemble des formules possibiliste multi-agents Φ' résultante sera le plus grand ensemble flou d'agents avec lequel b est déduite de Φ . Ce qui s'écrit plus formellement comme suit :

Théorème 7.5 (Réfutation) :

$$\Phi \models (b, \beta/B) \Leftrightarrow g - i - inc(\Phi \wedge (\neg b, 1/All)) \supseteq \beta/B$$

Ce qui peut être écrit de manière équivalente $g - i - inc(\Phi \wedge (\neg b, 1/All)) = \bigcup \{\beta/B, \Phi \models (b, \beta/B)\}$.

Preuve 7.7 (\Rightarrow):

Supposons que $\Phi \models (b, \beta/B)$, c'est-à-dire $\forall \pi$ satisfaisant Φ , $N(b) \supseteq \beta/B$ ce qui s'écrit encore : $\forall \omega$ tel que $\omega \models \neg b, \pi(\omega) \subseteq \overline{\beta/B}$.

Soit π satisfait $\Phi \wedge (\neg b, 1/All)$, c'est-à-dire $\forall \omega$ tel que $\omega \models b, \pi(\omega) = 0/\emptyset$.

Or π satisfait aussi Φ et on a donc aussi $\forall \omega$ tel que $\omega \models \neg b, \pi(\omega) \subseteq \overline{\beta/B}$, d'où $N(\perp) = \bigcap \{\overline{\pi(\omega)}, \omega \models b \text{ ou } \omega \models \neg b\} \supseteq \beta/B$.

Nous déduisons alors que $g - i - inc(\Phi \wedge (\neg b, 1/All)) \supseteq \beta/B$.

(\Leftarrow): $g - i - inc(\Phi \wedge (\neg b, 1/All)) \supseteq \beta/B$, donc

$N(\perp) = \bigcap \{\overline{\pi(\omega)}, \omega \models b \text{ ou } \omega \models \neg b\} \supseteq \beta/B$, d'où :

$\forall \omega, \omega \models \neg b, \pi(\omega) \subseteq \overline{\beta/B}$, alors : π satisfait Φ .

Aussi, $\forall \omega, \omega \models b, \pi(\omega) = 0/\emptyset$. Donc : π satisfait $\Phi \wedge (\neg b, 1/All)$. Alors : $\Phi \models (b, \beta/B)$.

La méthode de réfutation par résolution en logique possibiliste multi-agents s'écrit alors :

Début

Soit Φ une base de connaissances possibiliste multi-agents et b une formule.

Mettre Φ sous forme clausale (Soit \mathcal{C} la forme clausale obtenue) ainsi que $\neg b$ (Soit c_1, \dots, c_p les clauses obtenues) ;

$\mathcal{C}' \leftarrow \mathcal{C} \cup \{(c_1, 1/All), \dots, (c_p, 1/All)\}$;

Chercher une déduction de (\perp, \mathcal{EF}^*) par résolution à partir de \mathcal{C}' , avec \mathcal{EF}^* le plus grand ;

Si ($\mathcal{EF}^* = \alpha/A$) **Alors**

$\Phi \models (b, \alpha/A)$

$\{\mathcal{C}' \text{ est-à-dire } EnsF(b, \Phi) = \mathcal{EF}^*\}$

Fin Si

Fin

Algorithme 11: Algorithme de réfutation par résolution possibiliste multi-agents

Chaque déduction de (\perp, \mathcal{EF}) sera appelée \mathcal{EF} -réfutation ; toute réfutation avec un ensemble d'agents flou le plus grand sera appelée réfutation optimale.

Exemple 7.3 *Supposons que nous disposons de la base de connaissances multi-agents possibiliste \mathcal{C} composée des clauses multi-agents possibiliste suivantes :*

$$C_1 : (p \vee q, 1/All);$$

$$C_2 : (\neg q, 0.9/A);$$

$$C_3 : (p \vee r, 0.3/A);$$

$$C_4 : (\neg r, 0.3/B).$$

Initialement, il n'existe aucune réfutation à partir de \mathcal{C} . Soit maintenant à trouver le plus grand ensemble d'agents qui croient la formule p avec le plus grand degré. Soit alors \mathcal{C}' l'ensemble des clauses possibilistes multi-agents équivalent à $\mathcal{C}' = \mathcal{C} \cup \{(\neg p, 1/All)\}$. Il existe dans ce cas deux réfutations possibles à partir de \mathcal{C}' , qui sont illustrées par la Figure 7.1.

Nous concluons que $\mathcal{C} \models (p, All)$.

7.5.2 Correction et complétude de la réfutation par résolution

Le théorème suivant exprime la correction de la réfutation par résolution dans la logique possibiliste multi-agents.

Théorème 7.6 *Pour toute base de connaissances possibiliste multi-agents Φ , pour toute formule b , pour toute forme clausale \mathcal{C}' de $\Phi \wedge (\neg b, 1/All)$, pour tout $\alpha \in [0, 1]$ et pour toute ensemble $A \in 2^{All}$, nous avons :*

$$[\mathcal{C}' \vdash_{Res} (\perp, \alpha/A)] \Rightarrow [\Phi \models (b, \alpha/A)]$$

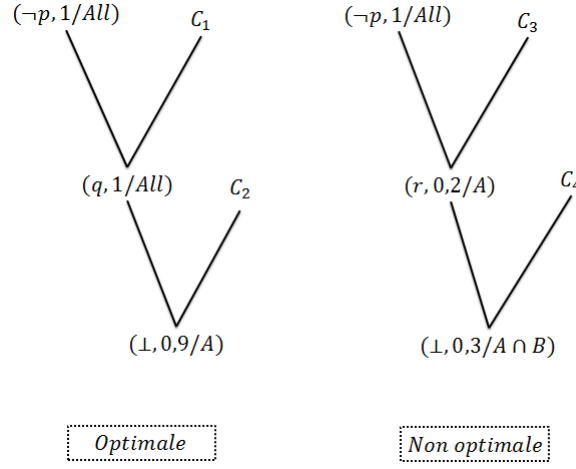


FIGURE 7.1 – Réfutation optimale vs réfutation non optimale de l'exemple 7.3.

Preuve 7.8

$$\begin{aligned}
\Phi \vdash_{Res} (b, \beta/B) \text{ par réfutation} &\Rightarrow \mathcal{C}' \vdash_{Res} (\perp, \beta/B) \\
&\Rightarrow \mathcal{C}' \models (\perp, \beta/B) \text{ (7.3 et 7.4)} \\
&\Rightarrow g - i - inc(\mathcal{C}') \supseteq \beta/B
\end{aligned}$$

Or \mathcal{C}' est une forme clausale de $\Phi \wedge (\neg b, 1/All)$ donc $i - inc(\Phi \wedge (\neg b, 1/All)) \supseteq g - i - inc(\mathcal{C}') \supseteq \beta/B$ et donc par le (théorème 7.5) : $[\Phi \models (b, \beta/B)]$.

Les théorèmes suivants expriment la complétude de la résolution pour la réfutation dans la logique clausale possibiliste multi-agents et la logique possibiliste multi-agents :

Théorème 7.7 *Pour tout ensemble de clauses possibilistes multi-agents \mathcal{C}' , si $g - i - inc(\mathcal{C}') = \alpha/A$ avec $\alpha/A \neq 0/\emptyset$ alors $\mathcal{C}' \vdash_{Res} (\perp, \alpha/A)$.*

Preuve 7.9 $g - i - inc(\mathcal{C}') = \alpha/A$

$\Rightarrow \mathcal{C}' \equiv \mathcal{C}'_{\alpha/A}$ (proposition 5.10) et donc $g - i - inc(\mathcal{C}'_{\alpha/A}) \supseteq \alpha/A$.
 $\Rightarrow \mathcal{C}'_{\alpha/A}$ est inconsistante (théorème 5.1).
 \Rightarrow Il existe une réfutation par résolution \mathcal{R}^* à partir de $\mathcal{C}'_{\alpha/A}$ (complétude de la résolution pour la réfutation en logique classique).

Construisons alors la réfutation \mathcal{R} à partir de \mathcal{R}^* en rétablissant les sous-ensembles flous : toute étape de résolution $c_p, c_q \vdash c_r$ est remplacée par $(c_p, \alpha_p/A_p), (c_q, \alpha_q/A_q) \vdash (c_r, \alpha_r/A_r)$ où les sous-ensembles flous sont calculés comme suit :

- c_i ($i = p$ ou $i = q$) est une clause initialement contenue dans $\mathcal{C}'_{\alpha/A}$ alors α_i/A_i est son sous-ensemble flou dans $\mathcal{C}'_{\alpha/A}$;
- sinon, α_i/A_i est le sous-ensemble associé à c_i dans l'étape de résolution qui a précédemment donnée c_i ;
- enfin, α_r/A_r est calculée par $\alpha_r/A_r = \alpha_p/A_p \cap \alpha_q/A_q$.

La réfutation \mathcal{R} ainsi obtenue est une suite d'étapes de résolution appliquant la règle (A-B résolution Graduelle), c'est donc une réfutation par résolution possibiliste multi-agents.

Par ailleurs, toutes les clauses initialement contenues dans $\mathcal{C}'_{\alpha/A}$ ont un sous-ensemble flou plus grand ou égal à α/A . Par induction sur la réfutation (en remarquant que la seule règle d'inférence utilisée, appliquée à deux clauses dont le sous-ensemble flou est $\supseteq \alpha/A$) produit une clause dont le sous ensemble flou est toujours $\supseteq \alpha/A$), toutes les clauses dérivées au cours de la réfutation ont un même sous ensemble $\supseteq \alpha/A$. Ceci est vrai en particulier pour la clause produite à la dernière étape, c'est-à-dire \perp .

Nous avons donc dérivé par résolution possibiliste multi-agents $(\perp, \beta/B)$ avec $\beta/B \supseteq \alpha/A$ à partir de $\mathcal{C}'_{\alpha/A}$, donc à partir de \mathcal{C}' c'est-à-dire que $\mathcal{C}' \vdash_{Res} (\perp, \beta/B)$ avec $\beta/B \supseteq \alpha/A$.

Par la règle de (affaiblissement sous ensembliste graduelle), nous pouvons donc enfin inférer $(\perp, \alpha/A)$ d'où $\mathcal{C}' \vdash_{Res} (\perp, \alpha/A)$.

Théorème 7.8 Pour toute base de connaissance possibiliste multi-agents Φ , pour toute formule b , pour toute forme clausale \mathcal{C}' de $\Phi \wedge (\neg b, 1/All)$ et $\forall \beta/B \neq 0/\emptyset$ nous avons :

$$[\Phi \models (b, \beta/B)] \Rightarrow [\mathcal{C}' \vdash_{Res} (\perp, \beta/B)]$$

Preuve 7.10 $[\Phi \models (b, \beta/B)]$

$$\begin{aligned} &\Rightarrow g - i - inc(\Phi \wedge (\neg b, 1/All)) \supseteq \beta/B \text{ (théorème 7.5)} \\ &\Rightarrow g - i - inc(\mathcal{C}') = g - i - inc(\Phi \wedge (\neg b, 1/All)) \text{ (théorème 7.1)} \supseteq \beta/B. \end{aligned}$$

De $\mathcal{C}' \vdash_{Res} (\perp, g - i - inc(\mathcal{C}'))$ donné par (théorème 7.7) et $g - i - inc(\mathcal{C}') \supseteq \beta/B$, nous déduisons $\mathcal{C}' \vdash_{Res} (\perp, \beta/B)$.

Définition 7.3 Une réfutation significative est une réfutation d'un ensemble flou non vide.

7.6 Stratégie de résolution

Nous avons vu que le problème majeur pour la déduction possibiliste multi-agents consiste à calculer le plus grand ensemble flou d'agents. Ce dernier, peut être calculé en recherchant une réfutation par résolution la plus grande, d'un ensemble de clause possibilistes multi-agents. Dans cette, section nous abordons le problème de rechercher une réfutation optimale en pratique.

Définition 7.4 Une stratégie de résolution possibiliste multi-agents est un **algorithme** qui a pour objectif de trouver une réfutation optimale d'un ensemble de clauses de la logique clausale possibiliste multi-agents. Une stratégie de résolution possibiliste multi-agents \mathcal{S} est :

- **complète** si et seulement si pour n'importe quel \mathcal{C} tel que $g - i - inc \supset 0/\emptyset$, il existe une réfutation optimale par résolution à partir de \mathcal{C} compatible avec les critères de \mathcal{S} .

- **directement complète** si et seulement si son application à \mathcal{C} tel que $g - i - inc \supset 0/\emptyset$, mène nécessairement à une réfutation optimale en un temps fini.
- **réellement directement complète** si non seulement son application à \mathcal{C} tel que $g - i - inc \supset 0/\emptyset$ non seulement mène à une réfutation optimale en un temps fini, mais aussi qu'elle soit reconnue comme optimale.
- **décidable** si et seulement si elle est réellement directement complète et si de plus elle s'arrête après un temps fini même si \mathcal{C} est graduellement individuellement inconsistant.
- **informée** si le choix des clauses pour l'étape de résolution à venir est guidé par les ensembles flous d'agents.
- **aveugle** si le choix des clauses pour l'étape de résolution à venir n'est pas guidé par les ensembles flous d'agents.

7.6.1 Stratégie informée en logique possibiliste multi-agents

Nous définissons dans cette section une classe de stratégie informée dite : stratégie linéaire de type *pseudo* – A^* .

7.6.1.1 Réfutations par résolution possibiliste multi-agents linéaires

La définition de la stratégie de résolution linéaire pour la logique possibiliste multi-agents est la même que pour que la logique classique : soit \mathcal{C} un ensemble graduellement individuellement inconsistant de clauses de la logique clausale possibiliste multi-agents ; nous distinguons une clause C_0 de \mathcal{C} (choisie telle que $\mathcal{C} - C_0$ soit graduellement individuellement consistant) qui est appelée clause centrale initiale. Les stratégies linéaires admettent seulement les résolutions entre : une clause CC descendante de C_0 (appelée clause centrale) et une clause qui est soit : parmi les clauses de \mathcal{C} (dites clauses d'entrée) ou descendante C_0 . Une réfutation linéaire une réfutation par résolution obtenue par l'application d'une stratégie linéaire.

Proposition 7.1 *Soit \mathcal{C} un ensemble graduellement individuellement inconsistant de clauses possibiliste multi-agents. Parmi les réfutations optimales de \mathcal{C} il en existe au moins une qui soit linéaire.*

Preuve 7.11

Cependant ce résultat ne veut pas dire que si l'on prend une clause quelconque C_0 de \mathcal{C} pour clause centrale initiale, alors une réfutation linéaire optimale utilisant C_0 pour clause centrale initiale sera une réfutation optimale de \mathcal{C} . Pour cela il faut s'assurer que $\mathcal{C} - C_0$ est multi-agents consistante (c'est-à-dire $g - i - inc(\mathcal{C} - \{C_0\}) = 0/\emptyset$).

Exemple 7.4 *Soit la base possibiliste multi agents Φ composée des clauses suivantes :*

- $C_1 : (a, 0.9/A)$
- $C_2 : (\neg a, 0.8/B)$
- $C_3 : (b, 0.3/C)$
- $C_4 : (\neg b, 0.1/D)$

Si nous choisissons C_1 comme clause centrale initiale, nous aurons $(\perp, 0.8/A \cap B)$. Cependant, cette réfutation n'est pas optimale car si nous choisissons C_3 comme clause centrale initiale nous aurons aussi $(\perp, 0.1/C \cap D)$.

Cette dernière condition est garantie dans le cas où \mathcal{C} est obtenue par l'ajout à une base de connaissances possibiliste multi-agent, multi-agents consistante une clause qui correspond à la négation de la formule b à prouver. Cette clause est alors choisie comme clause centrale initiale. Si la négation de b donne plusieurs clauses alors il est possible de prendre pour clause centrale initiale successivement chacune des clauses issue de $\neg b$, et alors retenir le plus grand ensemble flou d'agents des réfutations optimales obtenues.

La graduelle individuelle inconsistance de \mathcal{C} peut donc être déterminée en utilisant une stratégie linéaire.

7.6.1.2 Stratégies linéaires informées

La recherche d'une réfutation par résolution par une stratégie linéaire peut s'exprimer en terme de recherche arborescente dans un espace d'états. Un état est défini par une clause centrale et l'ensemble des clauses centrales ancêtres de celle-ci.

A chaque état de l'arbre de recherche, est associé l'ensemble flou d'agents de la dernière clause centrale produite.

Par conséquent, il faudra rechercher un état objectif avec le plus grand ensemble flou d'agents, où un état objectif a la forme : $(C_1 \dots, (\perp, \beta/B))$.

La recherche d'une réfutation optimale peut être vue comme une recherche d'un état objectif optimal avec un cout union-intersection flou. A savoir :

- L'état initial S_0 est défini par la clause centrale initiale C_0 et a pour coût $g(S_0) = EnsF(S_0)$;
- Le coût associé à l'arc $(C_0 C_1 \dots C_i) \rightarrow (C_0 C_1, \dots, C_i C_{i+1})$ est l'ensemble flou associé à C'_{i+1} ;
- Le coût global du chemin $(C_0) \rightarrow (C_0 C_1) \rightarrow \dots \rightarrow (C_0, \dots, C_{i+1})$ est l'intersection floue des coût de ses arcs élémentaires, qui peut être aussi écrit : $g(C_0, \dots, C_{i+1}) = g(C_0, \dots, C_i) \cap EnsF(C'_{i+1})$
- Les états objectifs sont des états $(C_0 C_1 \dots C_i)$ tels que $C_i = (\perp, \alpha_i/A_i)$ avec $\alpha_i/A_i \neq 0/\emptyset$;
- L'état $(C_0 C_1 \dots C_i)$ est développé en produisant toutes les résolvantes de C_i autorisées par la stratégie linéaire.

La recherche d'une clause vide avec le plus grand ensemble flou d'agents est équivalente à la recherche d'un chemin avec un coût maximal.

Cependant, il existe plusieurs différences :

- Le coût ici est à maximiser et non pas à minimiser. En effet, le but est de trouver le plus grand ensemble flou d'agents qui croient une formule.
- Les coûts ne sont pas additifs, mais ce sont des ensembles flous combinés par l'opérateur d'intersection floue.

- Etant donné que seul un ordre partiel existe entre les sous-ensembles flous, plusieurs états objectifs existent. Ces derniers seront ensuite combinés avec l'opérateur d'union flou.
- Si un ordre d'inclusion existe entre les ensembles flous, le plus grand ensemble flou est pris et l'autre chemin n'est jamais exploré, contrairement à la recherche dans les espaces d'états.

La recherche d'une réfutation avec le plus grand ensemble flou d'agents est guidée par une fonction d'évaluation qui associe à chaque état S une valuation $f(S) \in [0, 1]^{All}$. Comme pour les algorithmes classiques A et A^* , f est une fonction avec deux autres fonctions g et h . g est le coût du chemin de l'état initial S_0 vers l'état courant S , et $h(S)$ une fonction heuristique estimant le coût de S vers un état objectif qui a le plus grand ensemble flou d'agents, et $f(S) = g(S) \cap h(S)$.

Contrairement aux algorithmes A et A^* habituels où $f = g + h$. L'état suivant à explorer sera sélectionné parmi l'ensemble des états maximisant f . Dans ce qui suit, nous donnons d'abord une formulation générale de l'algorithme de recherche, qui est indépendant du choix d'une fonction d'évaluation particulière.

Début

$Open \leftarrow \{S_0\};$

$Closed \leftarrow \{S_0\};$

$bEnsF = 0/\emptyset$

Tant que ($Open \neq \emptyset$) **faire**

Choisir un état S_n dans Open maximisant f

Si (S_n est un état objectif) **Alors**

| $bEnsF = bEnsF \cup S_n$

Sinon

développer le nœud S_n en créant l'ensemble E'_n des états produits.

Si (dans l'ensemble E'_n il existe des sous-ensembles flous incluent dans d'autres) **Alors**

| les supprimer de E'_n

Fin Si

$E_n \leftarrow E'_n \setminus Closed$

$Open \leftarrow (Open - \{S_n\}) \cup E_n$

$Closed \leftarrow Closed \cup \{S_n\}$

calculer f pour chaque nouvel état de Open

Fin Si

Fait

Si ($Open = \emptyset$) **Alors**

| échec

Sinon

| afficher $bEnsF$.

Fin Si

Fin

Algorithme 12: Algorithme de réfutation par résolution possibiliste multi-agents linéaire

La fonction heuristique h est dite admissible si et seulement si

$$\forall S \quad h(S) \supseteq h^*(S)$$

où $h^*(S)$ est le coût réel d'un chemin optimal d'un état S à un état objectif. Une fonction h admissible est ainsi optimiste état donné que les couts ici sont à maximiser, comme que pour les fonctions d'évaluation du A^* traditionnelles. Notons qu'un exemple d'une fonction admissible est donnée par $h(S) = 1/All$ pour tout état S .

7.6.1.3 Une famille de fonctions d'évaluation admissibles

Dans cette section, nous proposons une collection de fonctions heuristiques admissibles.

Soit \mathcal{C} un ensemble de clauses possibilistes multi-agents, et $R(\mathcal{C})$ l'ensemble des clauses possibilistes multi-agents qui sont produites de \mathcal{C} par une application du principe de résolution.

Soient $R^1(\mathcal{C}) = R(\mathcal{C})$ et $\forall i, R^{i+1}(\mathcal{C}) = R(R^i(\mathcal{C}))$; et enfin $R^\infty(\mathcal{C}) = \cup_i R^i(\mathcal{C})$. $R^\infty(\mathcal{C})$ contient toutes les clauses possibilistes-multi-agents productibles obtenues en un nombre fini d'étapes à partir de \mathcal{C} .

Par convention, pour toute clause classique $c = [\neg]L_1 \vee [\neg]L_2 \vee \dots \vee [\neg]L_n$, on écrira $c = l_1 \vee l_2 \vee \dots \vee l_n$, ou $l_i = [\neg]L_i$; on écrira aussi, $\neg l = [L_i \text{ si } l = \neg L_i]$, $\neg l = [\neg L_i \text{ si } l = L_i]$.

Soit H_1 la fonction heuristique admissible suivante :

$$H_1(l) = \bigcup \{ \alpha/A, (c, \alpha/A) \in R^\infty(\mathcal{C}), \neg l \in c \}$$

Nous souhaitons obtenir une réfutation optimale de l'ensemble initial de clauses \mathcal{C} .

Considérons une réfutation de \mathcal{C} et soit $(c, \alpha/A)$ une clause possibiliste multi-agents utilisée dans cette réfutation.

Pour n'importe quel littéral l de c , la réfutation doit utiliser, à un moment ou un autre, une clause possibiliste multi-agents $(c', \beta/B)$ telle que c' contient $\neg l$.

Donc pour n'importe quel littéral l de c , n'importe quelle réfutation de \mathcal{C} utilisant c nécessairement cette dernière a un sous ensemble flou inférieur à : $H_1(l)$.

Cela signifie, en dénotant $h(C)$ pour $h(S)$ avec S est l'état dont la dernière clause est C , que :

$$\forall C = (c, \alpha/A), \forall l' \in c, h^*(C) \subseteq H_1(l')$$

Définie comme telle, la fonction heuristique H_1 ne parait pas directement calculable de \mathcal{C} . Cependant, H_1 est directement calculable à partir de \mathcal{C} vu le résultat suivant :

Proposition 7.2

$$H_1(l) = \bigcup \{ \alpha/A, (c, \alpha/A) \in \mathcal{C}, \neg l \in c \}$$

Preuve 7.12 Elle est identique à la preuve 6.15

De là, H_1 est une fonction statique qui est calculée une fois pour toutes avant avant que l'algorithme ne soit exécuté.

Pour toute clause $C = (c, \alpha/A)$, Définissons :

$$h_1(C) = \begin{cases} 1/All & \text{si } c = \perp \\ \bigcap \{H_1(l), l \in c\} & \text{sinon} \end{cases}$$

Soit $C = (c, \alpha/A)$, de $\forall l \in c, H_1(l) \supseteq h^*(C)$, d'ici $h_1(C) \supseteq h^*(C)$. Ceci implique l'admissibilité de h_1 .

La suite de fonctions d'évaluation $(f_p)_{p \geq 0}$ est définie comme suit :

$$f_p : R^\infty(\mathcal{C}) \rightarrow [0, 1]^{All}$$

$$h_0(C) = 1/All$$

et $\forall p \geq 0$

$$f_p(C) = \cap [EnsF(C), h_p(C)]$$

$$h_{p+1}(C) = \bigcap_{l \in c} \bigcup_{C'=(c', \beta/B) \in \mathcal{C}, \neg l \in cl(c')} f_p(C')$$

Pour $p = 1$, nous retrouvons h_1 .

7.6.1.4 Exemple

Soit la base multi-agents possibiliste Φ constituée des clauses suivantes :

$$C_1 : (\neg p \vee q, 0, 6/A)$$

$$C_2 : (\neg p \vee q, 0, 6/B)$$

$$C_3 : (\neg p \vee r, 0, 9/A)$$

$$C_4 : (\neg q \vee r, 0, 6/All)$$

$$C_5 : (p, 0, 3/All)$$

$$C_6 : (q, 0, 7/A)$$

Nous remarquons que la projection classique Φ^* telle que $\Phi^* = \{a_i | (a_i, A_i/\alpha_i) \in \Phi, i = 1 \dots m\}$ est la même que pour l'exemple du chapitre précédent.

Si nous voulons retrouver le plus grand ensemble d'agents qui croient une formule r et avec le plus grand degré nous rajoutons $\neg r$ à Φ . Nous aurons alors $\Phi' = \Phi \cup \{(\neg r, 1/All)\}$.

Nous posons toujours $C_0 = (\neg r, 1/All)$ du fait que $\Phi - \{C_0\}$ est cohérente.

Etant donné que la projection classique de Φ est la même que pour l'exemple du chapitre refresmultiagents, nous aurons les deux mêmes chemins possibles 7.2 : soit le chemin avec la clause C_3 ou le chemin avec la clause C_4 . Or ici, les deux chemins seront exploités car les deux sous-ensembles flous $0, 9/A$ et $0, 6/All$ ne sont pas inclus l'un dans l'autre.

Soit alors l'état suivant $S_1 = (C_0, C_7)$ ayant le coût : $EnsF(C_3) \cap EnsF(C_0) = EnsF(C_3) = EnsF(C_7)$.

A partir de cet état (voir Figure 7.3), une seule clause contient le littéral p , et la clause vide est obtenue avec un coût : $EnsF(C_5) \cap EnsF(C_7) = EnsF(C_5) = EnsF(C_8) = 0.9/A$.

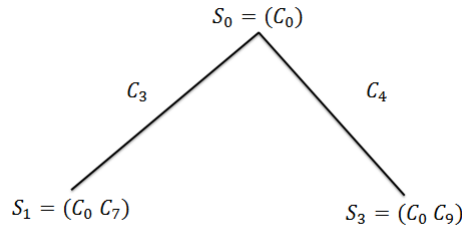


FIGURE 7.2 – Arbre de recherche multi-agents possibiliste -1-

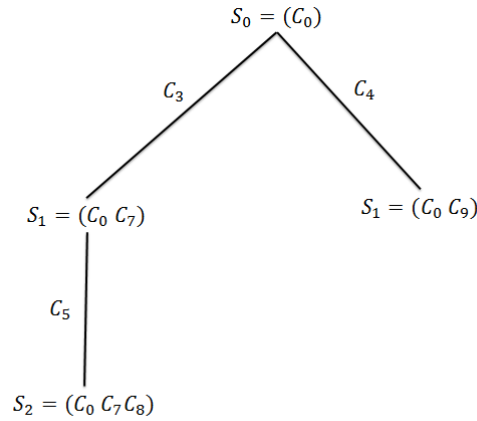


FIGURE 7.3 – Arbre de recherche multi-agents possibiliste -2-

Soit l'état suivant $S_3 = (C_0, C_9)$ ayant le coût : $EnsF(C_4) \cap EnsF(C_0) = EnsF(C_4) = EnsF(C_9) = 0.6/A$.

A partir de cet état (voir Figure 7.4), trois clauses contiennent le littéral q , à savoir : C_1 , C_2 et C_6 . Cependant, nous remarquons que le chemin : $C_0 \rightarrow C_4 \rightarrow C_1$ n'est pas exploré, car l'ensemble flou $0, 6/A$ est inclus dans l'ensemble flou $0, 7/A$ de l'état S_6 .

Soit l'état suivant $S_4 = (C_0, C_9, C_{10})$ ayant le coût : $EnsF(C_2) \cap EnsF(C_9) = EnsF(C_2) = EnsF(C_{10}) = 0.6/B$.

Il existe un seul chemin ensuite (voir Figure 7.5) avec la clause C_5 donc $S_5 = (C_0 C_9 C_{10} C_{11})$ et a comme coût : $Ens(C_{10}) \cap Ens(C_5) = Ens(C_{11}) = 0.3/B$. La clause C_{11} est une contradiction. De cela, le deuxième état objectif est atteint.

L'état suivant (voir Figure 7.6) : $S_6 = (C_0 C_9 C_{12})$, il a comme coût : $Ens(C_9) \cap Ens(C_6) = Ens(C_{12}) = 0.6/A$. Cependant, la clause C_{12} est une contradiction. Le dernier état objectif est atteint.

Nous concluons à la fin que $\Phi \models (r, 0, 6/A \cup 0, 3/B)$.

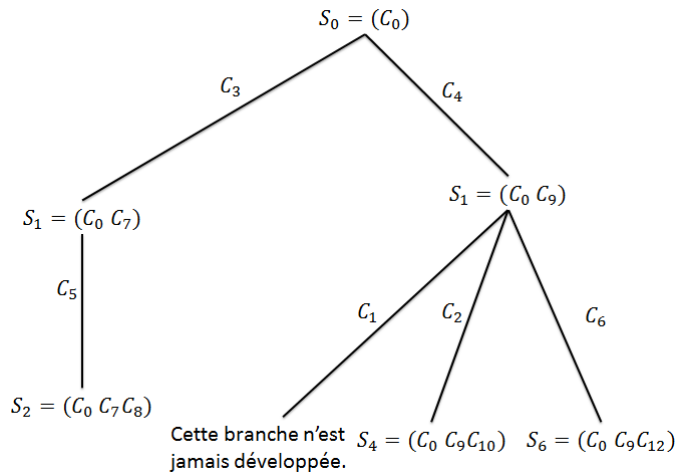


FIGURE 7.4 – Arbre de recherche multi-agents possibiliste -3-

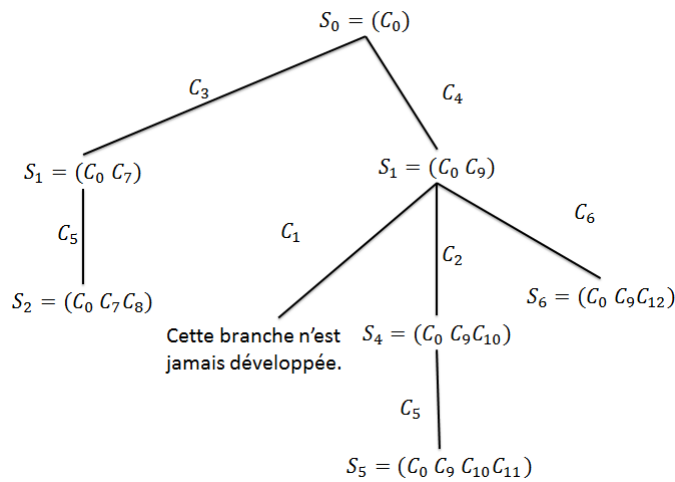


FIGURE 7.5 – Arbre de recherche multi-agents possibiliste -4-

7.7 Conclusion

Dans ce chapitre, nous nous sommes penchés sur le volet du raisonnement. Nous nous sommes intéressés aux mécanismes d'inférence en logique possibiliste multi-agents, une logique analogue à la logique possibiliste où les niveaux de certitude sont remplacés par des sous-ensembles flous d'agents. Nous avons présenté un algorithme de réfutation par résolution pour la logique possibilité multi-agents. Cet algorithme généralise l'inférence de la logique possibiliste mono-agents et utilise une stratégie de recherche basée A^* . Dans ce contexte, des réfutations avec les plus grands ensembles d'agents flous sont trouvées.

Dans le chapitre suivant, nous allons aborder l'étude expérimentale relative aux différents algorithmes développés dans cette thèse.

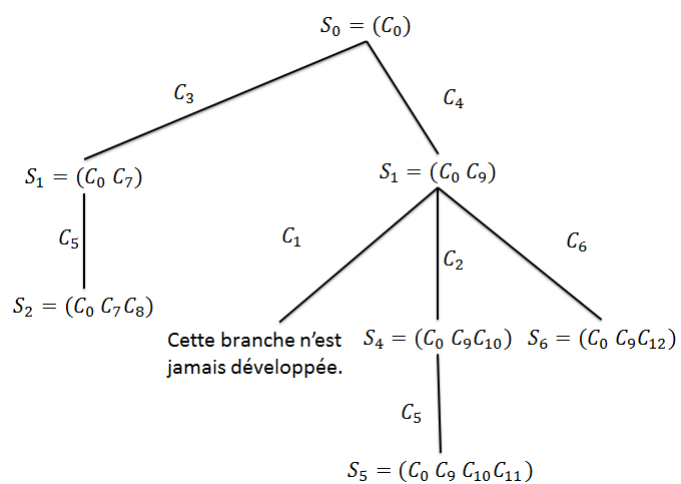


FIGURE 7.6 – Arbre de recherche -5-

Chapitre 8

Implémentation et Expérimentation

8.1 Introduction

Nous avons exposé dans les deux chapitres précédents, une nouvelle approche de réfutation par résolution dans le cadre de la logique multi-agents et la logique possibiliste multi-agents. Afin de valider l'approche proposée, ce chapitre est consacré à implémenter et à tester cette approche sur différentes bases de connaissances.

Pour cela, ce chapitre est organisé comme suit : la section suivante présente les principales classes implémentées en Java de l'approche de réfutation proposée pour la logique multi-agents et pour la logique possibiliste multi-agents. Dans la troisième section nous allons tester l'application en utilisant des exemples afin de vérifier le bon fonctionnement de l'application. La dernière section analyse le comportement de l'application tout en variant les paramètres de la base de connaissances multi-agents (respectivement, de la base de connaissances possibiliste multi-agents) tel que : le nombre de clauses, le nombre de variables, le nombre de groupes d'agents et le nombre d'agents de chaque groupe. Nous concluons ce chapitre par une petite analyse des résultats obtenus.

8.2 Implémentation

Dans cette section, nous abordons l'aspect de l'implémentation de l'algorithme de la réfutation par résolution pour la logique multi-agents ainsi que la logique possibiliste multi-agents avec le langage Java. Ce dernier est un langage orienté objet qui étend la notion de la structure du langage *C* pour introduire le concept de classe. Les principaux avantages de Java sont les suivants :

- Accès libre à tous les package programmées en Java.
- Grand nombre de fonctionnalités.
- Facilité de l'utilisation des langages objets.
- Portabilité des fichiers sources.
- Facilité de conversion des programmes C++ en Java.
- Combinaison de plusieurs paradigmes de programmation tels que le paradigme orienté objet (classes, héritage, encapsulation...), paradigmes fonctionnels (les procédures, les fonctions...), paradigmes de contrôles (les exceptions, contrôle d'erreurs...).

Les différentes classes de notre application sont :

- La classe `GroupeAgent` (voir Figure 8.1) : permet de représenter les agents d'un

```

package Applicaion;

import java.util.List;

public class group_agent {
    List <Integer> []agent_etat=new List[100];
}

```

FIGURE 8.1 – La classe group_agent

```

package Applicaion;

import java.util.List;

public class Clause {

    int []litteral=new int[100];
    List <group_agent>[]agent_clauses=new List[100];
    float degree;
}

```

FIGURE 8.2 – La classe Clause

```

package Applicaion;

public class DPLL {
    void init_assignment()
    int eval(int k, int n, Clause C[])
    boolean dp11(int profondeur, int k, int n, Clause C[])
    boolean main_process(int k, int n, Clause C[])
}

```

FIGURE 8.3 – La classe DPLL

```

package Applicaion;

import java.util.List;

public class refutation {
    List []Open =new List[100];
    List []Closed =new List[100];
    void creer_etat_initial()
    void choisir_clauses_adequate(Clause S)
    void Generer_etats(Clause S)
    String main_process()
}

```

FIGURE 8.4 – La classe refutation

groupe donné. Elle est caractérisée par une liste où son i ème élément représente l'absence (0), ou bien la présence (1) du i^{me} agent dans ce groupe.

- La classe `clause` (Figure 8.2) : Cette classe permet de représenter la structure d'une clause. Elle contient trois paramètres :

1. le premier paramètre est : un vecteur d'entier de littéraux où le i ème élément de ce vecteur peut prendre les valeurs 0, -1 et 1 : 0 si le i^{me} littéral n'appartient pas à la clause. 1 si le i ème littéral appartient à la clause avec la valeur vrai. -1 si le i ème littéral appartient à la clause avec la valeur faux.

2. Le deuxième paramètre est une liste qui représente le groupe d'agents qui croit cette clause. Chaque élément de cette liste est un groupe d'agents de la classe `GroupeAgent` présenté précédemment.

3. Le troisième est un nombre réel choisi entre 0 et 1 qui représente le degré de cette clause.

- La classe `DPLL` (Figure 8.3) : Cette classe permet de vérifier l'inconsistance de la base des clauses. Pour cela, nous considérons la base des clauses comme une instance SAT puis nous appliquons l'algorithme DPLL [Davis, Putnam 60] (l'algorithme le plus utilisé pour vérifier la satisfiabilité d'une instance SAT).

Cette classe est composée d'une procédure et trois fonctions, à savoir :

1. **`init_assignment()`** : cette procédure permet d'initialiser une instantiation des littéraux.

2. **`eval(k, n, C[])`** : cette fonction permet d'évaluer l'instanciation courante de n littéraux de la base de clause C . Elle retourne le nombre de clauses satisfaites parmi les k clauses de C .

3. **`dppl(profondeur,k, n, C[])`** : elle vérifie si l'instanciation avec la profondeur courante satisfait toutes les clauses ou pas. Si ce n'est pas le cas, elle passe par la profondeur suivante en instanciant le prochain littéral.

4. **`main_process(k, n, C[])`** : c'est l'algorithme principal de DPLL. Tout d'abord la procédure `init_assignment()` est appelé pour initialiser une instantiation ensuite un appel récursif à la fonction `dppl (profondeur,k, n, C[])` est effectué, jusqu'à obtenir une instantiation qui satisfait toutes les clauses. Si c'est le cas cette fonction retourne 'vrai' et la base de clauses est consistante. Si tous les littéraux sont instanciés et il existe une clause qui n'est pas satisfaite, la valeur 'faux' est retournée et la base de clauses est dite inconsistante.

- La classe `refutation` (Figure 8.4) : Cette classe permet de réfuter une formule donnée, deux structures `open` et `closed` sont utilisées. Cette classe est composée de trois procédures et une fonction comme suit :

1. **`creer_etat_initial()`** : cette procédure permet de générer l'état initial de l'arborescence.

2. **choisir_clauses_adequate(S)** : cette procédure permet de choisir les clauses adéquates avec l'état courant S.
3. **Generer_etats(S)** : cette procédure permet de générer les états fils de l'état S.
4. **main_process()** : C'est la fonction principale de notre algorithme. Tout d'abord, il y a vérification de la consistance de la base. Si cette dernière n'est pas consistante, alors une chaîne vide est retournée puis arrêt, sinon nous générons l'état initial et d'une manière récursive les clauses adéquates sont extraites, les états fils sont générés et les groupes d'agents d'un état final sont mis dans la chaîne résultat. Ce mécanisme est répété jusqu'à ce qu'aucun état ne soit généré. A la fin, la chaîne résultat est retournée. Ce processus peut être appliqué pour les deux algorithmes de réfutation (multi-agents et possibiliste multi-agents). Le seul changement est au niveau des pondérations.

8.3 Validation de l'application

Afin de pouvoir estimer les performances de l'application, une étude expérimentale a été menée et ce pour la logique multi-agents et la logique possibiliste multi-agents. L'étude expérimentale s'est déroulée en deux phases :

- La première phase consistait à générer des petites bases afin de pouvoir dérouler le test manuellement.
- La seconde phase consistait à générer aléatoirement des bases de connaissances de grandes instances.

8.3.1 Tests sur des bases de connaissances de petites tailles

Une série de tests a été effectuée en considérant un ensemble de base de connaissances multi-agents et possibiliste multi-agents de petites instances comme suit :

1. Tests sur des bases de connaissances de petites tailles pour la logique multi-agents

- Exemple 01 :
Considérons la base des connaissances constituée des clauses suivantes :
 - $C_1 : (\neg p \vee q, A)$;
 - $C_2 : (\neg r, All)$;
 - $C_3 : (p, All)$.

Le plus grand ensemble d'agents qui croient q est le groupe d'agents A . Pour ce faire :

- La clause $(\neg q, All)$ est rajoutée à la base.
- Il faut par la suite chercher l'ensemble de clauses contenant le littéral q .
- La seule clause contenant le littéral q est la clause C_1 . Par résolution, la résolvente $(\neg p, A)$ est ainsi obtenue.

- Le processus est alors réitéré avec la clause C_3 qui permet d'aboutir à la résolvente (\perp, A) .
- Exemple 02 :
Considérons la base des connaissances constituée des clauses suivantes :
 - $C_1 : (\neg p \vee q, B)$;
 - $C_2 : (\neg p \vee q, A)$;
 - $C_3 : (\neg p \vee r, B)$.
 - $C_4 : (p, All)$.

Le plus grand ensemble d'agents qui croient r est le groupe d'agents B . Pour ce faire :

- La clause $(\neg r, All)$ est rajoutée à la base.
- Il faut par la suite chercher l'ensemble de clauses contenant le littéral r .
- La seule clause contenant le littéral r est la clause C_3 . Par résolution, la résolvente $(\neg p, B)$ est ainsi obtenue.
- Le processus est alors réitéré avec la clause C_4 qui permet d'aboutir à la résolvente (\perp, B) .
- Exemple 03 :
Considérons la base des connaissances constituée des clauses suivantes :
 - $C_1 : (\neg p \vee q, A)$;
 - $C_2 : (\neg p \vee q, B)$;
 - $C_3 : (\neg p \vee r, A)$;
 - $C_4 : (\neg q \vee r, All)$;
 - $C_5 : (p, All)$;
 - $C_6 : (q, A)$.

Le plus grand ensemble d'agents qui croient r est le groupe d'agents $A \cup B$. Pour ce faire :

- La clause $(\neg r, All)$ est rajoutée à la base.
- Il faut par la suite chercher l'ensemble de clauses contenant le littéral r .
- Il existe deux clauses contenant le littéral r : la clause C_3 et la clause C_4 :
 - Le chemin de la clause C_3 n'est pas développé car l'ensemble A est inclus dans l'ensemble All .
 - Le chemin de la clause C_4 produit la clause $(\neg q, All)$.
- Il existe trois clauses contenant le littéral q : la clause C_1 , la clause C_2 et C_6 :
 - Le chemin en considérant la clause C_1 produit la clause $(\neg p, A)$. La seule clause qui contient le littéral p est la clause C_5 , ce qui génère une contradiction avec le groupe A .
 - Le chemin en considérant la clause C_2 produit la clause $(\neg p, B)$. La seule clause qui contient le littéral p est la clause C_5 , ce qui génère une contradiction avec le groupe B .
 - Le chemin en considérant la clause C_6 génère une contradiction avec le groupe A .

– Exemple 04 :

Considérons la base des connaissances constituée des clauses suivantes :

- $C_1 : (\neg p \vee q, All)$.
- $C_2 : (p, A)$
- $C_3 : (\neg q, B)$.

Cette base de connaissances est inconsistante :

L'inconsistance de la base a été vérifiée par l'algorithme DPLL. En effet, quatre cas possibles sont considérés :

- (a) Si $(p = vrai \text{ et } q = vrai)$ alors la clause C_3 est fausse.
- (b) Si $(p = vrai \text{ et } q = faux)$ alors la clause C_1 est fausse.
- (c) Si $(p = faux \text{ et } q = vrai)$ alors les clauses C_2 et C_3 sont fausses.
- (d) Si $(p = faux \text{ et } q = faux)$ alors la clause C_2 est fausse.

– Exemple 05 :

Considérons la base des connaissances constituée des clauses suivantes :

- $C_1 : (\neg p \vee q, A)$.
- $C_2 : (p, B)$.
- $C_3 : (q, C)$.

Le plus grand ensemble d'agents qui croient q est le groupe d'agents $(A \cap B) \cup C$.
Pour ce faire :

- La clause $(\neg q, All)$ est rajoutée à la base.
- Il faut par la suite chercher l'ensemble de clauses contenant le littéral q .
- Il existe deux clauses contenant le littéral q : la clause C_1 et la clause C_3 :
 - Le chemin en considérant la clause C_1 produit la clause $(\neg p, A)$. La seule clause qui contient le littéral p est la clause C_2 , ce qui génère une contradiction avec le groupe $A \cap B$.
 - Le chemin en considérant la clause C_3 génère une contradiction avec le groupe C .

2. Tests sur des bases de connaissances de petites tailles pour la logique possibiliste multi-agents

– Exemple 01 :

Considérons la base des connaissances possibiliste multi-agents constituée des clauses suivantes :

- $C_1 : (p \vee \neg r, 0.5/A)$;
- $C_2 : (q, 0.7/B)$;
- $C_3 : (\neg p, 0.7/A)$.

Le plus grand ensemble d'agents qui croient $\neg r$ avec le plus grand degré est le groupe d'agents A avec un degré 0.5. Pour ce faire :

- La clause $(r, 1/All)$ est rajoutée à la base.

- Il faut par la suite chercher l'ensemble de clauses contenant le littéral $\neg r$.
- La seule clause contenant le littéral $\neg r$ est la clause C_1 . Par résolution, la résolvente $(p, 0.5/A)$ est ainsi obtenue.
- Le processus est alors réitéré avec la clause C_3 qui permet d'aboutir à la résolvente $(\perp, 0.5/A)$.

– Exemple 02 :

Considérons la base des connaissances possibiliste multi-agents constituée des clauses suivantes :

- $C_1 : (\neg r, 0.2/A)$.
- $C_2 : (p \vee q \vee r, 0.9/All)$.
- $C_3 : (\neg q, 0.1/B)$.

Le plus grand ensemble d'agents qui croient p avec le plus grand degré est le groupe d'agents $A \cap B$ avec un degré 0.1. Pour ce faire :

- La clause $(\neg p, 1/All)$ est rajoutée à la base.
- Il faut par la suite chercher l'ensemble de clauses contenant le littéral p .
- La seule clause contenant le littéral p est la clause C_2 . Par résolution, la résolvente $(q \vee r, 0.9/All)$ est ainsi obtenue.
- La seule clause contenant le littéral $\neg r$ est la clause C_1 . Par résolution, la résolvente $(q, 0.2/A)$ est ainsi obtenue.
- Le processus est alors réitéré avec la clause C_3 qui permet d'aboutir à la résolvente $(\perp, 0.1/A \cap B)$.

– Exemple 03 :

Considérons la base des connaissances possibiliste multi-agents constituée des clauses suivantes :

- $C_1 : (\neg q, 0.5/A)$;
- $C_2 : (\neg q, 0.3/B)$;
- $C_3 : (\neg p \vee q \vee r, 0.8/All)$;
- $C_4 : (\neg p \vee q, 0.5/All)$;
- $C_5 : (\neg r, 0.4/All)$.

Le plus grand ensemble d'agents qui croient $\neg p$ avec le plus grand degré est le groupe d'agents A avec un degré 0.4 et le groupe d'agents B avec un degré 0.3. Pour ce faire :

- La clause $(p, 1/All)$ est rajoutée à la base.
- Il faut par la suite chercher l'ensemble de clauses contenant le littéral $\neg p$.
- Il existe deux clauses contenant le littéral $\neg p$: la clause C_3 et la clause C_4 :
 - Le chemin de la clause C_4 n'est pas développé car l'ensemble $0.5/All$ est inclus dans l'ensemble $0.8/All$.
 - Le chemin de la clause C_3 produit la clause $(q \vee r, 0.8/All)$.
- Il existe deux clauses contenant le littéral $\neg q$: la clause C_1 et la clause C_2 :
 - Le chemin en considérant la clause C_1 produit la clause $(r, 0.5/A)$. La seule clause qui contient le littéral $\neg r$ est la clause C_5 , ce qui génère une contradiction avec le groupe $0.4/A$.
 - Le chemin en considérant la clause C_2 produit la clause $(r, 0.3/B)$. La seule clause qui contient le littéral $\neg r$ est la clause C_5 , ce qui génère une

contradiction avec le groupe $0.3/B$.

- Exemple 04 :
Considérons la base des connaissances possibiliste multi-agents constituée des clauses suivantes :
- $C_1 : (p, 0.7/All)$;
- $C_2 : (q, 0.2/All)$;
- $C_3 : (\neg q \vee \neg q \vee r, 0.7/A)$;
- $C_4 : (r, 0.3/B)$;
- $C_5 : (\neg q \vee r, 0.9/C)$.

Le plus grand ensemble d'agents qui croient r avec le plus grand degré est le groupe d'agents A avec un degré 0.2 et le groupe d'agents B avec un degré 0.3 et le groupe d'agents C avec un degré 0.2. Pour ce faire :

- La clause $(\neg r, 1/All)$ est rajoutée à la base.
- Il faut par la suite chercher l'ensemble de clauses contenant le littérel r .
- Il existe trois clauses contenant le littéral r : la clause C_3 , la clause C_4 et la clause C_5 :
 - Le chemin de la clause C_3 produit la clause $(\neg p \vee \neg q, 0.7/A)$. La seule clause qui contient le littéral p est la clause C_1 , ce qui produit la clause $(\neg q, 0.7/A)$. La seule clause qui contient le littéral q est la clause C_2 , ce qui génère une contradiction avec le groupe $0.2/A$.
 - Le chemin en considérant la clause C_4 génère une contradiction avec le groupe $0.3/B$.
 - Le chemin en considérant la clause C_5 produit la clause $(\neg q, 0.9/C)$. La seule clause qui contient le littéral q est la clause C_2 , ce qui génère une contradiction avec le groupe $0.2/C$.

8.3.2 Tests sur des bases de connaissances générées aléatoirement

Cette phase de validation consiste à générer aléatoirement des bases multi-agents et des bases possibiliste multi-agents en faisant varier les paramètres suivants :

- nombre de clauses
- nombre de variables
- nombre d'agents
- nombre de groupe d'agents

1. Scénarios avec des bases de connaissances multi-agents

- Scénario 1 : Bases de connaissances multi-agents de petites tailles
La Figure 8.5 illustre le comportement de l'algorithme de réfutation en variant le nombre de clauses de 10 à 200. D'après cette figure, le temps d'exécution augmente proportionnellement avec le nombre de clauses. A titre d'exemple, le temps d'exécution atteint 0,12 sec lorsque le nombre de clauses est égale à 200 avec 15 groupe d'agents.
- Scénario 2 : Bases de connaissances multi-agents de tailles moyennes
La Figure 8.6 illustre le comportement de l'algorithme de réfutation en variant le nombre de clauses de 500 à 2500. D'après cette figure, le temps d'exécution

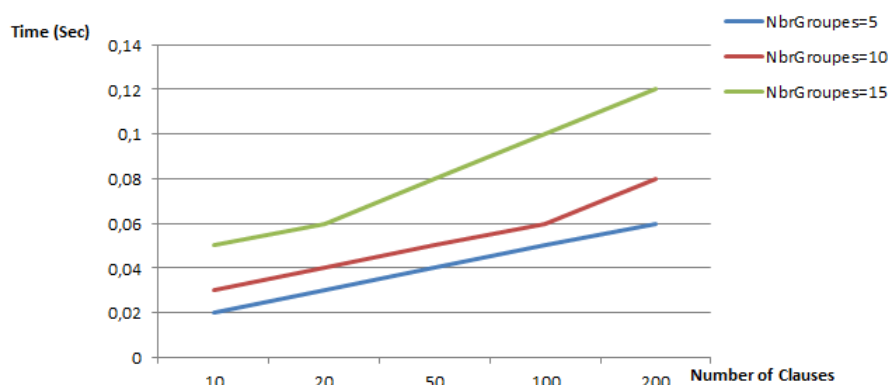


FIGURE 8.5 – Temps CPU (Sec) du processus de réfutation en utilisant des bases de connaissances multi-agents de petites tailles.

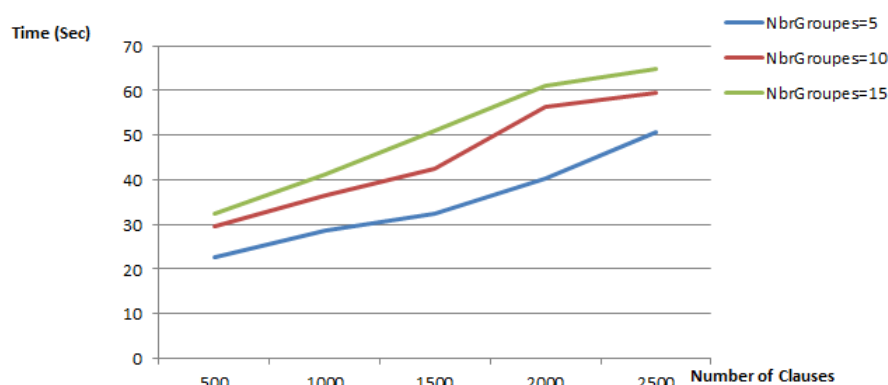


FIGURE 8.6 – Temps CPU (Sec) du processus de réfutation en utilisant des bases de connaissances multi-agents de tailles moyennes.

augmente proportionnellement avec le nombre de clauses. A titre d'exemple, le temps d'exécution atteint 65 *sec* lorsque le nombre de clauses est égale à 2500 avec 15 groupe d'agents.

- Scénario 3 : Bases de connaissances multi-agents de grandes tailles

La Figure 8.7 illustre le comportement de l'algorithme de réfutation en variant le nombre de clauses de 500 à 5000. D'après cette figure, le temps d'exécution augmente proportionnellement avec le nombre de clauses. A titre d'exemple, le temps d'exécution atteint 345 *sec* lorsque le nombre de clauses est égale à 50000 avec 15 groupe d'agents.

2. Scénarios avec des bases de connaissances possibilistes multi-agents

- Scénario 1 : Bases de connaissances possibilistes multi-agents de petites tailles

La Figure 8.8 illustre le comportement de l'algorithme de réfutation en variant le nombre de clauses de 10 à 200. D'après cette figure, le temps d'exécution augmente proportionnellement avec le nombre de clauses. A titre d'exemple, le temps d'exécution atteint 0,24 *sec* lorsque le nombre de clauses est égale à 200 avec 15 groupe d'agents.

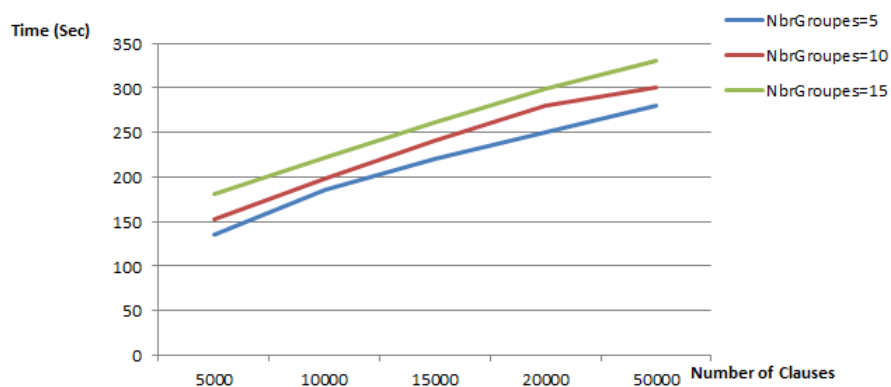


FIGURE 8.7 – Temps CPU (Sec) du processus de réfutation en utilisant des bases de connaissances multi-agents de grandes tailles.

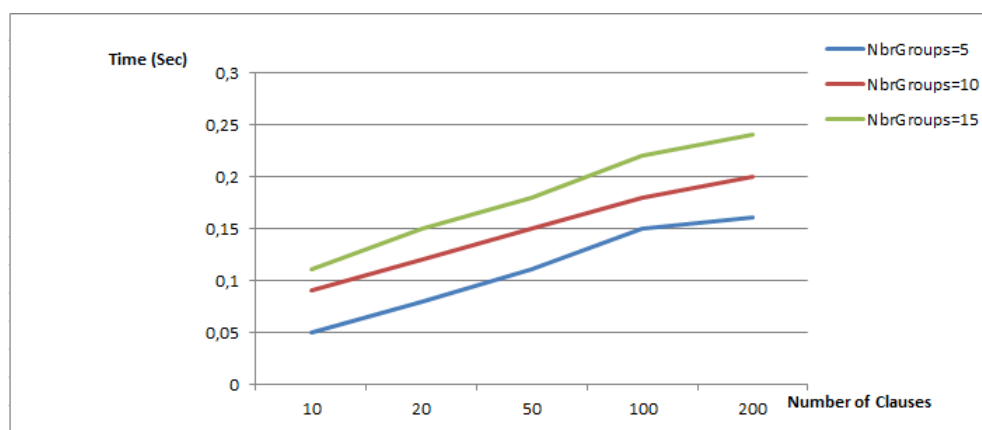


FIGURE 8.8 – Temps CPU (Sec) du processus de réfutation en utilisant des bases de connaissances possibilistes multi-agents de petites tailles.

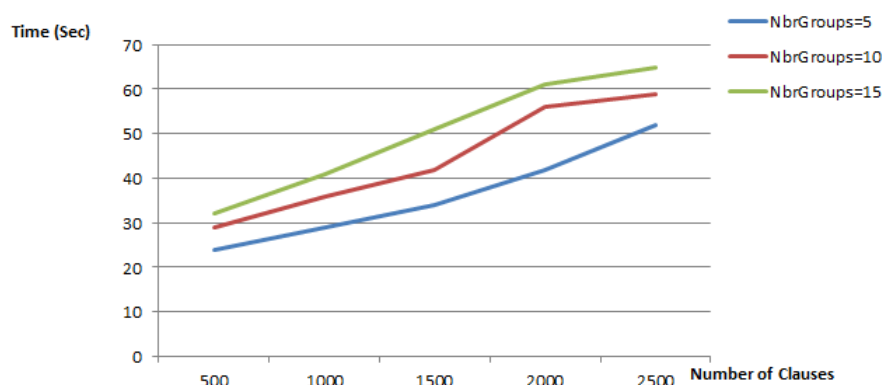


FIGURE 8.9 – Temps CPU (Sec) du processus de réfutation en utilisant des bases de connaissances possibilistes multi-agents de tailles moyennes.

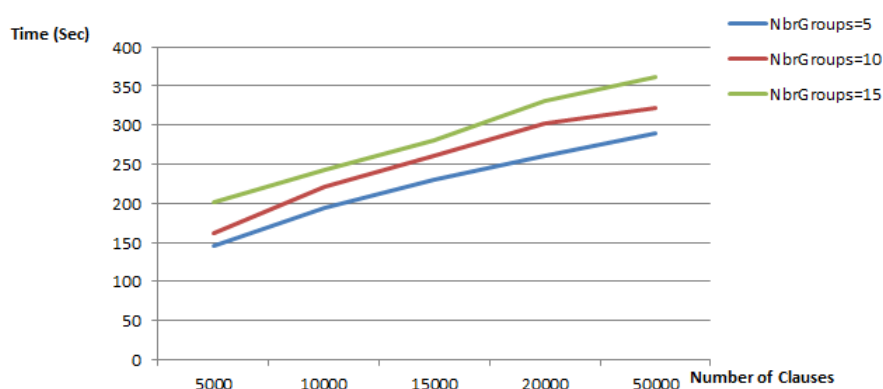


FIGURE 8.10 – Temps CPU (Sec) du processus de réfutation en utilisant des bases de connaissances possibilistes multi-agents de grandes tailles.

- Scénario 2 : Bases de connaissances possibilistes multi-agents de tailles moyennes
La Figure 8.9 illustre le comportement de l'algorithme de réfutation en variant le nombre de clauses de 500 à 2500. D'après cette figure, le temps d'exécution augmente proportionnellement avec le nombre de clauses. A titre d'exemple, le temps d'exécution atteint 68 sec lorsque le nombre de clauses est égale à 2500 avec 15 groupe d'agents.
- Scénario 3 : Bases de connaissances possibilistes multi-agents de grandes tailles
La Figure 8.10 illustre le comportement de l'algorithme de réfutation en variant le nombre de clauses de 5000 à 50000. D'après cette figure, le temps d'exécution augmente proportionnellement avec le nombre de clauses. A titre d'exemple, le temps d'exécution atteint 360 sec lorsque le nombre de clauses est égale à 50000 avec 15 groupe d'agents.

3. Comparaison entre la réfutations par résolution multi-agents linéaires et sa contrepartie possibiliste multi-agents

Dans le but de comparer les deux approches, d'autres expérimentations ont eu lieu en utilisant des bases de grandes tailles qui contiennent 50000 clauses, 30 variables et 15 groupes d'agents. En variant le nombre d'agents de 25 à 200, la figure 8.11

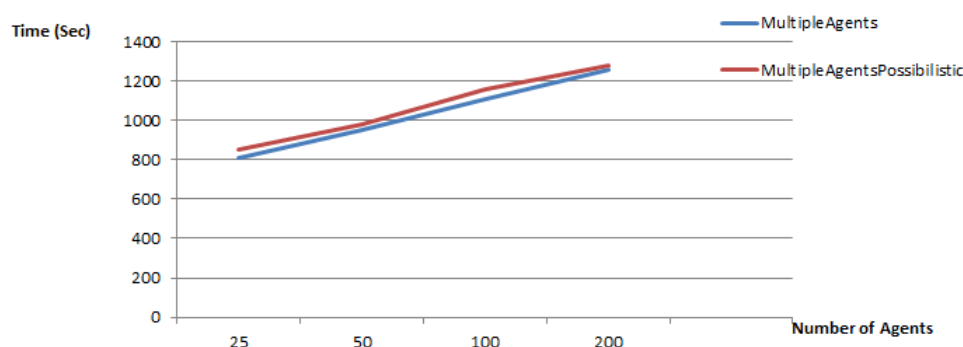


FIGURE 8.11 – Comparaison entre la réfutations par résolution multi-agents linéaires et sa contrepartie possibiliste multi-agents en termes de temps d'exécution (Sec)

montre que le temps d'exécution des bases de connaissances possibilistes multi-agents est légèrement grand comparé à des bases de connaissances multi-agents quel que soit le nombre d'agents.

Après ces expérimentations, nous pouvons conclure que la performance de notre approche de réfutation est fortement liée aux nombre de groupes d'agents. En effet, si nous augmentons le nombre de clauses le temps d'exécution augmente linéairement, tandis que lorsque nous augmentons le nombre de groupes d'agents, le temps d'exécution augmente exponentiellement. Cela peut être expliqué par le fait que la construction de l'arbre de réfutation est basée essentiellement sur les clauses adéquates de littéral réfuté et chaque branche de l'arbre représente une clause adéquate.

Nous pouvons conclure aussi que le temps d'exécution des bases de connaissances possibilistes multi-agents est légèrement grand comparé à des bases de connaissances multi-agents. Cela peut être expliqué par le fait que la construction de l'arbre de réfutation augmentée avec des degrés et des groupes d'agents consomme plus de temps par rapport à un arbre de réfutation augmenté seulement avec des groupes d'agents.

8.4 Conclusion

Dans ce chapitre, nous avons abordé l'implémentation en Java de notre approche de réfutation déjà proposée dans les deux chapitres précédents.

Afin d'analyser le comportement de notre approche, plusieurs tests ont été réalisés en concinsidérant différentes bases de connaissances multi-agents et possibiliste multi-agents. Tout en faisant varier le nombre de clauses, le nombre de variables, le nombre de groupes d'agents et le nombre d'agents de chaque de chaque groupe, le temps d'exécution augmente avec le nombre de clauses pour la logique multi-agents et pour la logique possibiliste multi-agents. Ce résultat est tout à fait justifié étant donné le principe de réfutation qui est basé sur les clauses adéquates.

Chapitre 9

Vers des applications multi-agents/ possibilistes multi-agents

9.1 Introduction

Après avoir présenté les différentes formulations relatives à la logique multi-agents et à la logique possibiliste multi-agents, il serait judicieux de faire une projection sur l'exploitation de ces modèles. Dans cette optique, nous présentons dans ce chapitre deux esquisses d'utilisation des logiques proposées à des domaines intéressants à savoir la révision de croyances et la prise en compte de la notion de confiance. Ainsi, la section suivante présente une esquisse d'une révision de croyances multi-agents. Et la section 2 étudie la prise en compte de la notion de confiance.

9.2 Révision de croyances en logique propositionnelle

L'un des thèmes de recherche importants en intelligence artificielle est la dynamique (le changement) des croyances. Dans de nombreuses applications, un agent reçoit des informations imparfaites, incomplètes, imprécises ou incertaines sur le monde auquel il est confronté, et il émet des hypothèses qui peuvent être contredites lors de l'arrivée d'une nouvelle information considérée comme plus fiable.

Ces hypothèses, qui reflètent la vision qu'un agent intelligent a du monde réel où il vit, et qu'il construit à partir d'informations qu'il reçoit (observations, témoignage) et aussi de son expérience, sont appelées croyances.

Quand une nouvelle information fiable vient contredire les croyances courantes de l'agent, la révision consiste alors à restaurer la cohérence de manière à intégrer la nouvelle information tout en modifiant le moins possible les croyances initiales.

Dans cette section, les croyances d'un agent sont représentées par des formules logiques. L'ensemble fini de telles formules est appelé base de croyances. La révision de croyances consiste à faire évoluer les croyances d'un agent qui reçoit une nouvelle information. Les propriétés attendues des opérateurs de révision de croyances se résument intuitivement par trois principes :

- Principe de succès : le changement doit réussir. En d'autres termes, il faut qu'après l'opération, la nouvelle information doit avoir le statut souhaité.

- Principe de cohérence : l'ensemble des croyances après la révision doit être cohérent.
- Principe de changement minimal : les croyances de l'agent doivent être modifiées le moins possible. En effet, il faut s'assurer qu'aucune information n'est éliminée de manière inconsidérée et qu'aucune information non souhaitée n'est ajoutée.

Dans la littérature, il existe plusieurs approches logiques pour la révision de croyances. Parmi ces approches, ils existent des approches dites syntaxiques où la plus grande importance est accordée à la façon de coder les croyances. La révision porte sur des bases de croyances finies, c'est-à-dire des ensembles finis de formules propositionnelles.

La plupart des approches syntaxiques reposent sur la construction de sous-bases cohérentes maximales selon différents critères [Benferhat et al. 93] [de Kleer 90] [Lehmann 95]. D'un point de vue dual, d'autres approches reposent sur le retrait minimal de formules, pour restaurer la cohérence avec l'information nouvelle à ajouter, comme l'approche basée sur les r -ensembles, c'est-à-dire des sous-ensembles minimaux de formules à retirer selon leurs cardinalités [Papini 92] [Wurbel et al. 00] [Benferhat et al. 10 a]. Dans ces approches, deux bases logiquement équivalentes peuvent être révisées différemment, ce qui veut dire que les connecteurs logiques sont interprétés de façons non classiques.

Par ailleurs, des approches dites cohérentistes sont distinguées. Dans ces approches, les croyances de l'agent sont supposées former une théorie logique, souvent représentée par une formule propositionnelle. La révision d'un ensemble de croyances revient à rechercher les modèles de la nouvelle information les plus proches des modèles de la formule représentant l'ensemble de croyances. Le principe du changement minimal est défini en termes de pré-ordre entre les formules du langage, comme dans l'approche AGM ou en terme de modèles comme dans l'approche de Katsuno et Mendelzon [Katsuno, Mendelzon 91] ou de Grove [Grove 88].

9.2.1 L'approche AGM

Alchourrón, Gärdenfors et Makinson ont proposé une formalisation des principes de révision en termes de postulats pour la révision de théories logiques [Alchourrón et al. 85] [Gärdenfors 88]. Pour un agent qui possède les croyances K , une formule ϕ ne peut avoir que trois statuts épistémiques différents, à savoirs :

- Soit $\phi \in K$: l'agent croit que ϕ est vrai. En effet, ϕ est acceptée par l'agent.
- Soit $\neg\phi \in K$: l'agent croit que ϕ est fausse. En effet, ϕ est rejetée par l'agent.
- Soit $\phi \notin K$ et $\neg\phi \notin K$: la vérité de ϕ est inconnue pour l'agent.

Les opérateurs de changement de croyances peuvent alors être définis comme des transitions entre ces différents statuts, comme illustrée par la figure 9.1.

Lorsque une formule passe du statut indéterminé à accepté (ou symétriquement refusé), cette transition est nommée expansion, notée $+$ car l'information est ajoutée. La transition inverse (de accepté/ refusé à indéterminé) est appelée contraction, notée $-$ car le but est d'enlever une information parmi les croyances de l'agent. Et lorsqu'elle passe directement du statut accepté à refusé (ou symétriquement de refusé à accepté), la transition est nommée révision, notée $*$. En effet, l'avis sur la véracité d'une information

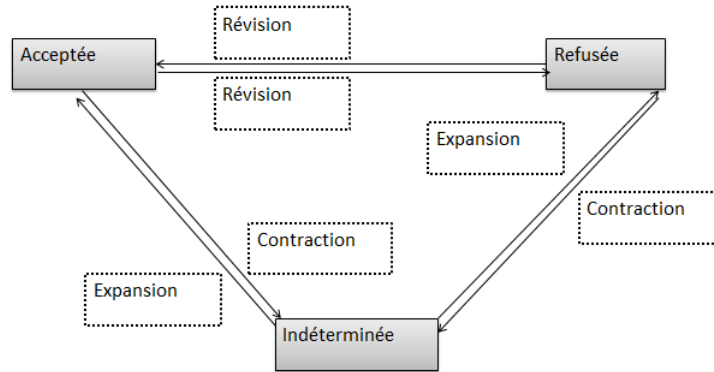


FIGURE 9.1 – Transitions entre statuts épistémiques.

est changée. Dans ce qui suit, nous nous focaliserons sur la révision et nous nous n'aborderons pas les opérations d'expansion ni de contraction. Un ensemble de propriétés que tout opérateur de changement raisonnable devrait satisfaire est proposé. Ces postulats reposent sur la nécessité de maintenir la cohérence des ensembles de croyances à travers des opérations de révision.

9.2.2 Postulats AGM

Un opérateur de révision $*$ est une fonction qui à partir d'une théorie K et d'une formule ϕ renvoie une nouvelle théorie $K * \phi$ qui vérifie les propriétés suivantes :

1. (K^*1) $K * \phi$ est une théorie (clôture) : le résultat de la révision est bien une théorie ;
2. (K^*2) $\phi \in K * \phi$ (succès) : la nouvelle information est vraie dans la nouvelle théorie ;
3. (K^*3) $K * \phi \subseteq K + \phi$ (inclusion) : implique que la révision par la nouvelle information ne peut pas ajouter de croyance qui ne soit une conséquence de la nouvelle information et de la théorie ;
4. (K^*4) si $\neg\phi \notin K$ alors $K + \phi \subseteq K * \phi$ (vacuité) : avec (K^*3) signifient que lorsque la nouvelle information n'est pas contradictoire avec la théorie initiale, alors la révision de la théorie se résume à l'expansion de cette théorie ;
5. (K^*5) $K * \phi = K_{\perp}$ ssi $\models \neg\phi$ (cohérence) : exprime le fait que la seule façon d'arriver à une théorie incohérente par une révision est de réviser par une information contradictoire ;
6. (K^*6) si $\models \phi \leftrightarrow \varphi$ alors $K * \phi = K * \varphi$ (extensionnalité) : le résultat de la révision ne dépend pas de la syntaxe de la nouvelle information ;
7. (K^*7) $K * (\phi \wedge \varphi) \subseteq (K * \phi) + \varphi$ (inclusion conjonctive) ;
8. (K^*8) si $\neg\varphi \notin K * \phi$ alors $(K * \phi) + \varphi \subseteq K * (\phi \wedge \varphi)$ (vacuité conjonctive) : avec le postulat (K^*7) ils sont appelés postulats supplémentaires. Ils assurent que la révision par une conjonction de deux informations revient à une révi possible.

9.2.3 Postulats KM

Afin de caractériser dans un même cadre différentes approches sémantiques, Katsuno et Mendelzon [Katsuno, Mendelzon 91] ont restreint le cadre AGM à la logique propositionnelle. Ils ont reformulé les postulats AGM pour la révision d'une formule propositionnelle, représentant un ensemble de croyances ou une théorie.

Soient γ , μ et δ des formules propositionnelles. L'opérateur \circ est un opérateur de révision s'il vérifie les postulats suivant :

1. (R1) $\gamma \circ \mu \models \mu$;
2. (R2) Si $\gamma \wedge \mu$ est cohérent alors $\gamma \circ \mu \equiv \gamma \mu$;
3. (R3) Si μ est cohérent alors $\gamma \circ \mu$ est cohérent ;
4. (R4) Si $\gamma_1 \equiv \gamma_2$ et $\mu_1 \equiv \mu_2$ alors $\gamma_1 \circ \mu_1 \equiv \gamma_2 \circ \mu_2$;
5. (R5) $(\gamma \circ \mu) \wedge \delta \models \gamma \circ (\mu \wedge \delta)$;
6. (R6) Si $(\gamma \circ \mu) \wedge \delta$ est cohérent alors $\gamma \circ (\mu \wedge \delta) \models (\gamma \circ \mu) \wedge \delta$.

Soit un opérateur de révision $*$ sur des théories et \circ un opérateur de révision sur des formules propositionnelles. L'opérateur $*$ correspond à l'opérateur \circ si quand $K = Cn(\gamma)$ (où $Cn(\gamma)$ est l'ensemble des conséquences logiques de γ), alors $K * \phi = Cn(\gamma \circ \phi)$. L'interprétation de ces postulats est alors comme suit : (R1) est équivalent à (K*2), (R2) est l'équivalent de (K*3) et (K*4) ; (R3) est le postulat de cohérence (K*5) et (R4) l'indépendance de la syntaxe (K*6). Enfin, (R5) et (R6) sont les équivalents de (K*7) et K*8 respectivement.

9.3 Approches valuées de la révision

Il existe également des approches valuées de la révision de croyances développée dans le cadre des théories de l'incertain, comme celle des possibilités [Dubois et al. 98]. Dans ces formalismes à base d'échelles de plausibilités, le changement de croyances par l'introduction d'une nouvelle information E_I , établissant que le monde réel est dans E_I , correspond à une modification de la distribution pl (π pour la théorie des possibilités) en pl' . En général, la distribution pl' résulte d'une opération de conditionnement $pl'(\omega) = pl(\omega|E_I)$. Dans la suite, l'indice I indique une nouvelle information.

L'opération de changement doit respecter les trois principes sous-tendant la révision, déjà évoqués en section 9.2 :

- Cohérence : pl' est de même nature que pl (préservation de la représentation) ;
- Succès : ce qui est observé est tenu pour certain après révision ;
- Principe de changement minimal : la distribution résultante pl' doit garder un maximum de traits communs avec pl , par exemple une distance entre elles doit être minimisée.

Ces approches permettent de définir la révision par une information incertaine. Le degré de certitude avec laquelle la nouvelle information doit être intégrée peut alors être exprimé. En d'autres termes, elles permettent d'exprimer à quel point les modèles de la nouvelle information doivent être préférés à ces contre-modèles.

Révision possibiliste : la révision dans le cadre possibiliste proposée par [Dubois, Prade 92] considère une distribution de possibilité π à valeurs sur une échelle ordinale et une nouvelle information μ totalement certaine ($N_I(\mu) = 1$). Les croyances associées à la distribution de possibilités π forment l'ensemble clos des formules $Bel(\pi) = \{\phi : \Pi(\phi) > \Pi(\neg\phi)\}$, dont les modèles sont $Mod(Bel(\pi)) = \{\omega | \pi(\omega) = 1\}$. Dans ce cadre, la révision possibiliste repose sur un conditionnement possibiliste ordinal :

$$\pi(\omega|\mu) = \begin{cases} 1 & \text{si } \pi(\omega) = \Pi(\mu) \text{ et } \omega \in Mod(\mu); \\ \pi(\omega) & \text{si } \pi(\omega) < \Pi(\mu) \text{ et } \omega \in Mod(\mu); \\ 0 & \text{si } \omega \notin Mod(\mu). \end{cases}$$

Cette révision possibiliste est conforme aux axiomes de base AGM, mais puisqu'elle considère la nouvelle information μ comme totalement certaine, les contres modèles de μ sont considérés comme totalement impossible, tandis que l'ordre relatif entre les modèles de μ est préservé. Cet opérateur vérifie les propriétés (R1) – (R6).

Des extension de cette approche au cas où l'information nouvelle est incertaine sont proposés dans [Dubois, Prade 97] [Benferhat et al. 10 b].

9.4 Esquisse d'une révision des croyances possibiliste multi-agents

9.4.1 Motivation

Pour illustrer le problème de la révision possibiliste multi-agents, nous prenons l'exemple suivant :

Supposons que des experts médicaux s'organisent en deux groupes :

- Groupe *A* des jeunes médecins généralistes ;
- Groupe *B* des jeunes médecins ORL.

Les connaissances des jeunes médecins généralistes sur la grippe sont les suivantes :

- Ils sont tout à fait certains que la grippe est contagieuse ;
- Il est relativement certains que la grippe est causée par des virus ;
- Ils sont tout à fait certains que la grippe affecte les être humain ;
- Ils pensent avec un très fort degré que la grippe n'affecte pas les oiseaux.

Les connaissances des jeunes médecins ORL sur la grippe sont les suivantes :

- Ils sont tout à fait certains que la grippe est contagieuse ;
- Il est relativement certains qu'une grippe fréquente peut être due une allergie.

Soient p, q, r, s, t des variables binaires représentant :

- p : la grippe est contagieuse ;
- q : la grippe est causée par des verus ;
- r : la grippe touche les être humain ;
- s : la grippe touche les oiseaux ;

- t : une grippe fréquente peut être une allergie.

La base de connaissances possibiliste multi-agents associée à ce problème est définie par :

$$\Phi = \{(p, 1/All), (q, 0.5/A), (r, 1/A), (\neg s, 0.7/A), (t, 0.2/B)\}$$

Supposons maintenant que les jeunes médecins généralistes apprennent en lisant les journaux médicaux que c'est certain à un très fort degré que la grippe affecte les oiseaux.

Dans ce cas-là, les jeunes médecins généralistes sont amenés à modifier leurs croyances. De ce fait, après l'ajout de l'information $(s, 0.9/A)$, la base possibiliste multi-agents devient individuellement inconsistante. L'objectif est donc de rétablir la consistance de la base tout en gardant la nouvelle information.

9.4.2 Le problème de la révision possibiliste multi-agents

Soit $\Phi = \{(a_1, \alpha_1/A_1), \dots, (a_n, \alpha_n/A_n)\}$ une base de connaissance possibiliste multi-agents. Soit $(b, \beta/B)$ la nouvelle information qui arrive.

Le problème de la révision possibiliste multi-agent consiste à calculer $i-inc(\Phi)$ afin de rétablir la consistance de la base Φ après l'intégration de la nouvelle information $(b, \beta/B)$. Il s'agit donc, de projeter l'information $(b, \beta/B)$ sur chaque groupe d'agents concerné dans la base.

9.4.2.1 Contrepartie sémantique de la révision possibiliste multi-agents

Sous un angle sémantique, la révision des croyances possibilistes multi-agents peut être exprimée par le moyen du conditionnement où la distribution de possibilité multi-agents graduée π et la nouvelle information $(p, \alpha/A)$ sont transformées en une nouvelle distribution de possibilité multi-agents graduée π' tel que :

$$\pi' = \pi(\omega|(p, \alpha/A)).$$

9.4.2.2 Contrepartie syntaxique de la révision possibiliste multi-agents

La contrepartie syntaxique de la révision des croyances possibilistes multi-agents consiste à effectuer directement la révision sur la base, sans passer par les distributions de possibilités multi-agents graduées.

En effet, il s'agit de rajouter une formule possibiliste multi-agents à la base avec un certain ensemble d'agents avec un certain niveau de certitude. Le problème est difficile car la base doit être modifiée de telle sorte que la nouvelle information soit maintenue au niveau prescrit sans qu'elle soit implicitement inférée par les formules possibilistes multi-agents de niveau de certitude plus important.

Cela revient à construire à partir de la base de connaissances possibiliste multi-agents Φ et de la nouvelle information $(p, \alpha/A)$, une nouvelle base de connaissance possibiliste multi-agents $\Phi' = \Phi \cup \{(p, \alpha/A)\}$ telle que :

$$\forall \omega, \pi_{\Phi'}(\omega) = \pi_{\Phi}(\omega|(p, \alpha/A)).$$

Avec $\pi_{\Phi'}$ (respectivement π_{Φ}) la distribution de possibilités multi-agents graduée associée à la base de connaissances Φ' (respectivement Φ).

9.5 Esquisse de la notion de confiance

Une autre application naturelle de la logique multi-agents concerne la modélisation de la notion de la confiance [Artz, Gil 07] [Pusey et All. 07] [Pichon et All. 12]. C'est ce que nous illustrons dans cette section.

Les systèmes de confiance sont des outils utilisés dans certains systèmes multi-agents du domaine de l'intelligence artificielle pour aider les utilisateurs à prendre des décisions alors même que les agents constituant ces systèmes peuvent être incompetents ou mal intentionnés. Par conséquent, un utilisateur a besoin d'une évaluation de ces agents par rapport à certains aspects avant d'interagir avec eux.

Un système de confiance est un système qui construit, pour un utilisateur donné, une telle évaluation. Typiquement, cette évaluation est obtenue sur la base de deux types d'information :

- Les résultats des interactions ultérieures entre l'utilisateur et les agents ;
- Les recommandations que certains agents fournissent à propos d'autres agents.

La position d'un agent a dans une telle évaluation peut être naturellement vue comme la confiance que l'utilisateur peut rationnellement accorder à l'agent a par rapport à certains aspects.

L'introduction de la confiance dans des systèmes d'agents a permis le développement de nouveaux modèles. Principalement, Il existe deux types de modèles de systèmes de confiance : Les modèles logiques et les modèles numériques.

Dans l'approche logique, le but n'est pas d'obtenir un système d'agents basé sur la confiance mais de tenter de caractériser ce qu'est la confiance, ce que cela signifie de faire confiance à quelqu'un, et quel est l'état mental d'un agent lorsqu'il a confiance en quelqu'un.

L'un des modèles théorique les plus importants de la confiance est le modèle cognitif de confiance de Castelfranchi et Falcone [Castelfranchi, Tan 01], désignés par $C\&F$ par la suite. De manière informelle, le modèle $C\&F$ définit la confiance comme une croyance individuelle du truster à propos de propriétés (telles que la capacité, l'intention, la disposition, etc.) que le premier attribue au second.

En accord avec la théorie de $C\&F$, et selon l'analyse menée dans [Herzig et All 10], la confiance est construite autour de quatre composants : un truster i , un trustee j , une action α de j , et un but φ de i .

Selon leurs définitions : i fait confiance à j pour exécuter α afin de réaliser le but φ si et seulement si : i a le but φ ; i croit que j est capable de faire α ; i croit que j , en

faisant α , va rendre φ vrai ; i croit que j a l'intention de faire α .

Par exemple, quand i fait confiance à j pour lui envoyer un certain produit p dans le but de posséder p , alors : i a le but de posséder p , et il croit que j est capable de lui envoyer p , que le fait que j lui envoie p réalisera son but de posséder p , et que j a bel et bien l'intention de le lui envoyer.

Un autre modèle théorique est le modèle proposé par Demolombe [Demolombe 04]. La confiance dans ce modèle est d'abord vue comme une attitude mentale d'un agent à l'égard d'un autre agent. Cette attitude est une sorte de croyance sur certaines propriétés d'un agent. L'auteur fait la différence entre : croyances, fortes croyances et connaissances. Pour la définition de la confiance, il se base sur le concept de fortes croyances. En effet, si un agent a ne croit pas que ses justifications sont vrai, alors cela veut dire que l'agent a a peut-être quelques doutes concernant la sincérité de l'agent b .

Ainsi, un agent a fait confiance à un agent b pour certaines propriétés, veut dire que l'agent a croit fortement que l'agent b satisfait cette propriété.

Deux sortes de propriétés sont définies :

- Les propriétés épistémiques telles que la sincérité, la coopérativité, la crédibilité, etc.
- Les propriétés déontiques et dynamiques telles que l'obéissance, l'activité, l'honnêteté, etc.

Par exemple, l'agent a fait confiance à b pour sa sincérité concernant p si et seulement si l'agent a croit fortement que l'agent b est sincère concernant p . Un agent b est sincère à l'égard de l'agent a pour p si et seulement si l'agent b informe l'agent a concernant p alors l'agent b croit p .

De plus, dans ce modèle l'auteur définit d'autres types de confiances, à savoir :

- La confiance graduée où l'agent a fait confiance à l'agent b à un certain degré. Par exemple, l'agent a peut être plus au moins certains en ce qui concerne le fait que l'agent b est sincère pour p .
- La confiance par rapport à un sujet donnée : en général la confiance n'est pas spécifique à une proposition p mais plutôt à toutes les propositions d'un sujet donné. Par exemple, l'agent a peut faire confiance à l'agent b à l'égard de sa sincérité pour toute les propositions qui concernent le sujet de l'aéronautique.
- La confiance conditionnelle ou un agent fait confiance à un autre agent seulement dans des situations particulières. Par exemple, si l'agent a est l'agent détecteur qui détecte si la porte est ouverte ou non. Cet agent a peut faire confiance à un autre agent b à l'égard de sa complétude pour le fait que la porte est ouverte seulement si il y a de l'électricité.

Un autre modèle théorique intéressant est celui proposé par Cholvy [Cholvy 11]. D'après Cholvy, évidemment, un agent peut croire une nouvelle information si cet agent fait confiance à la source d'information pour livrer des informations qui sont vraies. D'une manière équivalente, un agent peut croire une nouvelle information si cet agent fait confiance à la source d'information pour ne pas livrer des informations fausses.

Ainsi, savoir si la source d'information est digne de confiance ou non aide l'agent à avoir une position épistémique envers la nouvelle information.

Par exemple, dans le but de savoir s'il va pleuvoir ou non cette après-midi, la personne consulte le site web de la météo et apprend qu'effectivement il va pleuvoir. Si la personne fait confiance à ce site pour livrer des informations prévisionnelles correctes, alors la personne croit qu'il va pleuvoir. Inversement, supposons que la personne apprend dans un journal qu'il va pleuvoir. Si la personne sait que les prévisions de ce journal sont toujours fausses, alors la personne peut croire qu'il ne va pas pleuvoir.

Le modèle de Cholvy étudie comment un agent peut croire une nouvelle information quand elle est reportée à travers différentes sources.

Par exemple, c'est le cas si la personne a été informée par sa voisine que d'après le site web de la météo, il va pleuvoir. Ici, la voisine n'a pas dit à la personne qu'il va pleuvoir mais elle a seulement rapporté que le site web de la météo a mentionné qu'il va pleuvoir. Par conséquent, la confiance accordée à la voisine pour livrer des vraies prévisions n'est pas d'une grande utilité ici. Cependant, faisant confiance à la voisine pour dire la vérité en égard de ce qu'elle a lu dans le site web et faisant confiance au site web de la météo pour livrer des prévisions vrai, va permettre à la personne de croire qu'il va pleuvoir.

9.5.1 Formalisation de la confiance

Considérons une proposition p . L'agent a peut être, plus au moins certain que p est vrai, ou que p est faux, ou encore ne pas savoir si p est vrai, ou si p est faux. Sa confiance dans les dires de l'agent b peut être plus ou moins grande, ou au contraire, il peut douter de ses dires avec un niveau de défiance plus ou moins grand. Il peut aussi avoir une information imprécise, voire nulle, sur cette confiance / défiance. Nous pouvons aussi envisager que l'agent a considère que p est plus au moins possible, sans qu'il n'ait aucune certitude sur p , c.à.d. sans qu'il ne sache rien de la possibilité de $\neg p$.

Supposons que l'agent a croit p à un certain degré α . Supposons aussi, que l'agent b affirme que p est faux avec un certain degré de certitude β . Aussi, supposons que l'agent a a confiance au niveau γ en l'agent b .

Les deux premières informations correspondent à des formules standard de la logique possibiliste multi-agents : $(p, \alpha/a)$ et $(\neg p, \beta/b)$. La dernière information peut s'écrire sous la même forme : $(b, \gamma/a)$ en assimilant les agents a et b à des propositions. Cela conduit par inférence à : $(\neg p, \min(\beta, \gamma)/a)$.

Nous constatons que nous ne sommes pas dans une situation de révision classique, qui donne la priorité à l'information entrante. Selon les valeurs relatives de α et de $\min(\beta, \gamma)$, l'agent a va continuer à croire p , ou se mettre à croire que p est faux, avec un degré de certitude affaibli par rapport à β . Si $\alpha = \min(\beta, \gamma)$, l'agent a se retrouve dans un état de doute absolu ou p et $\neg p$ lui apparaissent chacun complètement possibles.

Notons aussi qu'un tel mécanisme simple de discounting et de gestion de l'incohérence par niveau ne modifie pas la confiance de l'agent a par rapport à l'agent b , si l'agent b va à l'encontre des croyances de a , ni ne la rend plus imprécise. Nous ne mo-

délisons pas non plus la simple défiance.

L'agent b peut envoyer un, ou plusieurs, argument a l'appui de son affirmation, c'est-à-dire un ensemble minimal cohérent de propositions qui impliquent $\neg p$. Certains composants de l'argument considéré peuvent être partagées par l'agent a , d'autres lui sont inconnus. Si l'argument de b comporte des éléments qui contredisent les croyances de a , cela amènera l'agent a à affaiblir la possibilité de $\neg p$ de manière d'autant plus forte que les croyances contredites de l'agent a seront fortes. Nous remarquons que les processus de révision, de prise en compte de la confiance et d'évaluations de la force des arguments sont étroitement liés.

9.5.2 Application liées au concept de confiance

Il existe plusieurs systèmes multi-agents dans lesquels il est utile d'incorporer un système de confiance. Voici quelques exemples :

- Grands wikis (Wikipédia, Planetmath, etc.) : les agents sont les contributeurs du wikis, c'est à dire ceux qui créent, effacent, ou modifient des articles. Un utilisateur est un lecteur ou un patrouilleur. Le rôle d'un patrouilleur est d'empêcher ou de réparer les actes de vandalisme. Il serait facile de modifier un wiki pour les contributeurs puissent fournir des opinions sur leurs pairs, en particulier à la suite des longues discussions qu'ils ont sur les sujets à controverse. Un système de confiance pourrait exploiter ces opinions pour aider les utilisateurs à choisir les articles à lire ou à valider.
- Les réseaux sociaux (Facebook, MySpace, etc.) : les agents sont des humains, des applications, des sociétés, des produits, etc. Les utilisateurs sont des agents. Certains agents peuvent être mal intentionnés. Donc, une évaluation des agents serait utile aux utilisateurs pour décider avec qui partager des informations personnelles.
- Les pages web et lien hypertextuels : les agents sont les pages web. Les utilisateurs sont des entités (humains, ordinateur, etc.) qui cherchent des informations ou des services sur le web. Un lien d'une page x vers une page y peut être vu comme un avis favorable de x envers y . Ces opinions peuvent être exploitées pour aider les utilisateurs à déterminer la confiance qu'ils peuvent avoir dans le fait qu'une page web contient des informations ou des services importants.

9.6 Conclusion

Nous avons présenté dans ce chapitre des applications relatives aux logiques développées au cours de cette thèse.

Dans ce cadre, nous avons vu que la révision de croyances se focalise sur la manière avec laquelle un groupe d'agents change ses croyances en adoptant de nouvelles croyances. En particulier, nous avons présenté comment cet aspect peut être articulé dans le cadre de la logique multi-agents. Puis, nous avons introduit la notion de confiance. En particulier, nous avons montré comment elle peut être appliquée dans le contexte de la logique possibiliste multi-agents. Enfin, nous avons présenté quelques exemples d'applications liées à la notion de la confiance.

Conclusion Générale

Bilan

Le thème général de cette thèse tourne autour de la logique multi-agents et de la logique possibiliste multi-agents pour représenter et raisonner sur des connaissances détenues par des groupes d'agents.

Ces deux logiques découlent de la logique possibiliste et manipulent des ensembles finis de formules propositionnelles pondérées par des ensembles d'agents (resp. par des ensembles flous d'agents).

D'une part, la logique multi-agents permet de décrire les croyances de sous-ensembles d'agents. Une formule de cette logique multi-agents est une paire (a, A) où a est une formule propositionnelle et A un sous-ensemble d'agents. Sémantiquement, elle permet d'exprimer qu'au moins tous les agents dans le sous-ensemble A croient que la proposition a est vraie.

D'autre part, la logique possibiliste multi-agents permet de décrire les croyances graduées d'un sous-ensemble d'agents. Une formule de cette logique possibiliste multi-agents est une paire $(a, \alpha/A)$ où a est une formule propositionnelle, $\alpha \in [0, 1]$ et A un sous-ensemble d'agents. Intuitivement, elle exprime qu'au moins tous les agents dans le sous-ensemble A croient que la proposition a est vraie à au moins un degré α .

Les travaux réalisés durant cette thèse apportent diverses contributions autour de la logique multi-agents et la logique possibiliste multi-agents tant sur le plan de la représentation des connaissances que sur le plan du raisonnement.

Sur le plan de la représentation des connaissances, ils ont permis de proposer une extension des notions de la distribution de possibilités, de la mesure de possibilité et de la mesure de nécessité de la logique possibiliste. En effet, la logique possibiliste multi-agents inclut la logique possibiliste comme cas particulier quand les croyances sont rapportées à un seul agent. Cependant, les sous-ensembles d'agents, éventuellement pondérés par des degrés de certitude forment une structure partiellement ordonnée, contrairement à la logique possibiliste où les degrés de certitude sont totalement ordonnés. Aussi, les opinions conflictuelles sont souvent rencontrées dans les groupes d'agents. Pour cela, la sémantique proposée permet de distinguer entre la consistance globale d'un ensemble d'agents et leurs consistance individuelle (avec ou sans degré). En outre, la logique multi-agents est proche de la logique propositionnelle classique. Ceci permet une extension de plusieurs résultats. Ainsi, nous avons montré que la logique multi-agents est axiomatisable. De plus, la logique possibiliste multi-agents est proche de la logique multi-agents et est aussi axiomatisable.

Sur le plan du raisonnement, ce travail a permis d'étudier une méthode de déduction automatique qui étend la résolution à la logique multi-agents et à la logique possibiliste multi-agents. Des résultats de correction et de complétude ont été montrés : la résolution multi-agents (resp. possibiliste multi-agents) est complète et correcte pour la résolution. Nous avons proposé une stratégie de réfutation par résolution prenant en compte les sous-ensembles associés aux clauses. Ensuite, nous avons étendu cette stratégie pour prendre en compte des sous-ensembles flous d'agents.

En marge de ces développements, nous avons implémenté les algorithmes mis-en-œuvre afin de pouvoir les évaluer.

Perspectives

Cette étude augure diverses perspectives.

En ce qui concerne la poursuite de l'étude des méthodes de déduction automatique, nous pensons dans un premier temps étendre les algorithmes de réfutation, basés sur le principe de résolution pour prendre en compte des informations supplémentaire sur les groupes d'agents de types : $A \subseteq B$ ou $A \cap B \neq \emptyset$. Nous envisageons ensuite d'exploiter le rôle symétrique joué par a et A . Ceci va nous permettre de répondre à des questions du genre : qu'est-ce qu'un groupe d'agents croient ? Enfin, nous envisageons une étude approfondie de la complexité des différents algorithmes.

Aussi, comme mentionné dans le chapitre 9, nous proposons d'approfondir l'utilisation de la logique multi-agents et la logique possibiliste multi-agents dans le processus complexe de la confiance. En effet, quand un agent reçoit une information d'un autre agent, la façon dont il intègre éventuellement cette information dépend de son propre état de croyance initial, de la confiance /défiance qu'il a vis-à-vis de l'autre agent [Prade 07]. Cela dépend aussi, éventuellement, des arguments que l'autre agent peut apporter à l'appui de son affirmation, et de la manière dont ces arguments rencontrent les croyances du premier agent [Falappa et All. 09] [Paglieri, Castelfranchi 04].

Sur les aspects formels, il sera intéressant d'étendre les principaux résultats de ce travail à une généralisation de la logique possibiliste standard qui est la logique possibiliste généralisée [Dubois, Prade 11]. Cette dernière va nous permettre de considérer les disjonctions et les négations de formules de multi-agents.

Dans une optique plus générale, nous envisageons de nous intéresser à étendre la logique multi-agents et la logique possibiliste multi-agents pour pouvoir représenter des informations comme : au plus tous les agents dans un certain sous ensemble croient a , ou des informations comme : il existe au moins un agent dans un sous ensemble qui croit a .

Par ailleurs, la synthèse des informations éventuellement contradictoires accessibles à un agent pourrait être faite à l'aide de l'approche décrite dans [Assaghir 11]. Elle est basée sur les structures de patron de l'analyse formelle de concepts.

Bibliographie

- [Assaghir 11] Assaghir, Z. Napoli, A. Kaytoue, M. Dubois, D. Prade, H. (2011). Numerical information fusion : Lattice of answers with supporting arguments. Proc. IEEE 23rd Inter. Conf. on Tools with Artificial Intelligence, ICTAI'11, Boca Raton, Nov.7-9, 621-628.
- [Alchourrón et All. 85] Alchourrón, C. Gärdenfors, P. and Makinson, D.(1985). On the logic of theory change : Partial meet contraction and revision functions. Journal of Symbolic Logic, 50(2), 510-530.
- [Alur 97] Alur, R. Henzinger T. A. Kupferman O. (1997). Alternating-time temporal logic. In Proceedings of the 38th Annual Symposium on Foundations of Computer Science. IEEE Computer Society Press, Los Alamitos, CA, 100-109.
- [Alur 99] Alur, R. Henzinger T. A. Kupferman. O. (1999). Alternating-time temporal logic. In Compositionality : The Significant Difference, Lecture Notes in Computer Science, Springer, Berlin, 23-60.
- [Alur 02] Alur, R., Henzinger, T. A., Kupferman, O. (2002). Alternating-time temporal logic. Journal of the ACM (JACM), 49(5), 672-713.
- [Bloch 96] Bloch, I. (1996). Incertitude, imprécision et additivité en fusion de données : point de vue historique. TS. Traitement du signal, 13(4), 267-288.
- [Dubois, Prade 94 a] Dubois, D., Prade, H. (1994). La fusion d'informations imprécises. TS. Traitement du signal, 11(6), 447-458.
- [Dubois, Prade 94 b] Dubois, D., Prade, H. (1994). Possibility theory and data fusion in poorly informed environments. Control Engineering Practice, 2(5), 811-823.
- [Smets 97] Smets, P. (1997). Imperfect information : Imprecision and uncertainty. In Uncertainty Management in Information Systems. Springer US. 225-254.
- [Bloch 03] Bloch, I. (2003). Fusion d'informations en traitement du signal et des images. Hermes Science Publications.
- [Zadeh 65] Zadeh, L. A. (1965). Fuzzy sets. Information and control, 8(3), 338-353.
- [Zadeh 78] Zadeh, L.(1978). Fuzzy sets as a basis for a theory of possibility. Fuzzy sets and systems, 1, 3-28.
- [Dubois, Prade 87 a] Dubois, D., Prade, H. (1987). Necessity measures and the resolution principle. Systems, Man and Cybernetics, IEEE Transactions on, 17(3), 474-478.
- [Dubois, Prade 87 b] Dubois, D., Prade, H. (1987). An introduction to possibilistic and fuzzy logics (with discussions). P. Smets, EH Mamdani, D. Dubois, H. Prade (Eds.), Non-Standard Logics for Automated Reasoning, Academic Press, New York, 287-326.
- [Belhadi et all. 13] Belhadi, A., Dubois, D., Khellaf-Haned, F., Prade, H. (2013). Multiple agent possibilistic logic. Journal of Applied Non-Classical Logics, 23(4), 299-320.

- [Belhadi et All. 14] Belhadi, A., Dubois, D., Khellaf-Haned, F., Prade, H. (2014). Algorithme d'inférence pour la logique possibiliste multi-agents. 23ème Rencontres francophones sur la Logique Floue et ses Applications. Cargèse.
- [Belhadi et All. 13] Belhadi, A., Khellaf-Haned, F., Prade, H. (2013). Reasoning about the opinions of groupes of agents. 11th edition of the European Workshop on Multi-agent Systems. Toulouse.
- [Konolige 86] Konolige, K.(1986). A deduction model of belief. Pitman/Morgan Kaufmann.
- [Cohen 90] Cohen, P. R., Levesque, H. J. (1990). Intention is choice with commitment. *Artificial intelligence*, 42(2), 213-261.
- [Bratman 87] Bratman, M. (1987). Intention, plans, and practical reason.
- [Bratman 90] Bratman, M. E. (1990). What is intention. *Intentions in communication*, 15-32.
- [Rao 91 a] Rao, A. S., Georgeff, M. P. (1991). Modeling rational agents within a BDI-architecture. *KR*, 91, 473-484.
- [Rao 91 b] Rao, A. S., George, M. P. (1991, August). Asymmetry thesis and side-effect problems in linear time and branching time intention logics. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, 498-504.
- [Pauly 01] Pauly, M. (2001). *Logic for social software*. Universiteit van Amsterdam.
- [Goranko 04] Goranko, V., Jamroga, W. (2004). Comparing semantics of logics for multi-agent systems. *Synthese*, 139(2), 241-280.
- [Goranko 06] Goranko, V., Van Drimmelen, G. (2006). Complete axiomatization and decidability of alternating-time temporal logic. *Theoretical Computer Science*, 353(1), 93-117.
- [Herzig 08] Herzig, A., Schwarzentruher, F. (2008). Properties of logics of individual and group agency. *Advances in modal logic*, 7, 133-149.
- [Kooi 08] Kooi, B., Tamminga, A. (2008). Moral conflicts between groups of agents. *Journal of Philosophical Logic*, 37(1), 1-21.
- [Dubois, Prade 80] Dubois, D. J. (1980). *Fuzzy sets and systems : theory and applications*. Academic press.
- [Dubois, Prade 83] Yager, R.R. (1983). An introduction to applications of possibility theory. *Human Systems Management*, 3, 246-249.
- [Dubois, Prade 86] Dubois, D., Prade, H. (1986). Possibilistic inference under matrix form. *Fuzzy Logic in Knowledge Engineering*, Verlag TUV Rheinland, Köln, 112-126.
- [Dubois et al. 88] Dubois, D., Lang, J., Prade, H. (1988). Advances in automated reasoning using possibilistic logic. Reprints of the Europ. Workshop on Logical Methods for Artificial Intelligence , 95-99.
- [Dubois et al. 94 a] Dubois, D., Lang, J., Prade, H. (1994). Automated reasoning using possibilistic logic : Semantics, belief revision, and variable certainty weights. *IEEE Transactions on Knowledge and Data Engineering*, 6(1), 64-71.
- [Dubois, Prade 85] Dubois, D., Prade, H. (1985). *Théorie des possibilité- Application a la représentation des Connaissances en Informatique*.
- [Froidevaux, Grossetête 90] Froidevaux, C., Grossetête, C. (1990). Graded default theories for uncertainty. In *Proceedings of the 9th European Conference on Artificial Intelligence*, Stockholm.

- [Dubois, Prade 90] Dubois, D., Prade, H. (1990). Resolution principles in possibilistic logic. *International Journal of Approximate Reasoning*, 4(1), 1-21.
- [Farreny, Ghallab 87] Farreny, H., Ghallab, M. *Éléments d'Intelligence Artificielle*, Hermès.
- [Dubois et al. 87] Dubois, D., Lang, J., Prade, H. (1987, August). Theorem proving under uncertainty : a possibility theory-based approach. In *Proceedings of the 10th international joint conference on Artificial intelligence*. Morgan Kaufmann Publishers Inc, 984-986.
- [Yager 86] Yager, R. R. (1986). Paths of least resistance in possibilistic production systems. *Fuzzy sets and systems*, 19(2), 121-132.
- [Dubois et al. 94 b] Dubois D., Lang J., Prade H. (1994). Possibilistic logic. In : *Handbook of Logic in Artificial Intelligence and Logic Programming*, Vol. 3 (D.M. Gabbay, C.J. Hogger, J.A. Robinson, D. Nute, eds.), Oxford University Press, 439-513.
- [Lang 00] Lang, J. (2000). Possibilistic logic : complexity and algorithms. In *Handbook of Defeasible Reasoning and Uncertainty Management Systems*. Springer Netherlands, 179-220.
- [Dubois, Prade 04] Dubois, D., Prade, H. (2004). Possibilistic logic : a retrospective and prospective view. *Fuzzy sets and systems*, 144(1), 3-23.
- [Smets 88] Smets, Ph. (1988). Belief functions. In : *Non-Standard logics for Automated Reasoning*, (Ph. Smets, E. H. Mamdani, D. Dubois, H. Prade, eds.), Academic Press, 253-277.
- [Dubois, Prade 07] Dubois, D., Prade, H. (2007). Toward multiple-agent extensions of possibilistic logic. In *Fuzzy Systems Conference. FUZZ-IEEE 2007*. IEEE International. IEEE, 1-6.
- [Robinson 65] Robinson, J. A. (1965). A machine-oriented logic based on the resolution principle. *Journal of the ACM (JACM)*, 12(1), 23-41.
- [Davis, Putnam 60] Davis, M., Putnam, H. (1960). A computing procedure for quantification theory. *Journal of the ACM (JACM)*, 7(3), 201-215.
- [Gödel 86] Gödel, K. (1986). *collected works*. S. Feferman, J. Dawson Jr., S. Kleene, G. Moore, R. Solovay, and J. van Heijenoort, editors. Oxford University Press, Inc., New York, NY, USA, 1929-1936.
- [Cook 71] Cook, S. (1971). The complexity of theorem-proving procedures. In *STOC*, 151-158.
- [Belhadi et All. 15] Belhadi, A., Dubois, D., Khellaf-Haned, F., Prade, H. (2015). La logique possibiliste multi-agent : une introduction. *24ème Rencontres francophones sur la Logique Floue et ses Applications*. Poitiers.
- [Colmerauer 90] Colmerauer, A. (1990, January). An introduction to Prolog III. In *Computational Logic*. Springer Berlin Heidelberg, 37-79.
- [Imai, Iséki 66] Imai, Y., Iséki, K. (1966). On axiom systems of propositional calculi. XIV. *Proceedings of the Japan Academy*, 42(1), 19-22.
- [Dubois, Prade 92] Dubois, D. Prade, H. (1992). Belief change and possibility theory. In P. Gärdenfors Eds, editor, *Belief Revision*. Cambridge University Press, 142-182.
- [Dubois, Prade 97] Dubois, D. Prade, H. (1997). A synthetic view of belief revision with uncertain inputs in the framework of possibility theory. *Int. J. Approx. Reasoning*, 295-324.

- [Artz, Gil 07] Artz, D., Gil, Y. (2007). A survey of trust in computer science and the semantic web. *Web Semantics : Science, Services and Agents on the World Wide Web*, 5(2), 58-71.
- [Pusey et All. 07] Pusey, B., Maitland, C., Tapia, A., Yen, J. (2007). A Survey of Trust Models in Agent Applications.
- [Rao, Georgeff 95] Rao, A. S., Georgeff M.(1995). BDI Agents : from theory to practice. In *Proceedings of the First International Conference on Multi-Agent Systems*, San Francisco, 312-319.
- [Chellas 80] Chellas, B. F. (1980). *Modal logic : an introduction*. Cambridge : Cambridge university press.
- [Pichon et All. 12] Pichon, F., Dubois, D., Denoeux, T. (2012). Relevance and truthfulness in information correction and fusion. *International Journal of Approximate Reasoning*, 53(2), 159-175.
- [Castelfranchi, Tan 01] Castelfranchi, C., Tan, Y.H. (2001). *Trust and deception in virtual societies*. Kluwer Academic Publishers, Dordrecht.
- [Herzig et All 10] Herzig, A., Lorini, E., HÅbner, J. F., Vercouter, L. (2010). A logic of trust and reputation. *Logic Journal of the IGPL*. 18(1), 214-244.
- [Demolombe 04] Demolombe, R. (2004). Reasoning about trust : A formal logical framework. In *Trust Management*. Springer Berlin Heidelberg, 291-303.
- [Cholvy 11] Cholvy, L. (2011). How strong can an agent believe reported information ?. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*. Springer Berlin Heidelberg. 386-397.
- [Falappa et All. 09] Falappa, M. Kern-Isberner, G. Simari, G. (2009). Belief revision and argumentation theory. In : *Argumentation in Artificial Intelligence* (I. Rahwan, G. Simari, eds.), Springer, 341-360.
- [Paglieri, Castelfranchi 04] Paglieri, F. Castelfranchi, C. (2004). Revising beliefs through arguments : Bridging the gap between argumentation and belief revision in MAS. In : *Proc. ArgMAS*, (I. Rahwan, P. Moraitis, C. Reed, eds.), LNCS 3366, Springer, 78-94.
- [Prade 07] Prade, H. (2007). A qualitative bipolar argumentative view of trust. *Proc. 1st Int. Conf. on Scalable Uncertainty Management (SUM 2007)*, Washington, DC, Oct. 10-12, (H. Prade, V. S. Subrahmanian, eds.), Springer, LNAI 4472, Springer, 268-276.
- [Dubois, Prade 11] Dubois, D., Prade, H. (2011). Generalized possibilistic logic. In *Scalable Uncertainty Management*. Springer, 428-432.
- [Benferhat et al. 93] Benferhat, S., Cayrol, C., Dubois, D., Lang, J., Prade, H. (1993). Inconsistency management and prioritized syntax-based entailment. In *IJCAI*, 640-645.
- [de Kleer 90] de Kleer, J. (1990). Using crude probability estimates to guide diagnosis. *Artificial Intelligence*, 45(3), 381-391.
- [Lehmann 95] Lehmann, D. (1995). Belief revision, revised. In *Proceedings of the 14th international joint conference on Artificial intelligence*, 1534-1540.
- [Papini 92] Papini, O. (1992). A complete revision function in propositional calculus. In *Proceedings of the 10th European conference on Artificial intelligence*. 339-343.
- [Wurbel et all. 00] Wurbel, E., Jeansoulin, R., Papini, O. (2000). Revision : an application in the framework of GIS. In *principles of knowledge representation and reasoning-international conference*. 505-518.

- [Belhadi et All. 16] Belhadi, A., Dubois, D., Khellaf-Haned, F., Prade, H. (2016). Reasoning with multiple-agent possibilistic logic. Scalable Uncertainty Management. Nice.
- [Benferhat et al. 10 a] Benferhat, S., Ben-Naim, J., Papini, O., Würbel, E. (2010). An answer set programming encoding of prioritized removed sets revision : application to GIS. Applied Intelligence, 60-87.
- [Katsuno, Mendelzon 91] Katsuno, H., Mendelzon, A. O. (1991). Propositional knowledge base revision and minimal change. Artificial Intelligence, 263-294.
- [Grove 88] Grove, A. (1988). Two modellings for theory change. Journal of philosophical logic, 157-170.
- [Alchourrón et al. 85] Alchourrón, C. E., Gärdenfors, P., Makinson, D. (1985). On the logic of theory change : Partial meet contraction and revision functions. The journal of symbolic logic, 510-530.
- [Gärdenfors 88] Gärdenfors, P. (1988). Knowledge in flux : Modeling the dynamics of epistemic states. The MIT press.
- [Dubois et al. 98] Dubois, D., Moral, S., Prade, H. (1998). Belief change rules in ordinal and numerical uncertainty theories. In Belief change, 311-392.
- [Dubois, Prade 92] Dubois, D., Prade, H. (1992). Belief change and possibility theory. Belief revision, 142-182.
- [Dubois, Prade 97] Dubois, D., Prade, H. (1997). A synthetic view of belief revision with uncertain inputs in the framework of possibility theory. International Journal of Approximate Reasoning, 295-324.
- [Benferhat et all. 10 b] Benferhat, S., Dubois, D., Prade, H., Annewilliams, M. A. (2010). A framework for iterated belief revision using possibilistic counterparts to Jeffrey's rule. Fundamenta Informaticae, 147-168.

Résumé

Dans le domaine de l'Intelligence Artificielle, la problématique de la représentation des connaissances incertaines et du raisonnement constitue un pôle important. Issue de la théorie des possibilités, développée par Dubois et Prade, la logique possibiliste se propose de prendre en compte l'incertitude dont sont entachées les connaissances.

Avec l'avènement du paradigme multi-agents, il s'est avéré naturel et nécessaire de développer une logique possibiliste afin de prendre en compte des croyances partagées par un ensemble d'agents.

Sur le plan de la représentation, cette thèse propose dans un premier temps une extension multi-agents de la logique propositionnelle. La logique multi-agents a pour but de modéliser les croyances d'un ensemble d'agents. Dans un second temps, une autre extension dans le cadre de la logique possibiliste collective graduée est proposée. Cette deuxième extension, nommée logique possibiliste multi-agents, permet de décrire les croyances graduées d'un sous-ensemble d'agents.

Du point de vue du raisonnement, un algorithme de réfutation par résolution pour une base de connaissances multi-agents a été développé. Ce dernier est une adaptation de l'algorithme A^* . Un algorithme d'inférence pour la logique possibiliste multi-agents est aussi développé. Ce dernier est une généralisation de l'algorithme d'inférence de la logique multi-agents. Les deux algorithmes proposés ont été mis en œuvre et une étude expérimentale a été ensuite menée. Les résultats ont été très prometteurs en utilisant différentes tailles d'instances des connaissances.

Mots Clés : Logique possibiliste, Incertitude, Logique multi-agents, Logique possibiliste multi-agents, Réfutation par résolution multi-agents, Réfutation par résolution possibiliste multi-agents.

Abstract

In Artificial Intelligence, the problem of representation and reasoning with uncertain knowledge constitutes an important pole. Issued from possibility theory, developed by Dubois and Prade, the possibilistic logic intends to take into account uncertainty related to knowledge.

With the advent of multiple agents paradigm, it has been natural and necessary to develop a possibilistic logic for beliefs shared by a set of agents.

In the field of the representation, this thesis presents in a first time a multiple agents extension of propositional logic. The multiple agents logic aims to modelize beliefs of a set of agents. In a second time, another extension in the framework of collective graded possibilistic logic is proposed. This second extension, called multiple agents possibilistic logic, allows to describe graded beliefs of a set of agents.

In the field of the reasoning, a refutation by resolution algorithm is developed for a multiple agents knowledge base. It is an adaptation of the A^* algorithm. An inference algorithm for possibilistic logic is also developed. The latter, is a generalization of a multiple agents logic inference algorithm. The two proposed algorithms have been implemented and an experimental study was then carried out. The results are very promising using different sizes of knowledge.

Keywords : Possibilistic logic, Uncertainty, Multiple agents logic, Multiple agents possibilistic logic, Multiple agents refutation by resolution, Multiple agents possibilistic refutation by resolution.