

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

Université d'Artois
Faculté Jean Perrin Lens

THESE EN COTUTELLE

présentée pour l'obtention du grade docteur en:

INFORMATIQUE

Spécialité:

Intelligence Artificielle

Par:

BOUTOUHAMI Khaoula



**Les modèles graphiques et logiques pour la gestion
des informations incohérentes et incertaines**

Soutenue publiquement le 3 décembre 2017, devant le jury composé de :

Aicha AISSANI	à l'USTHB	Président	Professeur
Faiza KHELLAF	à l'USTHB	Directrice de thèse	Professeur
Salem BENFERHAT	à l'Université d'Artois	Co-directeur de thèse	Professeur
Odile PAPINI	à l'Université Aix-Marseille	Examinatrice	Professeur
Bertrand MAZURE	à l'Université d'Artois	Examineur	Professeur
Narhimene BOUSTIA	à l'Université de Blida	Examinatrice	MCA
Farid NOUIOUA	à l'Université Aix-Marseille	Invité	MCA

Remerciements

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

Je remercie chaleureusement le professeur Odile PAPINI de m'avoir fait l'honneur d'accepter d'être le rapporteur de ce travail.

J'adresse mes remerciements au docteur Narhimene BOUSTIA d'avoir répondu à notre demande pour faire partie de ce jury en tant que rapporteur.

J'adresse mes remerciements au professeur Bertrand MAZURE d'avoir accepté d'examiner ce travail.

Je remercie fortement le docteur Farid NOUIOUA d'avoir bien voulu accepter d'examiner ce travail.

Je tiens à exprimer mes reconnaissances au professeur Hadja Faiza KHELLAF-HANED pour avoir accepté d'être ma directrice de thèse et pour ses conseils précieux ainsi que pour sa disponibilité.

Je voudrais exprimer ma profonde reconnaissance au professeur Salem BENFERHAT de m'avoir accordé sa confiance en proposant ce sujet de thèse et pour l'avoir dirigé.

Je remercie mes parents, mon frère, mes sœurs et tous les autres membres de ma famille qui n'ont jamais cessé de me soutenir.

Je remercie tous ceux qui m'ont encouragé.

Résumé

Abstract

In this thesis, we have studied logical and graphical models for the management of incoherent and uncertain information. In the first part, we have studied an extension of lightweight ontologies, encoded here in DL-Lite languages, for the product-based possibility theory framework. We first have introduced the language and the semantics used for representing uncertainty in lightweight ontologies. Then, we have shown that contrarily to a min-based possibilistic DL-Lite, query answering in a product-based possibility theory is a hard task. When the uncertainty is only considered at the ABox level, we have provided equivalent transformations between the inconsistency degree computation problem (the key notion in reasoning from a possibilistic DL-Lite knowledge base) and the weighted maximum 2-Horn SAT problem. In the general case, where the TBox may also be uncertain, we have modeled the inconsistency degree computation as an integer linear programming problem. Moreover, we have provided in both cases, an encoding of the problem of computing inconsistency degree in product-based possibility DL-Lite as a weighted set cover problem and we have used a greedy algorithm to compute an approximate value of the inconsistency degree. This encoding allows us to provide an approximate algorithm for answering instance checking queries in product-based possibilistic DL-Lite. Lastly, we have presented an experimental study where the different proposed solutions are compared. We have shown in particular the efficiency of the integer linear programming approach compared with the two other solutions based on the weighted Max-2-Horn-SAT and the approximate greedy algorithm.

In the second part of the thesis, we have implemented the problem of decision-making. We have implemented and, improved a new version of the possibilistic influence diagram decomposition process into two possibilistic networks without reduction, reducing the complexity of the resulting networks and thus improving the computation of the optimistic decision making process. In addition, we have proposed an approximate approach for the computation of decision under uncertainty within possibilistic networks. The computing of the optimal optimistic decision no longer goes through the junction tree construction step. Instead, it is performed by calculating the degree of normalization in the moral graph resulting from the merging of the possibilistic network codifying knowledge of the agent and that codifying its preferences.

Résumé

Dans cette thèse, nous avons étudié des modèles logiques et graphiques pour la gestion d'informations incohérentes et incertaines. Dans la première partie, nous avons étudié une extension des ontologies légères, exprimée ici dans les langages DL-Lite, dans le cadre de la théorie des possibilités basée sur le produit. Nous introduisons d'abord le langage et la sémantique utilisés pour représenter l'incertitude dans les ontologies légères. Nous montrons ensuite que, contrairement à la logique DL-Lite possibiliste basée sur l'opérateur min, le traitement des requêtes dans une théorie de possibilité basée sur le produit est une tâche difficile. Lorsque l'incertitude est considérée seulement au niveau des assertions de la ABox, nous fournissons des transformations équivalentes entre le problème du calcul du degré d'inconsistance (la notion clé dans le raisonnement à partir d'une base de connaissances DL-Lite possibiliste) et le problème Max-2-Horn-SAT pondéré. Dans le cadre général où la TBox peut également être incertaine, nous modélisons le calcul du degré d'inconsistance par un problème de programmation linéaire en nombres entiers. De plus, nous proposons dans les deux cas, un encodage du problème de calcul du degré d'inconsistance en terme d'un problème de couverture d'ensembles pondérés et nous utilisons un algorithme glouton pour calculer une valeur approximative du degré d'inconsistance. Enfin, nous présentons une étude expérimentale où les différentes solutions proposées sont comparées. Nous montrons en particulier l'efficacité de l'approche basée sur la programmation linéaire en nombres entiers par rapport aux deux autres approches basées sur le W-Max-2-Horn-SAT et l'algorithme glouton approximatif.

Dans la deuxième partie de la thèse, nous avons étudié le problème de la prise de décision. Nous avons implémenté une version améliorée du procédé de décomposition d'un diagramme d'influence possibiliste en deux réseaux possibilistes sans réduction, permettant de réduire la complexité des réseaux résultants et offrant ainsi une amélioration du processus de calcul des décisions optimales optimistes. De plus, nous avons proposé une approche approximative pour le calcul de la décision possibiliste qualitative en incorporant les techniques de fusion des réseaux possibilistes. Cette approche est efficace et elle est en particulier utile lorsque la génération des distributions de possibilités locales par l'algorithme standard est impossible ou prend un temps de réponse trop long.

Table des matières

Remerciements	1
Résumé	2
Table des figures	8
Liste des tableaux	10
Introduction Générale	11
I Etat de l'art	17
1 Gestion de l'incertitude	19
1.1 Introduction	19
1.2 Théories de l'incertain	20
1.2.1 Notations	20
1.2.2 La théorie des probabilités	21
1.2.2.1 Concepts de base	21
1.2.2.2 Indépendance et théorème de Bayes	22
1.2.3 La théorie des fonctions de croyance	23
1.2.3.1 Fonction de masse	24
1.2.3.2 Fonction de croyance	24
1.2.3.3 Fonction de plausibilité	25
1.2.3.4 La règle de combinaison de Dempster	26
1.2.4 La théorie des ensembles flous	26
1.2.4.1 Ensembles flous	27
1.2.4.2 Opération sur les ensembles flous	30
1.2.4.3 Implication floue et règles floues	32
1.2.4.4 Le contrôleur flou	32
1.2.5 La théorie des possibilités	33
1.2.5.1 Distribution possibiliste	33
1.2.5.2 Mesure de possibilité et mesure de nécessité	34
1.2.5.3 Conditionnement possibiliste	35
1.2.5.4 Indépendance possibiliste	35
1.3 Modèles possibilistes Logiques et graphiques	35
1.3.1 Modèle logique	36
1.3.1.1 La logique possibiliste standard	36
1.3.1.2 Distribution de possibilités associée à une base possibiliste	36

1.3.1.3	Degré d'inconsistance	37
1.3.2	Modèles graphiques	38
1.3.2.1	Réseaux possibilistes	38
1.3.2.2	Les arbres de décision possibilistes	40
1.3.2.3	Diagrammes d'influence possibilistes	40
1.4	Conclusion	45
2	Les logiques de description	46
2.1	Introduction	46
2.2	Le Web Sémantique	46
2.3	Ontologies	47
2.4	Les logiques de description	48
2.4.1	Des logiques de description simples	50
2.4.1.1	La logique de description \mathcal{AL}	50
2.4.1.2	La logique de description \mathcal{EL}	51
2.4.1.3	La logique de description \mathcal{ALC}	51
2.4.2	Services d'inférences standards	52
2.4.3	L'inconsistance dans la logique de description	53
2.5	La famille DL-Lite	54
2.5.1	Syntaxe et sémantique	54
2.5.2	Fermeture négative dans DL-Lite	56
2.5.3	Calcul des conflits	57
2.6	Extensions existantes de la logiques de description	58
2.7	Conclusion	59
II	Modèle logique pour extension DL-lite	60
3	Les fondements de la logique Pb-π-DL-Lite	62
3.1	Introduction	62
3.2	DL-Lite possibiliste basée sur le produit	63
3.2.1	La syntaxe de Pb- π -DL-Lite	63
3.2.2	La sémantique de Pb- π -DL-Lite	65
3.2.2.1	La distribution possibiliste	65
3.2.2.2	D'une base Pb- π -DL-Lite vers une distribution possibiliste	65
3.3	Mesure d'inconsistance	67
3.4	Réponse aux requêtes dans une Pb- π -DL-Lite	68
3.5	La fermeture négative en Pb- π -DL-Lite	72
3.6	Comparaison avec les extensions DL-Lite	73
3.7	Conclusion	75
4	Analyse de la complexité de la logiques Pb-π-DL-Lite	76
4.1	Introduction	76
4.2	Les problèmes calculatoire de décision utilisés dans cette thèse	77
4.2.1	Le probleme de Max-2-Horn-SAT pondéré	77
4.3	Le calcul du degré d'inconsistance	78
4.3.1	La transformation d'une base Pb- π -DL-Lite en une base de connaissances 2-Horn pondérée	78
4.3.2	L'algorithme du calcul du degré d'inconsistance	84

4.3.3	La transformation inverse d'une base de connaissances 2-Horn vers une base Pb- π -DL-Lite	86
4.4	Conclusion	87
5	Une approche approximative pour Pb-π-DL-Lite à base de couverture par ensembles	88
5.1	Introduction	88
5.2	Problème de couverture par ensembles	88
5.3	L'approche approximative pour calculer le degré d'inconsistance	90
5.3.1	La modélisation du problème	90
5.3.2	Algorithme pour le calcul du degré d'inconsistance	92
5.4	Algorithme approximatif pour la réponse aux requêtes de vérification d'instance	93
5.5	Étude expérimentale	95
5.5.1	Évaluation de l'algorithme approximatif	96
5.6	Conclusion	99
6	Extension du calcul des degrés d'inconsistance pour des TBOX incertaines	100
6.1	Introduction	100
6.2	La programmation linéaire	101
6.3	Extension aux TBOX incertaines	102
6.4	Calcul du degré d'inconsistance	103
6.4.1	La fermeture négative étendue	103
6.4.2	Modélisation du problème	105
6.5	Étude expérimentale	109
6.6	Conclusion	111
III	Modèles graphiques pour la prise de la décision	112
7	Les approches graphiques pour la prise de décision dans l'incertitude	114
7.1	Introduction	114
7.2	Concepts de base de la théorie de la décision	115
7.2.1	Les choix	115
7.2.2	Les préférences	115
7.3	Le calcul de la décision séquentielle	116
7.3.1	La démarche du calcul de la décision séquentielle	116
7.3.2	Algorithme du calcul de la décision séquentielle	119
7.3.3	Etude expérimentale	123
7.4	L'approche approximative pour la prise de la décision exclusive	131
7.4.1	La démarche du calcul de la décision approximative	131
7.4.1.1	Fusion des réseaux possibilistes	132
7.4.1.2	La construction du graphe moral	132
7.4.1.3	L'initialisation pour chaque décision d_i	133
7.4.1.4	Procédure de stabilité simple	133
7.4.1.5	Procédure de stabilité multiple	133
7.4.2	Algorithme approximatif	134
7.5	Conclusion	138
	Conclusion générale et Perspectives	139

Table des figures

1.1	Fonction d'appartenance en théorie des ensembles flous avec un ensemble d'univers discret	27
1.2	Fonction d'appartenance en théorie des ensembles flous avec un ensemble d'univers continu	28
1.3	Fonction d'appartenance triangulaire	28
1.4	Fonction d'appartenance singleton	29
1.5	Fonction d'appartenance gaussienne	29
1.6	Éléments caractéristiques d'un sous-ensemble flou	30
1.7	Fonction d'appartenance de l'union	31
1.8	Fonction d'appartenance de l'intersection	31
1.9	Modèle d'un DAG	39
1.10	La composante graphique du diagramme d'influence de l'Exemple 1.8	42
2.1	Les différentes familles de logiques de description	50
5.1	Le système de calcul du degré d'inconsistance	96
5.2	Les performances de l'algorithme approximative	97
5.3	$Inc_{Exc}(\mathcal{K})$ est calculé par l'algorithme exact et $Inc_{App}(\mathcal{K}_1)$ est calculé par l'algorithme approximatif	98
5.4	$Inc_{App}(\mathcal{K})$ et $Inc_{App}(\mathcal{K}_1)$ sont calculés par l'algorithme approximatif	98
6.1	Schéma récapitulatif du déroulement de l'expérimentation	109
6.2	Comparaison entre les temps de réponse des deux outils Clpex et W-Max-SAT	110
6.3	Comparaison entre Clpex et l'algorithme approximatif	111
7.1	Le calcul de décision en se basant sur la décomposition d'un diagramme d'influence	119
7.2	Le Diagramme d'influence	121
7.3	Le réseau de connaissances	122
7.4	Le réseau de préférences	122
7.5	Le réseau résultant de la fusion des deux réseaux (7.3 et 7.4)	122
7.6	L'arbre de jonction	122
7.7	Exemple d'un diagramme possibiliste de type 1	124
7.8	Exemple d'un diagramme possibiliste de type 2	125
7.9	Exemple d'un diagramme possibiliste de type 3	125
7.10	Exemple d'un diagramme possibiliste de type 4	126
7.11	Le temps de calcul de la décision optimiste pour les DIPs de type1	127
7.12	Le temps de calcul de la décision optimiste pour les DIPs de type2	128
7.13	Le temps de calcul de la décision optimiste pour les DIPs de type3	129

7.14	Le temps de calcul de la décision optimiste pour les DIPs de type4	130
7.15	Comparaison entre le temps de réponse du premier modèle avec et sans réduction	131
7.16	Les réseaux possibiliste de connaissances et de préférences de l'agent	135
7.17	Le DAG G_{\oplus}	136
7.18	Le graphe moral MG	137

Liste des tableaux

1.1	Exemples d'implications conjonctives	32
1.2	Distribution de possibilités associée à la base de connaissances de l'exemple 1.6	37
1.3	Distributions initiales	39
1.4	Distributions de possibilités jointes	39
1.5	Classification des diagrammes d'influence possibilistes	42
1.6	Les possibilistes conditionnelles de A_1	42
1.7	Les possibilistes conditionnelles de A_2	42
1.8	La fonction d'utilités $u(D_1, D_2, A_2)$	43
1.9	L'application de la règle de chaînage du diagramme d'influence possibiliste de la Figure 1.10	43
2.1	Syntaxe et Sémantique de termes de concept et de rôle	49
7.1	Résumé du problème de l'investissement immobilier	116
7.2	Les distributions de possibilités	121
7.3	Distribution de possibilité initiale ΠK_{\min}	135
7.4	Distribution de possibilité initiale ΠK_{\min}	135
7.5	Distribution de possibilité initiale ΠP_{\min}	135
7.6	Distribution de possibilité initiale ΠP_{\min}	136
7.7	Distribution de possibilité initiale ΠG_{\oplus}	136
7.8	Distribution de possibilité initiale ΠG_{\oplus}	136
7.9	Distribution jointes Π_{MG}	137
7.10	Distributions jointes Π_{MG} après la procédure d'initialisation	138

Introduction Générale

Dans de nombreuses applications, l'intérêt pour le développement et l'application des ontologies est croissant. La représentation et le traitement des ontologies sont très difficiles. Il existe de nombreux langages de représentation pour les ontologies. Parmi eux, se hissent les logiques de description.

Dans ces langages, la représentation des connaissances du domaine s'appuie sur les notions de concepts et de rôles. Des constructeurs, qui diffèrent selon la logique de description considérée, permettent de construire des concepts ou des rôles plus complexes à partir d'autres. Les relations entre concepts et rôles sont exprimées par des axiomes d'inclusion de concepts ou de rôles.

Historiquement, les logiques de description ont été conçues pour mettre en œuvre des raisonnements sur une ontologie (c'est-à-dire pour comparer deux classes ou un individu et une classe par subsumption). Malheureusement, à cause de leur complexité élevée, ces logiques sont peu adaptées au problème de l'interrogation d'une base de connaissances. Pour répondre à ce problème, de nouvelles logiques de description, moins expressives, mais ayant une complexité moindre, ont été proposées et baptisées logiques de description légères.

DL-Lite est une famille de logiques de description légères. Elle a été conçue pour exprimer des ontologies simples tout en conservant une complexité basse pour l'interrogation. Elle permet non seulement de vérifier la subsumption de concepts ou la satisfiabilité de la base de connaissances, comme c'était le cas pour les logiques de description standards, mais aussi de répondre à des requêtes complexes et notamment aux requêtes conjonctives.

L'incertitude est omniprésente dans le monde réel et le raisonnement en sa présence est l'un des défis les plus pressants dans le développement des systèmes intelligents. Il est donc difficile d'imaginer un vrai succès du projet du Web sémantique sans pouvoir représenter et raisonner sous l'incertitude. En fait, le traitement de l'incertitude est essentiel pour les ontologies du fait que les informations disponibles sur le Web conduisent naturellement à une évolution continue des ontologies et à un besoin permanent d'intégrer de données (alignement de schéma et d'instance) ou de l'apprentissage de l'ontologie. Cependant, les logiques de description ne sont pas appropriées pour traiter des informations imprécises ou incertaines.

Afin de résoudre cette limite, plusieurs extensions de logique de description ont été proposées en s'appuyant sur diverses extensions de la logique classique. Dans le passé, des extensions probabilistes, possibilistes et floues ont été proposées. Les approches probabilistes et possibilistes permettent de représenter des informations incertaines, alors que l'approche floue reflète l'imprécision et permet de représenter une connaissance imprécise. Ces extensions ont des différences fondamentales en termes de sémantique et donc dans les types d'informations qu'elles modélisent. Par exemple, les extensions floues et possibilistes de la

logique de description associent pour chaque formule un degré numérique compris dans l'intervalle $[0, 1]$, mais les significations de ces degrés diffèrent : en effet, dans le cas flou, ils sont interprétés en termes de degrés de vérité, alors que dans le cas de possibiliste, ils correspondent à des degrés d'incertitude.

Pour la théorie des possibilités, deux principales variantes ont été proposées : la théorie des possibilités basée sur le min (qualitative) et la théorie des possibilités basée sur le produit (quantitative).

Au niveau sémantique, ces deux théories partagent les mêmes définitions, y compris les concepts de distribution de possibilités, les mesures de nécessité, les mesures de possibilité et la définition des distributions de possibilités normalisées. Cependant, elles diffèrent dans la façon dont ils définissent le conditionnement et aussi dans la manière dont les degrés de possibilité sont définis par rapport aux interprétations.

Dans le contexte de l'extension possibiliste de DL-Lite, si les degrés d'incertitude représentent un codage de plausibilité entre les assertions (un pré-ordre total), alors la théorie possibiliste basée sur de min [13] est la plus appropriée. En revanche, si le degré d'incertitude représente un degré de surprise dans le sens des fonctions conditionnelles ordinales de Spohn ou un résultat de la transformation d'une distribution de probabilité en une distribution de possibilité, alors, c'est la théorie possibiliste basée sur le produit qui est la plus adaptée. Nous pouvons imaginer l'exemple suivant où les degrés de surprise sont présents dans la Tbox et la Abox. Supposons par exemple qu'il faille définir une connaissances de l'ontologie des hôtels du centre-ville de Paris. Par souci de simplicité, nous supposons que seulement deux types d'hôtels : hôtels quatre étoiles et hôtels une étoile. Supposons que, nous avons seulement deux commodités offertes par les hôtels sont disponibles : parking gratuit et piscine. Ensuite, nous pouvons avoir une ontologie incertaine où il sera surprenant de voir un hôtel une étoile avec un parking gratuit (respectivement une piscine). La logique DL-Lite possibiliste basé sur le produit permet de distinguer entre les différentes interprétations DL-Lite sur la base des éléments présents dans Abox \mathcal{A} . Par exemple, il peut considérer que les situations normales sont celles où nous avons des hôtels 4 étoiles avec piscine. Une situation surprenante est une situation où nous avons un hôtel une étoile avec une seule commodité (piscine ou parking gratuit). Et une situation très surprenante est d'avoir un hôtel d'une étoile avec les deux commodités.

Un autre problème crucial en intelligence artificielle est celui de la prise de décision. La nature humaine, fait que nous avons tendance à vouloir choisir les décisions qui maximisent notre satisfaction, même si on se retrouve dans des environnements où règnent parfois l'imprécision et/ou l'incertitude. Maximiser les satisfactions revient à choisir entre une liste d'alternatives possibles, compte tenu des connaissances d'un agent ainsi que ses préférences, en considérant les résultats possibles de sa décision exprimée par un ensemble d'utilités. Le choix de la stratégie optimale est aussi influencé par la nature du décideur qui peut être :

- Optimiste : le décideur choisit la décision qui a le gain maximal, même si c'est un choix risqué.
- Pessimiste : le décideur choisit la décision la moins risquée.
- Neutre : le décideur est neutre.

La prise de décision est un problème difficile et le choix de la meilleure décision possible fait intervenir de nombreux facteurs et de ce fait, ce choix est dans la plupart du temps délicat à effectuer. Ceci implique le besoin d'avoir des méthodes et des critères permettant à la fois de calculer les meilleurs choix et de justifier la manière dont ils ont été obtenus.

La théorie de la décision vise à comprendre, à modéliser et à expliquer cette tendance au

moyen de modèles logiques et graphiques tels que ceux offerts par la théorie des possibilités. Le modèle de décision possibiliste qualitative permet une expression progressive des préférences ainsi que des connaissances du décideur. Ce modèle propose deux critères d'utilités qualitatifs pour l'approche de décision sous incertitude : le critère de décision pessimiste et le critère de décision optimiste.

L'intérêt pour la problématique du calcul de décision qualitative ne cesse d'augmenter et plusieurs approches et modèles ont été proposés. Chaque méthode a pour objectif, en plus du calcul de décision, d'améliorer et de répondre aux inconvénients des autres méthodes, en termes de temps de calcul, du mode de présentation et de la facilité de compréhension.

Notre objectif dans cette thèse est double : d'un côté, nous étudions une nouvelle extension appelée DL-Lite possibiliste basée sur le produit, désignée par Pb- π -DL-Lite. À notre connaissance, il n'existe pas de travail de recherche qui étend les logiques de description, y compris les logiques de description légères telles que DL-Lite à la théorie des possibilités basée sur le produit. D'un autre côté, nous explorons le domaine de la prise de décision sous incertitude en utilisant des modèles graphiques possibilistes. Nous présentons deux modèles pour le calcul des décisions optimales optimistes.

Contributions

Dans le cadre de cette thèse, nous nous intéressons à étudier les modèles graphiques et logiques pour la gestion des connaissances incertaines et incohérentes.

La première contribution de la thèse concerne le modèle logique pour la gestion des informations incertaines. Nous nous intéressons à des logiques de description de faible complexité. Nous nous focalisons sur les logiques de type DL-lite avec un accent particulier sur les logiques DL-lite_{core}, DL-lite_R et DL-lite_F.

Nous avons d'abord étendu ces logiques, pour obtenir la logique DL-lite possibiliste quantitative, notée Pb- π -DL-Lite, dans laquelle l'incertitude concerne uniquement la partie assertionnelle (ABox) d'une base de connaissance DL-Lite et la partie décrivant les concepts et leurs relations (TBox) est supposée certaine.

Nous avons défini la syntaxe et la sémantique de Pb- π -DL-Lite. Nous avons étudié le problème du degré d'inconsistance d'une base de connaissances en Pb- π -DL-Lite et son rôle crucial dans les tâches d'inférence. Nous nous sommes concentrés en particulier sur le problème de vérification d'instances (instance checking).

Ensuite, nous avons considéré le problème de répondre à des requêtes conjonctives. Nous avons montré que cette tâche repose sur le problème de calcul du degré d'inconsistance, ce qui explique son rôle crucial pour les tâches d'inférence. Ce problème est NP-Difficile et une codification du problème en un problème 2-Horn-SAT a été développée. Cette transformation a été exploitée pour mettre au point un algorithme exact utilisant un prouver Weighted-Max-SAT pour calculer le degré d'inconsistance.

Du fait que le calcul du degré d'inconsistance est un problème NP-Difficile, nous avons proposé un encodage de notre problème en termes d'un problème de couverture par ensemble pondéré afin d'utiliser un algorithme approximatif opérant en temps polynomiale basé sur l'analyse des conflits pour calculer efficacement une valeur approximative du degré d'inconsistance et aussi pour répondre aux requêtes de vérification d'instance. Une étude expérimentale a été menée validant ainsi l'algorithme approximatif.

Dans un deuxième temps, nous avons généralisé cette approche pour que l'incertitude touche aussi bien la ABox que la Tbox. La généralisation comportera l'étude de la complexité, le calcul du degré d'inconsistance d'une base de connaissances et la réponse aux

requêtes conjonctives dans ce nouveau cadre. Pour le calcul du degré d'inconsistance, nous avons proposé un algorithme exact basé sur un codage du problème en termes d'un programme linéaire en nombres binaires. Nous avons également validé expérimentalement l'efficacité du choix de cette approche en la comparant avec les versions généralisées de l'algorithme exact à base de W-Max-SAT et de l'algorithme approché à base de couverture d'ensembles pondérés.

Une autre contribution de la thèse est située dans le cadre de la prise de décision sous incertitude en utilisant des modèles graphiques possibilistes. Nous présentons deux modèles pour le calcul des décisions optimales optimistes. Le premier modèle est basé sur la fusion de deux réseaux : le premier représente les connaissances de l'agent et le second représente les préférences de l'agent. Le deuxième modèle est basé sur les diagrammes d'influence possibilistes.

Nous avons implémenté dans cette thèse, une version améliorée du procédé de décomposition (utilisé dans le premier modèle) afin de réduire la complexité des réseaux résultants et ainsi permettre une amélioration du processus de calcul des décisions optimales optimistes et nous avons proposé une approche approximative pour la prise de la décision.

Organisation de la thèse.

Cette thèse comporte principalement trois parties.

La première partie est consacrée à l'étude de l'état de l'art lié à la problématique du calcul de degré de l'inconsistance sous incertitude. Cette partie est composée de deux chapitres abordant les principales théories de l'incertain et de l'imprécision ainsi que les modes logiques et graphiques de représentation des connaissances dans l'incertain. Un aperçu sur les logiques de description et les DL-Lite est également présenté par la suite.

La deuxième partie présente la plupart des contributions de cette thèse. Elle est composée de quatre chapitres. Le troisième chapitre expose une nouvelle extension de DL-Lite dans le cadre de la théorie des possibilités basée sur le produit. Le quatrième chapitre étudie ensuite la complexité de cette extension en présentant l'algorithme exact et l'algorithme approximatif pour le calcul du degré d'inconsistance.

Le cinquième chapitre comporte une expérimentation permettant la validation des algorithmes définis dans le chapitre précédent. Enfin le sixième chapitre présente une généralisation de notre approche pour une ontologie incertaine. Cela rend notre approche utile pour les applications où les relations terminologiques générales codées dans la TBox ne sont pas complètement fiables et peuvent avoir un certain niveau d'incertitude.

La troisième partie, comportant le septième chapitre, est dédiée globalement au calcul de la décision optimale optimiste en se basant sur des modèles graphiques. Nous utilisons en particulier, les réseaux de possibilités qualitatifs et les diagrammes d'influence qui sont des outils efficaces pour une représentation compacte des problèmes de décision sous incertitude.

Nous présentons l'algorithme du calcul de la décision séquentielle optimale optimisée en se basant sur la décomposition des diagrammes d'influence en deux réseaux possibilistes. Ensuite, nous présentons l'approche approximative pour le calcul de la décision dans les réseaux possibilistes sans passer par la construction de l'arbre de jonction. Au lieu d'une telle construction, notre algorithme calcule le degré de normalisation dans le graphe moral résultant de la fusion du réseau possibiliste codifiant les connaissances de l'agent et celui codifiant ses préférences.

Publications réalisées dans cette thèse.

Journaux internationaux

- Khaoula Boutouhami, Salem Benferhat, Faiza Khellaf, Farid Nouioua. Uncertain lightweight ontologies in a product-based possibility theory framework. International journal of approximate reasoning (IJAR) Vol. 88, pages. 237-258 (2017).
Lien : <https://doi.org/10.1016/j.ijar.2017.05.012>.
- Khaoula Boutouhami, Faiza Khellaf : optimal optimistic decision using an approximate graphical model. In International Journal of Artificial Intelligence and Applications (IJAIA) Vol. 6, No. 2, pages. (2015).
Lien : airconline.com/ijaia/V6N2/6215ijaia01.pdf

Conférences internationales

- Benferhat Salem, Khaoula Boutouhami, Faiza Khellaf, Fardi Nouioua. An integer linear programming approach for computing inconsistency degree in product-based possibilistic DL-Lite. In 11th International Conference on Scalable Uncertainty Management (SUM 2017), pp. 319-333, Granada, Spain, October 4-6, 2017.
- Benferhat Salem, Khaoula Boutouhami, Faiza Khellaf, Fardi Nouioua. Algorithms for quantitative-based possibilistic lightweight ontologies. In 29th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (IEA/AIE'16), pp. 364-372. Springer, Lecture Notes in Artificial Intelligence, 2-4 august 2016. Japan.
Lien : https://link.springer.com/chapter/10.1007/978-3-319-42007-3_31
- Benferhat Salem, Khaoula Boutouhami, Faiza Khellaf, Fardi Nouioua. Representing lightweight ontologies in a product-based possibility theory framework. In 8th International Conference on Soft Methods in Probability and Statistics (SMPS'16), Springer-Verlag, Advances in Intelligent Systems and Computing, pp. 45-52. Rome, San Pietro in Vincoli September 12-14, 2016.
Lien : https://link.springer.com/chapter/10.1007/978-3-319-42972-4_6.
- Khaoula Boutouhami, Faiza Khellaf : An approximate possibilistic graphical model for computing optimistic qualitative decision. In International Conference on Artificial Intelligence and Applications (AIFU2015), Dubai. Vol. 5, 2015, No. 33, pp. 183-1960.
Lien : <http://airccj.org/CSCP/vol5/csit53316.pdf>

Chapitre de livre

- Benferhat Salem, Khaoula Boutouhami, Faiza Khellaf, Ismahane Zeddigha. Qualitative Possibilistic Decisions : Decomposition and Sequential Decisions Making. In book : Agents and Artificial Intelligence, pp.171-188.

Première partie

Etat de l'art

Introduction de la première partie

Les travaux autour de la représentation de l'incertitude et de l'imprécision ne cessent de prendre de l'ampleur. Nous allons faire dans cette partie, un petit survol sur certains de ces modes de représentation.

Le premier chapitre est dédié à représenter les concepts de base des principales théories de l'incertain : la théorie des probabilités, la théorie des fonctions de croyance, la théorie des ensembles flous et la théorie des possibilités. Il existe deux modes de représentation de ces connaissances incertaines : les modèles logiques et les modèles graphiques. Dans notre thèse nous nous focalisons sur la logique possibiliste et les modèles graphiques possibilistes à savoir : les réseaux possibilistes, les arbres de décisions et les diagrammes d'influence.

Le deuxième chapitre est consacré aux logiques de description qui fournissent des bases théoriques solides aux ontologies grâce à leur sémantique claire et à leurs propriétés formelles. La famille DL-Lite bénéficie de bonnes propriétés de calcul tout en offrant des capacités intéressantes dans la représentation des connaissances terminologiques.

Chapitre 1

Gestion de l'incertitude

1.1 Introduction

La représentation de l'incertitude et de l'imprécision a reçu un grand intérêt en intelligence artificielle afin d'élargir les systèmes existants de représentation des connaissances pour faire face à la nature imparfaite de l'information [83, 82]. L'imperfection peut prendre différentes formes, parmi lesquelles nous citons : (i) l'incertitude et (ii) l'imprécision. L'incertitude est relative à la vérité d'une information et caractérise son degré de conformité à la réalité. Elle fait référence à la nature de l'objet et de ce fait elle concerne sa qualité, son essence ou son occurrence. Tandis que l'imprécision concerne le contenu de l'information : le manque d'exactitude en quantité, le manque de définition d'une proposition qui est ouverte à diverses interprétations ou qui a des frontières vagues et mal définies. Cette notion d'imprécision est souvent abusivement confondue avec celle de l'incertitude car les deux types d'imperfection sont souvent présents simultanément et l'un peut induire l'autre. L'incomplétude des informations issues de chaque source est la raison principale qui motive la fusion des deux notions d'incertitude et d'imprécision. L'information fournie par chaque source est généralement partielle, elle ne fournit qu'une vision du monde ou du phénomène observé, en ne mettant en évidence que certaines caractéristiques.

Au vu des nombreuses formes d'imperfection, plusieurs théories de représentation des données imparfaites ont été élaborées. La plus ancienne et la plus étudiée est la théorie des probabilités. Celle-ci est pertinente pour la représentation de tout phénomène aléatoire que nous pouvons l'approcher de manière quantitative. Elle portera plus sur la représentation de l'incertitude. Depuis la fin de la seconde guerre mondiale, l'approche subjective est devenue prédominante dans le monde. En 1965, Zadeh introduit la théorie des ensembles flous qui est une alternative importante à la théorie des probabilités pour l'imprécis. Dempster, en 1967, présente la théorie de l'évidence qui sera ensuite reprise et formalisée par Shafer en 1976. Cette théorie est aussi appelée théorie des fonctions de croyance ou encore théorie de Dempster-Shafer. Elle permet de modéliser l'ignorance. Puis plus tard, la théorie des possibilités, a été développée sous l'impulsion de Didier Dubois et Henri Prade [125], pour permettre la transition entre l'imprécision et l'incertitude.

Après le développement des théories de l'incertain au niveau sémantique, deux types de modèles pour représenter de façon compacte les connaissances ont été développés : les modèles logiques et les modèles graphiques.

Les modèles logiques de l'incertain permettent d'utiliser un langage formel pour la description des connaissances et le raisonnement automatique. Les modèles graphiques sont utilisés

dans de nombreux problèmes de décision. Ils offrent une grande expressivité et efficacité de calcul pour la représentation des incertitudes des connaissances et des préférences.

La structure du graphe exprime les spécificités du problème traité et est utilisée pour propager des informations et optimiser des décisions. Plusieurs modèles graphiques peuvent être utilisés pour le problème de la prise de décision séquentielle, tels que : les arbres de décision, les diagrammes d'influence et les systèmes basés sur les évaluations.

Dans ce chapitre, nous présentons quelques approches proposées pour représenter les connaissances incertaines. La section 2 présente les concepts de base des principales théories de l'incertain : la théorie des probabilités, la théorie des fonctions de croyance, la théorie des ensembles flous et la théorie des possibilités. La section 3 présente deux modes de représentation des connaissances incertaines, à savoir le mode logique et le mode graphique.

1.2 Théories de l'incertain

L'incertitude est présente dans de nombreux problèmes de la vie réelle. Le traitement de l'incertitude joue un rôle crucial dans l'intelligence artificielle [82]. De nos jours, de nombreuses applications et domaines manipulent des quantités massives de connaissances incontrôlables et imparfaites, en raison du manque de connaissances précises. Ces imperfections résultent de l'utilisation de sources d'information peu fiables et/ou de l'agrégation des opinions de différents experts [83].

Il est important de faire une distinction entre les deux formes majeures de l'imperfection : (i) incertitude et (ii) imprécision. Selon la nature de l'imperfection, plusieurs théories ont été proposées afin de modéliser, de représenter et de raisonner sur les données imparfaites. La théorie des probabilités a été la première à être utilisée pour traiter l'incertitude [90]. Cette théorie n'est pas un outil approprié pour prendre en compte le cas d'ignorance totale [79]. En outre, la théorie des probabilités ne peut pas gérer les situations dans lesquelles les experts sont incapables de fournir toutes les valeurs numériques d'un paramètre imparfait donné. Par conséquent, des théories alternatives ont été développées pour la quantification de l'incertitude et la modélisation des connaissances incomplètes. Les théories les plus visibles sont la théorie des ensembles flous [124], la théorie des fonctions de croyance et la théorie des possibilités [125]. La théorie de l'évidence et la théorie des ensembles flous représentent deux alternatives à la théorie des probabilités. La théorie de l'évidence a été développée pour mieux prendre en compte la non-spécificité et la discordance des informations, par rapport à la théorie des probabilités. La théorie des ensembles flous a été développée pour modéliser l'information vague et les concepts qui font référence à des intervalles aux limites mal définies.

Dans cette section, nous présentons ces différentes théories qui sont toutes mobilisables pour représenter des données imparfaites. Nous entamons d'abord la plus ancienne : la théorie des probabilités, puis la théorie de l'évidence et enfin les cadres généraux de la théorie des ensembles flous et celle des possibilités.

1.2.1 Notations

Nous donnons tout d'abord quelques notations qui seront utilisées dans la suite de cette thèse.

- $V = \{X_1, X_2, \dots, X_n\}$ désigne un ensemble de n variables décrivant un domaine étudié.

- D_i Désigne un domaine supposé fini associé à la variable X_i .
- Par_A Désigne les parents du nœud A dans un graphe.
- x_{ik} est une instance, un état (une valeur possible) de X_i .
- A_i Désigne un sous-ensemble d'instances d'une variable X_i ($A_i \subseteq D_i$).
- $\Omega = D_1 \times \dots \times D_n$ désigne l'univers du discours, qui est le produit cartésien de tous les domaines des variables de V .
- Chaque élément $I \in \Omega$ s'appelle interprétation ou événement et est désigné par un tuple (x_{1k}, \dots, x_{nk}) .
- $2^{card(D_i)}$ est l'ensemble de tous les sous-ensembles de D_i y compris aussi \emptyset et D_i .
- Par_A Désigne l'ensemble des parents du noeud A.
- G_N Désigne la composante graphique d'un graphe acyclique orienté.

1.2.2 La théorie des probabilités

La théorie des probabilités traite l'information imparfaite. Elle a été utilisée pour représenter toutes sortes d'imperfection puisqu'il s'agissait du seul cadre existant de traitement de l'incertitude. Le manque d'information est remplacé par une composante aléatoire.

Dans la théorie des probabilités, nous cherchons à déduire les probabilités d'événements plus compliqués ou des résultats d'expériences plus complexes à partir d'un ensemble de résultats possibles d'une expérience et leurs probabilités respectives.

1.2.2.1 Concepts de base

La notion de probabilité est liée à celle des expériences aléatoires. Une expérience est aléatoire si nous ne pouvons pas prédire avec certitude son résultat. Le résultat d'une expérience aléatoire est un élément ω de l'ensemble Ω de tous les résultats possibles, appelé univers des possibles ou référentiel.

Dans cette sous section nous présentons quelques concepts de base et propriétés de la théorie des probabilités.

Définition 1.1 Espace de probabilités. *Un espace de probabilités formel est défini par le triplet (Ω, F, P) où : Ω est l'espace de réalisation, F est l'ensemble des événements en S et P est une mesure de probabilités définie sur les éléments de F .*

La mesure de probabilités P définie comme $P : 2^\Omega \rightarrow [0, 1]$ a certaines propriétés considérées essentielles dans le formalisme bayésien :

- $p(\emptyset) = 0, p(\Omega) = 1$
- Pour chaque événement $A, 0 \leq P(A) \leq 1$, si la variable A représente une connaissance certaine alors $P(A) = 1$.
- Pour tout événement A et B , si $A \subseteq B$ alors $P(A) \leq P(B)$ (*principe de la monotonie*).
- Pour tout événement A et $B, P(A \cup B) = P(A) + P(B) - P(A \cap B)$ où $P(A \cap B)$ est appelée probabilité jointe et elle est notée par $P(A, B)$. Plus généralement, nous avons :

$$P(\cup_{h=1}^n A_h) = \sum_{h=1}^n P(A_h) \text{ si } A_1, \dots, A_n \text{ sont mutuellement exclusives.} \quad (1.1)$$

- Pour tout événement $A, P(A) = 1 - P(\bar{A})$ où \bar{A} est le complément de A .

La notion du conditionnement probabiliste permet de modéliser les liens de dépendance entre deux événements. Plus précisément, à chaque fois qu'une nouvelle information arrive, la probabilité des événements disponibles peut être mise à jour. Ceci conduit au concept de probabilité conditionnelle qui permet de s'intéresser à la probabilité d'un événement A étant donné la réalisation d'un autre événement B . D'une manière formelle :

Définition 1.2 *Conditionnement probabiliste.* Soient A et B deux événements avec $P(B) > 0$, la probabilité conditionnelle de A sachant B est définie par :

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad (1.2)$$

A partir de cette définition, la probabilité de l'intersection de deux événements, appelée *probabilité jointe*, est déduite comme suit :

$$P(A \cap B) = P(A|B) * P(B) = P(B|A) * P(A) \quad (1.3)$$

Définition 1.3 (*Règle de chaînage*). Soient A_1, A_2, \dots, A_n n événements, la probabilité jointe associée aux événements (A_1, A_2, \dots, A_n) est définie comme le produit des n probabilités conditionnelles comme suit :

$$P(A_1, A_2, \dots, A_n) = P(A_1|A_2, \dots, A_{n-1}, A_n) * \dots * P(A_{n-1}|A_n) * P(A_n) \quad (1.4)$$

1.2.2.2 Indépendance et théorème de Bayes

Deux événements A et B sont indépendants si l'événement B n'affecte pas l'événement A , autrement dit, leurs probabilités ne changent pas quand nous les conditionnons l'un par rapport à l'autre. Formellement :

$$P(A|B) = P(A) \quad (1.5)$$

L'Equation 1.5 indique que si A et B sont indépendants, la distribution de probabilité jointe associée aux instances a et b , est égale au produit des distributions marginales $P(A)$ et $P(B)$. Plus généralement :

Définition 1.4 (*Indépendance*). Un ensemble d'événements A_1, A_2, \dots, A_n est dit indépendant si pour chaque sous ensemble A_i, A_j, \dots, A_m de ces événements, nous avons :

$$P(A_i, A_j, \dots, A_m) = P(A_i) * P(A_j) * \dots * P(A_m) \quad (1.6)$$

Définition 1.5 (*Indépendance conditionnelle*). Deux événements A et B , avec $P(A) > 0$ et $P(B) > 0$ sont dits indépendants dans le contexte C si :

$$P(A \cap B|C) = P(A|C) * P(B|C) \quad (1.7)$$

Le théorème de Bayes est une formulation mathématique du conditionnement permettant de mettre à jour une probabilité initiale $P(A)$ donnant ainsi lieu à une probabilité révisée $P(A|B)$.

Définition 1.6 (*Théorème de Bayes*). Le théorème de Bayes permet de mettre à jour une probabilité initiale $P(A)$ donnant ainsi lieu à une probabilité révisée $P(A|B)$, elle est

définie comme suit : avec $P(A) \neq 0$ et $P(B) \neq 0$.

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)} \quad (1.8)$$

La loi de Bayes représente la loi fondamentale du calcul des probabilités, où $P(A|B)$ est appelée probabilité postérieure ou conditionnelle. Elle exprime la croyance révisée concernant A après avoir observé B , tandis que $P(A)$ désigne la probabilité à priori qui exprime la croyance initiale sur A , et enfin $P(B|A)$ représente la fonction de vraisemblance de A .

Un problème souvent cité dans la théorie des probabilités est celui la représentation de l'ignorance. Ceci est dû au fait que l'état des connaissances, où il existe également un manque de certitude sur tous les événements susceptibles de se produire, ne peut être exprimé par une seule mesure de probabilité.

L'inférence bayésienne est généralement utilisée avec un certain nombre de restrictions structurelles sévères. Le problème général de trouver la probabilité postérieure d'une proposition dans un réseau bayésien est dans NP [37]. Certains auteurs ont proposé des procédures d'inférence qui sont moins précises et ont une complexité plus faible pour le cas moyen [97]. Haber et Brown présentent un algorithme itératif pour le cas général des réseaux d'inférences cycliques, tandis que Pearl discute des procédures simplifiées avec des modèles d'interaction spéciaux [62]. Certaines approches utilisent des heuristiques d'optimisation qui fournissent des solutions approximatives avec moins d'effort de calcul [1].

1.2.3 La théorie des fonctions de croyance

La théorie des fonctions de croyance (Dempster-Shafer) est une théorie mathématique de la preuve introduite par Dempster (1967) [39] et développée par Shafer (1976) [103] pour représenter l'état des connaissances en fonction de toutes les preuves disponibles. Cette théorie a identifié trois points de vue distincts qui sont des extensions ou des variantes du modèle proposé par Shafer. Des détails sur ces différentes interprétations sont proposés par Smets [108]. Le premier modèle repose sur la théorie des probabilités imprécises appelée aussi théorie des probabilités inférieures [120, 126]. Cette théorie suppose l'existence d'une mesure de probabilité P précise, mais non parfaitement connue. Le modèle peut alors être défini soit directement par P , soit par ses enveloppes inférieure et supérieure. Sous certaines conditions, la probabilité inférieure peut alors être considérée comme une fonction de croyance. Le second modèle proposé par Dempster [39, 40] est un cas particulier des probabilités imprécises. Ce modèle peut être rapproché des hints présentés par Kohlas et Monney [77]. L'interprétation logique de la théorie des fonctions de croyance proposée par Cholvy [36] peut être vue comme un cas particulier du modèle de Kohlas et Monney. Enfin le dernier modèle, présenté par Smets [107, 106], est une extension des fonctions de croyance, proposée au travers du Modèle des Croyances Transférables (MCT). L'approche proposée par Smets se distingue des approches précédentes de par son caractère fondamentalement non probabiliste.

1.2.3.1 Fonction de masse

Une fonction de masse m appelée aussi jeu de masse ou BBA^1 associée à une source d'information est définie par : $m : 2^\Omega \rightarrow [0, 1]$. La fonction de masse vérifie la propriété

1. Acronyme du mot anglais Basic Belief Assignment

suivante :

$$\sum_{A \subseteq \Omega} m(A) = 1 \quad (1.9)$$

Où 2^Ω est l'ensemble de puissance qui comprend tous les sous-ensembles possibles, y compris l'ensemble vide \emptyset . A est un ensemble donné de l'ensemble de puissance. La masse $m(A)$ appelée masse de croyance fondamentale exprime la proportion de la preuve disponible qui soutient la conviction qu'un élément particulier appartient uniquement à l'ensemble A . La masse $m(\emptyset)$ est appelée masse conflictuelle ou conflit.

La principale différence entre la théorie des fonctions de croyance (Dempster-Shafer) et la théorie des probabilités est la façon dont le raisonnement de la crédibilité gère les preuves qui soutiennent les propositions focales complexes.

Les propriétés de l'attribution de probabilité de base sont :

- La masse de l'ensemble vide \emptyset est égale à 0, ($m(\emptyset) = 0$).
- Tout sous-ensemble de Ω vérifiant $m(A) \neq 0$ est dit élément focal de m .
- Le couple (Ω, m) est appelé corps de la croyance.
- Si $m(A) = 1$ et $\forall B \neq A \in 2^\Omega, m(B) = 0$, alors A est certain dans le sens où un de ces éléments a la valeur cherchée. Dans ce cas nous ne savons pas quel est l'élément concerné sauf si A est un singleton.
- Si $A = \Omega$, alors l'information $m(A) = 1$ correspond à une situation d'ignorance totale.

A partir d'une fonction de masse, plusieurs autres fonctions peuvent être déduites.

Exemple 1.1 *Un expert tente de définir des stratégies financières pour une entreprise économique. IL atteste que la stratégie S1 est optimale à 20% et les stratégies S2 est efficace à 30%. La modélisation dans le cadre des fonctions de croyance est donnée comme suit :*

Le sous-ensembles possibles est $\Omega = \{S1, S2\}$.

Les masses associées à chaque sous-ensemble possible sont :

- $m_1(S1) = 0.2$
- $m_1(S2) = 0.3$
- $m_1(\Omega) = 0.5$

1.2.3.2 Fonction de croyance

La fonction de croyance dénotée Bel évalue l'intensité de l'implication d'une formule à partir des croyances disponibles. Elle est définie comme suit :

Définition 1.7 (Fonction de croyance) *Soit $\Omega = \{w_1, w_2, \dots, w_n\}$ un ensemble fini, généralement appelé cadre de discernement. Une fonction de croyance Bel est une fonction de 2^Ω vers $[0, 1]$ définie par :*

$$Bel(A) = \sum_{B \subseteq A} m(B). \quad (1.10)$$

La fonction de crédibilité satisfait les relations suivantes :

- $Bel(\emptyset) = 0$.
- $Bel(\Omega) = 1$.
- $Bel(A) + Bel(\bar{A}) \leq 1$
- $Bel(A \cup B) \geq Bel(A) + Bel(B) - Bel(A \cap B)$.
- Règle de monotonie : Si $A \subseteq B$ alors $Bel(A) \leq Bel(B)$.

Exemple 1.2 *Nous reprenons l'Exemple 1.1. Les degrés de crédibilité sont :*

- $Bel(\emptyset) = 0$
- $Bel(S1) = 0.2$
- $Bel(S4, S3) = 0.3$
- $Bel(\Omega) = 1$ puisque la somme des masses de tous les sous-ensembles du cadre de discernement est toujours égale à 1.

1.2.3.3 Fonction de plausibilité

La fonction de plausibilité quantifie la quantité maximale de croyance qui pourrait être donnée à un sous-ensemble A de l'univers du discours. En d'autres termes, il contient ces parties de croyance qui ne contredisent pas A (qui n'ont pas une intersection vide avec A). Elle est définie par : $Pl : 2^\Omega \rightarrow [0, 1]$

$$Pl(A) = \sum_{B \cap A \neq \emptyset} m(B) = 1 - Bel(\bar{A}). \quad (1.11)$$

Le degré de plausibilité de A représente tout le degré de crédibilité qui n'est pas attribué à \bar{A} . En d'autres termes, A n'est quelque peu crédible que dans la mesure où l'événement opposé n'est pas très plausible.

La fonction de plausibilité est aussi la duale de la fonction de crédibilité. Ainsi dans le cadre du monde fermé, quelques propriétés intéressantes peuvent être exprimées :

- $Pl(\emptyset) = 0$.
- $Pl(\Omega) = 1$.
- $Pl(A) + Pl(\bar{A}) \geq 1$
- Règle de monotonie : Si $A \subseteq B$ alors $Pl(A) \leq Pl(B)$.
- $Pl(A \cap B) \geq Pl(A) + Pl(B) - Pl(A \cup B)$.

L'intervalle $[Bel(A), Pl(A)]$ est appelé intervalle de confiance et sa longueur est une mesure de l'ignorance que nous avons sur un événement A et son complémentaire.

Les deux cas extrêmes de connaissances dans la théorie de la fonction de croyance sont exprimés par :

- Ignorance totale : $m(\Omega) = 1$ et $\forall A \neq \Omega, m(A) = 0$.
- Certitude totale : $m(\{\omega\}) = 1$ pour un élément particulier de Ω et $\forall A \neq \{\omega\}, m(A) = 0$.

Le cas de l'imprécision est donné par : $m(A) = 1$ pour certains $A \subset \Omega$ et $m(B) = 0, \forall B \subseteq \Omega, B \neq A$. Dans ce cas, nous avons un élément focal unique.

Exemple 1.3 Nous reprenons l'Exemple 1.1. Les degrés de plausibilité sont :

- $Pl(\emptyset) = 0$.
- $Pl(S1) = m(S1) + m(\Omega) = 0.7$,
- $Pl(S2) = m(S2) + m(\Omega) = 0.8$,
- $Pl(\Omega) = Pl(\text{fff}) = 1$.

1.2.3.4 La règle de combinaison de Dempster

Dans le cas de données imparfaites, la fusion de données est une solution intéressante pour l'obtention d'informations plus pertinentes. La théorie des fonctions de croyance offre des outils appropriés de fusion. En effet, dans un cas multi-sources, il est possible de les combiner afin d'en extraire une information plus fiable.

Soient deux sources S_1 et S_2 , supposées fiables, dont les informations sont représentées respectivement par les fonctions de masse cognitivement indépendantes m_1 et m_2 . m_\oplus désigne

le résultat de la combinaison conjonctive qui représente l'agrégation des informations issues des deux sources considérées :

$$m_{\oplus} = m_1 \oplus m_2 \quad (1.12)$$

Formellement, la combinaison conjonctive de deux fonctions de masse s'écrit de la manière suivante :

$$\forall A \neq \Omega, m_{\oplus}(A) = \sum_{C \cap B = A} m_1(C) * m_2(B) \quad (1.13)$$

Cette règle vérifie un certain nombre de propriétés comme l'associativité, la commutativité et elle admet un élément neutre $m(\Omega) = 1$. Une des particularités de cette règle est qu'elle génère une fonction de masse non normalisée ($m(\emptyset) > 0$). Cette condition, n'étant pas possible dans un monde fermé, il est nécessaire de procéder à une étape de normalisation. La normalisation de la combinaison conjonctive est connue sous le nom *règle de combinaison orthogonale* ou sous la désignation de règle de combinaison de Dempster. Elle s'écrit de la manière suivante :

$$m_{\oplus}(A) = \frac{1}{1 - K} \sum_{C \cap B = A} m_1(C) * m_2(B) \quad (1.14)$$

où K est défini comme :

$$K = \sum_{A \cap B = \emptyset} m_1(A) * m_2(B) \quad (1.15)$$

K représente la masse conflictuelle ($m(\emptyset)$) prenant ses valeurs entre $[0, 1]$. Généralement, cette masse traduit le degré de contradiction entre les sources fusionnées. Pour $K = 1$, S_1 et S_2 sont considérées comme complètement conflictuelles et les sources ne peuvent pas être fusionnées. Au contraire, si $K = 0$ les sources sont parfaitement en accord.

La théorie des fonctions de croyance peut être considérée comme une généralisation de la théorie des probabilités bayésiennes. La théorie des fonctions de croyance permet de modéliser les problèmes de choix incertain. Elle ne considérerait pas la connaissance incomplète. Cette hypothèse du monde fermé a été trop restrictive pour de nombreuses applications pratiques. Smets décrit une théorie de la crédibilité qui traite de l'incomplétude. Chaque fois que de nouvelles preuves sont disponibles, toutes les assignations de probabilité de base sont modifiées pour tenir compte de cette nouvelle preuve. Cette révision est effectuée par la règle de conditionnement de Dempster. Le problème général de la déduction des fonctions de croyance est NP-difficile.

1.2.4 La théorie des ensembles flous

La théorie des sous-ensembles flous est une généralisation de la théorie des ensembles classiques. La théorie des sous-ensembles flous a été introduite par Lofti Zadeh en 1965 [124]. L'idée de Zadeh était de pouvoir manipuler et modéliser d'une manière mathématique des informations exprimées en langage naturel [22].

1.2.4.1 Ensembles flous

La théorie des sous-ensembles flous permet de s'affranchir du caractère binaire ou booléen de l'appartenance à un ensemble, en introduisant une notion d'appartenance graduelle [124]. Ainsi, nous pouvons définir un sous-ensemble A de S par sa fonction d'appartenance μ_A , qui associe à tout élément x de S une valeur réelle $\mu_A(x)$ dans l'intervalle $[0, 1]$. Un sous-ensemble classique est un sous-ensemble flou dont la fonction d'appartenance a une forme

particulière, puisqu'elle prend ses valeurs uniquement dans le doublet $\{0, 1\}$.

$$\mu_A(x) = \begin{cases} 1 & \text{si } x \in A \\ 0 & \text{sinon} \end{cases} \quad (1.16)$$

Si la fonction d'appartenance ne donne que les valeurs de l'ensemble $\{0, 1\}$, alors l'ensemble qu'elle génère n'est pas flou.

Dans le cas où l'ensemble d'univers U est discret, l'ensemble flou A peut être défini par une liste de paires ordonnées : degré d'appartenance / élément de l'ensemble. Formellement :

$$A = \sum_{i=1}^n \mu_A(x_i) / x_i \quad (1.17)$$

où $\mu_A(x_i)$ représente le degré d'appartenance de x_i à l'ensemble flou A .

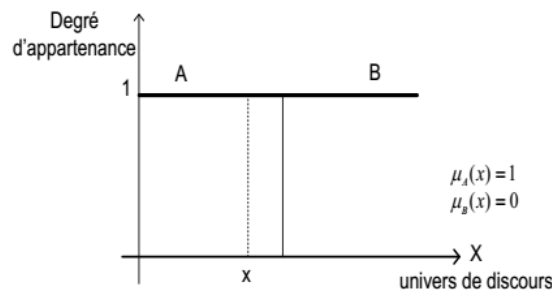


FIGURE 1.1 – Fonction d'appartenance en théorie des ensembles flous avec un ensemble d'univers discret

Dans le cas où l'ensemble d'univers U est continu, l'ensemble flou A peut être représenté par :

$$A = \int \mu_A(x) / x \quad (1.18)$$

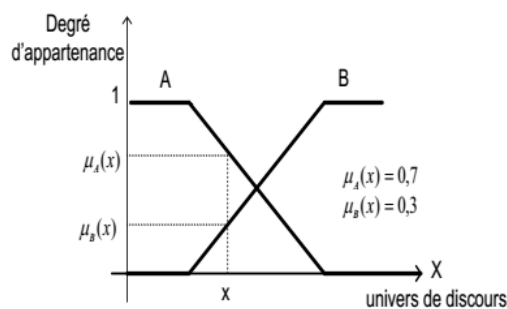


FIGURE 1.2 – Fonction d'appartenance en théorie des ensembles flous avec un ensemble d'univers continu

Le type d'ensemble flou obtenu dépend du type de la fonction d'appartenance. Les fonctions d'appartenance pourraient être classées en deux groupes : Ligne droite ou forme gaussienne. Les fonctions couramment utilisées sont :

- **Triangulaire** : Elle est définie par une valeur inférieure a , une valeur supérieure b et une valeur modale m de sorte que $a < m < b$. Formellement :

$$\mu_A(x) = \begin{cases} 0 & \text{si } x \leq a \\ (x - a)/(m - a) & \text{si } x \in]a, m] \\ (b - x)/(b - m) & \text{si } x \in]m, b] \\ 0 & \text{si } x > b \end{cases} \quad (1.19)$$

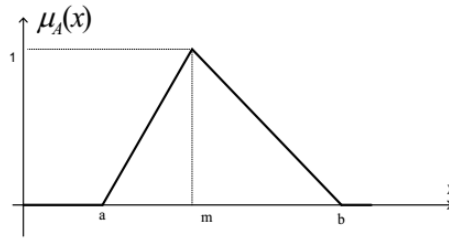


FIGURE 1.3 – Fonction d'appartenance triangulaire

- **Singleton** : La fonction d'appartenance prend la valeur 0 dans tout l'univers de discours sauf au point m où elle prend la valeur 1. Formellement :

$$\mu_A(x) = \begin{cases} 0 & \text{si } x \neq m \\ 1 & \text{si } x = m \end{cases} \quad (1.20)$$

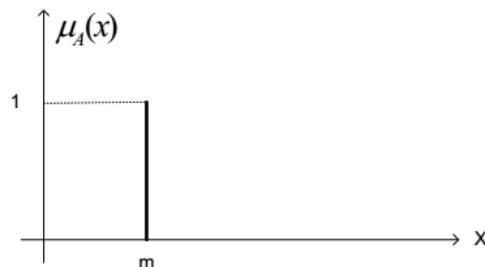


FIGURE 1.4 – Fonction d'appartenance singleton

- **Gaussienne** : La fonction d'appartenance est définie par sa moyenne m et son écart-type σ . Formellement :

$$\mu_A(x) = e^{-\frac{1}{2} \left\{ \frac{x-m}{\sigma} \right\}^2} \quad (1.21)$$

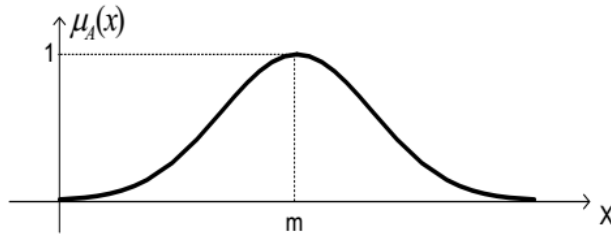


FIGURE 1.5 – Fonction d'appartenance gaussienne

Un sous-ensemble flou comporte certains éléments caractéristiques essentielles qui sont :

- Le support d'un sous-ensemble flou A , notée $Supp(A)$, est l'ensemble de tous les éléments qui lui appartiennent au moins un petit peu :

$$Supp(A) = \{x \in S | \mu_A(x) > 0\}. \quad (1.22)$$

- Le noyau d'un sous-ensemble flou A , noté $Noy(A)$, est l'ensemble de tous les éléments qui lui appartiennent totalementment :

$$Noy(A) = \{x \in S | \mu_A(x) = 1\}. \quad (1.23)$$

- La hauteur d'un sous-ensemble flou A de S , notée $h(A)$, est la valeur maximale atteinte sur tout le support de A . Elle est définie à l'aide de l'opérateur $Supp$, telle que :

$$h(A) = Supp_{x \in S} \mu_A(x). \quad (1.24)$$

- L' α -coupe d'un sous-ensemble flou A de S , notée A_α , est le sous-ensemble classique composé des éléments qui appartiennent à A avec un degré au moins égal à α :

$$A_\alpha = \{x \in S | \mu_A(x) \geq \alpha\}. \quad (1.25)$$

Cette notion d' α -coupe permet de faire le lien entre sous-ensemble flou et sous ensemble classique.

La Figure 1.6 illustre la plupart de ces caractéristiques.

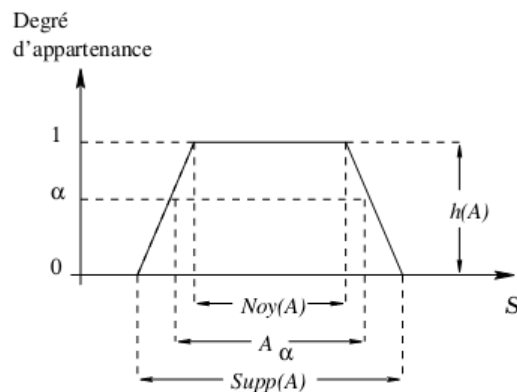


FIGURE 1.6 – Éléments caractéristiques d'un sous-ensemble flou

1.2.4.2 Opération sur les ensembles flous

Les sous-ensembles classiques peuvent être vus comme des cas particuliers des sous-ensembles flous. L'algèbre booléenne et ses connecteurs de base (NON, ET, OU) peut alors être considérée comme une restriction de l'algèbre floue.

Les opérateurs disponibles dans le cadre de la théorie des sous-ensembles flous sont nombreux et ont une propriété importante du point de vue de la fusion d'informations. Ils donnent un résultat qui a la même interprétation en termes d'imprécision que leurs entrées. Les opérations les plus courantes sont :

Définition 1.8 Inclusion. *Un sous-ensemble flou A est inclus dans un autre sous-ensemble flou B ($A \subseteq B$) si et seulement si tout élément x de S qui appartient à A appartient aussi à B avec un de degré au moins aussi grand :*

$$A \subseteq B \Leftrightarrow \forall x \in S \mu_A(x) \leq \mu_B(x) \quad (1.26)$$

Définition 1.9 Égalité. *Deux sous-ensembles flous A et B de S sont dits égaux s'ils ont des fonctions d'appartenance égales en tout point de S :*

$$A = B \Leftrightarrow \forall x \in S \mu_A(x) = \mu_B(x) \quad (1.27)$$

Définition 1.10 Union *L'union de deux sous-ensembles flous A et B de S est le sous-ensemble qui constitue des éléments de S affectés du plus grand des degrés avec lesquels ils appartiennent à A et B*

$$\forall x \in S \mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)) \quad (1.28)$$

La Figure 1.7 donne un exemple de graphe de la fonction d'appartenance de l'union des deux sous-ensembles flous A et B .

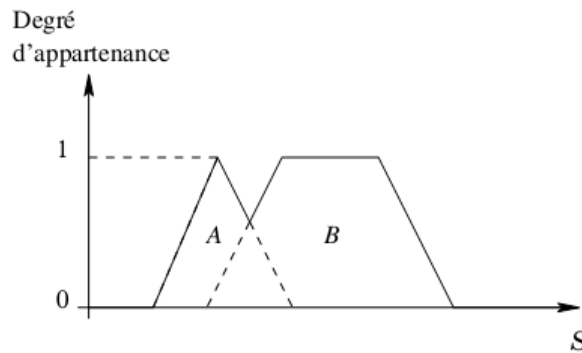


FIGURE 1.7 – Fonction d'appartenance de l'union

Définition 1.11 Intersection : *L'intersection de deux sous-ensembles flous A et B de S est le sous-ensemble qui est constitué des éléments de S affectés du plus petit des degrés avec lesquels ils appartiennent à A et à B :*

$$\forall x \in S \mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)) \quad (1.29)$$

La Figure 1.8 donne un exemple de graphe de la fonction d'appartenance de l'intersection des deux sous-ensembles flous A et B .

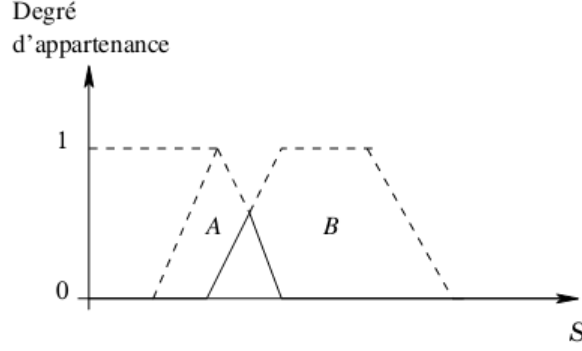


FIGURE 1.8 – Fonction d'appartenance de l'intersection

Définition 1.12 Complément : Le Complément d'un sous-ensemble flou A de S , notée \bar{A} , est défini par sa fonction d'appartenance :

$$\forall x \in S \quad \mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (1.30)$$

D'autres opérateurs sont définis à l'aide d'une norme triangulaire et d'une conorme triangulaire définies comme suit :

Définition 1.13 *t*-norme. Une norme triangulaire "*t*-norme" est une fonction $T : [0, 1] \times [0, 1] \rightarrow [0, 1]$ vérifiant, pour tout x et y dans $[0, 1]$, les propriétés suivantes :

- T est commutative : $T(x, y) = T(y, x)$,
- T est associative : $T(x, T(y, z)) = T(T(x, y), z)$,
- T est croissante : $T(x, y) \leq T(z, t)$ si $x \leq z$ et $y \leq t$,
- $T(x, 1) = x$.

Définition 1.14 *t*-conorme. Une conorme triangulaire "*t*-conorme" est une fonction $\perp : [0, 1] \times [0, 1] \rightarrow [0, 1]$ vérifiant, pour tout x et y dans $[0, 1]$, les propriétés suivantes :

- \perp est commutative : $\perp(x, y) = \perp(y, x)$,
- \perp est associative : $\perp(x, \perp(y, z)) = \perp(\perp(x, y), z)$,
- \perp est croissante : $\perp(x, y) \leq \perp(z, t)$ si $x \leq z$ et $y \leq t$,
- $\perp(x, 0) = x$.

1.2.4.3 Implication floue et règles floues

La notion d'implication entre deux propositions floues est généralisée par rapport à la logique classique. Du point de vue du raisonnement, une implication définit une règle floue :

Si X est x_1 Alors Y est b

La première partie de la règle, " X est x_1 ", s'appelle la prémisse et la seconde partie, " Y est b " la conclusion. La prémisse peut être composée de plusieurs propositions, reliées par des opérateurs de conjonction ou de disjonction. Les variables X et Y représentent les variables physiques, souvent caractéristiques d'un processus à commander. x_1 et b sont des valeurs linguistiques qui sont représentées par des ensembles flous.

D'un point de vue formel, l'implication exprime la relation R entre la prémisse et la conclusion de la règle et son degré de vérité est défini par la fonction d'appartenance μ_R telle que :

$$\forall \{x, y\} \in S_X \times S_Y, \mu_R(x, y) = I(\mu_{x_1}(x)\mu_b(y)) \quad (1.31)$$

où $I : [0, 1] \rightarrow [0, 1]$ est une fonction correspondant à l'implication floue.

Ce type particulier d'implication est très utilisé en commande floue. Il s'agit des implications conjonctives qui considèrent que la relation entre les univers de la prémisse et de la conclusion de la règle est de type conjonctif. Dans ce cas, la fonction I est représentée par un opérateur de type t-norme. Deux exemples d'implications conjonctives sont donnés dans le tableau 1.1.

Implication	Expression
Mamdani	$\min(\mu_A(x), \mu_B(y))$
Larsen	$\mu_A(x) \cdot \mu_B(y)$

TABLE 1.1 – Exemples d'implications conjonctives

1.2.4.4 Le contrôleur flou

Les applications de la commande floue sont nombreuses dans le contexte industriel. La commande floue a pour but de gérer un processus en fonction d'une consigne donnée. Elle permet de raisonner avec ses connaissances tout en les combinant avec les données numériques disponibles, et d'intégrer des comportements locaux.

L'élément central d'un système de commande floue est le contrôleur flou. A partir d'une base de connaissances représentée avec des sous-ensembles flous, il fournit une commande unique et précise en fonction d'une entrée précise donnée, issue d'un ou plusieurs capteurs. Le contrôleur flou utilise une expression proche du langage humain sous forme de règles. Le traitement réalisé par un contrôleur flou est divisé en trois tâches essentielles :

1. **La fuzzification** : revient à caractériser au moyen de sous-ensembles flous les valeurs des variables d'entrées représentant l'état du système et fournies par des capteurs
2. **L'inférence floue** : consiste à fournir une caractérisation pour chaque règle puis faire une synthèse pour parvenir à une décision.
3. **La defuzzification** : le but de la defuzzification est de trouver une sortie précise qui corresponde au mieux à la caractérisation floue (La transformation de la partie floue de l'inférence en une valeur numérique y en sortie).

La logique floue permet de décrire et de raisonner sur des concepts vagues, mais elle est moins appropriée pour faire face à des connaissances incertaines [78]. Certaines extensions de cette théorie ont été proposées pour surmonter cette lacune. Une approche pour gérer l'incertitude est la logique floue du type-2 [78]. Elle est basée sur le concept d'ensembles flous de type-2, introduit dans [125]. Dans les ensembles flous normaux, les degrés d'appartenance sont représentés par des nombres réels $[0, 1]$.

Cependant, les ensembles flous du type-2 peuvent être utilisés dans des situations où il y a incertitude quant aux degrés d'adhésion, comme lorsque la forme exacte de la fonction d'appartenance est inconnue. Cette approche peut être considérée comme une approximation de second ordre par rapport à la logique floue qui représente une approximation de premier ordre. Le raisonnement flou ne répond pas à nos exigences en matière de représentation de connaissances contradictoires et d'ignorance partielle.

1.2.5 La théorie des possibilités

La théorie des possibilités a été introduite par Lotfi Zadeh en 1978 [125] puis développée par Dubois et Prade [45]. La théorie des possibilités est la plus simple pour traiter et représenter des informations imparfaites qui sont incertaines et imprécises telles que celles exprimées par les humains. Avant de présenter les bases de la théorie des possibilités, intéressons nous tout d'abord au terme possibilité.

Dans la théorie des possibilités, le terme possibilité peut supporter diverses interprétations [76]. Les plus rencontrées sont : la faisabilité, la plausibilité et la vision logique de la "possibilité" est celle de la cohérence avec l'information disponible [49].

En théorie des possibilités, il existe deux variantes [49] :

- Théorie des possibilités qualitative : pour cette variante, une distribution de possibilités associe à chaque événement de l'univers du discours Ω un degré de possibilité "ordinal" où seul l'ordre des valeurs est important.
- Théorie des possibilités quantitative : pour cette variante, l'intervalle unitaire $[0,1]$ est numérique. Ainsi, les degrés de possibilités sont des valeurs numériques qui peuvent être manipulées par les opérateurs arithmétiques.

La théorie des possibilités qualitative et la théorie des possibilité quantitative partagent des notions communes telles que : la définition des mesures de nécessité et de possibilité, la définition de la condition de normalisation. Cependant, elles diffèrent dans la façon dont elles définissent le conditionnement et aussi dans la manière dont les degrés de possibilités sont définis par rapport aux interprétations.

Dans cette thèse, nous nous sommes intéressés essentiellement à la théorie des possibilités quantitative.

1.2.5.1 Distribution possibiliste

Soit $\nu = \{A_1, A_2, \dots, A_n\}$ un ensemble de variables binaires. Chaque élément $I \in \Omega$ s'appelle une interprétation qui représente un état du monde possible.

La notion de base de la théorie des possibilités est la distribution de possibilité [125], notée π , qui est une fonction de l'ensemble Ω vers une échelle totalement ordonnée qui a une valeur maximale et une valeur minimale. Cette échelle est l'intervalle unitaire $[0,1]$.

- $\pi(I) = 1$ signifie qu'il est totalement possible que I soit le monde réel (ou que I est totalement satisfaisante).
- $1 > \pi(I) > 0$ signifie que I est quelque peu possible (ou satisfaisante).
- $\pi(I)=0$ signifie que I n'est certainement pas le monde réel (ou est totalement insatisfaisante ou impossible).
- $\pi(I) > \pi(I')$ veut dire que I est plus plausible que I' .

Une distribution de possibilité π est dite "*normalisée*" s'il existe une interprétation I telle que $\pi(I) = 1$, c.à.d, il existe au moins une solution qui est cohérente avec les croyances disponibles, ou bien il existe au moins une solution qui satisfait toutes les préférences [121].

1.2.5.2 Mesure de possibilité et mesure de nécessité

Etant donnée une distribution de possibilités, deux mesures sont définies sur un ensemble d'interprétations, représenté ici par une formule, comme suit [50] :

Le degré de possibilité d'une formule Φ , noté $\Pi(\phi)$ permet d'évaluer la *cohérence* ou la *possibilité* de cette formule. Ce degré correspond au plus grand degré de possibilité associé

aux modèles de ϕ . Formellement, $\Pi(\phi)$ est définie comme suit :

$$\Pi(\phi) = \max\{\pi(I) : I \models \phi\} \quad (1.32)$$

Exemple 1.4 *Dans le domaine des prévisions météorologiques, si le mois en cours est Août, alors il est totalement possible que le ciel soit dégagé vu qu'il n'y a pas d'informations qui contredisent ce fait (exemple prévisions d'orages).*

Le degré de possibilité Π satisfait les propriétés suivantes [76] :

- $\Pi(\perp) = 0$.
- $\Pi(\top) = 1$.
- $\Pi(\phi) = 0$: ϕ est impossible.
- $\Pi(\phi) = 1$: ϕ est complètement possible.
- $\Pi(\phi) = \Pi(\neg\phi) = 1$: ϕ et $\neg\phi$ sont totalement possibles.
- $\max(\Pi(\phi), \Pi(\neg\phi)) = 1$: ϕ ou $\neg\phi$ doit être possible.

Le degré de nécessité de la formule ϕ , noté $N(\phi)$, évalué à quel point ϕ est inférée à partir des informations disponibles exprimées par π .

$N(\phi)$ est défini par :

$$N(\phi) = 1 - \Pi(\neg\phi) = \min\{1 - \pi(I) : I \not\models \phi\} \quad (1.33)$$

Exemple 1.5 *John reçoit un e-mail de la part de son collègue Adel, en lui disant qu'il est à Alger. Si quelqu'un demande à John : est-il là Adel? La réponse sera : Non. car il est impossible que Adel soit en même temps là et à Alger. Cela est équivalent à dire : nécessairement Adel n'est pas là car il est à Alger.*

La relation duale entre Π et N exprime le fait que plus ϕ est certaine moins $\neg\phi$ devient cohérente avec les informations disponibles. Si $N(\phi) = 0$ alors ϕ n'est pas certaine, néanmoins ϕ peut être possible. Si $N(\phi) = 1$ alors ϕ est certainement vraie. Si ϕ est quelque peu certaine alors son contraire n'est pas du tout certain ($N(\phi) > 0 \Rightarrow N(\neg\phi) = 0$). Si ϕ n'est pas complètement possible alors ϕ est complètement incertaine ($\Pi(\phi) < 1 \Rightarrow N(\phi) = 0$).

Le degré de nécessité de la conjonction de deux formules $\phi \vee \psi$ est donnée comme suit :

$$\forall\phi, \forall\psi, N(\phi \vee \psi) = \min(N(\phi), N(\psi)) \quad (1.34)$$

1.2.5.3 Conditionnement possibiliste

Le conditionnement a été initialement introduit en théorie des probabilité par l'équation de Bayes [73]. Le conditionnement est un concept de base pour mettre à jour les pondérations initiales attribuées aux connaissances représentées par la distribution des possibilités après l'arrivée d'une informations certaines, notée ϕ [76].

Deux types de conditionnement dans la théorie des possibilités ont été définis selon l'échelle utilisée [64] : le conditionnement qualitatif et le conditionnement quantitatif.

Dans le contexte qualitatif, le meilleur modèle de ϕ reçoit le degré maximal de possibilité :

$$\Pi(I|\phi) = \begin{cases} 1 & \text{si } \pi(I) = \Pi(\phi) \text{ et } I \models \phi \\ \pi(I) & \text{si } \pi(I) < \Pi(\phi) \text{ et } I \models \phi \\ 0 & \text{sinon} \end{cases} \quad (1.35)$$

Dans le contexte quantitatif, tous les éléments de ϕ sont augmentés proportionnellement :

$$\Pi(I|\phi) = \begin{cases} \frac{\pi(I)}{\Pi(\phi)} & \text{si } I \models \phi \\ 0 & \text{sinon} \end{cases} \quad (1.36)$$

1.2.5.4 Indépendance possibiliste

La relation d'indépendance possibiliste est basée sur le conditionnement possibiliste [64]. Deux événements φ et ϕ sont indépendants conditionnellement à ψ si :

$$\prod(\varphi|\phi, \psi) = \prod(\varphi|\psi) \quad (1.37)$$

La théorie de la possibilité comporte deux types de conditionnement, cela conduit à deux définitions de l'indépendance possibiliste :

- Indépendance possibiliste qualitative

$$\forall \varphi, \phi, \psi \quad \prod(\varphi \wedge \phi|\psi) = \min(\prod(\varphi|\psi), \prod(\phi|\psi)) \quad (1.38)$$

- Indépendance possibiliste quantitative

$$\forall \varphi, \phi, \psi \quad \prod(\varphi \wedge \phi|\psi) = \prod(\varphi|\psi) * \prod(\phi|\psi) \quad (1.39)$$

1.3 Modèles possibilistes Logiques et graphiques

De nombreux modèles en intelligence artificielle ont été développés afin de permettre la représentation et la déduction en présence d'incertitude et/ou d'incohérence pour prendre en compte l'information incertaine [75]. Ces modèles se répartissent en deux principales familles : la famille des modèles logiques [46, 92] et la famille des modèles graphiques [27, 4]. Dans les deux cas, le traitement des connaissances incertaines implique souvent l'affectation d'une pondération aux connaissances.

Le principe des logiques pondérées est d'associer à chaque formule du langage un poids qui a une signification différente selon la logique utilisée. Ces logiques manipulent donc un ensemble de couples (ϕ, α) où ϕ est une formule logique (e.g. propositionnelle) et α est le poids associé à cette formule. Parmi ces logiques, nous nous intéressons à la logique pondérée possibiliste, qui est un modèle fiable pour représenter des connaissances incertaines et imprécises.

Par ailleurs, plusieurs modèles graphiques ont été développées. Pour ces modèles, une quantification numérique est associée à la composante graphique qui représente la structure d'un problème.

Dans cette section, nous nous focalisons sur les modèles basés sur la théorie des possibilités.

1.3.1 Modèle logique

1.3.1.1 La logique possibiliste standard

La logique possibiliste standard est une logique de l'incertain qui s'appuie sur la théorie des possibilités pour modéliser de manière assez naturelle les informations imprécises et/ou incertaines [12]. La logique possibiliste associe un degré de possibilités à des interprétations

[43]. D'une manière formelle, une base de connaissances propositionnelle possibiliste est donnée par :

$$\Sigma = \{(\phi_i, \alpha_i) : i = 1, \dots, n\} \quad (1.40)$$

où ϕ_i est une formule propositionnelle et α_i est un nombre réel dans l'intervalle $[0, 1]$ qui représente le degré de nécessité minimale associé à ϕ_i i.e., $(N(\phi_i) \geq \alpha_i)$. α_i est le niveau de certitude de ϕ_i . La logique possibiliste dispose de deux variantes : la logique possibiliste qualitative et la logique possibiliste quantitative. La logique possibiliste qualitative est une logique de l'incertain connue pour raisonner avec des connaissances incomplètes et partiellement inconsistantes. Elle est basée sur l'idée d'ordre. Contrairement à la logique possibiliste qualitative, les quantifications numériques attachées aux formules, en logique possibiliste quantitative, n'induisent pas une relation d'ordre. En effet, la pondération associée à chaque formule permet juste de quantifier son degré d'incertitude.

1.3.1.2 Distribution de possibilités associée à une base possibiliste

La représentation sémantique des informations exprimées par Σ est donnée au moyen d'une distribution de possibilités qui associe à chaque interprétation possible son degré de possibilité par rapport aux informations de Σ [43]. Etant donné une base possibiliste Σ , nous pouvons générer une distribution de possibilité unique, notée π_Σ , où les interprétations qui satisfont toutes les formules de Σ auront le plus grand degré de possibilité à savoir 1 et les autres interprétations seront ordonnées par rapport à la formule de plus grand poids qu'elles falsifient [42].

D'une manière formelle, la distribution de possibilité associée à une formule pondérée (ϕ_i, α_i) est définie par (1.41) [42]; $\forall \omega \in \Omega$:

$$\pi_{(\phi_i, \alpha_i)}(\omega) = \begin{cases} 1 - \alpha_i & \text{si } \omega \not\models \phi_i \\ 1 & \text{sinon} \end{cases} \quad (1.41)$$

Considérons à présent le cas général, où la base de connaissances contient plus d'une formule incertaine. La distribution de possibilités associée à une base de connaissances possibiliste Σ est la combinaison des distributions de possibilités associées à chaque formule pondérée (ϕ_i, α_i) de Σ [76].

$\forall \omega \in \Omega$:

$$\pi_\Sigma(I) = \bigoplus_{\langle \phi_i, \alpha_i \rangle \in \Sigma} \pi_{(\phi_i, \alpha_i)}(I) \quad (1.42)$$

Où \bigoplus est une opération de fusion.

Comme nous l'avons vu précédemment dans la section 1.2.5, la théorie des possibilités offre deux contextes de travail : contexte quantitatif et qualitatif.

Selon le contexte de la base possibiliste, nous distinguons deux instanciations possibles de l'opérateur \bigoplus (min ou produit) donnant naissance à deux types différents de distribution de possibilités :

- **Distribution de possibilités associée à une base possibiliste qualitative**

$$\pi_\Sigma(\omega) = \begin{cases} 1 & \text{si } \forall (\phi_i, \alpha_i) \in \Sigma, \omega \models \phi_i \\ \min_{(\phi_i, \alpha_i) \in \Sigma, \omega \not\models \phi_i} (1 - \alpha_i) & \text{sinon} \end{cases} \quad (1.43)$$

• **Distribution de possibilités associée à une base possibiliste quantitative**

$$\pi_{\Sigma}(\omega) = \begin{cases} 1 & \text{si } \forall(\phi_i, \alpha_i) \in \Sigma, \omega \models \phi_i \\ \prod_{(\phi_i, \alpha_i) \in \Sigma, \omega \not\models \phi_i} (1 - \alpha_i) & \text{sinon} \end{cases} \quad (1.44)$$

Exemple 1.6 Soit la base de connaissances possibiliste standard suivante :

$$\Sigma = \{(a \vee f \vee e, 0.3), (\neg f \vee d, 0.5), (\neg e, 0.9), (\neg a \vee f, 0.1)\}.$$

La distribution de possibilités est donnée dans la table 1.6.

	ω	$\pi_{min}(\omega)$	$\pi_*(\omega)$
w_1	$a \wedge f \wedge e \wedge d$	0.1	0.1
w_2	$a \wedge f \wedge e \wedge \neg d$	0.5	$0.5 * 0.1$
w_3	$a \wedge f \wedge \neg e \wedge d$	1	1
w_4	$a \wedge f \wedge \neg e \wedge \neg d$	0.5	0.5
w_5	$a \wedge \neg f \wedge e \wedge d$	0.1	$0.1 * 0.9$
w_6	$a \wedge \neg f \wedge e \wedge \neg d$	0.9	$0.1 * 0.9$
w_7	$a \wedge \neg f \wedge \neg e \wedge d$	0.9	0.9
w_8	$a \wedge \neg f \wedge \neg e \wedge \neg d$	0.9	0.9
w_9	$\neg a \wedge f \wedge e \wedge d$	0.1	0.1
w_{10}	$\neg a \wedge f \wedge e \wedge \neg d$	0.5	$0.1 * 0.5$
w_{11}	$\neg a \wedge f \wedge \neg e \wedge d$	1	1
w_{12}	$\neg a \wedge f \wedge \neg e \wedge \neg d$	0.5	0.5
w_{13}	$\neg a \wedge \neg f \wedge e \wedge d$	0.1	0.1
w_{14}	$\neg a \wedge \neg f \wedge e \wedge \neg d$	0.5	$0.5 * 0.1$
w_{15}	$\neg a \wedge \neg f \wedge \neg e \wedge d$	0.7	0.7
w_{16}	$\neg a \wedge \neg f \wedge \neg e \wedge \neg d$	0.7	0.7

TABLE 1.2 – Distribution de possibilités associée à la base de connaissances de l'exemple 1.6

1.3.1.3 Degré d'inconsistance

Une caractéristique importante de la logique possibiliste est sa capacité à faire face à l'inconsistance. L'inconsistance correspond à l'absence d'une interprétation satisfaisant toutes les formules de la base de connaissances [43].

Une base possibiliste Σ est dite cohérente (normalisée) s'il existe une interprétation I telle que $\pi(I) = 1$. Σ est totalement incohérente, si chaque interprétation est impossible, c'est-à-dire que chaque interprétation falsifie au moins une formule certaine. Sinon, Σ est considérée comme quelque peu incohérente. Formellement, ce degré est décrit par l'équation (1.45) :

$$Inc(\Sigma) = 1 - \max_{I \in \Omega} (\pi_{\Sigma}(I)) \quad (1.45)$$

Nous rappelons que le degré de normalisation reflète dans quelle mesure la connaissance disponible (codée par π) est cohérente. Lorsque la base de connaissance Σ est dite normalisée. Cela signifie qu'il existe une interprétation I telle que $h(I) = 1$. Cela signifie également que I satisfait tous les éléments de la base de connaissances.

1.3.2 Modèles graphiques

En plus des représentations logiques, il existe des représentations graphiques de l'incertitude dans la théorie des possibilités. Dans cette section, nous présentons les réseaux

possibilistes, les arbres de décision [67], les diagramme d'influence [68] et les systèmes basés sur les évaluations [104]. Ces techniques permettent une représentation simple et dense des problèmes de décision séquentielle et sont aussi des techniques simples et intuitives pour le traitement des problèmes de décision de grande taille [28, 59].

1.3.2.1 Réseaux possibilistes

Un réseau possibiliste est une adaptation du réseau probabiliste (Bayésien), dans le sens où nous utilisons la même structure graphique qui est le graphe acyclique direct (*DAG*), noté G_N . Un DAG est composé d'un ensemble de nœuds, représentés par des variables et un ensemble d'arcs. Chaque arc reliant deux nœuds peut avoir une direction optionnelle qui lui est affectée. Un arc orienté relie un nœud parent à un nœud enfant explicitant une relation (cause, effet). Cette structure graphique est une représentation qualitative du réseau [76]. A chaque variable est associée une distribution a priori ou conditionnelle de possibilité discrète, dont la définition dépend du nombre d'arcs entrant dans sa composition. Cette phase illustre la représentation quantitative du réseau, noté π_N .

En plus, les graphes acycliques directs fournissent une représentation compacte des relations d'indépendance introduites par les distributions conditionnelles. Ces dernières, doivent respecter la contrainte de normalisation :

- Pour tout nœud racine A, la distribution de possibilités a priori vérifie :

Soit D_A le un domaine supposé fini associé à la variable A :

$$\forall a \in D_A, \max_{a \in D_A} \Pi(a) = 1. \quad (1.46)$$

- Pour tout nœud A tel que $Par_A \neq \emptyset$ la distribution de possibilités conditionnelle vérifie :

$$\forall a \in D_A, \max_{a \in D_A} \Pi(a|Par_A) = 1. \quad (1.47)$$

Dans un réseau possibiliste, les modes de conditionnement prédéfinis conduisent à établir deux méthodes de calcul de la distribution jointe. Il est évident que l'une sera basée sur l'opérateur minimum tandis que l'autre sera basée sur l'opérateur produit. Ainsi, nous remarquons que la distribution jointe quantitative est similaire à celle présentée dans les réseaux bayésiens [76]. Ce qui n'implique pas bien sûr que la méthode d'inférence soit identique.

Soit un réseau possibiliste $N = (G_N, \pi_N)$ définit sur un ensemble de variables $V = \{A_1, A_2, \dots, A_N\}$.

Définition 1.15 *La règle de chaînage basée sur le minimum ou règle de chaînage qualitative définit une distribution de possibilité jointe associée à un réseau possibiliste. Elle est exprimée par :*

$$\forall a_i \in D_{A_i}, \forall Par_{a_i} \in Par_{A_i}, \pi_{min}(a_1, a_2, \dots, a_N) = \min_{i=1..N} \Pi(a_i|Par_{a_i}) \quad (1.48)$$

Exemple 1.7 *Soit réseau possibiliste, illustré par la Figure 1.9 et dont les table de distributions de possibilités initiales est donnée par la tables 1.3. La distribution de possibilité jointe est donnée par la Table 1.4.*

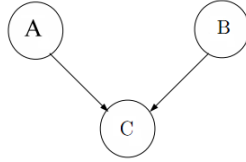


FIGURE 1.9 – Modèle d'un DAG

A	$\pi(A)$	B	$\pi(B)$	A	B	C	$\pi(C A \wedge B)$
a_1	0.1	b_1	0.6	a_1	b_1	c_1	0.5
a_2	1	b_2	1	a_1	b_1	c_2	1
				a_1	b_2	c_1	1
				a_1	b_2	c_2	0.2
				a_2	b_1	c_1	0.3
				a_2	b_1	c_2	1
				a_2	b_2	c_1	0
				a_2	b_2	c_2	1

TABLE 1.3 – Distributions initiales

A	B	C	$\pi_{min}(A \wedge B \wedge C)$
a_1	b_1	c_1	0.1
a_1	b_1	c_2	0.1
a_1	b_2	c_1	0.1
a_1	b_2	c_2	0.1
a_2	b_1	c_1	0.3
a_2	b_1	c_2	0.2
a_2	b_2	c_1	0
a_2	b_2	c_2	1

TABLE 1.4 – Distributions de possibilités jointes

1.3.2.2 Les arbres de décision possibilistes

Les arbres de décision possibilistes ont la même composante graphique que les arbres de décision probabilistes [32]. Ils sont composés d'un ensemble de noeuds N et d'un ensemble d'arcs ε . Comme pour les arbres de décision probabilistes, l'ensemble des noeuds N est composé de trois types de noeuds $N = D \cup C \cup LN$ où D est l'ensemble des noeuds décision, C l'ensemble des noeuds chance et LN est l'ensemble des feuilles [32].

Par contre la composante numérique n'est pas la même :

- Les arcs issus de noeuds de chance sont quantifiés par des degrés de possibilités dans le contexte de leurs parents.

Pour chaque $C_i \in C$, la distribution de possibilités conditionnelle π_i sur $Succ(C_i)$, $\forall N \in Succ(C_i), \pi_i = \Pi(N | chemin(C_i))$.

Pour chaque noeud C_i , une loterie possibiliste L_{C_i} lui est associée.

- Une utilité est associée à chaque noeud feuille.

La résolution d'un arbre décision revient à la construction de la stratégie optimale $\delta^* \in \Delta$. Ces stratégies peuvent être évaluées et comparées grâce à la notion de réduction de loterie

possibiliste : chaque noeud de chance peut être considéré comme une loterie simple ou comme une loterie composée. Chaque stratégie est donc une loterie composée et peut être réduite à une simple loterie équivalente.

1.3.2.3 Diagrammes d'influence possibilistes

Les diagrammes d'influence possibilistes (DIP) sont la contre-partie possibiliste des diagrammes d'influence (ID). Ils permettent de modéliser sous une forme compacte des problèmes de décision séquentielle sous incertitude qualitative dans le cadre de la théorie des possibilités. Selon la quantification des nœuds de chance et de décision, nous distinguons deux types de diagrammes d'influence possibilistes à savoir : les homogènes et les hétérogènes.

Un DIP noté ΠID_{\otimes}^u , a la même composante graphique qu'un diagramme d'influence standard. Il possède un ensemble de noeuds $N = D \cup C \cup V$, tel que D est l'ensemble des nœuds de décision, C est l'ensemble des noeuds de chance et V l'ensemble des noeuds d'utilités et d'un ensemble d'arcs A .

Pour ce qui est de la partie numérique, elle s'appuie sur la logique possibiliste :

- Pour chaque nœud $C_i \in C$, nous devons fournir des degrés de possibilités conditionnels $\pi(c_{ij}|Par(C_i))$, pour chaque instance c_{ij} de C_i dans le contexte de ses parents. Afin de satisfaire la contrainte de normalisation, ces distributions conditionnelles doivent vérifier :

$$\max_{c_{ij} \in D_{c_j}} \Pi(c_{ij}|Par(C_i)) = 1 \quad (1.49)$$

Dans ce qui suit ΠID_*^u (rep ΠID_{min}^u) dénote les diagrammes d'influence possibilistes où les distributions de possibilités conditionnelles sont modélisées dans le cadre quantitatif (resp. qualitatif).

- Pour chaque nœud d'utilité $V_i \in V$, un ensemble d'utilité U est défini dans le contexte de chaque instantiation $Par(V_i)$ de ses parents $Par(V_i)$. Dans ce qui suit ΠID_{\otimes}^* (resp ΠID_{\otimes}^{min}) dénote les diagrammes d'influence possibilistes où les utilités sont quantitatifs (resp. qualitatives).
- Les nœuds de décision dans les DIP ne sont pas quantifiés (comme pour les DI standards).

Les nœuds de chance agissent différemment des noeuds de décision, étant donné qu'ils ne sont pas quantifiés. Pour un noeud de chance C_i et un noeud de décision D_i , ça n'a pas de sens de considérer $\Pi(c_{ij}, d_{ij})$. En effet, ce qui est significatif est $\Pi(c_{ij}|do(d_{ij}))$, tel que $do(d_{ij})$ est un opérateur particulier défini par Pearl [96]. En utilisant la règle de chaînage relative aux réseaux possibilistes [11] et aux diagrammes d'influence standards [91], la règle de chaînage suivante pour les DIP peut être déduite :

$$\pi(C|D) = \bigotimes_{C_i \in C} \Pi(C_i|Pa(C_i)) \quad (1.50)$$

tel que \bigotimes est l'opérateur *min* dans le cas de la théorie des possibilités qualitative ou l'opérateur *produit* dans le cas de la théorie des possibilités quantitative.

Comme pour les diagrammes d'influence standards, la preuve générale de l'Equation 1.50, concernant la règle de chaînage des diagrammes d'influence possibilistes, peut être

faite en considérant une configuration particulière d'une décision d . Si cette configuration est insérée dans le diagramme d'influence possibiliste alors nous aurons un réseau possibiliste représentant $\Pi(C|d)$. En utilisant la règle de chaînage des réseaux possibilistes [9], nous obtenons deux cas par rapport à l'interprétation de l'échelle de l'incertitude :

1. Pour les attributions quantitatives, $\Pi(C|d)$ est le produit de toutes les possibilités potentielles attachées aux variables de décision instanciées à d .
2. Pour les attributions qualitatives, $\Pi(C|d)$ est le minimum de toutes les possibilités potentielles attachées aux variables de décision instanciées à d .

En d'autres termes, les diagrammes d'influence possibilistes sont une représentation compacte de la distribution conjointe relative aux noeuds de chance conditionnés par une configuration des noeuds de décision. Différentes combinaisons entre la quantification des noeuds de chance et les noeuds d'utilité dans les diagrammes d'influence offrent plusieurs types de diagrammes d'influence possibilistes qui peuvent être regroupées en deux classes principales [60] :

- **Les diagrammes d'influence possibilistes homogènes**, où les noeuds de chance et les noeuds d'utilités sont quantifiés dans le même cadre. Dans cette classe, nous pouvons distinguer deux variantes :
 - *Diagramme d'influence possibiliste basé sur le produit*, dénoté par IID_* , où les dépendances entre les noeuds de chance et les noeuds d'utilité sont quantifiées dans un environnement numérique authentique.
 - *Diagramme d'influence possibiliste basé sur le min*, dénoté par IID_{\min}^{\min} , où les dépendances entre les noeuds de chance et les noeuds d'utilité sont quantifiées dans un cadre qualitatif utilisé pour coder un ordre entre les différents états du monde [60].
- **Les diagrammes d'influence possibilistes hétérogènes**, où les noeuds de chance et d'utilité ne sont pas quantifiés dans le même cadre. Selon cette quantification, il y a deux types de diagrammes d'influence possibilistes hétérogènes qui seront dénotés par IID_*^{\min} et IID_{\min}^* .

Les différents types de diagrammes d'influence possibilistes sont résumés dans la tables suivante :

$U \Pi$	Qualitatif	Quantitatif
Qualitatif	IID_{\min}^{\min}	IID_*^{\min}
Quantitatif	IID_{\min}^*	IID_*

TABLE 1.5 – Classification des diagrammes d'influence possibilistes

L'exemple suivant montre un diagramme d'influence possibiliste basé sur le min (min-based)

Exemple 1.8 *Le diagramme d'influence de la Figure 1.10, est défini par $D = \{D_1, D_2\}$, $V = \{U\}$ et $C = \{A_1, A_2\}$.*

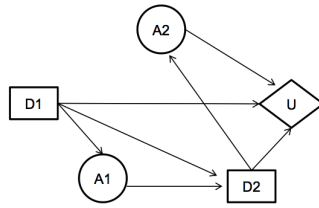


FIGURE 1.10 – La composante graphique du diagramme d'influence de l'Exemple 1.8

Les possibilités conditionnelles sont représentées par les Tables 1.6 et 1.7. La table 1.8 représente l'ensemble des utilités du noeud d'utilité U .

La table 1.9, représente l'application de la règle de chaînage du diagramme d'influence possibiliste de la Figure 1.10 en utilisant l'équation 1.50 dans le cadre qualitatif de la théorie des possibilités.

A_1	D_1	$\pi(A_1 D_1)$
T	T	1
T	F	0.4
F	T	0.2
F	F	1

TABLE 1.6 – Les possibilistes conditionnelles de A_1

A_2	D_2	$\pi(A_2 D_2)$
T	T	0.3
T	F	1
F	T	1
F	F	0.4

TABLE 1.7 – Les possibilistes conditionnelles de A_2

D_1	D_2	A_2	$u(D_1, D_2, A_2)$
T	T	T	0.2
T	T	F	0.3
T	F	T	0.4
T	F	F	0.6
F	T	T	1
F	T	F	0
F	F	T	0.1
F	F	F	0.7

TABLE 1.8 – La fonction d'utilités $u(D_1, D_2, A_2)$

A_1	A_2	D_1	D_2	$\pi(A_1, A_2 D_1, D_2)$
T	T	T	T	0.3
T	T	T	F	1
T	T	F	T	0.3
T	T	F	F	0.4
T	F	T	T	1
T	F	T	F	0.4
T	F	F	T	0.4
T	F	F	F	0.4
F	T	T	T	0.2
F	T	T	F	0.2
F	T	F	T	0.3
F	T	F	F	1
F	F	T	T	0.2
F	F	T	F	0.2
F	F	F	T	1
F	F	F	F	0.4

TABLE 1.9 – L’application de la règle de chaînage du diagramme d’influence possibiliste de la Figure 1.10

Evaluation des diagrammes d’influence possibilistes DIP

Peu de travaux se sont intéressés au problème de l’évaluation d’un diagramme d’influence possibiliste. Garcia et al. [54] ont proposé deux méthodes pour l’évaluation des DIP en utilisant l’utilité pessimiste et optimiste. Leur premier travail consiste en une méthode indirecte basée sur une transformation du DIP vers un arbre de décision, et dans le second travail, ils ont proposé un algorithme d’élimination de variables.

Evaluation des DIP en utilisant les arbres de décision

Un DIP peut être transformé en un arbre de décision possibiliste en utilisant une méthode de transformation similaire à celle proposée pour le cas de diagrammes d’influence standards. Cette transformation peut donner lieu à de nouvelles dépendances entre les nœuds de chance qui seront quantifiés en utilisant les distributions de possibilités initiales. Les valeurs d’utilités sont les mêmes que celles définies dans le diagramme d’influence et seront affectées à chaque feuille dans l’arbre de décision. Une fois l’arbre de décision construit, il doit être évalué afin de trouver la stratégie optimale.

Evaluation des DIP en utilisant les réseaux possibilistes

L’idée de cette méthode d’évaluation est d’adapter la méthode de Cooper [38] en transformant le diagramme d’influence initial en un réseau possibiliste, ensuite de l’utiliser pour effectuer des calculs de manière locale via les algorithmes de propagation. En effet, les réseaux possibilistes [54] sont équivalents aux réseaux bayésiens et peuvent être définis dans une version "min-based" ou "product-based".

Dans le travail présenté dans [59], les auteurs proposent une évaluation des DIP avec plusieurs nœuds d’utilité en utilisant un pré-traitement sur le diagramme d’influence avant sa transformation en un réseau possibiliste. L’étape de pré-traitement consiste en la réduction

du nombre de noeuds d'utilité en noeud (noté V_r) qui va hériter des parents de tous les noeuds d'utilité. Le noeud d'utilité V_r aura le minimum d'utilité. Formellement :

$$u(V_r|Par(V_r)) = \min_{i=1..k} u(V_i|Par(V_r)) \quad (1.51)$$

L'idée clé de l'algorithme proposé consiste à transformer les noeuds de décision et d'utilité en des noeuds de chance afin d'obtenir des réseaux possibilistes, et ensuite d'effectuer la propagation dans cette structure secondaire.

Les nouveaux noeuds de chance obtenus à partir de la transformation des noeuds de décision doivent être caractérisés par l'ignorance totale à savoir :

$$\Pi(d_{ij}|Par(D_i)) = 1, \forall D_i \in D \quad (1.52)$$

Les noeuds d'utilité vont être transformés en un nouveau noeud de chance binaire qui peut être quantifié en fonction de la nature des utilités. Nous pouvons distinguer deux cas :

1. Si les utilités et les possibilités sont commensurables et l'échelle d'incertitude est $[0, 1]$, les noeuds de chance issus de la transformation des noeuds d'utilités sont quantifiés par les deux équations suivantes :

$$\Pi(V_r = V|Par(V_r)) = u(Par(V_r)) \quad (1.53)$$

Et

$$\Pi(V_r = F|Par(V_r)) = 1 \quad (1.54)$$

2. Si les utilités et les possibilités ne sont pas commensurables alors chaque utilité doit être transformée à l'échelle $[0, 1]$:

$$\Pi(V_r = V|Par(V_r)) = \frac{u(Par(V_r)) - U_{min}}{U_{max} - U_{min}} \quad (1.55)$$

Et

$$\Pi(V_r = F|Par(V_r)) = 1 \quad (1.56)$$

U_{max} (reps. U_{min}) est l'utilité maximale dans $U(Par(V_r))$ (reps. l'utilité minimale dans $U(Par(V_r))$).

1.4 Conclusion

Dans ce chapitre, nous avons présenté les concepts de base des principales théories de l'incertain : la théorie des probabilités, la théorie des fonctions de croyance, la théorie des ensembles flous et la théorie des possibilités. Nous avons introduit de nombreux concepts importants relatifs à chaque théorie. Ensuite nous nous sommes intéressés plus en détail à la théorie des possibilités qui offre un cadre approprié pour traiter de l'incertitude de manière aussi bien qualitative que quantitative. Enfin, nous avons présenté brièvement les principes des modèles logiques de l'incertain en mettant l'accent sur la logique possibiliste avant de passer aux modèles graphiques qui apportent de l'expressivité et de l'efficacité de calcul pour la représentation des incertitudes, à savoir : les réseaux possibilistes, les arbres de décisions et les diagrammes d'influence. Le chapitre suivant présente la logique de description et la famille de la logique de description légère DL-Lite.

Chapitre 2

Les logiques de description

2.1 Introduction

La représentation des ontologies, qui joue un rôle crucial pour le succès du projet du web sémantique, exige des langages de modélisation et des services de raisonnement intelligents pour supporter les problèmes de conception, de gestion, de récupération, et d'intégration. Dans plusieurs applications réelles, un système moderne de représentation de connaissances est utile s'il a la possibilité non seulement de traiter de grandes masses de données mais aussi de fournir des langages de requêtes expressifs.

Afin d'atteindre ces objectifs, les logiques de description offrent un formalisme approprié. Elles constituent un axe de recherche actif dans le domaine de la représentation des connaissances avec de multiples applications pratiques. L'orientation principale de la recherche sur les logiques de description consiste à développer des théories et des systèmes pour représenter les connaissances structurées et pour raisonner sur ces connaissances. Les systèmes de logique de description fournissent aux utilisateurs plusieurs possibilités d'inférence permettant d'inférer des connaissances implicites à partir des connaissances explicites stockées dans une base de connaissances. L'inférence élémentaire sur les expressions de concepts en logique de description est la subsomption entre deux concepts et sur les individus et la détermination si un individu donné est une instance d'un certain concept.

Ce chapitre commence par présenter brièvement la définition du web sémantique et la notion d'ontologie. Ensuite, nous continuerons avec les logiques de description qui jouent un rôle important pour la représentation des ontologies. La section suivante se focalise sur la famille des logiques de description légères DL-Lite. Nous présentons leur syntaxe et leur sémantique, puis nous rappelons la notion de fermeture négative et son utilisation pour le calcul des conflits. Enfin, nous terminons ce chapitre en passant en revue les principales extensions existantes des logiques de description pour tenir compte les connaissances incertaines.

2.2 Le Web Sémantique

Le concept du Web sémantique se réfère à la vision du web de demain dans laquelle les utilisateurs devraient être déchargés d'une grande partie de leurs tâches de recherche et d'exploitation, grâce à l'augmentation des capacités des machines à accéder au contenu des ressources et à effectuer des raisonnements sur celles-ci.

Le Web sémantique est une infrastructure permettant l'utilisation de connaissances formalisées et du contenu informel existant dans le Web actuel. Cette infrastructure repose

sur certains consensus, par exemple, sur les langages de représentation ou sur les ontologies utilisées [65]. Ainsi, elle permet l'interopérabilité et les transformations entre les différents formalismes et les différentes ontologies. Grâce à la formalisation des connaissances, la mise en œuvre de raisonnements complexes peut être facile, tout en offrant des garanties supérieures pour leur fiabilité. Sur une base de sémantiques bien définies, le Web sémantique peut fournir aux utilisateurs des services automatisés et avancés [80].

Dans le contexte du Web sémantique, la plupart des langages développés se basent sur XML grâce à son caractère de métalangage. Cependant, XML est limité, car il ne dispose pas d'une sémantique, ce qui nécessite le développement d'autres langages pour le Web sémantique. Parmi les langages de représentations du Web sémantique, nous trouvons : RDF, RDF Schéma (RDFS) et OWL. Nous présentons brièvement la définition de ces langages :

- RDF¹ est un modèle graphique pour décrire formellement les ressources du Web et leurs méta-données, afin de permettre le traitement automatique de ces descriptions. RDF est basé sur des vocabulaires formels et précis. Il permet une certaine interopérabilité entre les applications échangeant des informations informelles et non structurées sur le Web, ce qui rend le traitement automatisé de l'information sur le Web plus pertinent. Cependant, il ne permet pas à l'utilisateur de définir des vocabulaires qui comportent une sémantique de description de ressources. Un schéma RDF avec des concepts de base peut offrir cette fonctionnalité en utilisant des classes de ressources.
- RDFS² ajoute à RDF la possibilité de définir des hiérarchies de classes et de propriétés en utilisant les attributs *rdfs : domain* et *rdfs : range* [80]. Pour chaque domaine est associé un schéma identifié par un préfixe particulier. Les instances de ressources sont décrites en utilisant le vocabulaire donné par les classes définies dans ce schéma. Ensuite, une interprétation opérationnelle est donnée aux applications. RDFS fournit des primitives pour représenter les structures, décrire les concepts utilisés dans des déclarations RDF et décrire également un ensemble de contraintes sur les objets et les valeurs triples.
- Le langage d'ontologie OWL est un standard basé sur la logique de description³. Il est conçu pour décrire et représenter un domaine particulier de la connaissances, en définissant leurs catégories et leurs relations, mais aussi pour identifier les individus et confirmer leurs caractéristiques et de raisonner sur ces classes et individus en se basant sur une sémantique formelle.

2.3 Ontologies

Les ontologies jouent un rôle important en intelligence artificielle en général et dans le domaine du Web sémantique en particulier. Elles sont employées comme une forme de représentation de la connaissance au sujet d'un monde ou d'une certaine partie du monde. La définition suivante caractérise le mieux une ontologie :

Une ontologie est une spécification formelle, explicite d'une conceptualisation partagée [58].

1. <http://www.w3.org/RDF/>
2. <http://www.w3.org/TR/rdf-schema/>
3. <http://www.w3.org/standards/techs/owl>

Une conceptualisation est une abstraction d'un domaine d'application. Elle identifie les concepts pertinents de ce domaine. Une conceptualisation partagée signifie que les concepts dans l'ontologie sont acceptés par une communauté d'utilisateurs. L'ontologie est une spécification explicite parce que tous les concepts et les contraintes utilisées de ces concepts sont explicitement définis. La spécification est formelle puisque l'ontologie doit être représentée par un langage formel avec lequel la machine peut opérer. Pour cette définition, une ontologie fournit un vocabulaire partagé, incluant les concepts importants, les propriétés, les définitions, et les contraintes qui sont employés dans un domaine communicatif entre les personnes et les systèmes d'applications hétérogènes et distribués [95].

Selon Gruber, cinq critères nous permettent de mettre en évidence les aspects importants d'une ontologie : la clarté, la cohérence, l'extensibilité, la déformation d'encodage minimale et l'engagement ontologique minimal.

Il existe différents critères de classification des ontologies : le domaine d'application, le but de leur usage, le niveau de complétude, le niveau de formalisme, l'objet de conceptualisation, etc.

2.4 Les logiques de description

"Les logiques de descriptions" est un nom récent donné à une famille de formalismes de représentation de connaissances. Les logiques de description (LDs) sont des fragments décidables de la logique du premier ordre. Leur fonction est de permettre aux utilisateurs de représenter convenablement des connaissances structurées de telle sorte que : (1) la représentation soit sans ambiguïté et ait un sens bien défini et (2) des connaissances implicites peuvent être déduites à partir des connaissances explicites [7].

Les logiques de description forment une famille de formalismes de représentation des connaissances terminologiques d'un domaine d'application d'une manière structurée et bien formelle. Elles définissent les notions importantes d'un domaine par les descriptions de concepts [94].

Les expressions sont construites à partir des concepts atomiques et des rôles atomiques utilisant les constructeurs de concept et de rôle fournis par une logique de description particulière. Des descriptions complexes peuvent être construites à partir d'inducteurs avec des constructeurs conceptuels. Dans la notation abstraite, nous utilisons les lettres A et B pour les concepts atomiques, la lettre R pour les rôles atomiques et les lettres C et D pour les descriptions de concepts.

Les principaux constructeurs de concepts et de rôles sont :

- **intersection de concepts** notée $C \sqcap D$, est composée des individus qui appartiennent à la fois au concept C et au concept D .
- **union de concepts** notée $C \sqcup D$, est le concept regroupant les individus des concepts C et D .
- **restriction universelle** notée $\forall R.C$, est composée des individus qui ne sont en relation par le rôle R qu'avec des individus du concept C .
- **restriction existentielle** notée $\exists R.C$, est composée des individus qui sont en relation par le rôle R à un individu du concept C .
- **restriction existentielle non qualifiée** notée $\exists R$: il s'agit du concept composé des individus qui sont en relation par le rôle R à un individu quelconque.
- **négation d'un concept** notée $\neg C$, est le complémentaire du concept C dans la base. Elle représente l'ensemble des individus qui n'appartiennent pas au concept C .

- **inverse d'un rôle** noté R^- : si un individu a est en relation avec un individu b par le rôle R alors b est en relation avec a par le rôle R^- .

Les logiques de description utilisent aussi le concept universel \top auquel appartiennent tous les individus et le concept \perp auquel n'appartient aucun individu.

La composante d'assertions ABox, décrit un état spécifique du domaine d'application en termes de concepts et de rôles. En fait, dans une ABox, nous introduisons des individus (en leur donnant des noms) ainsi que leurs propriétés. La ABox contient uniquement des assertions de la forme : $C(a)$ et $R(a, b)$ où C un concept et R un rôle et a, b sont des noms d'individus.

La sémantique de la logique de description en termes d'interprétations fixe la signification des termes dans le vocabulaire. Une interprétation consiste en un domaine non vide et une fonction d'interprétation, ce qui donne un sens aux termes dans le vocabulaire. Chaque individu est interprété comme un objet dans le domaine. Chaque concept, qui correspond intuitivement à un ensemble d'objets dans une classe est ordinairement interprété comme un sous-ensemble du domaine. Enfin, chaque rôle correspond à une relation binaire entre les objets.

Une interprétation $I = (\Delta^I, \cdot^I)$ consiste en un ensemble non-vidé du domaine d'interprétation Δ^I et d'une fonction d'interprétation \cdot^I qui associe :

- à chaque concept A , un ensemble A^I de Δ^I ,
- à chaque rôle R , un ensemble R^I de $\Delta^I \times \Delta^I$.
- à chaque individu b , un élément b^I de Δ^I ,

La fonction d'interprétation est étendue aux descriptions de concept comme indiqué dans la Table 2.1.

Syntaxe	Sémantique
\top	Δ^I
\perp	\emptyset
$\neg C$	$\Delta^I \setminus C^I$
$C \sqcap D$	$C^I \cap D^I$
$\forall R.C$	$\{x \in \Delta^I \mid \forall y : (x, y) \in R^I \rightarrow y \in C^I\}$
$\exists R.C$	$\{x \in \Delta^I \mid \exists y : (x, y) \in R^I \wedge y \in C^I\}$

TABLE 2.1 – Syntaxe et Sémantique de termes de concept et de rôle

Un axiome d'inclusion $C \sqsubseteq D$ se traduit par la formule logique $\forall X(C(X) \rightarrow D(X))$ et un axiome d'équivalence $C \equiv D$ par $\forall X(C(X) \leftrightarrow D(X))$. Les Tboxes les plus simples considérées en logique de description sont restreintes à un ensemble de définitions de concepts, c'est-à-dire un ensemble acyclique d'axiomes d'équivalence de la forme $A \equiv C$ où A est un concept atomique apparaissant en partie gauche d'un seul axiome d'équivalence.

L'interprétation I satisfait l'assertion de concept $C(a)$ si $a^I \in C^I$, et I satisfait l'assertion de rôle $R(a, b)$ si $(a^I, b^I) \in R^I$. Une interprétation I satisfait la ABox \mathcal{A} (I est un modèle de \mathcal{A}) si I satisfait toute assertion dans \mathcal{A} .

Une interprétation I est un modèle de \mathcal{K} si elle est un modèle de tous les axiomes de \mathcal{T} et toutes les assertions de \mathcal{A} .

Plus les logiques de description sont expressives, plus les chances sont grandes pour que les problèmes d'inférence soient non décidables ou de complexité très élevée. La Figure 2.1,

présente les différentes familles de logiques de description avec leurs principales caractéristiques en terme d'expressivité.

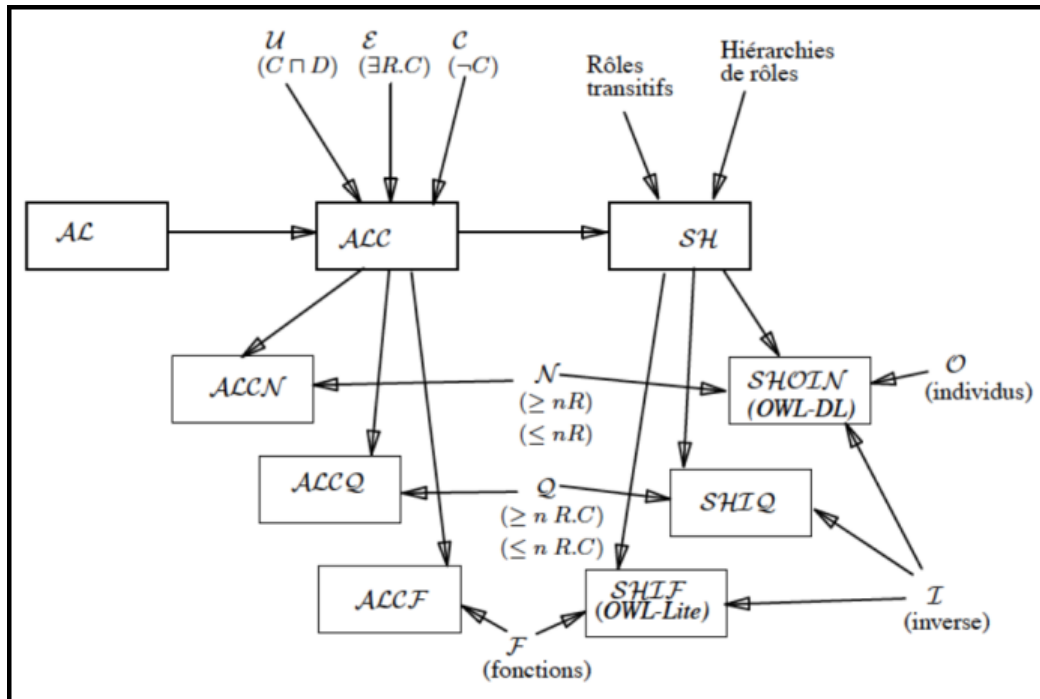


FIGURE 2.1 – Les différentes familles de logiques de description

2.4.1 Des logiques de description simples

Dans ce qui suit, nous présentons brièvement les deux types de logiques de base : \mathcal{AL} , et \mathcal{EL} . Pour ces deux logiques de description minimales, le test de subsomption entre expressions de concepts est polynomial. Elles contiennent toutes les deux le constructeur de conjonction $C_1 \sqcap C_2$.

2.4.1.1 La logique de description \mathcal{AL}

Les descriptions de concepts dans le langage \mathcal{AL} sont générées selon les règles syntaxiques suivantes :

C, D	\rightarrow	A		(concept atomique)
		\perp		(concept bas)
		\top		(concept universel)
		$\neg A$		(négation atomique)
		$C \sqcap D$		(intersection)
		$\forall R.C$		(restriction de valeur)
		$\exists R.T$		(quantification existentielle limitée)

Il est à noter, que dans le langage \mathcal{AL} , la négation n'est appliquée qu'aux concepts atomiques et le concept universel n'est permis qu'au quantificateur existentiel. D'autre part, le sous langage de \mathcal{AL} obtenu en supprimant la négation atomique est appelé FL - alors que le sous langage de FL - obtenu en supprimant le quantificateur existentiel limité est appelé FL_0 .

Exemple 2.1 Soient *Personne* et *Femme* deux concepts atomiques. $Personne \sqcap Femme$ et $Personne \sqcap \neg Femme$ sont des concepts décrivant intuitivement les personnes qui sont Femmes et les personnes qui ne sont pas Femmes. A partir du rôle atomique *enfant*, nous pouvons construire les concepts : $Personne \sqcap \exists enfant.T$ et $Personne \sqcap \forall enfant.Femme$, dénotant respectivement les personnes qui ont au moins un enfant et les personnes dont tous les enfants sont Femmes.

Nous pouvons aussi décrire les personnes qui n'ont pas d'enfants en utilisant le concept bottom par : $Personne \sqcap \forall enfant.\perp$

D'autres langages, plus expressifs, peuvent être définis en rajoutant d'autres constructeurs au langage \mathcal{AL} à savoir :

- **Les restrictions de nombres**, notées par $\geq nR$ (restriction au moins) et par $\leq nR$ (restriction au plus) où n représente un entier positif.

Avec ces nouveaux constructeurs, nous pouvons par exemple décrire les personnes ayant soit pas plus d'un enfant soit au moins trois enfants dont un est une Femme, comme suit :

$$Personne \sqcap (\leq 1a - enfant \sqcup (\geq 3a - enfant \sqcap \exists a - enfant.Femme))$$

2.4.1.2 La logique de description \mathcal{EL}

Pour des raisons historiques, les premières logiques de description étudiées favorisaient l'utilisation de la restriction universelle à la restriction existentielle [6]. Pourtant la restriction universelle est un facteur important de complexité pour les problèmes de raisonnements même basiques. Ceci a motivé l'étude de \mathcal{EL} , dont l'objectif était de gérer de grandes ontologies en permettant le raisonnement "intra TBox" (satisfiabilité, subsomption de concepts...) en temps polynomial. De plus, son expressivité s'avère suffisante pour un certain nombre de cas pratiques importants tels que l'ontologie biomédicale SNOMED.

La TBox quant à elle peut contenir des inclusions de concepts de la forme :

$$C \sqsubseteq D \tag{2.1}$$

où C et D sont des concepts construits selon la règle suivante :

$$C ::= \top \mid A \mid C \sqcap D \mid \exists R.C \tag{2.2}$$

où A est un concept atomique, C et D des concepts construits et R un rôle.

Il existe un nombre important d'extensions de \mathcal{EL} dont la plus fréquemment rencontrée pour l'interrogation d'une base de connaissances est \mathcal{ELHI} . \mathcal{ELHI} étend \mathcal{EL} en autorisant l'utilisation des rôles inverses et l'inclusion de rôles dans la TBox $R \sqsubseteq S$ pour l'inclusion du rôle R dans le rôle S, qui signifie que deux individus en relation par le rôle R le sont aussi par le rôle S.

2.4.1.3 La logique de description \mathcal{ALC}

\mathcal{ALC} est obtenue à partir de \mathcal{EL} par l'ajout de la négation complète. De ce fait, \mathcal{ALC} permet de construire des unions de concepts $C \sqcup D$, des restrictions de valeurs et permet aussi d'exprimer les concepts \perp et \top . Dans \mathcal{ALC} , le test de subsomption et la reconnaissance d'instances peuvent se réduire via la négation au test de satisfiabilité qui a été montré EXPTIME-complet.

\mathcal{ALC} est souvent considéré comme la logique de description prototype parce qu'elle correspond à un fragment de la logique du premier ordre facile à identifier, et aussi parce qu'elle

est une variante syntaxique de la logique modale de base \mathcal{K} . Par exemple l'information suivante : "les départements d'enseignement ne peuvent être dirigés que par des professeurs et seuls des professeurs ou des chercheurs peuvent enseigner à des étudiants de Master " peut-être présenter en \mathcal{ALC} sous la forme des deux axiomes d'inclusion suivants :

$$\begin{aligned} & \exists \text{Dirige}. \text{DeptEnseignement} \sqsubseteq \text{Professeur} \\ & \exists \text{Enseigne}. \text{A}. \text{EtudiantMaster} \sqsubseteq \text{Professeur} \sqcup \text{Chercheur}. \end{aligned}$$

2.4.2 Services d'inférences standards

Les logiques de description fournissent des services de raisonnement. Dans différents domaines d'application, il peut y avoir différents types de connaissances implicites que nous voulons les déduire à partir des connaissances explicites dans une base de connaissances. Ces différents types d'inférence sont formalisés comme étant un problème de raisonnement dont la résolution nécessite le développement et l'implémentation d'algorithmes adéquats. Un système de logique de description stocke non seulement des terminologies et des assertions, mais offre également les services d'inférence. Les services d'inférence sont réalisés aussi bien sur la TBox que sur la ABox.

Étant donné \mathcal{T} une TBox, C et D deux concepts, les tâches typiques du raisonnement sur \mathcal{T} comportent :

- **La subsomption** : Soient C, D des descriptions de concept. $D \sqsubseteq C$, par rapport à une TBox $\Delta_{\mathcal{T}}$ si et seulement si $C^I \subseteq D^I$ pour tout modèle I . C, D sont équivalentes, par rapport à une TBox $\Delta_{\mathcal{T}}$ si et seulement si $C \sqsubseteq D$ et $D \sqsubseteq C$.
- **La satisfiabilité** : Soit C une description de concept. C est satisfiable par rapport à une TBox $\Delta_{\mathcal{T}}$ si et seulement si il existe un modèle I tel que $C^I \neq \emptyset$.
Il y a une relation étroite entre les inférences de subsomption et de satisfiabilité. Effectivement, le problème de satisfiabilité peut être réduit au problème de subsomption. Une description de concept C est insatisfiable si et seulement si $C \sqsubseteq \perp$.
- **La vérification d'équivalence entre deux concepts** : Deux concepts C et D sont équivalents, $C \equiv D$, par rapport à \mathcal{T} si et seulement si $C^I \equiv D^I$ pour tout modèle I de la TBox \mathcal{T} . Dans ce cas, nous notons $C \equiv_{\mathcal{T}} D$ ou $\mathcal{T} \models C \equiv D$.
- **La vérification de disjonction entre deux concepts** : Deux concepts C et D sont disjoints, écrit $C \neq D$, par rapport à une TBox \mathcal{T} si et seulement si $C^I \cap D^I = \emptyset$ pour tous les modèles I de la TBox \mathcal{T} .

Le raisonnement sur une ABox se focalise sur le test de la correction d'un modèle du domaine. Il permet effectuer les deux tâches suivantes :

- **La vérification d'instance** Soient $\Delta_{\mathcal{T}}$ une TBox et C une description de concept. Soient Δ_A une ABox et a un individu. L'individu a est une instance de C par rapport à la TBox et à la ABox si seulement si $a^I \in C^I$ pour tous les modèles I de $\Delta_{\mathcal{T}}$ et de Δ_A .
- **La vérification de consistance** Une ABox \mathcal{A} est consistante par rapport à une TBox \mathcal{T} , s'il existe une interprétation qui est un modèle de la ABox \mathcal{A} et la Tbox \mathcal{T} .

La satisfiabilité d'une base de connaissances est l'une des tâches de raisonnement les plus importantes. De nombreuses autres tâches, telles que l'implication d'un axiome ou la satisfiabilité d'un concept, entre autres peuvent être réduites au test de satisfiabilité d'une base de connaissances. Une tâche de raisonnement approfondie largement utilisée, et non réductible à la satisfiabilité d'une base de connaissances, est la réponse aux requêtes. Elle consiste à interroger une base de connaissances afin de récupérer les données souhaitées. Nous nous intéressons souvent à une forme très utilisée de requêtes appelées requêtes conjonctives. Nous reviendrons dans le chapitre suivant sur la notion de requêtes conjonctives et leur traitement dans le cadre de l'extension possibiliste quantitative des logiques DL-Lite, proposée dans cette thèse.

2.4.3 L'inconsistance dans la logique de description

Les ontologies problématiques se répartissent en deux catégories : inconsistantes et incohérentes.

- Une ontologie est inconsistante si elle n'a pas de modèle.
- Une ontologie est incohérente si TBox contient au moins un concept insatisfaisable, c'est-à-dire un concept qui est interprété comme l'ensemble vide dans tous les modèles du TBox.

Pour réparer des ontologies inconsistantes, un certain nombre de techniques ont été proposées. En particulier, dans [81], les auteurs ont étudié la réponse aux requêtes consistantes sur les ontologies DL-Lite en fonction de la technique de réparation [21], en utilisant des réparations de suppressions minimales.

Le travail présenté dans [100] est basé sur la suppression d'assertions de la ABox. La restauration de la consistance d'une ontologie est réalisée en identifiant et en éliminant les ensembles de conflits minimaux. Un autre travail récent [93] propose une approche pour une détection efficace de l'inconsistance dans les bases de connaissances DL-Lite distribuées. Il s'appuie sur la génération des requêtes dans le paramètre distribué. Les requêtes de conflit constituent essentiellement des conflits considérés par Lembo et al [93].

Une grande quantité de recherches a été consacrée à l'étude des incohérences dans les ontologies en général, sans distinguer la partie de données de la TBox. Le travail dans [72] propose une technique pour restaurer la consistance des ontologies inconsistantes en générant des sous-ensembles inconsistants minimaux. Une approche similaire rapportée dans [61] fournit des méthodologies pour extraire des sous-ontologies minimales inconsistantes et maximales consistantes. Les auteurs de [69] traitent de l'identification des dérivations de contradiction à l'aide de règles de contrainte d'intégrité définies dans un programme logique. Pour résoudre les inconsistances détectées, ils génèrent tous les candidats minimaux inactifs résolus.

En ce qui concerne les ontologies incohérentes, une tentative décrite dans [102] utilise des logiques de défauts pour relâcher les axiomes qui provoquent une incohérence et montre comment les logiques de description probabilistes peuvent être utilisées pour résoudre les conflits. D'autres approches antérieures pour déboguer des terminologies et résoudre des incohérences ont été proposées dans [102]. Les techniques les plus importantes pour le raisonnement paraconsistant dans les ontologies sont basées sur l'utilisation de valeurs de vérité supplémentaires. Cette sémantique multi-valeurs est définie pour les logiques de description. Dans [3], les auteurs suggèrent d'appliquer des requêtes uniquement sur un fragment consistant de l'ontologie. L'identification des fragments consistants est basée sur une fonction de

sélection, qui peut être définie par une pertinence syntaxique ou sémantique.

2.5 La famille DL-Lite

La famille DL-Lite a été introduite dans [5]. Elle a été proposée pour exprimer des ontologies simples mais bien utiles. En effet, la famille DL-Lite permet d'atteindre un équilibre permettant d'assurer un niveau d'expressivité satisfaisant tout en conservant une complexité basse pour les tâches d'interrogation, notamment la réponse à des requêtes complexes comme requêtes conjonctives, ainsi que pour les tâches de raisonnement comme la vérification de la subsumption de concepts ou la satisfiabilité de la base de connaissances.

Comme dans le cas des logiques de descriptions standards, pour représenter la connaissance en DL-Lite, nous utilisons un couple composé des ensembles suivants : une ABox représentant les connaissances spécifiques aux individus et une TBox représentant les connaissances terminologiques, générales à tous les individus. Les logiques de description légères manipulent les trois notions de base qui sont les individus, les rôles, et les concepts. Les individus sont décrits à l'aide de concepts et de rôles pour représenter la base de faits et constituent la ABox. Un concept représente une catégorisation d'un individu tandis que le rôle représente une relation entre deux individus.

2.5.1 Syntaxe et sémantique

La famille DL-Lite_{core} constitue le noyau de tous les autres membres de la famille DL-Lite [35, 34, 33]. Elle permet d'exprimer une négation très restreinte puisqu'elle ne peut porter que sur un rôle ou un concept basique et ne peut apparaître qu'en partie droite d'une inclusion. De plus, seule la restriction existentielle non qualifiée de rôles basiques est autorisée. Il existe plusieurs membres dans la famille DL-Lite dont les plus simples sont DL-Lite_{core} et DL-Lite_R permettant de constituer la base du langage du web sémantique OWL 2 QL.

En logique DL-Lite_{core}, le schéma conceptuel est décrit en termes de concepts, de rôles et d'attributs (valeurs de propriétés de concepts), avec les notations suivantes :

$$\begin{array}{ll} B \rightarrow A | \exists R & C \rightarrow B | \neg B \\ R \rightarrow P | P^- & E \rightarrow R | \neg R \end{array} \quad (2.3)$$

où A est un concept atomique (c'est à dire un concept désigné par un nom), B un concept basique, C un concept général. P un rôle atomique, P^- est l'inverse d'un rôle atomique P , E un rôle basique et R un rôle général.

Une base de connaissances DL $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ représente le domaine d'intérêt en termes de deux composants, une TBox \mathcal{T} et une ABox \mathcal{A} . La terminologie Tbox \mathcal{T} dans DL-Lite_{core} est formée d'un ensemble fini d'assertions d'inclusion de la forme :

$$B \sqsubseteq C$$

Autrement dit, des concepts généraux sont autorisés à se produire sur le côté droit des assertions d'inclusion, alors que seuls les concepts de base peuvent se produire sur le côté gauche des assertions d'inclusion.

La composante d'assertions ABox dans la logique DL-Lite_{core}, est formée d'un ensemble fini d'assertions de la forme :

$A(a)$ et $P(a, b)$

L'objet désigné par la constante a est une instance du concept A et la paire d'objets désignée par la paire de constantes (a, b) est une instance du rôle P .

Parmi les extensions de $\text{DL-Lite}_{\text{core}}$, deux extensions particulières sont à distinguer, à savoir DL-Lite_R et DL-Lite_F . Ces sont deux dialectes de DL-lite qui diffèrent sur les axiomes supplémentaires autorisés.

La logique DL-Lite_R étend la logique $\text{DL-Lite}_{\text{core}}$ par la possibilité de spécifier dans la TBox des axiomes d'inclusion de rôles de la forme :

$$R \sqsubseteq E$$

La logique DL-Lite_F étend la logique $\text{DL-Lite}_{\text{core}}$ par la possibilité de spécifier dans la TBox des rôles fonctionnels de la forme :

$$(\text{funct } R)$$

Il est important de noter que les concepts et les rôles généraux sont seulement autorisés en partie droite des inclusions. Les axiomes de fonctionnalité ne peuvent également porter que sur des rôles de base. La partie gauche est constituée seulement de concepts et de rôles de base.

Il a été montré dans [35] que le test de subsomption est polynomial pour des Tboxes de DL-lite_R ou de DL-lite_F . DL-lite_R a été récemment incorporée dans la version OWL2 (W3C, 2009) de OWL en tant que profil appelé OWL2 QL. Ce profil est une extension de la partie schéma de RDFS (W3C, 2004a).

Exemple 2.2 Soit l'ontologie $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ donnée ci-dessous :

Nous considérons *Superviseur* et *Doctorant* deux concepts qui contiennent intuitivement la liste des superviseurs et des étudiants de doctorat respectivement. Nous considérons *est_Superviseur_de* un rôle qui répertorie les enseignants qui encadrent des doctorants. a , b et h sont des individus.

$$\mathcal{T} = \{ \text{Superviseur} \sqsubseteq \neg \text{Doctorant}, \exists \text{est_Superviseur_de} \sqsubseteq \text{Superviseur}, \\ \exists \text{est_Superviseur_de}^- \sqsubseteq \text{Doctorant} \}.$$

$$\mathcal{A} = \{ \langle \text{Doctorant}(b), \text{Superviseur}(a), \\ \text{est_Superviseur_de}(h, a), \text{Superviseur}(b) \rangle \}.$$

L'axiome $\text{Superviseur} \sqsubseteq \neg \text{Doctorant}$ déclare qu'un superviseur ne peut jamais être un doctorant. L'assertion $\text{est_Superviseur_de}(h, a)$ signifie que le superviseur h encadre le doctorant a .

La sémantique d'une expression $\text{DL-Lite}_{\text{core}}$ est donnée en termes d'interprétations. Une interprétation $I = (\Delta^I, \cdot^I)$ est formée du domaine d'interprétation non vide Δ^I et d'une fonction d'interprétation \cdot^I . La fonction \cdot^I associe un sous-ensemble de Δ^I à chaque concept ou domaine de valeurs, et un sous ensemble de $\Delta^I \times \Delta^I$ à chaque rôle ou attribut.

La sémantique d'une TBox est définie comme suit : une interprétation I est un modèle d'une inclusion $C \sqsubseteq D$, si $C^I \subseteq D^I$. D'une manière analogue, I est un modèle d'une inclusion de rôles $R \sqsubseteq E$ si $R^I \subseteq E^I$. Une interprétation $I = (\Delta^I; \cdot^I)$ est un modèle d'un axiome de fonctionnalité (*funct* R) si la relation binaire R^I est une fonction, c'est-à-dire : $(o, o_1) \in R^I$ et $(o, o_2) \in R^I$ implique $o_1 = o_2$. On dit que l'interprétation I est un modèle d'une TBox si I est un modèle de tous ses axiomes.

De même, la sémantique d'une ABox est définie comme suit : Une interprétation I est un modèle d'une assertion $C(a)$ (resp. $P(a, b)$) $I \models C(a)$ (resp. $I \models P(a, b)$), si $a^I \in C^I$ (resp. $(a^I, b^I) \in P^I$). Une interprétation est un modèle d'une ABox si elle est un modèle de toutes ses assertions.

Enfin, une interprétation I est un modèle d'une base de connaissances $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ ($I \models \mathcal{K}$), si elle est modèle de \mathcal{T} et de \mathcal{A} . \mathcal{K} est dite satisfiable si elle a au moins un modèle.

2.5.2 Fermeture négative dans DL-Lite

Dans DL-Lite, une TBox \mathcal{T} d'une base de connaissances $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ peut être considérée comme composée des axiomes d'inclusion positifs (de la forme $C \sqsubseteq D$), notée \mathcal{T}_p , et des axiomes d'inclusion négatifs (de la forme $C \sqsubseteq \neg D$), notée \mathcal{T}_n , soit $\mathcal{T} = \mathcal{T}_p \cup \mathcal{T}_n$. Un concept important dans DL-Lite pour identifier les éventuels conflits et répondre aux requêtes est celui de la fermeture négative d'une TBox.

La fermeture négative d'une TBox \mathcal{T} , notée par $Cln(\mathcal{T})$, combine \mathcal{T}_p et \mathcal{T}_n pour dériver tous les axiomes d'inclusion négatifs implicites à partir de \mathcal{T}_p et \mathcal{T}_n . $Cln(\mathcal{T})$ est obtenu en appliquant à plusieurs reprises les règles suivantes jusqu'à atteindre un point fixe [5] :

Règle 2.1 *Tous les axiomes négatifs dans \mathcal{T} sont dans $Cln(\mathcal{T})$.*

Règle 2.2 *Si $\langle B1 \sqsubseteq B2 \rangle$ est dans \mathcal{T} et $\langle B2 \sqsubseteq \neg B3 \rangle$ est dans $Cln(\mathcal{T})$, alors ajouter $\langle B1 \sqsubseteq \neg B3 \rangle$ à $Cln(\mathcal{T})$.*

Règle 2.3 *Si $\langle B1 \sqsubseteq B2 \rangle$ est dans \mathcal{T} et $\langle B3 \sqsubseteq \neg B2 \rangle$ est dans $Cln(\mathcal{T})$, alors ajouter $\langle B1 \sqsubseteq \neg B3 \rangle$ à $Cln(\mathcal{T})$.*

Règle 2.4 *Si $\langle R1 \sqsubseteq R2 \rangle$ est dans \mathcal{T} et $\langle \exists R2 \sqsubseteq \neg B \rangle$ est dans $Cln(\mathcal{T})$ ou $\langle B \sqsubseteq \neg \exists R2 \rangle$ est dans $Cln(\mathcal{T})$, alors ajouter $\langle \exists R1 \sqsubseteq \neg B \rangle$ à $Cln(\mathcal{T})$.*

Règle 2.5 *Si $\langle R1 \sqsubseteq R2 \rangle$ est dans \mathcal{T} et $\langle \exists R_2^- \sqsubseteq \neg B \rangle$ est dans $Cln(\mathcal{T})$ ou $\langle B \sqsubseteq \neg R_2^- \rangle$, alors ajouter $\langle R_1^- \sqsubseteq \neg B \rangle$ à $Cln(\mathcal{T})$.*

Règle 2.6 *Si $\langle R1 \sqsubseteq R2 \rangle$ est dans \mathcal{T} et $\langle R2 \sqsubseteq \neg R3 \rangle$ est dans $Cln(\mathcal{T})$ ou $\langle R3 \sqsubseteq \neg R2 \rangle$, alors ajouter $\langle R1 \sqsubseteq \neg R3 \rangle$ à $Cln(\mathcal{T})$.*

Règle 2.7 *Dans le cas où \mathcal{T} est dans $DL-Lite_R$, lorsque l'un des axiomes $\langle \exists R \sqsubseteq \neg \exists R \rangle$, $\langle \exists R^- \sqsubseteq \neg \exists R^- \rangle$ ou $\langle R \sqsubseteq \neg R \rangle$ est dans $Cln(\mathcal{T})$, alors ajouter les trois axiomes à $Cln(\mathcal{T})$. Dans le cas où \mathcal{T} est dans $DL-Lite_F$, lorsque l'un des axiomes $\langle \exists R \sqsubseteq \neg \exists R \rangle$ ou $\langle \exists R^- \sqsubseteq \neg \exists R^- \rangle$ est dans $Cln(\mathcal{T})$, alors ajouter les deux axiomes à $Cln(\mathcal{T})$.*

Ces règles sont principalement des règles de transitivité entre les axiomes positifs et les axiomes négatifs .

Exemple 2.3 *Considérons la base de connaissances DL-Lite \mathcal{K} composée de la TBox \mathcal{T} et la ABox \mathcal{A} suivantes :*

$$\mathcal{T} = \{B \sqsubseteq \neg C, A \sqsubseteq C, F \sqsubseteq B\} \text{ et}$$

$$\mathcal{A} = \{A(b), B(a), C(a), F(c), B(b)\}.$$

La fermeture négative $Cln(\mathcal{T})$ de \mathcal{T} est la suivante :

$$Cln(\mathcal{T}) = \{B \sqsubseteq \neg C, A \sqsubseteq \neg B, F \sqsubseteq \neg C\}.$$

2.5.3 Calcul des conflits

La fermeture négative joue un rôle crucial pour vérifier la consistance d'une base de connaissances DL-Lite, pour la réponse aux requêtes et pour le calcul des conflits.

Il a été montré dans [5] que $\langle \mathcal{T}, \mathcal{A} \rangle$ est consistante si et seulement si $\langle \text{Cln}(\mathcal{T}), \mathcal{A} \rangle$ est consistant. Par conséquent, pour vérifier la consistance, la TBox \mathcal{T} peut être remplacée par sa fermeture négative $\text{Cln}(\mathcal{T})$.

Un résultat similaire concerne le calcul des conflits où $\langle \text{Cln}(\mathcal{T}), \mathcal{A} \rangle$ peut être utilisée au lieu de $\langle \mathcal{T}, \mathcal{A} \rangle$.

Soit $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ une base de connaissances DL-Lite. Un conflit est un sous-ensemble \mathcal{C} minimal de l'ABox \mathcal{A} qui est inconsistant avec le TBox \mathcal{T} . Plus précisément :

- $\langle \mathcal{T}, \mathcal{C} \rangle$ est inconsistant
- $\forall \mathcal{C}' \subset \mathcal{C}, \langle \mathcal{T}, \mathcal{C}' \rangle$ est consistant.

Il a été montré dans [5] que tous les conflits possibles dans une base de connaissances DL-Lite sont au plus binaires, ils sont composés d'au plus de deux assertions ABox (ils sont de taille 1 seulement dans le cas où la Tbox \mathcal{T} est incohérent, c'est-à-dire qu'elle admet au moins un concept vide). Une fonctionnalité intéressante utilisant les langues DL-Lite est le fait que le calcul de l'ensemble de tous les conflits est polynomial.

L'ensemble des conflits est calculé à partir de la fermeture négative. Pour chaque $A \sqsubseteq \neg B \in \text{Cln}(\mathcal{T})$, l'interrogation de la Abox \mathcal{A} se fait en recherchant des paires $(A(x), B(x))$. Plus précisément, l'ensemble des conflits est donné par :

$$\zeta = \{(A(x), B(x)) \mid A(x) \in \mathcal{A}, B(x) \in \mathcal{A}, A \sqsubseteq \neg B \in \text{Cln}(\mathcal{T})\}$$

Exemple 2.4 *Considérons la base de connaissances \mathcal{K} de l'Exemple 2.3. Les ensembles des conflits sont calculés comme suit :*

A partir de : $A(b) \in \mathcal{A}, B(b) \in \mathcal{A}$ et $A \sqsubseteq \neg B \in \text{Cln}(\mathcal{T})$, nous obtenons $\mathcal{C}_1 = \{\langle A(b), B(b) \rangle\}$, et a partir de : $B(a) \in \mathcal{A}, C(a) \in \mathcal{A}$ et $B \sqsubseteq \neg C \in \text{Cln}(\mathcal{T})$, nous obtenons $\mathcal{C}_2 = \{\langle B(a), C(a) \rangle\}$.

Nous avons un autre axiome négatif $F \sqsubseteq \neg C$ mais les deux assertions $C(a), F(c)$ impliquent différents individus.

Par conséquent, l'ensemble des conflits est $\zeta = \{\mathcal{C}_1, \mathcal{C}_2\}$

2.6 Extensions existantes de la logiques de description

Au cours des dernières décennies, le traitement de l'incertitude sur le Web sémantique est devenu un problème important. Comme motionnée dans la chapitre 1, la logique floue, la logique possibiliste et la logique basée sur la théorie des probabilités constituent les principales modélisations de l'incertitude. Dans cette section, nous rappelons les principales approches proposées dans la littérature pour la gestion de l'incertitude des logiques de description.

La logique de description probabiliste

De nombreuses approches ont été proposées pour étendre les logiques de description avec un raisonnement probabiliste, telle que P-SHOIN(D) qui est la logique de description probabiliste la plus expressive [56, 87, 89]. Sa syntaxe utilise la notion de contrainte conditionnelle probabiliste pour exprimer les connaissances probabilistes en plus des axiomes [84]. Une

contrainte conditionnelle est une expression de la forme $(D|C)[l, u]$. Elle représente une relation d'inclusion entre deux concepts C et D avec une probabilité entre l et u , où l et u sont des nombres réels appartenant à l'intervalle $[0, 1]$.

Chaque base de connaissances probabiliste se compose d'un ensemble fini d'axiomes d'une TBox probabiliste. Cette dernière se compose d'une TBox standard et d'un ensemble fini de contraintes conditionnelles définies sur des individus probabilistes. Sa sémantique est basée sur la notion d'implication lexicographique définie dans le cadre du raisonnement probabiliste par défaut [85, 86].

Dans [63, 71, 52], une extension probabiliste de la logique de description *ALC*, basée sur la notion d'implication logique probabiliste, a été proposée. Cette extension permet de représenter une connaissances probabiliste terminologique des concepts et des rôles. Les principaux services de raisonnement considérés, y compris la vérification de la cohérence et l'implication lexicographique pour les bases de connaissances probabilistes, sont également basés sur des notions de cohérence et d'implication lexicographique telles que définies dans le raisonnement probabiliste par défaut [86].

La logique de description floue

Le travail sur l'extension floue des langages d'ontologie a également reçu beaucoup d'attention. L'un des premiers travaux dans ce domaine a été proposé par Yen dans [122]. Ensuite, Straccia [115] a introduit un raisonnement assertionnel et a fourni des algorithmes de raisonnement pour plusieurs tâches de raisonnement [116]. Il a également proposé une extension floue de DL-SHOIN(D) [118].

Dans [112], les auteurs ont proposé des extensions floues des langage OWL et OWL 2 [114]. En outre, ils ont fourni des algorithmes de raisonnement pour certains DLs expressifs, tels que SHIN [113]. Des réductions pour les DLs découpés ont également été envisagées dans [24, 23, 25, 26, 117]. Il est bien connu que différentes familles d'opérateurs flous conduisent à des DLs flous avec des propriétés différentes [111]. D'un point de vue sémantique, les logiques de description floues se rapportent directement au travail de Zadeh sur la théorie des ensembles flous [124].

La logique de description possibiliste

En revanche, il y a relativement peu de travaux sur la combinaison de la logique possibiliste et de la logique de description. Toutes les extensions possibilistes existantes sont basées sur la théorie de possibilité qualitative (basée sur le min). Elles ont été proposées par Hollunder [66], Dubois et al. [44] puis développées par Qi et Pan dans [99] qui a également fourni des algorithmes pour plusieurs services d'inférence.

Dans [99], les auteurs ont défini la syntaxe et la sémantique de la logique de description possibiliste. Ils ont attaché à chaque axiome et assertion un degré de certitude qui est utilisé par la suite pour calculer le degré d'inconsistance de la base de connaissances DL possibiliste. Ensuite, ils ont proposé deux services d'inférence pour les logiques de description possibles [98]. Plus récemment, dans [13], les auteurs ont proposé une extension des principaux fragments de DL-Lite. Ils ont étudié la question de révision de bases de connaissances DL-Lite possibilistes et ils ont également proposé un opérateur de fusion assertionnel basé sur le min pour traiter le cas où les assertions de la ABox proviennent de plusieurs sources d'information [14].

2.7 Conclusion

Dans la première partie de ce chapitre, nous avons introduit la définition du web sémantique et les ontologies qui ont pris une place centrale dans les applications basées sur les connaissances. Ensuite, nous avons présenté une grande famille de formalismes permettant la représentation d'ontologies formelles et le raisonnement avec ces ontologies (les logiques de description). Puis, nous nous sommes focalisés sur la famille DL-Lite qui sont des logiques légères (à basse complexité) conçues pour les applications manipulant de gros volumes de données où l'interrogation est la tâche de raisonnement la plus importante. Enfin nous avons présenté brièvement les principales extensions des logiques de description développées pour tenir compte de l'incertitude dans les bases de connaissances.

Le chapitre suivant étudie la nouvelle extension appelée la logique DL-Lite possibiliste basée sur le produit, désignée par $Pb-\pi$ -DL-Lite.

Deuxième partie

Modèle logique pour extension DL-lite

Introduction de la deuxième partie

Les ontologies jouent un rôle important pour le succès du web sémantique. Elles peuvent être modélisées par les logiques de description. Traiter l'incertitude dans la logique de description et en particulier dans la famille DL-Lite est un sujet de recherche important. Récemment, plusieurs travaux ont été développés proposant de combiner les théories de l'incertain avec les logiques de description.

A notre connaissance, il n'existe pas de travaux qui combinent la logique de description avec la théorie des possibilités quantitative. Dans cette partie, nous proposons une nouvelle extension des principaux fragments de DL-Lite pour faire face à l'incertitude associée aux axiomes en utilisant le cadre de la théorie des possibilités basée sur le produit.

Chapitre 3

Les fondements de la logique Pb- π -DL-Lite

3.1 Introduction

La dynamique de l'information disponible sur le Web conduit naturellement à une évolution continue des ontologies et à un besoin permanent de les fusionner ou de les aligner. En conséquence, nous sommes souvent confrontés à l'incertitude dans l'information utilisée. Il est alors essentiel d'étendre DL-Lite afin de représenter et de gérer des informations incertaines. Plusieurs approches ont été développées dans la littérature pour étendre les logiques de description avec le raisonnement sur l'incertitude, y compris : les logiques de description probabilistes [56, 87, 89, 84, 85, 86, 63, 71, 52], les logiques de description possibilistes [66, 44, 99, 13] et les logiques de description floues [122, 115, 116, 112, 114, 113]. Plusieurs alternatives ont été proposées pour chacune de ces extensions.

Toutes les extensions existantes des logiques de description possibilistes sont basées sur la théorie des possibilités qualitative (basée sur le min). Elles ont d'abord été proposées par Hollunder [66], Dubois et al. [44] puis développées par Qi et Pan dans [99] qui ont également fourni des algorithmes pour mettre au point plusieurs services d'inférence. Plus récemment, dans [13], les auteurs ont proposé une généralisation des principaux fragments de DL-Lite pour gérer l'incertitude dans le cadre de la théorie possibiliste qualitative. Ils ont montré que DL-Lite possibiliste qualitative min est une simple extension de DL-Lite standard, puisque les réponses aux requêtes conjonctives et le calcul des conflits se font toujours en un temps polynomial dans le nouveau cadre. Par ailleurs, les auteurs ont étudié la révision de DL-Lite possibiliste basée sur le min et ont également proposé un opérateur de fusion assertionnel pour traiter le cas où les assertions de la ABox proviennent de plusieurs sources d'information. [14].

Comme nous l'avons déjà indiqué précédemment, la théorie des possibilités a été proposée dans deux contextes : la théorie des possibilités qualitative (basée sur le min) et la théorie des possibilités quantitative (basée sur le produit) [49].

La théorie des possibilités basée sur le min est appropriée lorsque l'échelle d'incertitude encode seulement un ordre de plausibilité (un pré-ordre total) par rapport aux interprétations ou aux formules.

La théorie des possibilités basée sur le produit est appropriée lorsque les degrés d'incertitude représentent des degrés de surprise ou reflètent le résultat de la transformation d'une distri-

bution de probabilité en une distribution de possibilité. Par conséquent, dans le contexte de DL-Lite, si les degrés d'incertitude représentent un encodage plausible entre les assertions (un pre-ordre total), alors la logique DL-Lite possibiliste basée sur min est la mieux adaptée. En revanche, si le degré d'incertitude représente un degré de surprise dans le sens de Spohn des fonctions ordinales conditionnelles [109, 110] alors la distribution des possibilités basée sur le produit est plus appropriée [51].

En fait, dans [109], Spohn a introduit le concept de fonctions ordinales conditionnelles dans le contexte du changement de croyances. Les fonctions ordinales conditionnelles de Spohn comme les cadres théoriques possibles, offrent des alternatives intéressantes pour représenter des degrés d'incertitude qui ne représentent pas des probabilités exactes. Ces degrés d'incertitude peuvent représenter des degrés de surprise ou simplement un ordre sur les informations disponibles. Ils peuvent également représenter des probabilités qualitatives ou abstraites ou même des probabilités infinitésimales ([46, 2]).

Dans le contexte des ontologies incertaines et des assertions incertaines, il n'est pas toujours vrai que les degrés d'incertitude représentent des probabilités exactes. Par conséquent, dans ce contexte, l'utilisation de fonctions ordinales conditionnelles ou de cadres théoriques de possibilité ont également un sens et doivent être explorées.

Ce chapitre est organisé de la manière suivante. La section 3.2 discute la nouvelle extension DL-Lite possibiliste basée sur le produit, désignée par Pb- π -DL-Lite, où nous présentons sa syntaxe et sa sémantique. La section 3.3 décrit comment calculer le degré d'incohérence d'une base de connaissances Pb- π -DL-Lite. Dans la section 3.4, nous traitons le problème de la réponse aux requêtes dans Pb- π -DL-Lite et la section 3.5 présente le calcul de la fermeture négative pour une base de connaissances Pb- π -DL-Lite avec TBox certaine. La dernière section, présente une comparaison entre notre extension et les extensions déjà développées.

3.2 DL-Lite possibiliste basée sur le produit

Dans cette section, nous définissons la nouvelle extension DL-Lite possibiliste basée sur le produit, dénotée par Pb- π -DL-Lite. Nous commençons par présenter la syntaxe ensuite nous montrons comment générer la distribution de possibilités associée à une base de connaissances Pb- π -DL-Lite [15, 29, 16].

Dans cette thèse, nous nous focalisons sur le membre principal de la famille DL-Lite, à savoir, DL-Lite_{core}. Cependant, cette extension peut être étendue à tout autre fragment de DL-Lite vérifiant les deux exigences suivantes :

- i) Chaque conflit doit impliquer au plus deux assertions de la ABox et
- ii) l'ensemble des conflits doit être calculé en un temps polynomial.

En l'occurrence, les fragments DL-Lite_R et DL-Lite_F vérifient tous les deux les deux conditions précédentes.

3.2.1 La syntaxe de Pb- π -DL-Lite

La syntaxe de la DL-lite possibiliste basée sur le produit est représentée par la notion de base de connaissances Pb- π -DL-Lite, dénotée par \mathcal{K} .

Définition 3.1 *Une base de connaissances Pb- π -DL-Lite $\mathcal{K} = \{\langle \phi_i, \alpha_i \rangle : i = 1, \dots, n\}$ est un ensemble fini de formules possibilistes de la forme $\langle \phi_i, \alpha_i \rangle$, où ϕ_i est un axiome de la TBox ou une assertion de la ABox et α_i représente le degré de certitude de ϕ_i .*

Dans la définition 3.1, nous ne considérons que les connaissances qui sont au moins quelque peu certaines. Cela signifie que tous les degrés de certitude appartiennent à l'intervalle $]0, 1]$. La notion de base de connaissances DL-lite possibiliste est définie dans l'esprit de la logique possibiliste standard [48]. Dans cette extension, ϕ_i désigne une formule DL-Lite au lieu d'une formule logique propositionnelle.

Une base de connaissances Pb- π -DL-Lite est également représentée par un couple $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ où \mathcal{T} (resp. \mathcal{A}) est un ensemble d'axiomes de TBox (resp. d'assertions de ABox) avec degrés de certitudes non nuls.

Dans ce qui suit, nous considérons que la terminologie \mathcal{T} est certaine, ce qui signifie qu'il n'y a pas d'incertitude sur les relations générales entre les concepts et les rôles, mais uniquement sur les assertions de la ABox. Par conséquent, la base terminologique est supposée stable et ne doit pas être mise en doute en présence d'incohérences. En effet, il existe des contextes d'applications telles que OBDAO (Ontology-Based Data Access OBDA) ou OMQA (Ontology Mediated Query Answering), où cette hypothèse est tout à fait raisonnable [29]. Dans ces contextes, il existe une quantité énorme de données provenant de différentes sources. Ces données peuvent être incertaines. Cependant, la terminologie est de petite taille et elle est souvent considérée comme stable et incontestable.

Exemple 3.1 *Nous considérons les deux concepts Professeur et Etudiant qui contiennent intuitivement la liste des professeurs et des étudiants de doctorat respectivement. Nous considérons le rôle Enseigne qui répertorie les couples de la forme (enseignant, enseigné) et le rôle aTuteur qui regroupe les couples (doctorant, encadrant). Nous considérons a et b deux individus.*

Soit Pb- π -DL-Lite \mathcal{K} une base de connaissances composée d'une TBox \mathcal{T} et une ABox \mathcal{A} . Dans cet exemple, le choix des degrés de certitude est fait de façon arbitraire et ne reflète pas des vrais degrés de surprise issues d'un contexte réel. Cet exemple sera utilisé tout le long de la thèse.

La TBox \mathcal{T} contient les axiomes suivants :

$$\begin{aligned} &\langle \exists \text{Enseigne} \sqsubseteq \text{Professeur}, 1.0 \rangle, & \langle \exists \text{aTuteur} \sqsubseteq \text{Etudiant}, 1.0 \rangle, \\ &\langle \exists \text{Enseigne}^- \sqsubseteq \text{Etudiant}, 1.0 \rangle, & \langle \exists \text{aTuteur}^- \sqsubseteq \text{Professeur}, 1.0 \rangle, \\ &\langle \text{Professeur} \sqsubseteq \neg \text{Etudiant}, 1.0 \rangle. \end{aligned}$$

La base d'assertions \mathcal{A} contient les assertions suivantes :

$$\begin{aligned} &\langle \text{Professeur}(\text{John}), 0.95 \rangle, & \langle \text{Enseigne}(\text{Jean}, \text{John}), 0.87 \rangle, \\ &\langle \text{Etudiant}(\text{Jean}), 0.90 \rangle, & \langle \text{aTuteur}(\text{John}, \text{Jean}), 0.65 \rangle, \\ &\langle \text{Professeur}(\text{Marry}), 0.98 \rangle, & \langle \text{Enseigne}(\text{Jean}, \text{Marry}), 0.8 \rangle. \end{aligned}$$

L'axiome $\langle \text{Professeur} \sqsubseteq \neg \text{Etudiant}, 1.0 \rangle$ déclare qu'un professeur ne peut jamais être un étudiant avec un degré de certitude égal à 1. Le degré 1 signifie que l'information est complètement certaine.

Notons que si les degrés de certitude associés à la base d'assertion \mathcal{A} sont tous égaux à 1, alors la base de connaissances n'est rien d'autre qu'une base de connaissances DL-Lite classique.

3.2.2 La sémantique de Pb- π -DL-Lite

Comme dans la logique possibiliste standard, la sémantique de la base de connaissances Pb- π -DL-Lite $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ est définie par le concept d'une distribution de possibilité $\pi_{\mathcal{K}}$ sur l'ensemble de toutes les interprétations de Ω .

3.2.2.1 La distribution possibiliste

Une distribution de possibilités est une fonction de l'ensemble Ω vers une échelle totalement ordonnée qui a une valeur maximale et une valeur minimale. Cette échelle est l'intervalle unitaire $[0,1]$. Le degré de possibilité $\pi(I)$ d'une interprétation I représente le degré de compatibilité ou le degré de cohérence de l'interprétation I .

$\pi(I) = 1$ signifie que I est totalement possible, i.e., I est un modèle de chaque axiome de \mathcal{T} et chaque assertion de \mathcal{A} . Lorsque $\pi(I) = 0$, cela signifie que I est impossible et elle est totalement inconsistant avec la base de connaissances \mathcal{K} .

Une distribution possibiliste π est normalisée s'il existe au moins une interprétation totalement possible : $\exists I \in \Omega$ tel que $\pi(I) = 1$. Sinon, π est dite sous-normalisée. Dans ce cas, l'expression suivante définit le degré de cohérence h de π :

$$h(\pi) = \max_{I \in \Omega} \pi(I) \quad (3.1)$$

3.2.2.2 D'une base Pb- π -DL-Lite vers une distribution possibiliste

Dans cette sous section, nous montrons comment induire une distribution possibiliste sur les interprétations DL-Lite à partir d'une base de connaissances DL-Lite possibiliste basée sur le produit \mathcal{K} .

Nous commençons par le cas le plus simple, où la base d'assertions ABox ne contient qu'une assertion incertaine $\langle \phi, \alpha \rangle$.

Si une interprétation I falsifie un axiome terminologique du TBox \mathcal{T} , sous hypothèse que tous les axiomes de la TBox sont considérés comme complètement certains, alors I est considérée comme impossible $\pi(I) = 0$.

Supposons maintenant que I satisfait tous les éléments de \mathcal{T} , alors, nous distinguons deux cas :

- I satisfie ϕ . Dans ce cas, I est complètement consistante et $\pi_\phi(I) = 1$.
- I falsifie ϕ , alors $\pi_\phi(I) = 1 - \alpha$.

La fonction d'échelle inversée exprime le fait que plus ϕ est certaine, I est moins plausible. Plus formellement :

$$\pi_\phi(I) = \begin{cases} 1 & \text{si } I \models \phi \text{ et } \forall \langle \varphi, 1 \rangle \in \mathcal{T}, I \models \varphi. \\ 0 & \text{si } \exists \langle \varphi, 1 \rangle \in \mathcal{T}, I \not\models \varphi \\ 1 - \alpha & \text{sinon} \end{cases} \quad (3.2)$$

Dans le cas général, la ABox $\mathcal{A} = \{\langle \phi_i, \alpha_i \rangle | i = 1 \dots n\}$ peut contenir plus d'une assertion incertaine. Une façon de définir $\pi_{\mathcal{K}}(I)$ est de combiner les distributions de possibilités associées à chaque assertion $\pi_{\phi_i}(I)$ de \mathcal{A} , à savoir :

$$\pi_{\mathcal{K}}(I) = \bigoplus_{\langle \phi_i, \alpha_i \rangle \in \mathcal{A}} \pi_{\phi_i}(I). \quad (3.3)$$

L'opérateur \bigoplus est une opération de fusion. L'opérateur de fusion \bigoplus choisi doit satisfaire les deux exigences minimales suivantes :

- i) si $\forall \langle \phi_i, \alpha_i \rangle \in \mathcal{A}$, $\pi_{\phi_i}(I) = 1$ alors $\pi_{\mathcal{K}}(I) = 1$ et
- ii) si $\exists \langle \phi_i, \alpha_i \rangle \in \mathcal{A}$, s.t. $\pi_{\phi_i}(I) = 0$ alors $\pi_{\mathcal{K}}(I) = 0$.

Une exigence supplémentaire est la propriété dite *strict-pareto* définie comme suit :

Supposons que $\forall \langle \phi_i, \alpha_i \rangle \in \mathcal{A}$, $\pi_{\phi_i}(I) \neq 0$ et $\pi_{\phi_i}(I') \neq 0$, où I et I' sont deux interprétations. Alors, une opération de fusion \bigoplus satisfait la propriété *strict-pareto-ordering* :

Si $\forall(\phi_i, \alpha_i), \pi_{\phi_i}(I) \geq \pi_{\phi_i}(I')$ et $\exists(\phi_i, \alpha_i), \pi_{\phi_i}(I) > \pi_{\phi_i}(I')$ alors $\pi_{\mathcal{K}}(I) > \pi_{\mathcal{K}}(I')$.

Dans cette thèse, nous considérons $\bigoplus = \prod$ qui satisfait les trois propriétés ci-dessus. La distribution possibiliste $\pi_{\mathcal{K}}(I)$ est définie alors comme suit :

Définition 3.2 *Pour tout $I \in \Omega$,*

$$\pi_{\mathcal{K}}(I) = \begin{cases} 1 & \text{si } \forall \langle \phi_i, \alpha_i \rangle \in \mathcal{K}, I \models \phi_i \\ \prod_{\langle \phi_i, \alpha_i \rangle \in \mathcal{K}, I \not\models \phi_i} (1 - \alpha_i) & \text{sinon} \end{cases} \quad (3.4)$$

où \models est la relation de satisfaction entre une interprétation DL-Lite et une formule DL-Lite.

La distribution possibiliste définie par l'Equation 3.4 représente la distribution possibiliste la moins spécifique (la plus prudente) satisfaisant \mathcal{K} .

Exemple 3.2 *Considérons la base de connaissances DL-Lite \mathcal{K} composée de la TBox \mathcal{T} et la ABox \mathcal{A} .*

$$\mathcal{T} = \{ \langle B \sqsubseteq \neg C, 1 \rangle, \langle A \sqsubseteq \neg C, 1 \rangle, \} \text{ et}$$

$$\mathcal{A} = \{ \langle A(a), \alpha_1 \rangle, \langle B(b), \alpha_2 \rangle, \langle C(a), \alpha_3 \rangle, \langle C(b), \alpha_4 \rangle \}.$$

Nous prenons l'interprétation I sur le domaine $\Delta = \{a, b\}$. avec : $.^I : A^I = \{a\}, B^I = \{a\}, C^I = \{b\}$.

D'après l'Equation 3.2, nous avons :

- $\pi_{A(a)}(I) = 1$
- $\pi_{B(b)}(I) = 1 - \alpha_2$
- $\pi_{C(a)}(I) = 1 - \alpha_3$
- $\pi_{C(b)}(I) = 1$

La distribution possibiliste $\pi_{\mathcal{K}}(I)$ est donnée alors par :

$$\pi_{\mathcal{K}}(I) = \prod_{\langle \phi_i, \alpha_i \rangle \in \mathcal{A}} \pi_{\phi_i}(I) = (1 - \alpha_2) * (1 - \alpha_3).$$

Dans la théorie des possibilités [47], il existe différentes interprétations de l'échelle d'incertitude. Une interprétation possible des degrés de certitude est de les considérer comme des "degrés de surprise" comme c'est le cas dans les fonctions ordinales conditionnelles de Spohn [109, 110].

Dans ce contexte, les degrés d'incertitude associés aux assertions prennent leurs valeurs sur une échelle spécifique de la forme $\{0, 1 - k^{-1}, 1 - k^{-2}, \dots, 1\}$ où k représente un nombre entier positif. En théorie des probabilités, k est défini comme étant infinitésimal dans une échelle de probabilité spécifique.

Les degrés de possibilité des interprétations $\pi(I)$ prennent leurs valeurs dans l'ensemble $\{1, k^{-1}, k^{-2}, \dots, 0\}$. Dans ce contexte, $\pi(I) = 1$ correspond à la situation normale. $\pi(I) = k^{-1}$ représente la première situation surprenante. Le degré de surprise est par nature cumulatif (ou additif) : plus les assertions sont falsifiées par une certaine interprétation, plus l'interprétation sera surprenante. Lorsque les degrés d'incertitude associés aux assertions prennent leurs valeurs sur l'échelle d'incertitude $\{0, 1 - k^{-1}, 1 - k^{-2}, \dots, 1\}$, nous pouvons facilement vérifier que le degré de possibilité associé à une interprétation est de la forme :

$$\begin{aligned}
\pi(I) &= \prod_{\langle \phi, 1-k^{-i} \rangle \in \mathcal{A}, I \neq \phi} (1 - (1 - k^{-i})). \\
&= \prod_{\langle \phi, 1-k^{-i} \rangle \in \mathcal{A}, I \neq \phi} (k^{-i}). \\
&= k^{-\sum_{\langle \phi, 1-k^{-i} \rangle \in \mathcal{A}, I \neq \phi} i}.
\end{aligned}$$

Par conséquent, nous obtenons clairement une mesure additive : plus une interprétation falsifie les assertions dans \mathcal{A} , moins elle sera plausible. Nous notons également que dans cette interprétation du degré de surprise, l'entier i (dans le degré d'incertitude $1 - k^i$) peut être considéré comme le prix à payer si une assertion n'est pas satisfaite. Dans ce cas, les meilleures interprétations sont celles qui minimisent le coût total des assertions falsifiées.

Exemple 3.3 *Considérons à nouveau l'Exemple 3.1. Le Tableau 3.3 donne un exemple des degrés de possibilité, obtenus par l'Equation 3.4, pour trois interprétations sur le domaine $\Delta = \{Jean, John\}$.*

I	I	$\pi_{\mathcal{K}}(I)$
I_1	$Professeur^I = \{John\}, Etudiant^I = \{Jean\},$ $aTuteur^I = \{(Jean, John)\}, Enseigne^I = \{(John, Jean)\}$	0.0455
I_2	$Professeur^I = \{Jean\}, Etudiant^I = \{John\},$ $aTuteur^I = \{(John, Jean)\}, aEnseigne^I = \{(Jean, John)\}$	0.005
I_3	$Professeur^I = \{John, Jean\}, Etudiant^I = \{Jean, John\},$ $aTuteur^I = \{(Jean, John)\} aEnseigne^I = \{(John, Jean)\}$	0

L'interprétation I_1 falsifie l'assertion $\langle Enseigne(Jean, John), 0.87 \rangle$ et l'assertion $\langle aTuteur(b, a), 0.65 \rangle$. Ainsi, son degré de possibilité est $\pi_{\mathcal{K}}(I_1) = (1 - 0.87) \times (1 - 0.65) = 0.0455$. L'interprétation I_2 falsifie les deux assertions : $\langle Professeur(b), 0.95 \rangle$, $\langle Etudiant(a), 0.9 \rangle$. Nous notons que $\pi_{\mathcal{K}}(I_3) = 0$ puisque I_3 falsifie un axiome certain : $\langle Professor \sqsubseteq \neg Student, 1.0 \rangle$. Aucune des trois interprétations n'est un modèle de \mathcal{K} .

3.3 Mesure d'inconsistance

L'inconsistance se produit souvent lors de la construction, l'évolution et/ou la fusion d'ontologies. Manipuler les inconsistances, en particulier, gérer l'inconsistance en logique de description DL-Lite est de plus en plus reconnu comme un sujet de recherche important. nous distinguons en fait les notions d'incohérence et la notion d'inconsistance d'une base de connaissances DL-Lite. Alors que l'incohérence porte sur la TBox, l'inconsistance porte sur la ABox étant donné une TBox.

Définition 3.3 (*concept insatisfiable*). Soit $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ une base de connaissances. Un concept C est dit insatisfiable si seulement si, pour chaque interprétation I qui est un modèle de \mathcal{K} , $C^I = \emptyset$.

Définition 3.4 (*base de connaissances Incohérente*). Une base de connaissances $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ est incohérente si sa TBox \mathcal{T} est incohérente. La TBox \mathcal{T} est dite incohérente si elle contient un concept insatisfiable.

L'incohérence d'une base de connaissances DL-Lite correspond à l'absence d'un modèle satisfaisant toutes les formules de la TBox sans avoir de concept dont le nombre d'instances associées par ce modèle est nul . Par exemple, une base de connaissances ayant comme TBox $\mathcal{T} = \{A \sqsubseteq B, A \sqsubseteq \neg B\}$ est incohérente parce que A est insatisfiable.

Définition 3.5 (*base de connaissances inconsistante*). Une base de connaissances $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ est inconsistante s'il n'existe aucune interprétation qui soit un modèle pour \mathcal{T} et \mathcal{A} .

Contrairement au cadre standard où la notion de consistance est binaire, cette notion dans le cadre possibiliste devient graduelle. Une base de connaissances \mathcal{K} est totalement inconsistante, si le degré de possibilité de chaque interprétation est égale à 0, c'est-à-dire que chaque interprétation falsifie au moins une formule complètement certaine. \mathcal{K} est complètement consistante s'il y a au moins une interprétation qui satisfait toutes les formules. Mise à part ces deux cas extrêmes, la base de connaissances \mathcal{K} est considérée comme quelque peu inconsistante.

Le degré d'inconsistance d'une base de connaissances, désigné par $Inc(\mathcal{K})$, est un nombre réel entre 0 et 1. Plus formellement :

Définition 3.6 Soit Ω l'ensemble de toutes les interprétations possibles. Nous considérons \mathcal{K} une base de connaissances Pb- π -DL-Lite et $\pi_{\mathcal{K}}$ la distribution de possibilité induite par \mathcal{K} obtenue par l'équation 3.4. Le degré d'inconsistance de \mathcal{K} , désigné par $Inc(\mathcal{K})$, est défini sémantiquement comme suit :

$$Inc(\mathcal{K}) = 1 - \max_{I \in \Omega} (\pi_{\mathcal{K}}(I)) \quad (3.5)$$

La dualité entre le degré d'inconsistance et le degré de normalisation est exprimée par : $Inc(\mathcal{K}) = 1 - h(\pi)$. À savoir, le degré d'inconsistance d'une base de connaissances Pb- π -DL-Lite est donné par l'inverse du degré de normalisation de la distribution de possibilité π associée à \mathcal{K} . En particulier, le degré d'inconsistance est égal à 0 s'il existe au moins une interprétation entièrement possible (donc $h(\pi) = 1$) et ce degré est égal à 1 si toutes les interprétations sont totalement impossibles ($h(\pi) = 0$).

Exemple 3.4 Nous pouvons vérifier que le degré d'inconsistance de la base de connaissances \mathcal{K} présentée dans l'Exemple 3.1 est : $Inc(\mathcal{K}) = 1 - \max_{I \in \Omega} (\pi_{\mathcal{K}}(I)) = 1 - 0.0455$.

3.4 Réponse aux requêtes dans une Pb- π -DL-Lite

Une requête conjonctive $q(\vec{x})$ est sous la forme :

$$q(\vec{x}) \leftarrow \vec{x} . \exists \vec{y} \text{ conj}_i(\vec{x}, \vec{y}) \quad (3.6)$$

où \vec{x} et \vec{y} sont des vecteurs de variables. Les variables de \vec{x} sont appelées variables distinguées alors que celles de \vec{y} sont appelées variables non-distinguées.

La notation $\text{conj}_i(\vec{x}, \vec{y})$ dénote une conjonction d'atomes de la forme $A(a)$ ou $P(a, b)$ où A et P sont respectivement un concept atomique et un rôle atomique de \mathcal{K} et a, b sont des constantes ou des variables dans \vec{x} ou \vec{y} .

Lorsque la requête ne contient pas de variable distinguée (\vec{x} est vide), $q(\vec{x})$ est dite requête booléenne.

Une requête conjonctive peut être redéfinie d'une manière équivalente à partir d'une famille de requêtes booléennes, chacune d'elles résultant de l'instanciation du vecteur de variables distinguées. Dans cette section, nous nous limiterons à des requêtes booléennes.

Une requête booléenne de base¹, a la forme :

$$q \leftarrow \exists \vec{z}_i \bigwedge_{i=1}^n B_i(\vec{y}_i) \quad (3.7)$$

où B_i est un concept atomique ou un rôle atomique, \vec{z}_i est un ensemble de variables et \vec{y}_i est un individu ou une variable de \vec{z}_i (lorsque B_i est un concept) ou une paire de variables, ou une paire d'individus ou une variable et un individu (lorsque B_i est un rôle).

Le concept d'une mesure de nécessité est défini par :

$$N_\pi(q) = 1 - \max_{I \in \Omega, I \neq q} (\pi_{\mathcal{K}}(I)) \quad (3.8)$$

$N_\pi(q)$ représente dans quelle mesure q est certain compte tenu des connaissances disponibles. Lorsque \mathcal{K} est consistante, c'est-à-dire que $\pi_{\mathcal{K}}$ est normalisée, $N_\pi(q) > 0$ (à savoir q est quelque peu acceptée) si et seulement si chaque modèle des axiomes de \mathcal{T} et des assertions de \mathcal{A} est également un modèle de q .

De même, $N_\pi(q) = 1$ (q est entièrement acceptée) si et seulement si pour tout I tel que $I \neq q$, nous avons $\pi_{\mathcal{K}}(I) = 0$. Lorsque π est sous-normalisée ou inconsistante, alors q est considéré comme un peu acceptée si et seulement si $N(q) > Inc(\mathcal{K})$. Plus précisément :

Définition 3.7 Soit \mathcal{K} une base de connaissances Pb- π -DL-Lite et soit $\pi_{\mathcal{K}}$ la distribution possibiliste associée à \mathcal{K} . Soit N_π la mesure de nécessité induite par $\pi_{\mathcal{K}}$. Soit q une requête booléenne.

$$\mathcal{K} \models_\pi q \text{ si et seulement si } N_\pi(q) > Inc(\mathcal{K}).$$

Il a été montré dans [33] que dans le contexte de DL-Lite, répondre à une requête q revient simplement à évaluer une union de requêtes conjonctives sur la ABox vue comme une base de données relationnelle. En effet, le traitement des requêtes est divisé en deux étapes indépendantes : (1) la requête est d'abord traitée par un processus qui permet de la reformuler en considérant les axiomes de la TBox. Cette reformulation est donnée dans [33] par l'algorithme *PerfectRef*(q, \mathcal{T}). (2) la requête reformulée est ensuite évaluée en utilisant l'algorithme *Answer* [33], indépendamment de la TBox, i.e., sur la partie ABox, vue comme une base de données relationnelle.

La vérification d'instance représente le cas le plus simple pour répondre à une requête. Elle consiste à vérifier si un individu (resp. une paire d'individus) est une instance d'un concept (resp. d'un rôle) donné. La proposition suivante montre comment répondre à une requête de vérification d'instance en utilisant le concept de degré d'inconsistance.

Proposition 3.1 Soit $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ une base de connaissances Pb- π -DL-Lite, B un concept (resp. R un rôle) et soient a, b deux individus. Soit D_B (resp. D_R) un concept atomique (resp. rôle atomique) n'apparaissant pas dans \mathcal{T} :

- i) $N(B(a)) = Inc(\mathcal{K}_1)$ (resp. $N(R(a, b)) = Inc(\mathcal{K}_1)$) où
 - $\mathcal{K}_1 = \langle \mathcal{T}_1, \mathcal{A}_1 \rangle$ avec $\mathcal{T}_1 = \mathcal{T} \cup \{(D_B \sqsubseteq \neg B, 1)\}$
 - (resp. $\mathcal{T}_1 = \mathcal{T} \cup \{(D_R \sqsubseteq \neg R, 1)\}$) et
 - $\mathcal{A}_1 = \mathcal{A} \cup \{(D_B(a), 1)\}$ (resp. $\mathcal{A}_1 = \mathcal{A} \cup \{(D_R(a, b), 1)\}$).

1. a grounded query

- ii) $B(a)$ (resp. $R(a, b)$) est une conséquence de \mathcal{K} , dénotée par $\mathcal{K} \models_{\pi} B(a)$ (resp. $\mathcal{K} \models_{\pi} R(a, b)$), if $\text{Inc}(\mathcal{K}_1) > \text{Inc}(\mathcal{K})$.

Dans la Proposition 3.1, \mathcal{K}_1 représente une base de connaissances Pb- π -DL-Lite obtenue de \mathcal{K} en ajoutant l'hypothèse que $B(a)$ (resp. $R(a, b)$) est certainement fausse (exprimé par un nouveau concept D_B (resp. un nouveau rôle D_R)). Il est à noter que l'item ii) suit immédiatement l'item i) de la Proposition 3.1 et la Définition 3.7.

Preuve 3.1 Pour monter que $\text{Inc}(\mathcal{K}_1) = N(B(a))$ (un raisonnement similaire est valable pour $\text{Inc}(\mathcal{K}_1) = N(R(a, b))$), nous rappelons l'équation 3.5 :

$$\text{Inc}(\mathcal{K}) = 1 - \max_{I \in \Omega}(\pi_{\mathcal{K}}(I)).$$

Soit I une interprétation. Nous considérons deux cas :

- Lorsque $I \models D(a)$ et $I \models D \sqsubseteq \neg B$ (par conséquent $I \not\models B(a)$) alors $\text{Inc}(\mathcal{K}_1) = 1 - \max_{I \in \Omega}(\pi_{\mathcal{K}_1}(I))$.
- Lorsque $I \models B(a)$ alors $\text{Inc}(\mathcal{K}_1) = 0$.

Par conséquent, en utilisant l'équation 3.8 :

$$\text{Inc}(\mathcal{K}_1) = 1 - \max_{I \in \Omega, I \not\models B(a)}(\pi_{\mathcal{K}}(I)) = N(B(a)) \blacksquare$$

Exemple 3.5 Considérons le concept *Etudiant* (resp. *Enseignant*) dont les individus sont des étudiants (resp. enseignants) et soit *Enseigne* un rôle qui relie les enseignants à leurs élèves. Soient a et b deux individus.

Considérons la base de connaissances Pb- π -DL-Lite \mathcal{K} composée de la TBox \mathcal{T} et la ABox \mathcal{A} suivantes :

$$\begin{aligned} T &= \{ \langle \text{Enseignant} \sqsubseteq \neg \text{Etudiant}, 1.0 \rangle, \langle \text{Enseignant} \sqsubseteq \exists \text{Enseigne}, 1.0 \rangle \}, \\ A &= \{ \langle \text{Enseignant}(a), 0.5 \rangle, \langle \text{Etudiant}(a), 0.2 \rangle, \langle \text{Etudiant}(b), 0.7 \rangle \}. \end{aligned}$$

Nous pouvons facilement vérifier que l'interprétation qui a le degré de possibilité maximum, ne falsifie que l'assertion $(\text{Etudiant}(a), 0.2)$. La meilleur interprétation I_1 sur le domaine $\Delta = \{a, b\}$ est donné comme suit : $\text{Enseignant}^I = \{a\}$ et $\text{Etudiant}^I = \{b\}$, $\pi_{\mathcal{K}}(I_1) = (1 - 0.2)$. Le degré d'inconsistance de cette base est $\text{Inc}(\mathcal{K}) = 1 - \max(\pi_{\mathcal{K}}(I)) = 0.2$.

Considérons la requête suivante :

$$q(x) \leftarrow \exists y \text{ Enseigne}(x, y) \tag{3.9}$$

Pour répondre à cette requête, la première étape consiste à extraire l'ensemble des requêtes booléennes représentant la requête $q(x)$ en instanciant x par les valeurs possibles (a et b).

Nous obtenons les deux requêtes booléennes q_1 et q_2 suivantes :

$$q_1 \leftarrow \exists y \text{ Enseigne}(a, y)$$

$$q_2 \leftarrow \exists y \text{ Enseigne}(b, y)$$

L'idée est que si la réponse à la requête q_1 (resp. à q_2) est positive, alors a (resp. b) est une réponse pour la requête $q(x)$.

Nous commençons par la requête q_1 . La reformulation de la requête q_1 en utilisant l'algorithme *PerfectRef* défini dans [33] permet d'ajouter la requête suivante :

$$q_3 \leftarrow \text{Enseignant}(a)$$

Lorsque l'évaluation directe de q_1 ou q_3 sur l'assertion $ABox$ donne une réponse positive, la réponse finale à q_1 est positive.

Nous évaluons la requête q_3 . Puisque la requête q_3 est une requête de vérification d'instance, nous utilisons la Proposition 3.1 pour y répondre.

Nous introduisons un nouveau concept $D_{Enseignant}$ et nous construisons la nouvelle base de connaissances $K_1 = \langle T_1, A_1 \rangle$ avec :

- $T_1 = T \cup \{ \langle D_{Enseignant} \sqsubseteq \neg Enseignant, 1.0 \rangle \}$ et
- $A_1 = A \cup \{ \langle D_{Enseignant}(a), 1.0 \rangle \}$.

Il est facile de vérifier que le degré d'inconsistance de K_1 est $Inc(K_1) = 0.5$.

$Inc(K_1) > Inc(K)$, nous en déduisons que $Enseignant(a)$ est une conséquence de \mathcal{K} . L'évaluation de la requête q_1 se fait d'une manière similaire.

La réponse finale à la requête q_1 est positive et a est une réponse pour la requête de départ $q(x)$.

Considérons à présent la requête q_2 .

La reformulation de la requête q_2 nous permet d'ajouter la requête q_4 suivante :

$$q_4 \leftarrow Enseignant(b)$$

En suivant le même raisonnement que celui de la requête q_3 pour évaluer la requête q_4 , nous construisons la nouvelle base de connaissances $K_2 = \langle T_2, A_2 \rangle$ avec :

- $T_2 = T \cup \{ \langle D_{Enseignant} \sqsubseteq \neg Enseignant, 1.0 \rangle \}$ et
- $A_2 = A \cup \{ \langle D_{Enseignant}(b), 1.0 \rangle \}$.

Nous pouvons vérifier que le degré d'inconsistance de K_2 est $Inc(K_2) = Inc(K) = 0.2$. Nous en déduisons que $Enseignant(b)$ n'est pas une conséquence de la base de connaissances \mathcal{K} .

Pour évaluer la requête q_2 , nous devons d'abord construire un ensemble de requêtes booléennes de base en remplaçant la valeur de y par les valeurs possibles.

La réponse à la requête q_2 est alors positive si la réponse a au moins une des requêtes obtenues est positive. Nous obtenons les deux requêtes suivantes :

$$q_5 \leftarrow Enseigne(a, b)$$

$$q_6 \leftarrow Enseigne(b, b)$$

En suivant le même raisonnement à celui de la requête q_3 et q_4 , nous pouvons vérifier que ni $Enseigne(a, b)$ ni $Enseigne(b, b)$ n'est une conséquence de \mathcal{K} .

En conclusion, la requête $q(x)$ a une seule réponse qui est : $x = a$.

La Proposition 3.1 montre comment évaluer une requête de vérification d'instance en utilisant le concept du degré d'inconsistance. En fait, vérifier si une assertion $B(a)$ (resp. $R(a, b)$) est une conséquence d'une base de connaissances DL-Lite \mathcal{K} se résume à comparer deux degrés d'inconsistance :

- le degré d'inconsistance de la base initiale \mathcal{K} et
- le degré d'inconsistance de la base augmentée \mathcal{K}_1 qui résulte de \mathcal{K} en ajoutant le fait certain que a (resp. (a, b)) n'est pas une instance du concept B (resp. du rôle R).

Lorsqu'une requête est composée d'une conjonction d'assertions², il suffit d'appliquer la proposition 3.1, à chaque assertion. Ceci est possible grâce à la propriété de la mesure de nécessité suivante :

$$N(q_1 \wedge q_2) = \min(N(q_1), N(q_2)). \quad (3.10)$$

2. grounded queries

Soit $q = \bigwedge_{i=1,\dots,n} B_i(\vec{a}_i)$, où B_i est un concept ou un rôle, et \vec{a}_i est un individu ou une paire des individus. D'après l'équation 3.10 et la proposition 3.1, nous avons :

- i) $N(q) = \min(Inc(\mathcal{K}_{B_1}), \dots, Inc(\mathcal{K}_{B_m}))$, \mathcal{K}_{B_1} est le résultat de l'ajout à \mathcal{K} l'hypothèse que $B_i(\vec{a}_i)$ est certainement fausse.
- ii) q est une conséquence de \mathcal{K} si et seulement si $\forall i, Inc(\mathcal{K}_{B_i}) > Inc(\mathcal{K})$.

La Proposition 3.1 peut être également étendue pour traiter les requêtes booléennes et les requêtes conjonctives. Dans ce cas, le processus de réponse à des requêtes se résume à la reformulation de la requête q en utilisant l'algorithme *PerfectRef* proposé dans [35]. Cette étape conduit à un ensemble de requêtes Q . L'ensemble de réponses de la requête initiale est alors l'union des ensembles de réponses de toutes les requêtes de Q . Par conséquent, répondre à la requête q se résume à l'évaluation de chaque requête $q_i \in Q$ directement sur la ABox.

3.5 La fermeture négative en Pb- π -DL-Lite

L'objectif de cette section est de définir la fermeture négative d'une base de connaissances Pb- π -DL-Lite. La notion de la fermeture négative joue un rôle cruciale pour caractériser les concepts de la consistance et d'inférence à partir d'une base de connaissances Pb- π -DL-Lite [29].

Comme nous l'avons déjà mentionné, la terminologie \mathcal{T} est considérée comme totalement certaine et les assertions de la ABox sont incertaines. La base terminologique \mathcal{T} est supposée stable et ne doit pas être mise en doute en présence d'incohérences.

Nous rappelons que la fermeture négative combine \mathcal{T}_p et \mathcal{T}_n pour dériver tous les axiomes d'inclusion négatifs de \mathcal{T}_p et \mathcal{T}_n . Chaque axiome TBox a un degré de certitude égal à 1. Dans ce cas, la base terminologie correspond à une base de connaissances classique DL-Lite. La fermeture négative $Cln(\mathcal{T})$ peut être obtenue en appliquant simplement les règles de la fermeture négative standard définie dans [5] jusqu'à atteindre un point fixe.

Exemple 3.6 La fermeture négative $Cln(\mathcal{T})$ de \mathcal{T} de l'Exemple 3.1 est :

$$Cln(\mathcal{T}) = \{ \langle \text{Professeur} \sqsubseteq \neg \text{Etudiant}, 1.0 \rangle, \\ \langle \exists \text{Enseigne} \sqsubseteq \neg \text{Etudiant}, 1.0 \rangle, \\ \langle \exists \text{Enseigne}^- \sqsubseteq \neg \text{Professeur}, 1.0 \rangle, \\ \langle \exists a \text{Tuteur} \sqsubseteq \neg, \text{Professeur} 1.0 \rangle, \\ \langle \exists a \text{Tuteur}^- \sqsubseteq \neg \text{Etudiant}, 1.0 \rangle \}.$$

Proposition 3.2 Soient $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ une base de connaissances Pb- π -DL-Lite et $\mathcal{K}' = \langle Cln(\mathcal{T}), \mathcal{A} \rangle$, où $Cln(\mathcal{T})$ est la fermeture négative de \mathcal{T} . Nous avons :

$$Inc(\mathcal{K}) = Inc(\mathcal{K}'). \quad (3.11)$$

La preuve de cette proposition est montrée dans [35]. Elle stipule qu'une base de connaissances DL-Lite classique $\mathcal{K}_s = \langle \mathcal{T}_s, \mathcal{A}_s \rangle$ est inconsistante si et seulement si la base de connaissances $\langle Cln(\mathcal{T}_s), \mathcal{A}_s \rangle$ est inconsistante.

Du fait que la base TBox est totalement certaine, les conflits sont assertionnels et en plus ils sont binaires, c'est-à-dire ils impliquent au plus deux assertions de la ABox qui contredisent un axiome TBox négatif. La complexité du calcul des conflits est polynomiale.

Définition 3.8 Soit $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ une base de connaissances Pb- π -DL-Lite. Un conflit est un sous-ensemble minimal de la ABox \mathcal{A} qui est inconsistant avec la fermeture négative $Cln(\mathcal{T})$. Plus précisément :

- $\langle Cln(\mathcal{T}), \mathcal{C} \rangle$ est inconsistant
- $\forall \mathcal{C}' \subset \mathcal{C}$, $\langle Cln(\mathcal{T}), \mathcal{C}' \rangle$ est consistant.

Exemple 3.7 Considérons à nouveau la base de connaissances \mathcal{K} de l'Exemple 3.1. L'ensemble de conflit obtenu en appliquant le Définition 3.8 est :

- $\mathcal{C}_1 = \{ \langle \text{Professeur}(\text{John}), 0.95 \rangle, \langle \text{Enseigne}(\text{Jean}, \text{John}), 0.87 \rangle \}$,
- $\mathcal{C}_2 = \{ \langle \text{Etudiant}(\text{Jean}), 0.90 \rangle, \langle \text{Enseigne}(\text{Jean}, \text{John}), 0.87 \rangle \}$,
- $\mathcal{C}_3 = \{ \langle \text{Professeur}(\text{John}), 0.95 \rangle, \langle a\text{Tuteur}(\text{John}, \text{Jean}), 0.65 \rangle \}$,
- $\mathcal{C}_4 = \{ \langle \text{Etudiant}(\text{Jean}), 0.90 \rangle, \langle a\text{Tuteur}(\text{John}, \text{Jean}), 0.65 \rangle \}$
- $\mathcal{C}_5 = \{ \langle \text{Etudiant}(\text{Jean}), 0.90 \rangle, \langle \text{Enseigne}(\text{Jean}, \text{Marry}), 0.87 \rangle \}$,
- $\mathcal{C}_6 = \{ \langle \text{Professeur}(\text{Marry}), 0.98 \rangle, \langle \text{Enseigne}(\text{Jean}, \text{Marry}), 0.87 \rangle \}$,

Un conflit est clairement un sous-ensemble inconsistant d'informations. En effet, supprimer un élément de chaque conflit permet de restaurer la consistance. Pour notre Exemple, la consistance est rétablie si nous supprimons les assertions suivantes : $\langle \text{Professeur}(\text{John}), 0.95 \rangle$, $\langle \text{Etudiant}(\text{Jean}), 0.9 \rangle$ et $\langle a\text{Enseigne}(\text{Jean}, \text{Marry}), 0.87 \rangle$.

3.6 Comparaison avec les extensions DL-Lite

Dans le chapitre 2, nous avons présenté brièvement quelques approches développées, pour étendre les logiques de description avec le raisonnement incertain à savoir, les logiques de description probabilistes, les logiques de description possibilistes (basées sur le min) et les logiques de description floues. Pour chacune de ces extensions, nous avons présenté quelque alternatives. Dans cette section, nous présentons une comparaison entre notre extension et ces extensions qui existent dans la littérature.

- **Pb- π -DL-Lite Vs DL probabiliste**

La différence principale entre notre approche DL-Lite possibilite basée sur le produit et les DLs probabilistes réside dans le fait que dans notre approche un degré de nécessité est attaché à chaque axiome/assertion de la logique de description. En revanche, les logiques de description probabilistes (par exemple [56, 88]) attachent un intervalle de croyance reflétant la limite inférieure l et la limite supérieure u ($l, u \in [0, 1]$) à chaque assertion/axiome.

- **Pb- π -DL-Lite Vs DL floue**

La différence entre notre approche et les approches de DLs floues réside dans la nature des connaissances gérées : notre approche manipule des connaissances incertaines où l'incertitude est causée par l'absence d'information tandis que les DLs floues manipulent des connaissances imprécises où l'imprécision est causée par une imprécision inhérente à la description de certains termes. D'un point de vue syntaxique, les deux extensions associent un degré à un axiome/assertion de la logique de description. Cependant, ils diffèrent dans la définition de la sémantique, c'est-à-dire que la vérité d'un concept (ou d'un rôle) dans notre approche possibiliste est encore à deux valeurs (two-valued), alors que dans les logiques de description floues, la vérité d'un

concept (ou d'un rôle) est Multi-valuée (multi-valued).

- **Pb- π -DL-Lite Vs DL-Lite basé sur le min**

La première différence entre notre extension et la DL-Lite possibiliste basée sur le min a été déjà évoquée précédemment et concerne la manière dont l'échelle d'incertitude est utilisée. Plus précisément, lorsque les degrés d'incertitude représentent un codage de plausibilité entre les assertions, nous devons utiliser DL-Lite possibiliste basée sur min. En revanche, lorsque le degré d'incertitude représente un degré de surprise dans le sens de Spohn [109, 110] ou un résultat de transformation d'une distribution de probabilité en une distribution de possibilité [51], la DL-Lite possibiliste basée sur le produit est plus appropriée.

Une autre différence entre les deux approches, qui découle d'ailleurs de la première différence, concerne la manière dont la distribution de possibilité est générée. Enfin, du point de vue des réponses aux requêtes, DL-Lite possibiliste basée sur le min et DL-Lite possibiliste basée sur le produit fournissent des résultats différents.

Exemple 3.8 Soit $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ une base de connaissances DL-Lite possibiliste composée de la TBox \mathcal{T} et la ABox \mathcal{A} suivantes :

$$\mathcal{T} = \{ \langle A \sqsubseteq \neg B, 1.0 \rangle, \langle A \sqsubseteq \neg C, 1.0 \rangle \}$$

$$\mathcal{A} = \{ \langle A(b), 0.8 \rangle, \langle B(b), 0.7 \rangle, \langle C(b), 0.7 \rangle \}.$$

- **DL-Lite possibiliste basée sur le min** : le degré d'inconsistance de la base de connaissances \mathcal{K} est 0.7. En effet, l'interprétation I qui satisfait le mieux \mathcal{K} , satisfait $A(b)$ et falsifie $B(b)$ et $C(b)$. Par conséquent, son degré de possibilité est : $\pi(I) = 1 - \max(0.7, 0.7) = 0.3$.
- **DL-Lite possibiliste basée sur le produit** : Le degré d'inconsistance de la base de connaissances \mathcal{K} est de 0.8. L'interprétation J qui satisfait le mieux \mathcal{K} , satisfait $B(b)$ et $C(b)$ et falsifie $A(b)$. Son degré de possibilité est : $\pi(J) = 1 - 0.8 = 0.2$.

Avec DL-Lite possibiliste basée sur le min, l'assertion $A(b)$ est valide alors qu'avec DL-Lite possibiliste basée sur le produit, les assertions $B(b)$ et $C(b)$ sont valides.

À partir de cet exemple, nous pouvons voir que les deux résultats sont différents et cela renforce le fait que les deux approches ne sont pas comparables.

3.7 Conclusion

Dans ce chapitre, nous avons développé une extension des ontologies légères, encodées dans le langage DL-Lite, dans le cadre de la théorie des possibilités basée sur le produit. Le langage résultant est noté Pb- π -DL-Lite. Tout d'abord, le chapitre présente la syntaxe et la sémantique de Pb- π -DL-Lite. Ensuite, il aborde le problème de degré d'inconsistance d'une base de connaissances exprimée à l'aide de l'extension proposée Pb- π -DL-Lite et son rôle crucial dans les tâches d'inférence. Le chapitre met en évidence en particulier que le calcul du degré d'inconsistance est au cœur du processus de réponse à des requêtes dans de telles bases de connaissances. Dans le chapitre suivant, nous nous concentrerons sur les aspects algorithmiques et calculatoires du problème de calcul du degré d'inconsistance d'une base de connaissances Pb- π -DL-Lite avec TBox certaine.

Chapitre 4

Analyse de la complexité de la logiques Pb- π -DL-Lite

4.1 Introduction

Dans le chapitre précédent, nous avons présenté les fondements de la nouvelle extension développée Pb- π -DL-Lite ainsi que la notion du degré d'inconsistance et sa définition sémantique. Cependant, le calcul du degré d'inconsistance en se fendant sur sa définition sémantique nécessite un examen de toutes les interprétations possibles, qui sont en nombre exponentiel, afin de choisir celle qui maximise le degré de possibilité [16].

Dans ce chapitre, nous étudions la complexité de l'extension DL-Lite que nous avons proposé et nous détaillons l'approche développée pour le calcul du degré d'inconsistance. Cet approche est basée sur une hypothèse, assez forte, que seules les assertions de la ABox sont incertaines. Les axiomes de la TBox sont considérés comme totalement certains. Cependant, il existe des contextes d'applications telles que OBDA¹ ou OMQA², où cette hypothèse est raisonnable. Dans ce contexte, il existe une grande quantité de données provenant de différentes sources. Ces données peuvent être incertaines. Cependant, la terminologie est généralement de petite taille et elle est souvent considérée comme stable et incontestable. Cela étant dit, d'un point de vue théorique, notre approche peut être facilement étendue pour faire face à des ontologies incertaines. Cette généralisation est discutée dans le sixième chapitre.

L'approche présentée ici, nécessite d'abord le calcul des conflits binaires, dont la complexité reste polynomiale. Ensuite, une transformation de la base de connaissances Pb- π -DL-Lite en une base de connaissances propositionnelle est proposée. Le calcul du degré d'inconsistance pour des base de connaissances se base alors sur des tests de satisfiabilité utilisant des outils de la famille SAT.

Ce chapitre est organisé comme suit : la section 2 rappelle la définition des problèmes utilisés pour l'encodage du problème de calcul du degré d'inconsistance. La section 3 présente l'approche exacte développée dans cette thèse pour le calcul du degré d'inconsistance. Elle discute les transformations équivalentes entre une base de connaissances Pb- π -DL-Lite \mathcal{K} et une base de connaissances 2-Horn $\mathbb{B}_{\mathcal{K}}$, la formulation du problème du calcul d'un degré d'inconsistance dans le cadre des bases 2-Horn et l'algorithme exact qui permet d'obtenir le degré d'inconsistance d'une base de connaissances Pb- π -DL-Lite. Enfin la section 4 conclut

1. OntologyBased Data Access
2. Ontology Mediated Query Answering

le chapitre.

4.2 Les problèmes calculatoire de décision utilisés dans cette thèse

Avant d'aborder le problème de calcul du degré d'inconsistance d'une base de connaissances, nous présentons brièvement les définitions de classes de complexité qui permettent d'ordonner les problèmes en fonction des algorithmes qui existent pour les résoudre.

La classe de complexité P regroupe l'ensemble des problèmes de décision qui peuvent être résolus en un temps polynomial. Cette classe contient les problèmes de décision polynomiaux, c'est-à-dire que nous pouvons les résoudre à l'aide d'un algorithme déterministe en un temps polynomial par rapport à la taille de l'instance. S'il est vrai que la classe P contient de nombreux problèmes de décision, d'autres problèmes ne sont pas connus pour être polynomiaux.

La classe la plus vaste est la classe de complexité NP qui regroupe l'ensemble des problèmes de décision qui peuvent être résolus en un temps polynomial par une machine de Turing non déterministe. Les problèmes de la classe NP (pour Non déterministe Polynomial) sont les problèmes de décision pour lesquels la réponse « Oui » peut être décidée par un algorithme non déterministe en un temps polynomial par rapport à la taille de l'instance. De façon équivalente, c'est la classe des problèmes qui admet un algorithme polynomial qui, étant donné une solution du problème NP , est capable de répondre si « Oui » ou « Non » cette solution est une réponse au problème.

4.2.1 Le problème de Max-2-Horn-SAT pondéré

Le problème de satisfiabilité maximale (Max-SAT) est l'une des variantes d'optimisation associée au problème de satisfiabilité booléenne (SAT). Dans Max-SAT, il s'agit de déterminer la solution qui permet d'avoir le nombre maximal de clauses pouvant être satisfaites, où une clause est une disjonction de littéraux et un littéral est simplement une variable ou sa négation.

Dans le problème Max-2-Sat, chaque clause contient au plus deux littéraux. La tâche est toujours de trouver une affectation aux variables de la formule de telle sorte à maximiser le nombre de clauses satisfaites. Le problème Max-2-Sat est un problème d'optimisation classique. Sa version de décision a été prouvée NP -complet par Garey, Johnson et Stockmeyer [55]. Un problème plus général est Max-2-Sat pondéré. Dans ce cas, un poids positif est associé à chaque clause et le problème consiste à trouver l'affectation qui minimise la somme des poids des clauses non satisfaites.

Le problème W-Max-2-Horn-SAT est une variante du problème Max-2-SAT pondéré où toutes les clauses sont des clauses de Horn. Ainsi résoudre un problème W-Max-2-Horn-SAT consiste à trouver une affectation de valeurs booléennes aux variables propositionnelles qui maximisent le poids total des clauses satisfaites ou, de façon équivalente, minimise le poids total des clauses non satisfaites. Une base de connaissances pondérée 2-Horn est un ensemble de formules pondérées de la forme :

$$\theta = \{(\phi_i, k_i) : i = 1, \dots, n\}, \quad (4.1)$$

où ϕ_i est une clause 2-Horn, à savoir, une clause avec au plus deux littéraux et au plus un littéral positif. k_i est un nombre naturel appartenant à $N \cup \{\infty\}$. Lorsque $k_i = \infty$, la clause

ϕ_i correspondante doit être satisfaite (elle est dite clause dure). Dans le cas contraire, k_i reflète le degré d'importance de la clause correspondante.

La version de décision du problème W-Max-2-Horn-SAT est donnée comme suit :

Définition 4.1 *Soit θ une base de connaissances 2-Horn pondérée et R un entier positif.*

Le problème de décision du Max-2-Horn-SAT pondéré est défini comme suit :

Existe-t-il un sous-ensemble $\theta' \subseteq \theta$ consistant tel que :

- i) θ' contient chaque (ϕ, k) de θ tel que $k = \infty$ et*
- ii) $Weight(\theta') \geq R$?*

où $Weight(\theta') = \sum_{(\phi_i, k_i) \in \theta' \text{ } k_i \neq \infty} k_i$.

Notons que la maximisation de $Weight(\theta')$ est équivalente à la minimisation du coût de θ' dénoté $Cost(\theta')$ et défini par : $Cost(\theta') = \sum_{(\phi_i, k_i); i \in \theta \setminus \theta'} k_i$.

Exemple 4.1 *Soit la base de connaissances θ suivante :*

$$\theta = \{(a, \infty), (b, \infty), (\neg b \vee \neg a, 6), (a \vee \neg c, 2), (\neg b \vee a, \infty), (a \vee c, 7)\}.$$

Existe-t-il un sous-ensemble θ' consistant tel que : $Cost(\theta') < 5$? (ou $Weight(\theta') \geq 5$).

Pour assurer que $Cost(\theta') < 5$, nous devons éliminer que la formule $(a \vee \neg c; 2)$. Le sous-ensemble restant n'est pas consistant, donc la réponse est "non".

4.3 Le calcul du degré d'inconsistance

Dans cette section, nous présentons notre approche exacte pour le calcul du degré d'inconsistance. La première étape consiste à calculer l'ensemble des conflits assertionnels. Cela se fait en calculant d'abord la fermeture négative de \mathcal{T} obtenue en utilisant les règles présentées dans le chapitre précédent. Ensuite, cet ensemble de conflits est transformé en une base de connaissances 2-Horn sur laquelle le problème du calcul de degré d'inconsistance est formulé comme problème W-Max-2-Horn-SAT. Ceci nous permet alors d'utiliser un solveur W-Max-SAT pour résoudre le problème.

4.3.1 La transformation d'une base Pb- π -DL-Lite en une base de connaissances 2-Horn pondérée

La transformation d'une base Pb- π -DL-Lite en une base de connaissances 2-Horn est une étape importante pour le calcul. Cette transformation est donnée par la définition suivante :

Définition 4.2 *Soit $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ une base de connaissances Pb- π -DL-Lite. Soit ζ l'ensemble de tous les conflits présents dans \mathcal{K} . Soit F une fonction de changement d'échelle de $[0, 1]$ à \mathbb{N} . Chaque assertion $X(a)$ est associée à un symbole propositionnelle unique, désignée par X_a .*

La base de connaissances 2-Horn pondérée correspondant à \mathcal{K} , dénotée par $\mathbb{B}_{\mathcal{K}}$, est définie comme suit :

$$\mathbb{B}_{\mathcal{K}} = \{(D_a, F(\alpha)), (B_a, F(\beta)), (\neg D_a \vee \neg B_a, \infty) \mid \{(D(a), \alpha), (B(a), \beta)\} \in \zeta\}.$$

Dans la Définition 4.2, la formule logique $(\neg D_a \vee \neg B_a, \infty)$ est équivalente à l'axiome $\langle D \sqsubseteq \neg B, 1 \rangle$. Cette formule exprime une contrainte d'intégrité où D_a et D_b ne peuvent pas être acceptés tous les deux. Il est clair que la complexité de la construction de $\mathbb{B}_{\mathcal{K}}$ est linéaire.

Exemple 4.2 *Considérons à nouveau la base de connaissances \mathcal{K} de l'Exemple 3.1 du chapitre 3. Nous rappelons tout d'abord l'ensemble de conflits extrait de la base ζ :*

- $\mathcal{C}_1 = \{\langle \text{Professeur}(\text{John}), 0.95 \rangle, \langle \text{Enseigne}(\text{Jean}, \text{John}), 0.87 \rangle\}$,
- $\mathcal{C}_2 = \{\langle \text{Etudiant}(\text{Jean}), 0.90 \rangle, \langle \text{Enseigne}(\text{Jean}, \text{John}), 0.87 \rangle\}$,
- $\mathcal{C}_3 = \{\langle \text{Professeur}(\text{John}), 0.95 \rangle, \langle \text{aTuteur}(\text{John}, \text{Jean}), 0.65 \rangle\}$,
- $\mathcal{C}_4 = \{\langle \text{Etudiant}(\text{Jean}), 0.90 \rangle, \langle \text{aTuteur}(\text{John}, \text{Jean}), 0.65 \rangle\}$
- $\mathcal{C}_5 = \{\langle \text{Etudiant}(\text{Jean}), 0.90 \rangle, \langle \text{Enseigne}(\text{Jean}, \text{Marry}), 0.8 \rangle\}$,
- $\mathcal{C}_6 = \{\langle \text{Professeur}(\text{Marry}), 0.98 \rangle, \langle \text{Enseigne}(\text{Jean}, \text{Marry}), 0.8 \rangle\}$

La base de connaissances 2-Horn \mathbb{B}_K correspondante est la suivante :

$$\begin{aligned} \mathbb{B}_K = \{ & (\text{Professeur_John}, F(0.95)), (\text{Professeur_Marry}, F(0.98)) \\ & (\text{Etudiant_Jean}, F(0.90)), \\ & (\text{Enseigne_Jean_Marry}, F(0.8)), \\ & (\text{Enseigne_Jean_John}, F(0.87)), \\ & (\text{aTuteur_Jean_John}, F(0.65)), \\ & (\neg \text{Enseigne_Jean_Marry} \vee \neg \text{Etudiant_Jean}, \infty), \\ & (\neg \text{Enseigne_Jean_Marry} \vee \neg \text{Professeur_Marry}, \infty), \\ & (\neg \text{Enseigne_Jean_John} \vee \neg \text{Etudiant_Jean}, \infty), \\ & (\neg \text{Enseigne_Jean_John} \vee \neg \text{Professeur_John}, \infty), \\ & (\neg \text{Professeur_John} \vee \neg \text{aTuteur_Jean_John}, \infty), \\ & (\neg \text{Etudiant_Jean} \vee \neg \text{aTuteur_Jean_John}, \infty). \} \end{aligned}$$

La fonction F utilisée dans la Définition 4.2 n'est pas unique. Nous rappelons que les poids de \mathcal{K} sont exprimés dans l'intervalle $[0, 1]$, alors que les poids de \mathbb{B}_K sont des nombres entiers.

Soit $\ell = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$ l'ensemble des différents degrés de certitude utilisés dans la base assertionnelle \mathcal{A} . La fonction F est une fonction croissante et doit satisfaire la propriété suivante :

Soit $A \subseteq \ell$

$$\begin{aligned} & \text{Si } \alpha = 1, \text{ alors } F(\alpha) = \infty, \\ & \text{Si } \forall \alpha \in A, \alpha \neq 1 \text{ alors } \sum_{\alpha_i \in A} F(\alpha_i) = F(1 - \prod_{\alpha_i \in A} (1 - \alpha_i)). \end{aligned} \quad (4.2)$$

Dans une base de connaissances DL-Lite possibilité quantitative, l'opérateur utilisé est le produit alors que l'outil W-Max-2-Horn-SAT utilise l'opération somme. La propriété précédente (4.2) exprime le comportement de la fonction F .

La proposition suivante montre que lorsque la fonction F satisfait l'équation (4.2) alors nous pouvons établir comment calculer le degré d'inconsistance $\text{Inc}(\mathcal{K})$ en utilisant la base de connaissances 2-Horn associée \mathbb{B}_K [29].

Proposition 4.1 *Soient \mathcal{K} une base de connaissances Pb- π -DL-Lite et \mathbb{B}_K sa base de connaissances 2-Horn pondérée. $\text{Inc}(\mathcal{K}) = \alpha$ si et seulement s'il existe un sous-ensemble consistant $\mathbb{B}'_K \subseteq \mathbb{B}_K$ tels que :*

- i) $\text{Cost}(\mathbb{B}'_K) = F(\alpha)$ et
- ii) pour chaque sous-ensemble consistant $\mathbb{B}''_K \subseteq \mathbb{B}_K$, $\text{Cost}(\mathbb{B}''_K) \geq \text{Cost}(\mathbb{B}'_K)$.

Preuve 4.1 *La preuve de cette proposition se fait en deux parties :*

- *La 1^{er} partie*

Supposons que $\text{Inc}(\mathcal{K}) = \alpha$. Soit J une interprétation ayant un degré de possibilité maximal. $\forall I \in \Omega, I \neq J$ alors $\pi_{\mathcal{K}}(J) \geq \pi_{\mathcal{K}}(I)$.

Soient $\{(D_1(\vec{a}_1), \gamma_1), \dots, (D_m(\vec{a}_m), \gamma_m)\} \in \mathcal{A}$ toutes les assertions telles que $J \not\models D_i(\vec{a}_i)$ où D_i est un concept et \vec{a}_i est un individu ou D_i est un rôle et \vec{a}_i est une paire d'individus. $\text{Inc}(\mathcal{K}) = \alpha = 1 - \prod_{i=1}^m (1 - \gamma_i)$.

Puisque pour $1 \leq i \leq m$, $J \not\models D_i(\vec{a}_i)$ si et seulement si $\mathcal{C} = \{(D_i(\vec{a}_i), \gamma_i), (B_i(\vec{a}_i), \beta_i)\}$ est un conflit de $D_i(\vec{a}_i)$ avec une certaine assertion $B_i(\vec{a}_i)$. Il s'ensuit que $\mathbb{B}'_{\mathcal{K}} = \mathbb{B}_{\mathcal{K}} \setminus \{(D_{i_{\vec{a}_i}}, F(\gamma_i)) | i \in \{1, \dots, m\}\}$ est consistante. De plus, nous avons :

$$\begin{aligned} \text{Cost}(\mathbb{B}'_{\mathcal{K}}) &= \sum_{i=1}^m F(\gamma_i) \\ &= F(1 - \prod_{i=1}^m (1 - \gamma_i)) \quad (\text{de l'Equation 4.2}) \\ &= F(\alpha). \end{aligned}$$

Pour chercher une contradiction, supposons qu'il y ait $\mathbb{B}''_{\mathcal{K}} \subseteq \mathbb{B}_{\mathcal{K}}$ tel que $\mathbb{B}''_{\mathcal{K}}$ est consistante et $\text{Cost}(\mathbb{B}''_{\mathcal{K}}) < \text{Cost}(\mathbb{B}'_{\mathcal{K}})$.

$\mathbb{B}''_{\mathcal{K}}$ a la forme $\mathbb{B}_{\mathcal{K}} \setminus \{(D'_{1_{\vec{a}_1}}, F(\gamma'_1)), \dots, (D'_{k_{\vec{a}_k}}, F(\gamma'_k))\}$ où $D'_i(\vec{a}_i)$ sont des assertions de \mathcal{A} et $\text{Cost}(\mathbb{B}''_{\mathcal{K}}) = \sum_{j=1}^k F(\gamma'_j)$. Du fait que $\text{Cost}(\mathbb{B}''_{\mathcal{K}}) < \text{Cost}(\mathbb{B}'_{\mathcal{K}})$ il s'ensuit que :

$$\sum_{j=1}^k F(\gamma'_j) < \sum_{i=1}^m F(\gamma_i) \quad (4.3)$$

A partir l'équation 4.2, il s'ensuit que :

$$F(1 - \prod_{j=1}^k (1 - \gamma'_j)) < F(1 - \prod_{i=1}^m (1 - \gamma_i)) \quad (4.4)$$

Le fait que F est croissant nous permet d'obtenir :

$$\prod_{j=1}^k (1 - \gamma'_j) > \prod_{i=1}^m (1 - \gamma_i) \quad (4.5)$$

Soit $J' \in \Omega$ une interprétation satisfaisant $\mathbb{B}''_{\mathcal{K}}$, c'est-à-dire que J' falsifie exactement $\{D'_j(\vec{a}_j) | j \in \{1, \dots, k\}\}$.

Mais, l'équation 4.5 signifie exactement que $\pi_{\mathcal{K}}(J') > \pi_{\mathcal{K}}(J)$, ce qui contredit le fait que α soit le degré d'inconsistance de \mathcal{K} .

• La 2^{ème} partie

Supposons qu'il existe $\mathbb{B}'_{\mathcal{K}} \subseteq \mathbb{B}_{\mathcal{K}}$ telle que $\mathbb{B}'_{\mathcal{K}}$ est consistante, $\text{Cost}(\mathbb{B}'_{\mathcal{K}}) = F(\alpha)$ et $\forall \mathbb{B}''_{\mathcal{K}} \subseteq \mathbb{B}_{\mathcal{K}}$, si $\mathbb{B}''_{\mathcal{K}}$ est consistante, alors $\text{Cost}(\mathbb{B}''_{\mathcal{K}}) \geq \text{Cost}(\mathbb{B}'_{\mathcal{K}})$.

Clairement, il existe $(D_1(\vec{a}_1), \gamma_1), \dots, (D_m(\vec{a}_m), \gamma_m) \in \mathcal{A}$ tel que : $\mathbb{B}'_{\mathcal{K}} = \mathbb{B}_{\mathcal{K}} \setminus \{(D_{i_{\vec{a}_i}}, F(\gamma_i)) | i \in \{1, \dots, m\}\}$, $\mathbb{B}'_{\mathcal{K}}$ est consistante et $\text{Cost}(\mathbb{B}'_{\mathcal{K}}) = \sum_{i=1}^m F(\gamma_i) = F(\alpha)$.

De l'équation 4.2, il s'ensuit : $\alpha = 1 - \prod_{i=1}^m (1 - \gamma_i)$.

Soit I une interprétation satisfaisant $\mathbb{B}'_{\mathcal{K}}$, c'est-à-dire que I falsifie exactement $\{D_i(\vec{a}_i) | i \in \{1, \dots, m\}\}$, alors : $\pi_{\mathcal{K}}(I) = \prod_{i=1}^m (1 - \gamma_i)$.

Pour chercher une contradiction, supposons que α n'est pas le degré d'inconsistance de \mathcal{K} .

Alors, il existe une interprétation $I' \in \Omega$ telle que $\pi_{\mathcal{K}}(I') > \pi_{\mathcal{K}}(I)$.

Nous considérons que $(D'_1(\vec{a}'_1), \gamma'_1), \dots, (D'_k(\vec{a}'_k), \gamma'_k) \in \mathcal{A}$ tel que $\{D'_j(\vec{a}'_j) | j \in \{1, \dots, k\}\}$ sont exactement les assertions falsifiées par I' . Alors, nous avons :

$$\pi_{\mathcal{K}}(I') = \prod_{j=1}^k (1 - \gamma'_j) > \pi_{\mathcal{K}}(I) = \prod_{i=1}^m (1 - \gamma_i) \quad (4.6)$$

Donc, $\mathbb{B}''_{\mathcal{K}} = \mathbb{B}_{\mathcal{K}} \setminus \{(D'_{j \rightarrow \vec{a}'_j}, F(\gamma'_j)) | j \in \{1, \dots, k\}\}$ est un sous-ensemble consistant de \mathbb{B} et

$$\text{Cost}(\mathbb{B}''_{\mathcal{K}}) = \sum_{j=1}^k F(\gamma'_j) = F(1 - \prod_{j=1}^k (1 - \gamma'_j)).$$

De la Proposition 4.6 et du fait que F est croissante nous avons : $F(1 - \prod_{j=1}^k (1 - \gamma'_j)) < F(1 - \prod_{i=1}^m (1 - \gamma_i))$, c'est-à-dire, $\text{Cost}(\mathbb{B}''_{\mathcal{K}}) < \text{Cost}(\mathbb{B}'_{\mathcal{K}})$. Contradiction. ■

Lorsque les degrés d'incertitude représentent un degré de surprise, à savoir lorsque l'échelle d'incertitude est $\{0, 1 - k^{-1}, 1 - k^{-2}, 1 - k^{-3}, \dots, 1\}$ alors une définition possible de la fonction F est :

$$F(\alpha) = -\log_k(1 - \alpha) \quad (4.7)$$

La valeur $F(\alpha)$ est un entier et la fonction F satisfait à la propriété générale donnée par l'équation 4.2.

Exemple 4.3 Soit $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ une base de connaissances Pb- π -DL-Lite. Soit $\ell = \{1, 1 - 2^{-1}, 1 - 2^{-2}, \dots, 1 - 2^{-n}\}$ l'échelle des degrés de certitude utilisée dans la base assertionnelle \mathcal{A} où :

$$\begin{aligned} \mathcal{T} &= \{\langle A \sqsubseteq \neg B, 1.0 \rangle, \langle A \sqsubseteq \neg C, 1.0 \rangle\} \text{ et} \\ \mathcal{A} &= \{\langle A(b), 1 - 2^{-3} \rangle, \langle B(b), 1 - 2^{-1} \rangle, \langle C(b), 1 - 2^{-4} \rangle\}. \end{aligned}$$

Sémantiquement, le degré d'inconsistance de \mathcal{K} est $\text{Inc}(\mathcal{K}) = 1 - 2^{-3}$.

La fermeture négative $\text{Cln}(\mathcal{T})$ de \mathcal{T} est :

$$\text{Cln}(\mathcal{T}) = \{\langle A \sqsubseteq \neg B, 1.0 \rangle, \langle A \sqsubseteq \neg C, 1.0 \rangle\}.$$

L'ensemble de conflit obtenu en appliquant le Définition 3.8 est :

- $\mathcal{C}_1 = \{\langle A(b), 1 - 2^{-3} \rangle, \langle B(b), 1 - 2^{-1} \rangle\}$,
- $\mathcal{C}_2 = \{\langle A(b), 1 - 2^{-3} \rangle, \langle C(b), 1 - 2^{-4} \rangle\}$

La base de connaissances 2-Horn pondérée obtenue par la fonction $F(\alpha) = -\log_k(1 - \alpha)$ est :

$$\mathbb{B}_{\mathcal{K}} = \{(A_b, 3), (B_b, 1), (C_b, 4), (\neg A_b \vee \neg B_b, \infty), (\neg A_b \vee \neg C_b, \infty)\}.$$

Après l'application de l'outil W-Max-2-Horn-SAT, nous obtenons $\text{cost}(\mathbb{B}_{\mathcal{K}}) = 3$. Le degré d'inconsistance de \mathcal{K} est donc $\text{Inc}(\mathcal{K}) = 1 - 2^{-3}$. Par conséquent, nous vérifions bien que $\text{Inc}(\mathcal{K})$ est récupéré à partir $\mathbb{B}_{\mathcal{K}}$.

Lorsque l'échelle des degrés de certitude utilisée est définie dans l'intervalle $[0, 1]$, c'est clair que la fonction $F(\alpha) = -\log_k(1 - \alpha)$ ne donne pas un nombre entier. Comme nous le verrons ci-après, le degré d'inconsistance $\text{Inc}(\mathcal{K})$, peut être récupéré en utilisant une valeur approximative de $F(\alpha)$.

Soit M un nombre entier qui représente le plus grand nombre de décimaux (nombre de chiffres après la virgule) dans une liste de nombres réels. Nous définissons $\lceil F(\alpha) \rceil^M$ comme étant l'approximation de la valeur $F(\alpha)$ avec une précision égale à M . Par exemple, si $\alpha = 0.3246$ et $M = 2$ alors $\lceil \alpha \rceil^M = 0.32$.

Le nombre M devrait être tel que :
Pour $A, B \subseteq \ell$ tel que $A \neq \emptyset, B \neq \emptyset$:

$$\prod_{\alpha_i \in A} (1 - \alpha_i) < \prod_{\beta_j \in B} (1 - \beta_j) \quad \text{Si et seulement si} \quad \sum_{\alpha_i \in A} \lceil F(\alpha_i) \rceil^M > \sum_{\beta_j \in B} \lceil F(\beta_j) \rceil^M \quad (4.8)$$

Lorsque la valeur M est fixée, pour obtenir des nombres entiers de $\lceil -\log_k(1 - \alpha) \rceil$, il suffit de multiplier $\lceil -\log_k(1 - \alpha) \rceil$ par 10^M , à savoir :

$$F_1(\alpha) = 10^M * \lceil -\log_k(1 - \alpha) \rceil^M$$

Malgré l'utilisation de valeurs approximatives de $-10^M * \log_k(1 - \alpha)$, nous pouvons facilement récupérer le degré d'inconsistance de \mathcal{K} $Inc(\mathcal{K})$, grâce à l'outil W-Max-2-Horn-SAT. En effet, nous rappelons que W-Max-2-Horn-SAT fournit comme résultat la meilleure interprétation I . Par conséquent, nous avons :

$$Inc(\mathcal{K}) = 1 - \prod_{(\phi_i, \alpha_i) \in \mathcal{K}, I \neq \phi_i} (1 - \alpha_i) \quad (4.9)$$

Exemple 4.4 *Considérons à nouveau la base de connaissances DL-Lite de l'exemple 4.3, en considérant l'échelle des degrés de certitude $[0, 1]$.*

$$\mathcal{T} = \{\langle A \sqsubseteq \neg B, 1.0 \rangle, \langle A \sqsubseteq \neg C, 1.0 \rangle\} \text{ et}$$

$$\mathcal{A} = \{\langle A(b), 0.8 \rangle, \langle B(b), 0.23 \rangle, \langle C(b), 0.74 \rangle\}.$$

Sémantiquement, le degré d'inconsistance de \mathcal{K} est $Inc(\mathcal{K}) = 1 - (1 - 0.23) * (1 - 0.74) = 0.7998$.

La fermeture négative $Cln(\mathcal{T})$ de \mathcal{T} est :

$$Cln(\mathcal{T}) = \{\langle A \sqsubseteq \neg B, 1.0 \rangle, \langle A \sqsubseteq \neg C, 1.0 \rangle\}.$$

L'ensemble de conflit obtenu en appliquant la Définition 3.8 est :

- $\mathcal{C}_1 = \{\langle A(b), 0.8 \rangle, \langle B(b), 0.23 \rangle\}$,
- $\mathcal{C}_2 = \{\langle A(b), 0.8 \rangle, \langle C(b), 0.74 \rangle\}$

La base de connaissances 2-Horn pondérée obtenue par la fonction $F(\alpha) = -\log_k(1 - \alpha)$ est la suivante :

$$\mathbb{B}_{\mathcal{K}} = \{ (A_b, -\log(1 - 0.8)), (B_b, -\log(1 - 0.23)), (C_b, -\log(1 - 0.74)), (\neg A_b \vee \neg B_b, \infty), (\neg A_b \vee \neg C_b, \infty) \}.$$

La fonction F ne donne pas des entiers puisque les valeurs de $-\log(1 - 0.8)$, $-\log(1 - 0.23)$ et $-\log(1 - 0.74)$ ne sont pas des nombres entiers. Nous devons d'abord préciser le nombre M et ensuite remplacer $F(\alpha)$ par $10^M * \lceil F(\alpha) \rceil^M$. Le nombre M doit satisfaire l'Equation 4.8 pour toutes les combinaisons de valeurs A et B .

Pour $\ell = \{0.8, 0.74, 0.23\}$ et pour M inférieur ou égal 3, nous avons :
 Pour $M = 3$ et pour $A, B \subseteq \ell$. Nous obtenons :

1. $(1 - 0.74) = 0.26 < (1 - 0.23) = 0.77$ et
 $\lceil -\log(1 - 0.74) \rceil^3 > \lceil -\log(1 - 0.23) \rceil^3$.
2. $(1 - 0.8) = 0.20 < (1 - 0.74) = 0.26$ et
 $\lceil -\log(1 - 0.8) \rceil^3 > \lceil -\log(1 - 0.74) \rceil^3$.
3. $(1 - 0.8) = 0.20 < (1 - 0.23) = 0.26$ et
 $\lceil -\log(1 - 0.8) \rceil^3 > \lceil -\log(1 - 0.23) \rceil^3$.
4. $(1 - 0.8)(1 - 0.74) = 0.052 < (1 - 0.23) = 0.77$ et
 $\lceil -\log(1 - 0.8) - \log(1 - 0.74) \rceil^3 > \lceil -\log(1 - 0.23) \rceil^3$.
5. $(1 - 0.8)(1 - 0.23) = 0.154 < (1 - 0.74) = 0.26$ et
 $\lceil -\log(1 - 0.8) - \log(1 - 0.23) \rceil^3 > \lceil -\log(1 - 0.74) \rceil^3$.
6. $(1 - 0.8) = 0.20 < (1 - 0.74)(1 - 0.23) = 0.2003$ et
 $\lceil -\log(1 - 0.8) \rceil^3 = \lceil -\log(1 - 0.74) - \log(1 - 0.23) \rceil^3$.

Nous remarquons que la contrainte n'est pas satisfaite pour toutes les combinaisons possibles de valeurs pour les sous-ensembles A et B .

En fait, il suffit de prendre $M = 4$ pour satisfaire la contrainte pour toutes les combinaisons de valeurs pour A et B . Pour la combinaison précédente, avec $M = 4$, nous obtenons :

$$(1 - 0.8) = 0.2 < (1 - 0.74)(1 - 0.23) = 0.2003 \quad \text{et}$$

$$\lceil -\log(1 - 0.8) \rceil^4 = 0.6989 > \lceil -\log(1 - 0.74) - \log(1 - 0.23) \rceil^4 = 0.6985.$$

La nouvelle base de connaissances 2-Horn pondérée obtenue par la fonction $10^4 * \lceil F(\alpha) \rceil^4$ est la suivante :

$$F(0.8) = 16094, F(0.23) = 02613 \text{ et } F(0.74) = 13470.$$

$$\mathbb{B}_K = \{ (A_b, 16094), (B_b, 02613), (C_b, 13470), \\ (\neg A_b \vee \neg B_b, \infty), (\neg A_b \vee \neg C_b, \infty) \}.$$

Après l'application de l'outil *W-Max-2-Horn-SAT*, nous obtenons $\text{cost}(\mathbb{B}_K) = 16083$. Le degré d'inconsistance de \mathcal{K} est $\text{Inc}(\mathcal{K}) = 1 - F^{-1}(\text{cost}(\mathbb{B}_K)) = 1 - e^{-1.6083} = 0.799$.

La proposition 4.1 est importante. Elle montre que le degré d'inconsistance de la base de connaissances DL-Lite Pb- π -DL-Lite peut être calculé en utilisant le coût $\text{cost}(\mathbb{B}_K)$ d'une solution du problème Max-2-Horn-SAT sur la base de connaissances pondérée associée. Cela nous donne un moyen pratique pour calculer le degré d'inconsistance en utilisant l'outil *W-Max-2-Horn-SAT*.

4.3.2 L'algorithme du calcul du degré d'inconsistance

Nous rappelons tout d'abord le problème que nous traitons. L'entrée de notre problème est un ensemble de formules pondérées $\mathbb{B} = \{(\psi_i, k_i) : i = 1, \dots, n\}$.

Notre objectif est de résoudre le problème suivant (P) :

P : « Recherche d'un sous-ensemble $X \subseteq \mathbb{B}$, tel que :

- i) $\{\psi_i | (\psi_i, k_i) \in X\}$ est consistant et X contient toutes les clauses dures, ayant le poids ∞ .
 - ii) il n'existe pas $Y \subseteq \mathbb{B}$ tel que $\{\psi_j | (\psi_j, k_j) \in Y\}$ est consistant et contient toutes les clauses dures et $Cost(Y) < Cost(X) : \sum_{(\psi_j, k_j) \notin Y} k_j < \sum_{(\psi_i, k_i) \notin X} k_i$
- Pour résoudre le problème "P", nous utilisons l'outil W-Max-2-Horn-SAT.

- L'entrée est une base \mathbb{B} et un nombre entier positif r
- La sortie une valeur booléenne :
 - L'outil retourne "Vrai" si $\exists Z \subseteq \mathbb{B}$ tel que $\{\psi_i : (\psi_i, k_i) \in Z\}$ est consistant, Z contient toutes les clauses dures et $Cost(Z) = \sum_{(\psi_i, k_i) \notin Z} k_i \leq r$.
 - L'outil retourne "Faux" sinon.
- Lorsque l'outil retourne "Vrai", il renvoie également une interprétation I d'où le sous-ensemble Z est défini de la façon suivante : $Z = \{(\psi_i, k_i) : (\psi_i, k_i) \in \mathbb{B} \text{ and } I \models \psi_i\}$.

L'algorithme utilise une recherche dichotomique. La raison de l'utilisation de la recherche dichotomique est la similarité entre notre problème et celui de la recherche d'une valeur donnée dans un vecteur trié d'entiers. Cependant, nous avons besoin de déterminer la valeur minimale et la valeur maximale.

La valeur minimale correspond à la situation où une consistance est restaurée en supprimant une formule (ψ_i, k_i) ayant le poids minimal k_i . c'est-à-dire :

$$Valeur_Minimale = \min_{(\psi_i, k_i) \in \mathbb{B}, k_i \neq \infty} k_i. \quad (4.10)$$

La valeur maximale possible est obtenue lorsque les formules de \mathbb{B} , autres que les formules dures, sont contradictoires deux à deux : $\forall(\phi_i, k_i) \in \mathbb{B}, \forall(\psi_j, k_j) \in \mathbb{B}$ tel que $k_i \neq \infty$ et $k_j \neq \infty$, nous avons : $\phi_i \wedge \psi_j$ est inconsistant. Dans ce cas, chaque sous-ensemble consistant de \mathbb{B} qui contient les formules dures de \mathbb{B} , peut contenir une seule formule de poids $k_i \neq \infty$. Le degré d'inconsistance dans ce cas est :

$$Valeur_Maximale = \sum_{(\psi_i, k_i) \in \mathbb{B}, k_i \neq \infty} k_i - \max_{(\psi_i, k_i) \in \mathbb{B}, k_i \neq \infty} k_i. \quad (4.11)$$

La valeur recherchée est incluse dans :

$$\ell = \{Valeur_Minimal, \dots, Valeur_Maximal\}.$$

Le calcul du degré d'inconsistance nécessite $\log_2(\sum_{(\psi_i, k_i) \in \mathbb{B}, k_i \neq \infty} k_i - \max_{(\psi_i, k_i) \in \mathbb{B}, k_i \neq \infty} k_i - \min_{(\psi_i, k_i) \in \mathbb{B}, k_i \neq \infty} k_i)$ appels à l'outil W-MAX-2-Horn-SAT.

L'algorithme 1 accepte comme entrée une base de connaissances Pb- π -DL-Lite et retourne son degré d'inconsistance. Cet algorithme calcule l'ensemble des conflits $\zeta \in \langle Cln(\mathcal{T}), \mathcal{A} \rangle$ en utilisant la fonction $Calculer_Conflits(Cln(\mathcal{T}), \mathcal{A})$. Ensuite, un pré-traitement est nécessaire pour assurer que $Inc(\mathcal{K}) \neq 1$, puisque cela signifie qu'il existe au moins un conflit qui implique des clauses dures, représentant ainsi un cas de conflit très fort. La fonction $bool = Verification_Fort_Conflit(\zeta)$ permet de vérifier s'il y a des conflits forts dans ζ . Si de tels conflits existent la fonction retourne "Vrai" sinon elle retourne "Faux". Si la valeur retournée est "Faux", l'algorithme transforme l'ensemble de conflits en une base de connaissances pondérée \mathbb{B}_K en utilisant la Définition 4.2. Ensuite, il appelle l'outil $W_Max_Horn_2_Sat$ sur la base \mathbb{B}_K en variant le nombre r . Cet outil renvoie la valeur "Vrai" s'il existe une interprétation avec un coût plus petit ou égal à r .

Algorithm 1 calcul le degré d'inconsistance

Require: $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$: une base de connaissances Pb- π -DL-Lite.

Ensure: Inc : le degré d'inconsistance.

```

 $\zeta = \text{Calculer\_Conflits}(\text{Cln}(\mathcal{T}), \mathcal{A})$ 
 $r \leftarrow 0$ 
 $bool \leftarrow \text{Vrai}$ 
 $Inc \leftarrow 0$ 
if  $\zeta \neq \emptyset$  then
   $Inc \leftarrow 1$ 
   $bool = \text{Verification\_Fort\_Conflit}(\zeta)$  {Vérifié l'existence d'un conflit fort}
  if  $bool = \text{Faux}$  then
     $\mathbb{B}_K = \text{Transformation\_Vers\_Une\_Base}(\zeta, \mathcal{A})$ 
     $l \leftarrow \min_{(\varphi_i, k_i) \in \mathbb{B}_K} (k_i)$ 
     $u \leftarrow \sum_{(\varphi_i, k_i) \in \mathbb{B}_K \text{ and } k_i \neq \infty} k_i - \max_{(\varphi_i, k_i) \in \mathbb{B}_K \text{ and } k_i \neq \infty} k_i$ 
    while  $(l < u)$  do
       $r \leftarrow (l + u) / 2$ 
      if  $(\text{Solveur\_W\_Max\_Horn\_2\_Sat}(\mathbb{B}_K, r) = \text{Vrai})$  then
         $u \leftarrow r$ 
      else
         $l \leftarrow r + 1$ 
      end if
    end while
  end if
   $Inc \leftarrow F^{-1}(r)$ 
end if
retourne  $(Inc)$ 

```

Exemple 4.5 *Considérons à nouveau la base de connaissances \mathcal{K} de l'Exemple 3.1. La base de connaissances 2-Horn \mathbb{B}_K correspondante en prenant $M = 3$ est la suivante :*

$$\mathbb{B}_K = \{ (Professeur_John, 2995), (Professeur_Marry, 3912), \\ (Etudiant_Jean, 2302), \\ (Enseigne_Jean_Marry, 1609), \\ (Enseigne_Jean_John, 2040), \\ (aTuteur_John_Jean, 1049), \\ (\neg aTuteur_John_Jean \vee \neg Etudiant_Jean, \infty), \\ (\neg aTuteur_John_Jean \vee \neg Professeur_John, \infty), \\ (\neg Enseigne_Jean_Marry \vee \neg Etudiant_Jean, \infty), \\ (\neg Enseigne_Jean_Marry \vee \neg Professeur_Marry, \infty), \\ (\neg Enseigne_Jean_John \vee \neg Etudiant_Jean, \infty), \\ (\neg Enseigne_Jean_John \vee \neg Professeur_John, \infty) \}.$$

L'algorithme consiste à utiliser une recherche dichotomique dans un intervalle allant de la valeur minimale $l = \min(22995, 3912, 1609, 2040, 1049) = 1049$ représentant le degré le plus faible apparaissant dans la base à la valeur maximale $u = \sum(22995, 3912, 1609, 2040, 1049) - \max(22995, 3912, 1609, 2040, 1049) = 13907 - 3912$ qui est la somme de tous les degrés de formules de la base \mathbb{B}_K autre que ∞ et autres que le degré maximal.

Nous rappelons que l'outil $W - Max - 2 - Horn - SAT$ retourne une valeur booléenne et

vérifie si une interprétation I est telle que la somme du coût des formules qu'elle falsifie est inférieure ou égale à un nombre entier positif r .

Cette exemple nécessite 14 itérations afin de récupérer la valeur finale de r .

1. $l = 1049$ et $u = 9995$. $r \leftarrow 5522$. La fonction retourne **Vrai**
2. $l = 1049$ et $u = 5522$. $r \leftarrow 3285$. La fonction retourne **Faux**
3. $l = 3286$ et $u = 5522$. $r \leftarrow 4404$. La fonction retourne **Faux**
4. $l = 4405$ et $u = 5522$. $r \leftarrow 4963$. La fonction retourne **Vrai**
5. $l = 4405$ et $u = 4963$. $r \leftarrow 4684$. La fonction retourne **Faux**
6. $l = 4685$ et $u = 4963$. $r \leftarrow 4824$. La fonction retourne **Vrai**
7. $l = 4685$ et $u = 4824$. $r \leftarrow 4754$. La fonction retourne **Vrai**
8. $l = 4685$ et $u = 4754$. $r \leftarrow 4719$. La fonction retourne **Vrai**
9. $l = 4685$ et $u = 4719$. $r \leftarrow 4702$. La fonction retourne **Vrai**
10. $l = 4685$ et $u = 4702$. $r \leftarrow 4693$. La fonction retourne **Faux**
11. $l = 4694$ et $u = 4702$. $r \leftarrow 4698$. La fonction retourne **Faux**
12. $l = 4694$ et $u = 4698$. $r \leftarrow 4696$. La fonction retourne **Faux**
13. $l = 4697$ et $u = 4698$. $r \leftarrow 4697$. La fonction retourne **Faux**
14. $l = 4698$ et $u = 4698$. nous remarquons que dans cette itération $u = l$. Fin de la boucle.

A la fin de la boucle, nous obtenons la valeur finale de r qui est : $\text{cost}(\mathbb{B}'_K) = 4698$. La dernière étape consiste à appliquer l'inverse de la fonction F pour obtenir une valeur dans l'intervalle $[0, 1]$: $\text{Inc}(\mathcal{K}) = 1 - e^{(-4698 \div 10^3)} = 0.99$

4.3.3 La transformation inverse d'une base de connaissances 2-Horn vers une base Pb- π -DL-Lite

Dans cette sous-section, nous présentons la transformation inverse d'une base de connaissances 2-Horn à une base de connaissances Pb- π -DL-Lite. Nous considérons comme entrée une base de connaissances 2-Horn où toutes les clauses binaires ont le même poids ∞ et toutes les clauses unaires ont un même poids différent de ∞ [29]. La transformation est décrite dans la définition suivante :

Définition 4.3 Soit \mathbb{B}_K une base de connaissances 2-Horn où toutes les clauses binaires ont le même poids ∞ et toutes les clauses unaires ont le même poids $\gamma < \infty$ (elles ont toutes la même importance).

Soit $\nu = \{x_1, x_2, \dots, x_n\}$ l'ensemble des symboles propositionnels présents dans la base \mathbb{B}_K . Chaque $x_i \in V$ est associé à un nouveau concept, notée x_i . Soit $\gamma \in]0, 1[$ tel que $\lambda = F^{-1}(\gamma) = 1 - 2^{(-\gamma)}$. Soit a l'individu unique utilisé dans la Abox \mathcal{A} .

La base de connaissances DL-Lite $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ obtenue de \mathbb{B}_K est construite comme suit : Pour chaque $\varphi \in \mathbb{B}_K$:

- Lorsque φ est de la forme $(\neg\phi \vee \neg\psi, \infty)$ alors ajouter à \mathcal{T} l'axiome : $(\phi \sqsubseteq \neg\psi, 1)$.
- Lorsque φ est de la forme $(\neg\phi \vee \psi, \infty)$ alors ajouter à \mathcal{T} l'axiome : $(\phi \sqsubseteq \psi, 1)$.
- Lorsque φ est de la forme (ϕ, γ) alors ajouter à \mathcal{A} l'assertion : $(\phi(a), \lambda)$.

Exemple 4.6 Soit \mathbb{B}_K une base de connaissances 2-Horn.

$\mathbb{B}_K = \{(B \vee \neg C, \infty); (B, \gamma); (C, \gamma); (A \vee \neg D, \infty); (E, \gamma)\}$.

La base de connaissances DL-Lite correspondante $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ est la suivante :

$\mathcal{T} = \{ \langle C \sqsubseteq B, 1 \rangle, \langle D \sqsubseteq A, 1 \rangle \}$.

$\mathcal{A} = \{ \langle C(a), \lambda \rangle, \langle E(a), \lambda \rangle, \langle B(a), \lambda \rangle \}$.

Proposition 4.2 soit \mathbb{B}_K une base de connaissances 2-Horn où toutes les clauses binaires sont certaines et toutes les clauses unaires ne sont pas certaines et ont le même degré de certitude. Soit \mathcal{K} la base de connaissances Pb- π -DL-Lite associée. $\text{Inc}(\mathcal{K}) = \alpha$ si et seulement s'il existe un sous-ensemble consistant $\mathbb{B}'_K \subseteq \mathbb{B}_K$ tel que $\text{Cost}(\mathbb{B}'_K) = F(\alpha)$ et pour chaque sous-ensemble consistant $\mathbb{B}''_K \subseteq \mathbb{B}_K$, $\text{Cost}(\mathbb{B}''_K) \geq \text{Cost}(\mathbb{B}'_K)$.

La preuve est similaire à celle de la Proposition 4.1. Cette proposition montre que, contrairement à la DL-lite possibiliste basée sur le min, définie dans [13], où la réponse à une requête se fait dans un temps polynomial, ce problème est traitable dans le cadre de la théorie des possibilités basée sur le produit. En effet, le problème de calcul du degré d'inconsistance nécessite un nombre logarithmique d'appels au problème de décision du *weightedMax-2-Horn-SAT* qui est NP-Complet. Ainsi, il est important de fournir des algorithmes approximatifs pour répondre aux requêtes.

4.4 Conclusion

Dans ce chapitre, nous avons rappelé le problème W-Max-2-Horn-SAT qui a été utilisé par la suite comme outil pour le calcul exact du degré d'inconsistance d'une base de connaissances Pb- π -DL-Lite. Pour cela, nous avons montré comment coder une base de connaissances Pb- π -DL-Lite comme base de clauses 2-Horn pondérées avant de montrer que le calcul du degré d'inconsistance peut être effectué à l'aide d'une procédure de recherche dichotomique évoquant à chaque itération une instance du problème W-Max-2-Horn-SAT. De plus, nous avons également montré la transformation inverse, c'est-à-dire le codage d'une base de clauses 2-Horn en une théorie DL-Lite. Ceci nous a permis de montrer que le problème du calcul du degré d'inconsistance d'une base de connaissances Pb- π -DL-Lite est un problème NP-Difficile puisqu'il nécessite un nombre logarithmiques d'appels au problème *W-MAX-2-Horn-SAT* qui est NP-Hard.

Chapitre 5

Une approche approximative pour Pb- π -DL-Lite à base de couverture par ensembles

5.1 Introduction

Dans le chapitre précédent, nous avons présenté l'algorithme pour calculer le degré d'inconsistance d'une base de connaissances Pb- π -DL-Lite. Cet algorithme nécessite un nombre logarithmique d'appels au problème de décision du *weightedMax - 2 - Horn - SAT* qui est NP-Complet. Comme l'approche exacte est NP-Difficile, une approche approximative est alors nécessaire afin de répondre aux requêtes conjonctives en un temps raisonnable. L'approche que nous présentons dans ce chapitre permet de calculer efficacement la valeur approximative du degré d'inconsistance et de répondre aux requêtes de vérification d'instance. Cette approche repose sur la modélisation du problème de calcul du degré d'inconsistance par un problème de couverture par ensembles pondérée.

Tout d'abord nous rappelons la définition du problème de couverture par ensembles pondérée (W-SCP). Ensuite, nous présentons la modélisation du problème du calcul de degré d'inconsistance en tant que problème de couverture par ensembles pondérée puis nous présentons l'algorithme approximatif proposé qui repose sur le principe d'un algorithme glouton polynomial et qui est bien connu comme un algorithme approximatif pour la résolution du W-SCP. Ce codage nous permet aussi de fournir un algorithme approximatif pour répondre aux requêtes de vérification d'instance. La dernière section est consacrée aux études expérimentales des deux algorithmes approximatifs : le premier concerne le calcul du degré d'inconsistance et le second, basé sur le premier algorithme, concerne la vérification d'instance.

5.2 Problème de couverture par ensembles

Le problème de couverture par ensembles est l'un des problèmes les plus anciens et les plus étudiés dans le domaine de l'optimisation combinatoire. Il a une forte valeur théorique et est également utilisé dans des applications pratiques telles que le test de dispositifs VLSI ou l'agencement d'équipage.

Le problème de couverture par ensembles est l'un des 21 problèmes NP-Complete de Krap

[74].

Définition 5.1 (Couverture par ensembles) *Étant donné un univers U de n éléments et une collection S de sous-ensembles de U : $S = \{S_1, \dots, S_k\}$. L'objectif est de trouver un sous-ensemble de S de cardinalité minimale qui couvre tous les éléments de U .*

Exemple 5.1 *Nous considérons un ensemble de cinq éléments à couvrir : $U = \{1, 2, 3, 4, 5\}$. Soit $S = \{\{1, 2\}, \{3, 4\}, \{4, 5\}, \{1, 2, 3\}\}$ une collection de sous-ensembles. nous essayons alors de couvrir tous les éléments de U avec un sous-ensemble de S . Par exemple $\{1, 2\}, \{3, 4\}, \{4, 5\}$ est une couverture, puisque chaque élément de U est dans au moins un de ces sous-ensembles. Nous pouvons vérifier que la couverture qui utilise le moins de sous-ensembles de S est $\{4, 5\}, \{1, 2, 3\}$.*

Le problème de couverture par ensembles pondérée "W-SCP" est une variante du problème de couverture par ensembles où les sous-ensembles de la collection utilisée ont des poids ou des coûts. L'idée est alors de trouver l'ensemble avec un coût minimal. La version d'optimisation du problème de couverture par ensembles pondère "W-SCP" est définie comme suit :

Définition 5.2 *Étant donné une collection ξ de sous-ensembles d'un univers U tel que, à chaque ensemble $s \in \xi$ est associé un poids $w_s \geq 0$.*

Un ensemble couvrant C est une sous collection de l'ensemble ξ telle que : $\bigcup_{s \in C} s = U$. L'objectif est de trouver un ensemble couvrant C de poids total $\sum_{s \in C} w_s$ minimum.

L'un des meilleurs algorithmes polynomiaux connus pour l'approximation de la solution du problème de couverture par ensembles pondérée est l'algorithme glouton.

L'algorithme glouton construit une couverture en choisissant à plusieurs reprises un ensemble s qui minimise le rapport entre le poids w_s de s et le nombre d'éléments de s qui ne sont pas encore couverts par les ensembles choisis (ou de façon équivalente, maximiser le rapport entre le nombre d'éléments de S et le poids w_s de s). Il a été montré dans [123] que l'algorithme glouton s'arrête et renvoie une couverture de poids définie au plus H_k fois le poids minimum de n'importe quelle couverture où H_k désigne le nombre harmonique de k^{th} donné par :

$$H(k) = \sum_{i=1}^k \frac{1}{i} \leq \ln(k) + 1.$$

Exemple 5.2 *Nous considérons un ensemble de cinq éléments à couvrir : $U = \{1, 2, 3, 4, 5\}$. Soit la collection S de sous-ensembles de U avec $S = \{\{1, 2\}, \{3, 4\}, \{4, 5\}, \{1, 2, 3\}\}$. A chaque sous-ensemble s de S , un poids W_s est associé comme suit : $W_{\{1,2\}} = 2$, $W_{\{3,4\}} = 1$, $W_{\{4,5\}} = 4$ et $W_{\{1,2,3\}} = 1$.*

Le but est de couvrir tous les éléments de U avec des sous-ensembles de S ayant un coût total minimum. Par exemple, $\{1, 2\}, \{3, 4\}, \{4, 5\}$ est un ensemble couvrant puisque chaque élément de U est dans au moins un de ces sous-ensembles. Le poids de cet ensemble couvrant est égal à 7.

En revanche, nous pouvons vérifier que le sous-ensemble $\{\{4, 5\}, \{1, 2, 3\}\}$ est un ensemble couvrant ayant le poids minimum qui est égal à 5.

5.3 L'approche approximative pour calculer le degré d'inconsistance

Dans cette section, nous présentons l'encodage du problème du calcul du degré d'inconsistance d'une base de connaissances Pb- π -DL-Lite en un problème de couverture par ensembles pondérée.

5.3.1 La modélisation du problème

Définition 5.3 Soit $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ une base de connaissances Pb- π -DL-Lite. Soient ζ l'ensemble de tous les conflits dans \mathcal{A} et M un nombre naturel suffisamment grand. Soit F une fonction de changement d'échelle définie par : $F(x) = -10^M * (\ln(1 - x))$.

La base de couverture par ensembles pondérée associée à \mathcal{K} est désignée par $SCP_{\mathcal{K}}$ et définie comme suit :

- L'univers est l'ensemble des conflits ζ .
- Pour chaque assertion $(X(\vec{\alpha}), \alpha)$ de \mathcal{A} est associé un ensemble noté $X_{\vec{\alpha}}$ ayant le poids $w_{X_{\vec{\alpha}}} = F(\alpha)$ et contenant les conflits de ζ impliquant $X(\vec{\alpha})$: $X_{\vec{\alpha}} = \{C \in \zeta \mid X(\vec{\alpha}) \in C\}$.
- la collection S est formée par les ensembles construits à partir de toutes les assertions de \mathcal{A} :

$$S = \{X_{\vec{\alpha}} \mid (X(\vec{\alpha}), \alpha) \in \mathcal{A}\} \quad (5.1)$$

À partir de la Définition 5.3, il est clair que la complexité en temps et en espace pour définir la collection d'ensembles S est linéaire.

Exemple 5.3 Nous considérons à nouveau l'Exemple 3.1. L'ensemble initial des conflits est $\zeta = \{C_1, C_2, C_3, C_4, C_5, C_6\}$ avec :

- $C_1 = \{\langle \text{Professeur}(\text{John}), 0.95 \rangle, \langle \text{Enseigne}(\text{Jean}, \text{John}), 0.87 \rangle\}$,
- $C_2 = \{\langle \text{Etudiant}(\text{Jean}), 0.90 \rangle, \langle \text{Enseigne}(\text{Jean}, \text{John}), 0.87 \rangle\}$,
- $C_3 = \{\langle \text{Professeur}(\text{John}), 0.95 \rangle, \langle \text{aTuteur}(\text{John}, \text{Jean}), 0.65 \rangle\}$,
- $C_4 = \{\langle \text{Etudiant}(\text{Jean}), 0.90 \rangle, \langle \text{aTuteur}(\text{John}, \text{Jean}), 0.65 \rangle\}$
- $C_5 = \{\langle \text{Etudiant}(\text{Jean}), 0.90 \rangle, \langle \text{Enseigne}(\text{Jean}, \text{Marry}), 0.8 \rangle\}$,
- $C_6 = \{\langle \text{Professeur}(\text{Marry}), 0.98 \rangle, \langle \text{Enseigne}(\text{Jean}, \text{Marry}), 0.8 \rangle\}$

L'ensemble $B = \{B_1, B_2, B_3, B_4, B_5, B_6\}$ contient toutes les assertions impliquées dans l'ensemble des conflits ζ avec :

- $B_1 = \langle \text{Professeur}(\text{John}), 0.95 \rangle$,
- $B_2 = \langle \text{Professeur}(\text{Marry}), 0.98 \rangle$,
- $B_3 = \langle \text{Etudiant}(\text{Jean}), 0.90 \rangle$,
- $B_4 = \langle \text{Enseigne}(\text{Jean}, \text{John}), 0.87 \rangle$,
- $B_5 = \langle \text{Enseigne}(\text{Jean}, \text{Marry}), 0.8 \rangle$,
- $B_6 = \langle \text{aTuteur}(\text{John}, \text{Jean}), 0.65 \rangle$.

Pour chaque assertion $B_i \in B$ ($1 \leq i \leq 6$) est associé un ensemble s_i contenant les conflits impliquant B_i avec le poids correspondant.

La collection de ces ensembles est $S = \{s_1, s_2, s_3, s_4, s_5, s_6\}$ où :

- $s_1 = \{C_1, C_3\}$ est l'ensemble associé à l'assertion B_1 . Son poids est $w_{s_1} = w_{B_1} = F(0.95) = 2995$.
- $s_2 = \{C_6\}$ et $w_{s_2} = F(0.98) = 3912$;
- $s_3 = \{C_2, C_4, C_5\}$ et $w_{s_3} = F(0.90) = 2302$;
- $s_4 = \{C_1, C_2, C_3\}$ et $w_{s_4} = F(0.87) = 2040$;
- $s_5 = \{C_5, C_6\}$ et $w_{s_5} = F(0.80) = 1609$.
- $s_6 = \{C_3, C_4\}$ et $w_{s_6} = F(0.65) = 1049$.

La Proposition 5.1 montre que le calcul du degré d'inconsistance d'une base de connaissances Pb- π -DL-Lite se résume à trouver une solution au problème de couverture par ensembles pondérée correspondant.

Proposition 5.1 Soient \mathcal{K} une base de connaissances Pb- π -DL-Lite, ζ l'ensemble de ses conflits et $SCP_{\mathcal{K}}$ l'instance W-SCP donnée par la Définition 5.3.

$Inc(\mathcal{K}) = \alpha$ si et seulement s'il existe une couverture C avec un poids total minimal $\sum_{s \in C} w_s = F(\alpha)$.

Preuve 5.1 La preuve de cette proposition se fait en deux parties :

• **La 1^{er} partie**

Soit $Inc(\mathcal{K}) = \alpha$. Soit J une interprétation ayant le degré de possibilité maximal, c'est-à-dire, $\forall I \in \Omega$, si $I \neq J$ alors $\pi_{\mathcal{K}}(J) \geq \pi_{\mathcal{K}}(I)$.

Soient $\{(D_1(\vec{a}_1), \gamma_1), \dots, (D_m(\vec{a}_m), \gamma_m)\} \in A$ toutes les assertions telles que $J \not\models D_i(\vec{a}_i)$.
 $Inc(\mathcal{K}) = \alpha = 1 - (1 - \gamma_1) \dots (1 - \gamma_m)$.

Puisque pour $1 \leq i \leq m$, $J \not\models D_i(\vec{a}_i)$ si et seulement si $C = \{(D_i(\vec{a}_i), \gamma_i), (B_i(\vec{a}_i), \beta_i)\}$ est un conflit entre $D_i(\vec{a}_i)$ et une certaine assertion $B_i(\vec{a}_i)$, il s'ensuit que la collection $C = \{D_{i_{\vec{a}_i}} \mid 1 \leq i \leq m\}$ est un ensemble couvrant (rappelons que $D_{i_{\vec{a}_i}}$ indique l'ensemble des conflits impliquant $D_i(\vec{a}_i)$).

Maintenant, supposons qu'il existe une collection C' d'ensembles de conflits tel que C' est un ensemble couvrant et $\sum_{s' \in C'} w_{s'} < \sum_{s \in C} w_s$. Alors, $C' = \{D'_{j_{\vec{a}'_j}} \mid 1 \leq j \leq k\}$

pour un certain $k > 0$ où $(D'_j(\vec{a}'_j), \gamma'_j)$ sont des assertions de \mathcal{A} et à partir du fait que $\sum_{s' \in C'} s' < \sum_{s \in C} s$ nous avons :

$F(\gamma'_1) + \dots + F(\gamma'_k) < F(\gamma_1) + \dots + F(\gamma_m)$ ce qui signifie que :

$$(1 - \gamma_1) \times \dots \times (1 - \gamma_m) < (1 - \gamma'_1) \times \dots \times (1 - \gamma'_k) \quad (5.2)$$

Soit $J' \in \Omega$ une interprétation qui falsifie exactement $\{D'_j(\vec{a}'_j) \mid 1 \leq j \leq k\}$. D'après (5.2), nous obtenons $\pi_{\mathcal{K}}(J') > \pi_{\mathcal{K}}(J)$, ce qui contredit le fait que α soit le degré de consistance de \mathcal{K} .

• **La 2^{eme} partie**

Supposons que C est un ensemble couvrant de poids total minimal $\sum_{s \in C} w_s$. Il existe alors $(D_1(\vec{a}_1), \gamma_1), \dots, (D_m(\vec{a}_m), \gamma_m) \in A$ tel que : $C = \{D_{i_{\vec{a}_i}} \mid 1 \leq i \leq m\}$.

Soit I une interprétation qui falsifie exactement $\{D_i(\vec{a}_i) \mid i \in \{1, \dots, m\}\}$, alors : $\pi_{\mathcal{K}}(I) = (1 - \gamma_1) \dots (1 - \gamma_m)$.

Supposons que α n'est-ce pas le degré d'incohérence de \mathcal{K} . Alors, il existe une interprétation $I' \in \Omega$ tel que $\pi_{\mathcal{K}}(I') > \pi_{\mathcal{K}}(I)$.

Il s'ensuit qu'il y a $(D'_1(\vec{a}'_1), \gamma'_1), \dots, (D'_k(\vec{a}'_k), \gamma'_k) \in A$ tel que $\{D'_i(\vec{a}'_i) \mid 1 \leq i \leq k\}$ sont exactement les assertions falsifiées par I' . Nous avons :

$\pi_{\mathcal{K}}(I') = (1 - \gamma'_1) \dots (1 - \gamma'_k) > \pi_{\mathcal{K}}(I) = (1 - \gamma_1) \dots (1 - \gamma_m)$. Alors :

$C' = \{D'_{i_{\vec{a}'_i}} \mid 1 \leq i \leq k\}$ est un ensemble couvrant mais du fait que $\pi_{\mathcal{K}}(I') > \pi_{\mathcal{K}}(I)$, nous

pouvons vérifier que : $\sum_{s' \in C'} w_{s'} < \sum_{s \in C} w_s$.

Ce résultat a contredit le fait que C est un ensemble couvrant de poids total minimal ■

La Proposition 5.1 fournit un moyen pratique et efficace pour calculer une valeur approximative du degré d'inconsistance d'une base de connaissances Pb- π -DL-Lite en appliquant l'algorithme glouton du problème de couverture par ensembles pondérée.

5.3.2 Algorithme pour le calcul du degré d'inconsistance

L'algorithme 6 montre comment calculer le degré d'inconsistance d'une base de connaissances Pb- π -DL-Lite à l'aide de la méthode approximative glouton pour le problème de couverture par ensembles pondérée [123].

L'algorithme accepte comme entrée une base de connaissances Pb- π -DL-Lite et retourne la valeur approximative de son degré d'inconsistance. Cet algorithme calcule l'ensemble des conflits $\zeta \in \langle \text{Cln}(\mathcal{T}), \mathcal{A} \rangle$. Ensuite, il extrait l'ensemble B de toutes les assertions impliquées dans ζ , et pour chaque assertion, il construit un ensemble ayant comme poids celui de l'assertion et contenant tous les conflits impliquant cette assertion. Il applique la méthode glouton afin de trouver l'ensemble couvrant de poids minimal et retourne la valeur de ce poids.

Algorithm 2 Méthode approximative pour le calcul du degré d'inconsistance

Require: $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$: une base de connaissances Pb- π -DL-Lite

Ensure: Inc : Valeur approximative du degré d'incohérence de \mathcal{K}

$\zeta \leftarrow \text{Calculer_Conflits}(\mathcal{T}, \mathcal{A})$; {Calculer les conflits}

$Cost \leftarrow 0$;

$B \leftarrow \text{Extract_Assertions}(\zeta)$; {Extrait les assertions se produisant dans ζ }

while ($B \neq \emptyset$) **do**

$max \leftarrow 0$;

for ($x_i \in B$) **do**

$Nb_conf \leftarrow \text{Calculer_Nb_conf}(\zeta, x_i)$;

if ($max < Nb_conf/w_{x_i}$) **then**

$max \leftarrow Nb_conf/w_{x_i}$;

$\phi \leftarrow x_i$;

end if

end for

$Cost \leftarrow Cost + w_\phi$; { Nous prenons l'assertion ϕ qui maximise le rapport entre le nombre de conflits qui l'impliquent et son poids }

$\zeta \leftarrow \text{Supprimer_Conflit}(\zeta, \phi)$; {Supprimer de ζ tous les conflits impliquant ϕ }

$B \leftarrow \text{Extract_Assertions}(\zeta)$; {Recalculer l'ensemble des assertions qui restent impliquées dans le nouvel ensemble réduit de conflits}

end while

$Inc \leftarrow F^{-1}(Cost)$;

return(Inc)

Exemple 5.4 Nous continuons avec l'Exemple 3.1. Nous appliquons l'algorithme 6 comme suit :

— **La première itération.** Nous avons :

— $B = \{B_1, B_2, B_3, B_4, B_5, B_6\}$,

— $\zeta = \{C_1, C_2, C_3, C_4, C_5, C_6\}$ et

— $Cost = 0$.

Pour chaque B_i , nous calculons $|s_i| \div w_{s_i}$ et nous obtenons :

— Pour B_1 nous avons : $|s_1| \div w_{s_1} = 2 \div 2995$

— Pour B_2 nous avons : $2 \div 3912$

— Pour B_3 nous avons : $3 \div 2302$

— Pour B_4 nous avons : $3 \div 2040$

— Pour B_5 nous avons : $2 \div 1609$

— Pour B_6 nous avons : $2 \div 1049$

Nous pouvons vérifier que B_6 a la valeur maximale. Nous ajoutons à la variable $Cost$ le poids w_{s_6} . Sa nouvelle valeur est alors : $Cost = 1049$.

L'étape suivante consiste à supprimer de ζ tous les conflit impliquant B_6 .

— **La deuxième itération**

— $B = \{B_1, B_2, B_3, B_4, B_5\}$,

— $\zeta = \{C_1, C_2, C_5, C_6\}$ et

— $Cost = 1049$.

A cette étape, B_5 maximise $|s_i| \div w_{s_i}$. La nouvelle valeur de $Cost$ est de $1049 + 1609$.

Nous supprimons de l'ensemble de conflit courant ζ tous les conflit impliquant B_5 .

— **La troisième itération**

— $B = \{B_1, B_3, B_4\}$,

— $\zeta = \{C_1, C_2\}$ et

— $Cost = 1049 + 1609$.

B_3 maximise $|s_i| \div w_{s_i}$. La nouvelle valeur de $Cost$ est de $1049 + 1609 + 2040$.

Nous supprimons de l'ensemble de conflit courant ζ tous les conflit impliquant B_3 .

— **La quatrième itération**

— $B = \{\emptyset\}$,

— $\zeta = \{\emptyset\}$ et

— $Cost = 1049 + 1609 + 2040$.

L'ensemble de conflits est vide, la dernière étape consiste à calculer le degré d'inconsistance de \mathcal{K} en utilisant la fonction inverse de F . Nous obtenons $Inc(\mathcal{K}) = F^{-1}(Cost) = 1 - e^{-Cost/10^3} = 1 - 0.009 = 0.99$.

Nous remarquons que c'est le même résultat obtenu en appliquant l'approche exacte.

5.4 Algorithme approximatif pour la réponse aux requêtes de vérification d'instance

Comme indiqué dans la Proposition 3.1, répondre à une requête de vérification d'instance revient à calculer les degrés d'inconsistance des deux bases de connaissances. Le degré d'inconsistance de la base de connaissances Pb- π -DL-Lite initiale est calculé une seule fois, alors que chaque requête nécessite un calcul du degré d'inconsistance de la seconde base de connaissances [15].

Pour des raisons d'efficacité, nous préférons utiliser, autant que possible, l'algorithme approximatif 6 plutôt que l'algorithme exact.

Une méthode approximative pour calculer un degré d'inconsistance nous permet d'avoir immédiatement un algorithme approximatif pour la réponse aux requêtes.

L'algorithme approximatif prend comme entrées une base de connaissances Pb- π -DL-Lite $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, une requête de vérification d'instance q et le degré d'inconsistance de la base \mathcal{K} $Inc(\mathcal{K})$. En effet, cet algorithme est appelé pour chaque nouvelle requête. Il retourne comme sortie une réponse booléenne (*Oui* ou *Non*) [15].

Exemple 5.5 Soient *Eleve* (resp. *Enseignant*, *Directeur*) des concepts qui représentent les individus qui sont des élèves (resp. enseignants ou directeurs). Soit *enseigne_a* le rôle qui relie les enseignants à leurs élèves. Soient a , b et c trois individus.

Considérons la base de connaissances Pb- π -DL-Lite \mathcal{K} composée de la TBox \mathcal{T} et de la ABox \mathcal{A} suivantes :

Algorithm 3 Algorithme approximatif pour la réponse aux requêtes

Require: $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$: une base de connaissance Pb- π -DL-Lite ;

q : requête ;

Inc : Le degré d'inconsistance de K

Ensure: Réponse boolean : Oui / Non.

$\mathcal{K}_1 = \langle \mathcal{T}_1, \mathcal{A}_1 \rangle \leftarrow K \cup \neg Q$;

$\zeta \leftarrow \text{Calculer_Conflits}(\mathcal{T}, \mathcal{A})$;

$\zeta_1 \leftarrow \text{Calculer_Conflits}(\mathcal{T}_1, \mathcal{A}_1)$;

$Free \leftarrow$ l'ensemble des assertions de \mathcal{A} non impliquées dans ζ

if ($|\zeta_1| = |\zeta|$) **then**

$Reponse \leftarrow Non$; $\{Q$ n'est pas une conséquence de $\mathcal{K}\}$

else

if (ζ Implique des éléments de $Free$) **then**

$Reponse \leftarrow Oui$; $\{Q$ est une conséquence de $\mathcal{K}\}$

else

$Inc_1 = \text{Methode_approximative_calcul_degre_Inconstance}(\mathcal{T}_1, \mathcal{A}_1)$;

if ($Inc_1 > Inc$) **then**

$Reponse \leftarrow Oui$;

else

$Reponse \leftarrow Non$;

end if

end if

end if

return ($Reponse$);

$\mathcal{T} = \{ \langle Eleve \sqsubseteq \neg Enseignant, 1.0 \rangle,$
 $\langle Directeur \sqsubseteq Enseignant, 1.0 \rangle,$
 $\langle \exists enseigne_a \sqsubseteq Enseignant, 1.0 \rangle,$
 $\langle \exists enseigne_a^- \sqsubseteq Eleve, 1.0 \rangle \}.$

$\mathcal{A} = \{ \langle Enseignant(b), 0.85 \rangle, \langle Eleve(a), 0.95 \rangle, \langle enseigne_a(a, b), 0.75 \rangle$
 $\langle Directeur(c), 0.25 \rangle \}.$

La fermeture négative $Cln(\mathcal{T})$ de \mathcal{T} est :

$Cln(\mathcal{T}) = \{ \langle Eleve \sqsubseteq \neg Enseignant, 1.0 \rangle,$
 $\langle \exists enseigne_a \sqsubseteq \neg Eleve, 1.0 \rangle,$
 $\langle \exists enseigne_a^- \sqsubseteq \neg Enseignant, 1.0 \rangle,$
 $\langle Directeur \sqsubseteq \neg Eleve, 1.0 \rangle,$
 $\langle \exists enseigne_a^- \sqsubseteq \neg Directeur, 1.0 \rangle \}.$

L'ensemble de conflits est $\zeta = \{C_1, C_2\}$ tels que :

- $C_1 = \{ \langle Enseignant(b), 0.85 \rangle, \langle enseigne_a(a, b), 0.75 \rangle \},$
- $C_2 = \{ \langle Eleve(a), 0.95 \rangle, \langle enseigne_a(a, b), 0.75 \rangle \}.$

Nous pouvons facilement vérifier que l'interprétation qui a le degré de possibilité maximal, ne falsifie que l'assertion $\langle enseigne_a(a, b), 0.75 \rangle$.

En utilisant la définition sémantique (Definition 3.5), le degré d'inconsistance de \mathcal{K} est : $Inc(\mathcal{K}) = 0.75$.

Nous pouvons vérifier que la valeur du degré d'inconsistance de \mathcal{K} obtenue en appliquant aussi bien l'algorithme exact que celui approximatif est aussi : $Inc(\mathcal{K}) = 0.75$.

Considérons la requête suivante :

$$q \leftarrow \text{Enseignant}(c)$$

Nous introduisons un nouveau concept $D_{\text{Enseignant}}$ et nous construisons la nouvelle base de connaissances $\mathcal{K}_1 = \langle \mathcal{T}_1, \mathcal{A}_1 \rangle$ avec :

- $\mathcal{T}_1 = \mathcal{T} \cup \{ \langle D_{\text{Enseignant}} \sqsubseteq \neg \text{Enseignant}, 1.0 \rangle \}$ et
- $\mathcal{A}_1 = \mathcal{A} \cup \{ \langle D_{\text{Enseignant}}(c), 1.0 \rangle \}$.

La base de connaissances \mathcal{K}_1 ajoute à \mathcal{K} le fait que $\text{Enseignant}(c)$ est certainement fausse. En utilisant la définition sémantique (Définition 3.5), le degré d'incohérence de la base augmentée \mathcal{K}_1 est $\text{Inc}(\mathcal{K}_1) = 0.81$.

Nous avons : $\text{Inc}(\mathcal{K}_1) > \text{Inc}(\mathcal{K})$. Nous déduisons alors que q est une conséquence de \mathcal{K} .

Après l'application de l'algorithme approximatif, le nombre de conflits dans ζ_1 de la base augmentée \mathcal{K}_1 est supérieur au nombre de conflits dans ζ de la base initiale \mathcal{K} . De plus, l'ensemble ζ_1 contient une nouvelle variable $\langle \text{Directeur}(c), 0.25 \rangle$. Ce qui permet de déduire que la requête q est une conséquence de \mathcal{K} .

5.5 Étude expérimentale

Dans cette section, nous exposons quelques expérimentations que nous avons effectuées afin de pouvoir évaluer les performances de notre outil développé pour le calcul du degré d'inconsistance et la réponse aux requêtes [29].

Pour l'implémentation des algorithmes, nous avons utilisé le langage de programmation JAVA qui est à la fois un langage de programmation et un environnement d'exécution sous Windows.

Plusieurs traitements sont nécessaires afin de pouvoir calculer le degré d'inconsistance. Ces traitements sont basés sur un ensemble d'algorithmes. Le système de calcul du degré d'inconsistance que nous avons implémenté, propose de calculer cette décision selon deux algorithmes. Afin de faciliter la lecture de ce chapitre, nous allons nommer :

- Le premier algorithme : l'algorithme exact basé sur l'outil W-Max-2-horn-Sat (décrit dans le chapitre 4, section 3).
- Le deuxième algorithme : l'algorithme approximatif basé sur l'algorithme glouton.

Le schéma suivant, permet d'expliquer de façon simple le fonctionnement de notre système.

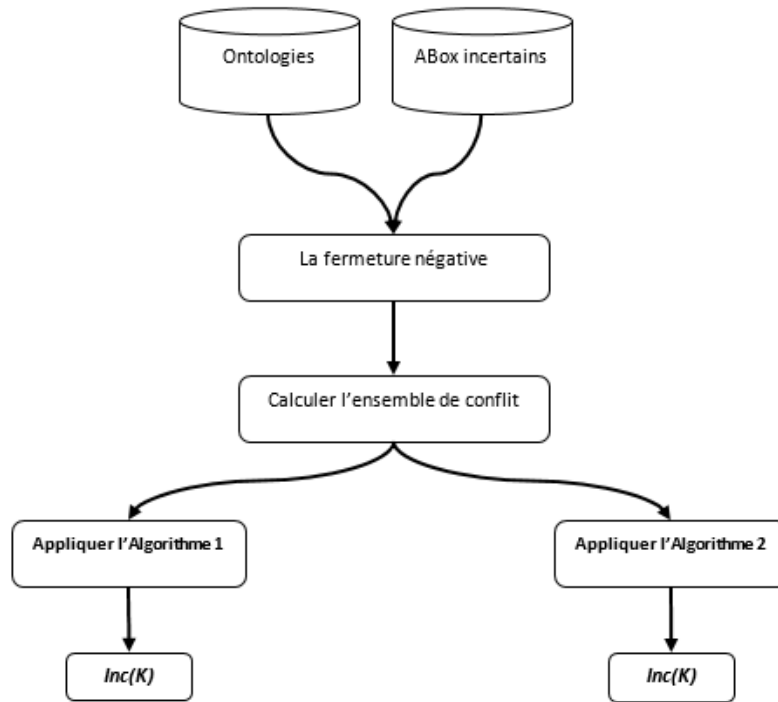


FIGURE 5.1 – Le système de calcul du degré d'inconsistance

5.5.1 Évaluation de l'algorithme approximatif

Pour évaluer la qualité des résultats donnés par le deuxième algorithme, nous avons effectué une étude expérimentale qui compare la valeur exacte du degré d'incohérence donné par l'Algorithme 1 et sa valeur approximative calculée par l'Algorithme 2. Plus précisément :

- Nous avons généré 400 instances aléatoires de bases de connaissances Pb- π -DL-Lite.
- Pour chaque base de connaissances, \mathcal{K} , nous avons calculé la valeur exacte du degré d'inconsistance $Inc(\mathcal{K})$ ainsi que sa valeur approximative.

Les résultats obtenus sont représentés dans la Figure 5.2. L'axe des X représente le nombre de chiffres après la virgule. Pour une abscisse x donné, l'ordonnée y correspondante sur l'axe des Y est le taux de réussite, c'est-à-dire le pourcentage où les deux algorithmes donnent le même résultat. D'après la figure 5.2, nous remarquons que les résultats sont très encourageants. En effet, pour un chiffre après la virgule, nous avons obtenu d'excellents résultats : environ 99,8% de succès total. Pour deux chiffres, nous avons obtenu environ 94,4% de succès total. Même en prenant huit chiffres après la virgule, le score reste élevé (81.2%).

Les résultats de la première expérience confirment bien notre objectif de validation et sont encourageants. Une deuxième expérience a été menée afin d'évaluer la qualité de l'algorithme 1 pour répondre à des requêtes de vérification d'instance.

Nous avons généré au hasard un grand nombre de paires (base de connaissances, requête). Nous avons divisé ces paires en trois classes en fonction du nombre de conflits présents dans les bases générées :

- **petite** : le nombre de conflits est inférieur à 50.
- **moyenne** : le nombre de conflits est entre 50 et 150.

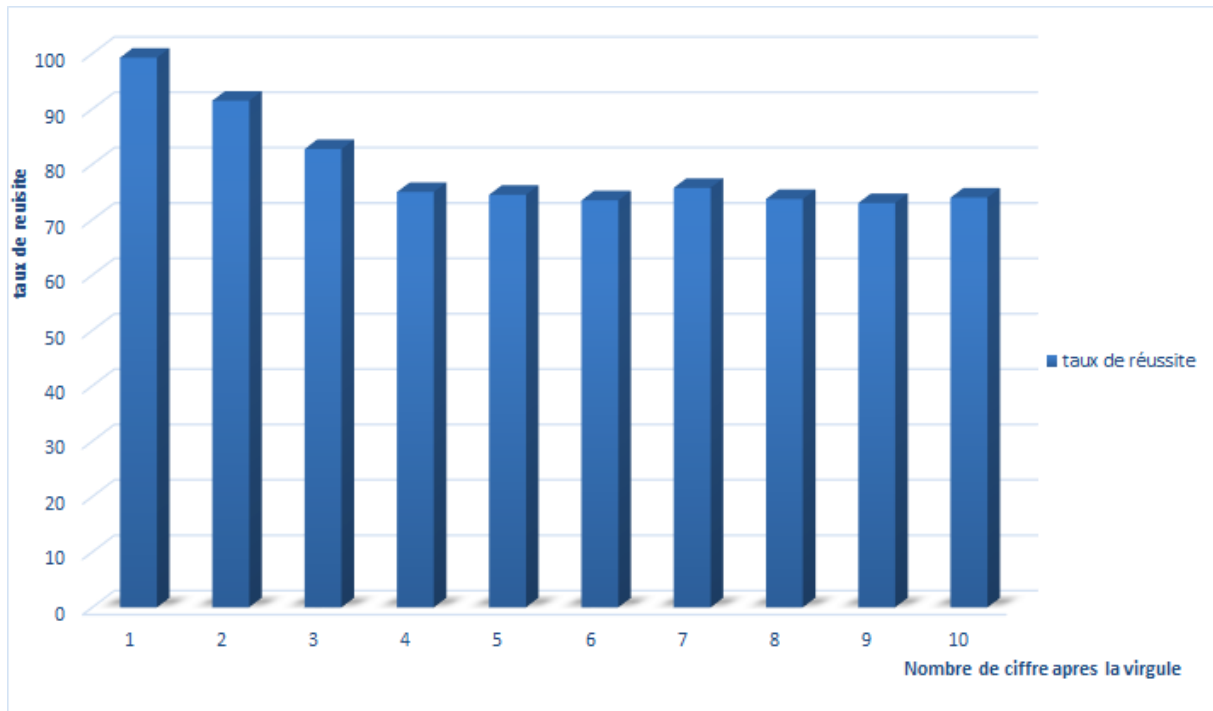


FIGURE 5.2 – Les performance de l’algorithme approximative

- **grande** : le nombre de conflits supérieure à 150.

Pour chacune des paires (base de connaissances, requête), la réponse exacte attendue est calculée. Le nombre de réponses correctes est calculé pour chaque classe de bases de connaissances et pour toutes les bases de connaissances générées. Les expériences sont menées pour deux cas :

1. Lorsque l’algorithme exact est utilisé pour calculer le degré d’inconsistance de la base de connaissances initiale ($Inc_{Exc}(\mathcal{K})$) et l’algorithme approximatif est utilisé pour calculer le degré d’inconsistance de la base augmentée ($Inc_{App}(\mathcal{K}_1)$).
2. Lorsque les deux degrés d’inconsistance des deux bases (initiale et augmentée) sont calculés par l’algorithme approximatif ($Inc_{App}(\mathcal{K})$ et $Inc_{App}(\mathcal{K}_1)$).

Les résultats obtenus sont donnés par les Figures 5.3 et 5.4.

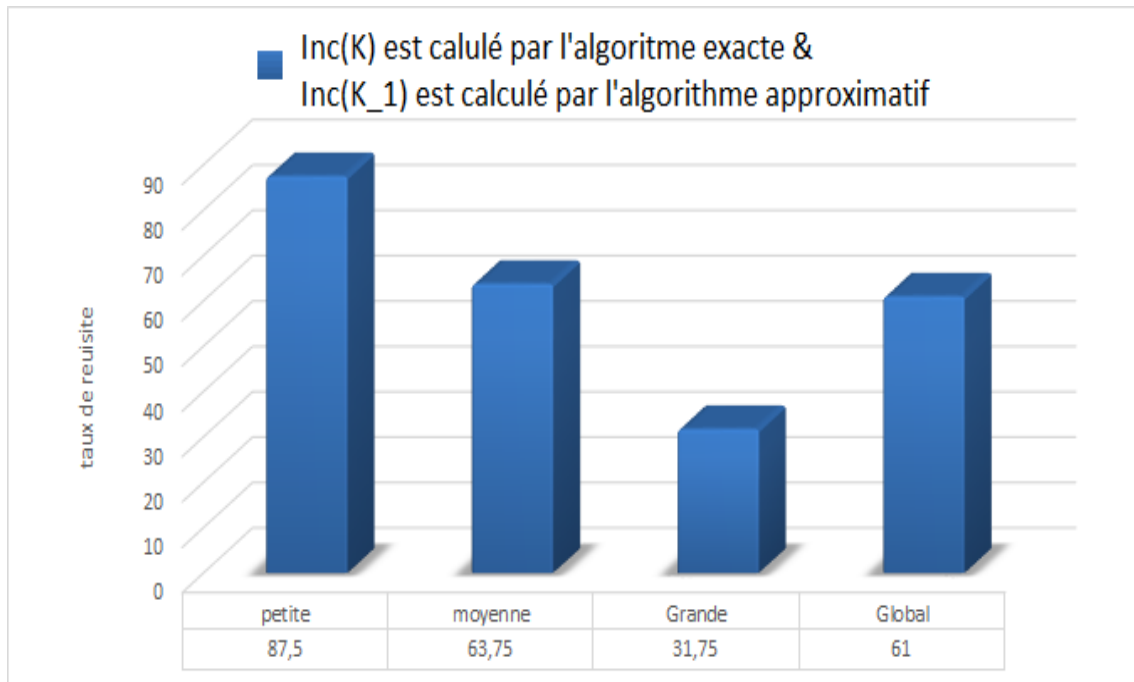


FIGURE 5.3 – $Inc_{Exc}(\mathcal{K})$ est calculé par l'algorithme exact et $Inc_{App}(\mathcal{K}_1)$ est calculé par l'algorithme approximatif

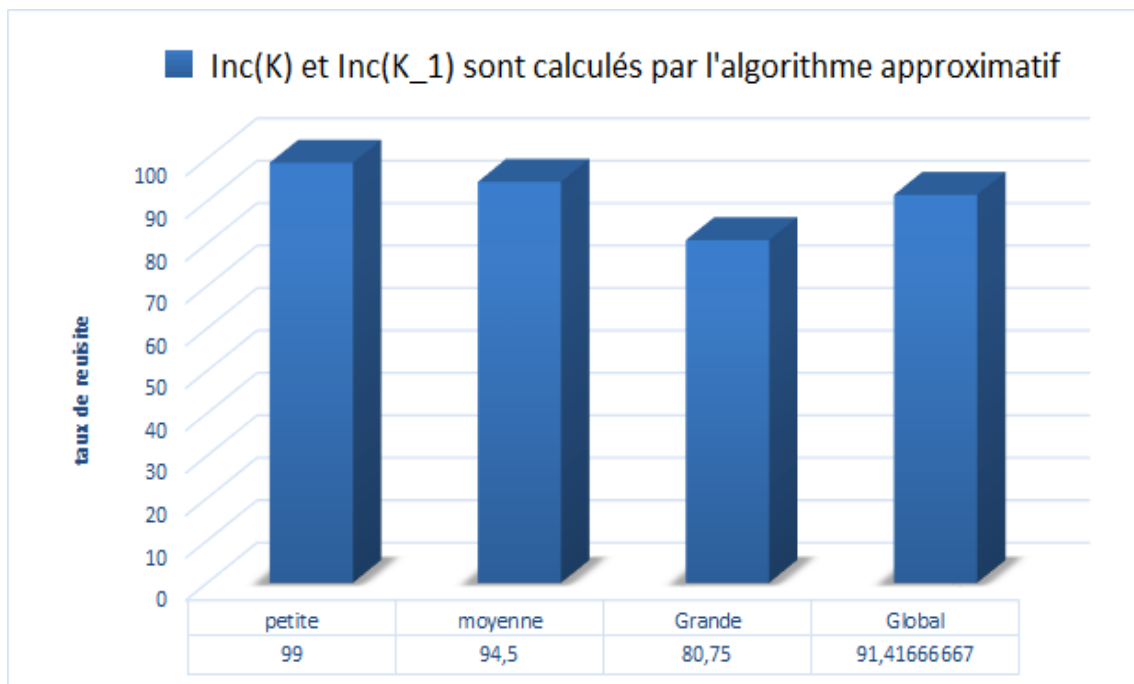


FIGURE 5.4 – $Inc_{App}(\mathcal{K})$ et $Inc_{App}(\mathcal{K}_1)$ sont calculés par l'algorithme approximatif

L'axe des X représente la classe des tailles des bases de connaissances d'entrée. Pour une abscisse x donné, l'ordonnée y correspondante sur l'axe des Y représente le taux de réussite. A partir de la Figure 5.3, nous remarquons que pour les bases de connaissances avec un petit nombre de conflits, l'algorithme approximatif donne de bons résultats mais une fois le nombre de conflits augmente la réponse devient fiable. La raison de ce résultat est dû à la méthode heuristique de l'algorithme approximatif.

Comme indiqué dans [123], la valeur approximative donnée par l'algorithme glouton est toujours supérieure ou égale à la valeur exacte. En fixant une valeur exacte $Inc_{Exc}(\mathcal{K})$ pour le degré d'inconsistance de la base initiale \mathcal{K} , l'écart avec la valeur approchée du degré d'inconsistance de \mathcal{K}_1 est toujours présent même dans le cas où la valeur exacte du degré d'inconsistance de \mathcal{K}_1 est égale à celle du degré d'inconsistance de \mathcal{K} .

En revanche, d'après la Figure 5.4, nous pouvons constater que les résultats sont très encourageants et que l'algorithme approximatif est satisfaisant. Pour les bases de connaissances de petite taille, il n'y a pas de différence si l'on utilise un algorithme exact ou approximatif pour calculer $Inc(\mathcal{K})$. Cependant, pour les bases de connaissances de tailles moyennes et de grandes tailles, l'utilisation de l'algorithme approximatif pour le calcul du degré d'inconsistance de la base initiale \mathcal{K}_1 donne de meilleurs résultats. Ces résultats étaient attendus. En effet, d'une part, comme nous l'avons déjà évoqué, la valeur approximative donnée par l'algorithme glouton est toujours supérieure ou égale à la valeur exacte. Utiliser une valeur approximative pour le degré d'inconsistance de \mathcal{K} permet d'éviter l'inconvénient cité ci-dessus pour la première approche. D'autre part, puisque le degré d'inconsistance de la nouvelle base de connaissances \mathcal{K}_1 ne peut qu'augmenter (tous les conflits de \mathcal{K} sont conservés dans \mathcal{K}_1), en utilisant deux calculs approximatifs, nous conservons une meilleure approximation de l'écart exact entre les deux valeurs exactes de $Inc(\mathcal{K})$ et $Inc(\mathcal{K}_1)$.

5.6 Conclusion

Dans ce chapitre, nous avons présenté un algorithme approximatif pour calculer efficacement une valeur approchée du degré d'inconsistance d'une base de connaissances Pb- π -DL-Lite. Cet algorithme a été utilisé par la suite pour donner des réponses approchées à des requêtes de vérification d'instance.

Nous avons également présenté une étude expérimentale pour l'évaluation de l'algorithme approximatif proposé. Les résultats obtenus mettent en évidence deux points importants. Le premier point est la validation de la qualité et de l'efficacité de l'algorithme approximatif proposé pour le calcul du degré d'inconsistance d'une base de connaissances Pb- π -DL-Lite. Le deuxième point est que pour la réponse aux requêtes de vérification d'instance, l'utilisation de l'algorithme approximatif pour le calcul des degrés d'inconsistance des deux bases de connaissances (initiale et augmentée) donne un très bon taux de réussite qui dépasse largement celui obtenu en utilisant l'algorithme approximatif uniquement pour le calcul du degré d'inconsistance de la base augmentée. Le chapitre suivant présente la généralisation de la nouvelle extension à des bases terminologiques Tbox et à des bases assertionnelles Abox incertaines.

Chapitre 6

Extension du calcul des degrés d'inconsistance pour des TBOX incertaines

6.1 Introduction

Dans le chapitre précédent, nous avons considéré que seules les assertions de la ABox peuvent être incertaines alors que les axiomes TBox sont considérés comme totalement sûrs. Ceci signifie que la base terminologique \mathcal{T} est stable et ne doit pas être mise en doute en présence d'inconsistance.

Dans ce chapitre, nous généralisons cette hypothèse pour que la base terminologiques et les assertions peuvent être incertaines. Cela rend notre approche utile pour les applications où les relations terminologiques générales codées dans la TBox ne sont pas complètement fiables et peuvent avoir un certain niveau d'incertitude. Notre contribution se résume comme suit :

- Lorsque la TBox est totalement certaine, le calcul de la fermeture négative se fait en appliquant les règles standards et les conflits sont binaires, c'est-à-dire qu'ils impliquent au plus deux assertions de la ABox qui contredisent un axiome négatif de la fermeture négative de la TBox. Nous montrons que face à une TBox incertaine, les conflits peuvent impliquer également des axiomes incertains de la TBox en plus des deux assertions de la ABox. Nous étendons alors les règles standards du calcul de la fermeture négative d'une TBox afin de garder la trace de tous les axiomes positifs de la TBox conduisant à un axiome négatif implicite ;
- Le calcul du degré d'inconsistance est au cœur du processus de réponse aux requêtes. Pour le calculer dans le nouveau cadre, nous proposons un nouvel algorithme exact basé sur la programmation linéaire en 0-1 et nous validons expérimentalement l'efficacité de cette nouvelle approche proposée.

Ce chapitre est organisé comme suit. La section 6.2 rappelle la définition de la programmation linéaire. Dans la section 6.3, nous introduisons la nouvelle forme des conflits dans notre cadre étendu et nous expliquons comment obtenir ces conflits en étendant les règles standards de la fermeture négative. Nous proposons également dans la section 6.4 une nouvelle modélisation du problème du calcul du degré d'inconsistance par un programme linéaire en nombres entiers. Dans la section 6.5, nous comparons l'approche proposée avec deux autres algorithmes proposés dans les travaux précédents.

6.2 La programmation linéaire

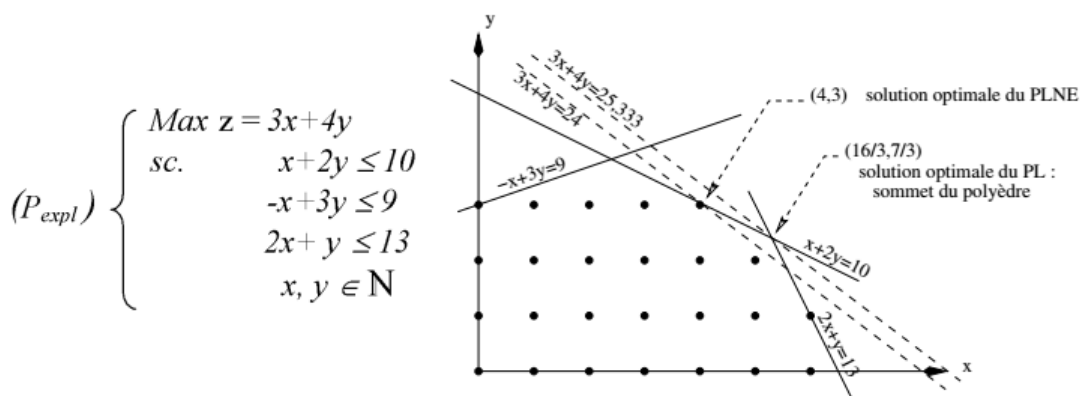
Les méthodes de la programmation linéaire sont souvent utilisées pour une résolution exacte de problèmes dont la fonction objective et les contraintes sont des fonctions linéaires.

Ces méthodes permettent de traiter des problèmes de grande taille grâce à leur efficacité. Un Programme Linéaire est un problème d'optimisation consistant à maximiser (ou à minimiser) une fonction objective linéaire de variables soumises à un ensemble de contraintes exprimées sous forme d'équations ou d'inéquations linéaires.

Un programme linéaire est mis sous forme standard si sa fonction objective est une fonction de maximisation et si toutes ses contraintes en dehors des contraintes de non-négativité sont des égalités. Un programme linéaire se présente donc comme suit :

$$P = \begin{cases} \max(\min) & z = \sum_{j=1}^n c_j x_j & \text{fonction objective} \\ \text{sous les contraintes} & \sum_{j=1}^n a_{ij} x_{ij} \leq b \\ & \sum_{j=1}^n a_{ij} x_{ij} \geq b \\ & \sum_{j=1}^n a_{ij} x_{ij} = b \\ & x_j \geq 0 \\ & j = 1, \dots, n \end{cases} \quad (6.1)$$

Les programmes linéaires en nombres entiers (PLNE) dans lesquels les variables prennent des valeurs entières sont souvent difficiles à résoudre, du fait notamment que l'espace de recherche n'est plus convexe mais discret. Toutefois, la programmation linéaire en nombres entiers est un outil très utile de modélisation des problèmes et de nombreuses approches de résolution ont été développées.



Exemple 6.1

Différentes approches ont été étudiées pour la recherche de solutions optimales entières des PLNE. Elles font appel essentiellement à des algorithmes de recherche arborescents par séparation et évaluation (branch and bound) et à des méthodes de coupes. Un moyen naturel pour résoudre un PLNE de très petite taille consiste à calculer l'ensemble des solutions entières du problème et à en retenir la meilleure. Cependant, au delà de quelques variables, cette énumération explicite devient impossible du fait de la combinatoire du problème. L'idée de la recherche arborescente par séparation et évaluation est d'effectuer une énumération implicite des solutions du PLNE.

Cette méthode consiste à séparer le problème en plusieurs sous-problèmes, puis à évaluer chaque sous-problème en calculant une borne supérieure (en maximisation) de sa valeur optimale et en la comparant à une valeur de référence. Le processus est alors réitéré uniquement sur les sous-problèmes dont l'évaluation est supérieure à la valeur de référence.

L'ensemble des solutions est ainsi représenté par une arborescence dans laquelle un grand nombre de nœuds sont éliminés. L'évaluation des sous-problèmes se fait souvent en résolvant (par le Simplex) leur relaxation linéaire.

Les méthodes de coupes ont pour but de trouver l'enveloppe convexe des solutions entières, c'est-à-dire le plus petit polyèdre contenant toutes les solutions entières du PLNE. En effet, si nous arrivons à déterminer cette enveloppe convexe, la résolution de la relaxation linéaire du PLNE réduit à cet ensemble donne une solution optimale entière.

Un cas particulier des PLNE est celui dans lequel les variables sont binaires et prennent leurs valeurs dans l'ensemble $\{0, 1\}$. nous parlons alors de programmes linéaires en 0-1. C'est cette classe de programmes qui nous intéresse dans ce chapitre.

6.3 Extension aux TBOX incertaines

Une base de connaissances Pb- π -DL-Lite $\mathcal{K} = \{\langle \phi_i, \alpha_i \rangle : i = 1, \dots, n\}$ se compose d'une TBox possibiliste \mathcal{T} et une ABox possibiliste \mathcal{A} .

Une base possibiliste TBox (resp. ABox) est un ensemble fini d'axiomes possibilistes de la forme $\langle \phi_i, \alpha_i \rangle$, où ϕ_i est un axiome DL-Lite (resp. une assertion DL-Lite) et $\alpha_i \in]0, 1]$ représente le degré de certitude.

La sémantique d'un Pb- π -DL-Lite est donnée par le concept d'une distribution de possibilité $\pi_{\mathcal{K}}$ sur l'ensemble de toutes les interprétations Ω . Une distribution possibiliste π est dite normalisée s'il existe au moins une interprétation totalement possible. Sinon π est dite sous-normalisée. Si une interprétation I est un modèle de chaque axiome de \mathcal{T} et chaque assertion de \mathcal{A} , son degré de possibilité est égal à 1. Cela reflète le fait que I est entièrement compatible avec $\langle \mathcal{T}, \mathcal{A} \rangle$. Cela signifie évidemment que $\langle \mathcal{T}, \mathcal{A} \rangle$ est consistante. Plus généralement, si une interprétation I falsifie certaines assertions de la Abox ou certains axiomes de la Tbox ou les deux, son degré de possibilité est inversement proportionnel au produit des poids des assertions et des axiomes qu'elle falsifie. Plus formellement :

Définition 6.1 $I \in \Omega$,

$$\pi_{\mathcal{K}}(I) = \begin{cases} 1 & \text{si } \forall \langle \phi_i, \alpha_i \rangle \in \mathcal{K}, I \models \phi_i \\ \prod_{\langle \phi_i, \alpha_i \rangle \in \mathcal{K}, I \not\models \phi_i} (1 - \alpha_i) & \text{sinon} \end{cases} \quad (6.2)$$

Dans le cas de l'extension possibiliste de DL-Lite avec TBox certaine, les principaux concepts utiles étaient facilement adaptés. Notamment, les notions de conflit et la façon de les calculer au moyen des règles standards définies pour DL-Lite restaient inchangées. En revanche, dans le cas de TBox incertaines la situation est différente. Tout d'abord, les règles de la fermeture standards ne sont plus suffisantes, car il faut garder trace des axiomes positifs conduisant à des conflits. De ce fait, le calcul des conflits est plus complexe. De plus, les conflits ne sont plus binaires. Même si le conflit proprement dit est toujours causé par une paire d'assertions de la ABox et un axiome négatif de la TBox, casser un tel conflit ne nécessite pas forcément exclusivement l'enlèvement d'une assertion de la ABox mais peut être également effectué en enlevant des axiomes incertains de la TBox. Dans ce qui suit, nous montrons comment calculer le degré d'inconsistance dans le cas des TBox incertaines.

6.4 Calcul du degré d'inconsistance

La première étape pour calculer le degré d'inconsistance consiste à calculer l'ensemble des conflits car la présence d'inconsistances est due à la présence de conflits.

Nous avons vu qu'aussi bien dans la logique DL-Lite classique que dans la logique DL-lite possibiliste basée sur le produit avec une TBox certaine, les conflits peuvent être obtenus en construisant la fermeture négative $Cln(\mathcal{T})$ de la TBox \mathcal{T} , en utilisant les règles standards données dans [5].

Dans notre situation où la TBox peut être incertaine, nous ne pouvons pas appliquer directement les règles standards sauf si nous définissons un opérateur permettant de propager les degrés de certitudes. L'autre alternative de solution consiste à proposer une extension des règles de la fermeture négative de façon à sauvegarder le(s) chemin(s) permettant l'obtention de chacun des axiomes négatifs présents dans la fermeture négative de la TBox \mathcal{T} .

6.4.1 La fermeture négative étendue

Dans cette section, nous expliquons comment étendre les règles standards afin de calculer la fermeture négative d'une DL-Lite possibiliste quantitative avec TBox incertaine.

La fermeture négative étendue, notée $Pb_Cln(\mathcal{T})$, se compose d'un ensemble de paires de la forme $[Axiom, Support]$ où : $Axiom$ est un axiome négatif non pondéré appartenant à la fermeture négative standard $Cln(\mathcal{T})$ et $Support \subseteq \mathcal{T}$ est un sous-ensemble d'axiomes pondérés permettant d'obtenir $Axiom$. Notons que $Support$ contient exactement un axiome négatif et un nombre positif ou nul d'axiomes positifs. Plus précisément, l'ensemble des règles adaptées utilisées pour générer la fermeture négative possibiliste basée sur le produit $Pb_Cln(\mathcal{T})$ est donné comme suit :

Règle 6.1 Pour chaque axiome négatif $\langle \Phi, \alpha \rangle$ dans la TBox \mathcal{T} , ajouter $[\Phi, \{\langle \Phi, \alpha \rangle\}]$ à $Pb_Cln(\mathcal{T})$.

La première règle indique que les axiomes négatifs dans \mathcal{K} sont ajoutés à $Pb_Cln(\mathcal{T})$.

Règle 6.2 Lorsque $\langle B_1 \sqsubseteq B_2, \alpha \rangle$ est dans \mathcal{T} alors, pour chaque $[B_2 \sqsubseteq \neg B_3, Support]$ ou $[B_3 \sqsubseteq \neg B_2, Support]$ dans $Pb_Cln(\mathcal{T})$, ajouter $[B_1 \sqsubseteq \neg B_3, Support \cup \{\langle B_1 \sqsubseteq B_2, \alpha \rangle\}]$ à $Pb_Cln(\mathcal{T})$.

Les règles 6.1 et 6.2 sont suffisantes pour $Pb\text{-}\pi\text{-DL-Lite}_{core}$.

Règle 6.3 Lorsque $\langle R_1 \sqsubseteq R_2, \alpha \rangle$ est dans \mathcal{T} alors, pour chaque $[\exists R_2 \sqsubseteq \neg B, Support]$ ou $[B \sqsubseteq \neg \exists R_2, Support]$ dans $Pb_Cln(\mathcal{T})$, ajouter $[\exists R_1 \sqsubseteq \neg B, Support \cup \{\langle R_1 \sqsubseteq R_2, \alpha \rangle\}]$ à $Pb_Cln(\mathcal{T})$.

Règle 6.4 Lorsque $\langle R_1 \sqsubseteq R_2, \alpha \rangle$ est dans \mathcal{T} alors, pour chaque $[\exists R_2^- \sqsubseteq \neg B, Support]$ ou $[B \sqsubseteq \neg \exists R_2^-, Support]$ dans $Pb_Cln(\mathcal{T})$, ajouter $[\exists R_1^- \sqsubseteq \neg B, Support \cup \{\langle R_1 \sqsubseteq R_2, \alpha \rangle\}]$ à $Pb_Cln(\mathcal{T})$.

Règle 6.5 Lorsque $\langle R_1 \sqsubseteq R_2, \alpha \rangle$ est dans \mathcal{T} alors, pour chaque $[R_2 \sqsubseteq \neg R_3, Support]$ ou $[R_3 \sqsubseteq \neg R_2, Support]$ dans $Pb_Cln(\mathcal{T})$, ajouter $[R_1 \sqsubseteq \neg R_3, Support \cup \{\langle R_1 \sqsubseteq R_2, \alpha \rangle\}]$ à $Pb_Cln(\mathcal{T})$.

Règle 6.6 Pour chaque $\langle \text{funct } R, \alpha \rangle$ dans \mathcal{T} , ajouter $[\text{funct } R, \{\langle \text{funct } R, \alpha \rangle\}]$ à $Pb_Cln(\mathcal{T})$.

Règle 6.7

- a) Pour $DL\text{-Lite}_R$: lorsque l'une des expressions $[\exists R \sqsubseteq \neg \exists R, Support]$, $[\exists R^- \sqsubseteq \neg \exists R^-, Support]$ ou $[R \sqsubseteq \neg R, Support]$ est dans $Pb_Cln(\mathcal{T})$, ajouter toutes ces trois expressions à $Pb_Cln(\mathcal{T})$.
- b) Pour $DL\text{-Lite}_F$: lorsque l'une des expressions $[\exists R \sqsubseteq \neg \exists R, Support]$ ou $[\exists R^- \sqsubseteq \neg \exists R^-, Support]$ est dans $Pb_Cln(\mathcal{T})$, ajouter toutes ces deux expressions à $Pb_Cln(\mathcal{T})$.

Exemple 6.2 Soit la base de connaissances $Pb\text{-}\pi\text{-DL-Lite}$ $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$. L'échelle d'incertitude de la base $Tbox$ ou de la base $Abox$ est définie dans l'intervalle $]0, 1]$.

La base des axiomes \mathcal{T} est :

$$\begin{aligned} &\langle \exists Enseigne \sqsubseteq Professeur, 0.89 \rangle, && \langle \exists aTuteur \sqsubseteq Etudiant, 0.85 \rangle, \\ &\langle \exists Enseigne^- \sqsubseteq Etudiant, 0.8 \rangle, && \langle \exists aTuteur^- \sqsubseteq Professeur, 0.96 \rangle, \\ &\langle Professeur \sqsubseteq \neg Etudiant, 1.0 \rangle. \end{aligned}$$

La base des assertions \mathcal{A} est :

$$\begin{aligned} &\langle Professeur(John), 0.95 \rangle, && \langle Enseigne(Jean, John), 0.87 \rangle, \\ &\langle Etudiant(Jean), 0.90 \rangle, && \langle aTuteur(John, Jean), 0.65 \rangle. \end{aligned}$$

La fermeture négative étendue de la nouvelle base $\mathcal{K}' Pb_Cln(\mathcal{T})$ est :

- $[Professeur \sqsubseteq \neg Etudiant, \{\langle Professeur \sqsubseteq \neg Etudiant, 1.0 \rangle\}]$
- $[\exists Enseigne \sqsubseteq \neg Etudiant, \{\langle \exists Enseigne \sqsubseteq Professeur, 0.89 \rangle, \langle Professeur \sqsubseteq \neg Etudiant, 1.0 \rangle\}]$
- $[\exists aTuteur^- \sqsubseteq \neg Etudiant, \{\langle \exists aTuteur^- \sqsubseteq Professeur, 0.96 \rangle, \langle Professeur \sqsubseteq \neg Etudiant, 1.0 \rangle\}]$
- $[\exists Enseigne^- \sqsubseteq \neg Professeur, \{\langle \exists Enseigne^- \sqsubseteq Etudiant, 0.8 \rangle, \langle Professeur \sqsubseteq \neg Etudiant, 1.0 \rangle\}]$
- $[\exists aTuteur \sqsubseteq \neg Professeur, \{\langle \exists aTuteur \sqsubseteq Etudiant, 0.85 \rangle, \langle Professeur \sqsubseteq \neg Etudiant, 1.0 \rangle\}]$.

La nouvelle forme des conflits est intuitivement dérivée de la fermeture négative étendue comme suit : l'expression $[Axiom, Support]$ peut donner naissance à un conflit : $\{\langle C(\vec{a}), \alpha \rangle, \langle D(\vec{b}), \beta \rangle\} \cup Support$ où $\langle C(\vec{a}), \alpha \rangle, \langle D(\vec{b}), \beta \rangle \in \mathcal{A}$ et $\{C(\vec{a}), D(\vec{b})\}$ représente un conflit binaire classique au sens de DL-Lite standard [5, 13]. En fait, $C(\vec{a}), D(\vec{b})$ créent une contradiction avec $Axiom$ qui peut être dérivé de la TBox en s'appuyant sur les versions non pondérées des axiomes de l'ensemble $Support$. Contrairement au cas d'une TBox certaine où ce conflit est cassé uniquement en retirant soit $C(\vec{a})$ soit $D(\vec{b})$, dans le cas d'une TBox incertaine, ce conflit peut également être cassé en retirant n'importe quel axiome incertain de l'ensemble $Support$. La nouvelle forme de conflit incorpore ainsi les deux assertions de la ABox qui causent le conflit mais aussi l'ensemble des axiomes de la TBox qui participent à la dérivation de l'axiome négatif implicite qui entre en contradiction directe avec ces deux assertions. Notons que pour un couple d'assertions formant un conflit binaire au sens classique, il peut y avoir plusieurs conflits qui partagent le même axiome négatif résultat mais qui diffèrent sur les supports conduisant à ce même axiome.

Formellement, la nouvelle forme de conflit dans le cadre de DL-Lite possibiliste quantitative avec une TBox incertaine est définie comme suit :

Définition 6.2 Soit $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ une base de connaissances $Pb\text{-}\pi\text{-DL-Lite}$ avec TBox incertaine et $Pb_Cln(\mathcal{T})$ la fermeture négative étendue de \mathcal{T} . Un conflit \mathcal{C} de \mathcal{K} est une sous-base de la forme :

$$\mathcal{C} = \{\langle C(\vec{a}), \alpha \rangle, \langle D(\vec{b}), \beta \rangle\} \cup Support \quad (6.3)$$

où :

- $[Axiom, Support] \in Pb_Cln(\mathcal{T})$.

- $\mathcal{C} = \{\langle C(\vec{a}), \alpha \rangle, \langle D(\vec{b}), \beta \rangle\} \cup \text{Support}$ est inconsistent et
- tout sous-ensemble strict \mathcal{C}' de \mathcal{C} est consistant.

Le calcul du degré d'inconsistance se calcule en déterminant un sous-ensemble d'assertions et/ou d'axiomes à partir de ceux impliqués dans l'ensemble des conflits tels que :

- Si ce sous-ensemble est supprimé, la consistance de la base est alors restaurée. Cela veut dire que la suppression de ce sous-ensemble permet de casser tous les conflits (en enlevant au moins un des éléments de chaque conflit).
- Ce sous-ensemble assure une valeur maximale du degré de possibilité de l'interprétation correspondante.
- Le degré d'inconsistance est l'inverse du degré de possibilité de cette interprétation.

Dans la section suivante, nous fournissons un codage du problème de calcul du degré d'inconsistance d'une base de connaissances Pb- π -DL-Lite en termes d'un programme linéaire en 0-1.

Exemple 6.3 L'ensemble des conflits de l'Exemple 6.2 est $\zeta = \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4\}$:

- $\mathcal{C}_1 = \{\langle \text{Professeur}(\text{John}), 0.95 \rangle, \langle \text{Enseigne}(\text{Jean}, \text{John}), 0.98 \rangle, \langle \exists \text{anseigne}^- \sqsubseteq \text{Etudiant}, 0.79 \rangle, \langle \text{Professeur} \sqsubseteq \neg \text{Etudiant}, 1.0 \rangle\}$
- $\mathcal{C}_2 = \{\langle \text{Professeur}(\text{John}), 0.95 \rangle, \langle \text{aTuteur}(\text{John}, \text{Jean}), 0.5 \rangle, \langle \exists \text{aTuteur} \sqsubseteq \text{Etudiant}, 0.77 \rangle, \langle \text{Professeur} \sqsubseteq \neg \text{Etudiant}, 1.0 \rangle\}$
- $\mathcal{C}_3 = \{\langle \text{Etudiant}(\text{Jean}), 0.9 \rangle, \langle \text{Enseigne}(\text{Jean}, \text{John}), 0.98 \rangle, \langle \exists \text{Enseigne} \sqsubseteq \text{Professeur}, 0.79 \rangle, \langle \text{Professeur} \sqsubseteq \neg \text{Etudiant}, 1.0 \rangle\}$
- $\mathcal{C}_4 = \{\langle \text{Etudiant}(\text{Jean}), 0.9 \rangle, \langle \text{aTuteur}(\text{John}, \text{Jean}), 0.5 \rangle, \langle \exists \text{aTuteur}^- \sqsubseteq \text{Professeur}, 0.75 \rangle, \langle \text{Professeur} \sqsubseteq \neg \text{Etudiant}, 1.0 \rangle\}$

6.4.2 Modélisation du problème

Dans cette partie, nous fournissons un codage du problème de calcul du degré d'inconsistance d'une base de connaissances Pb- π -DL-Lite en utilisant la programmation linéaire en 0-1.

La première étape du calcul du degré d'inconsistance est le calcul de l'ensemble de conflits en appliquant les règles de la fermeture négative étendue.

A partir de l'ensemble de conflits, la modélisation du problème en termes de programmation linéaire en 0-1 est décrite par la définition suivante :

Définition 6.3 Soient \mathcal{K} une base de connaissances Pb- π -DL-Lite, $\zeta = \{\mathcal{C}_1, \dots, \mathcal{C}_m\}$ l'ensemble de conflits de \mathcal{K} , $\Gamma = \{\langle \Phi_1, \alpha_1 \rangle, \dots, \langle \Phi_n, \alpha_n \rangle\}$ l'ensemble de tous les axiomes et les assertions incertains impliqués dans ζ . Soit F une fonction de changement d'échelle définie par¹ : $F(y) = -(\ln(1 - y))$.

S'il n'existe pas de conflit $\mathcal{C} \in \zeta$ tel que $\Gamma \cap \mathcal{C} = \emptyset$, alors le calcul du degré d'inconsistance de \mathcal{K} est modélisé par le programme linéaire en 0-1 $LP_{\mathcal{K}}$ défini par :

- Chaque formule $\langle \Phi_i, \alpha_i \rangle \in \Gamma$ est représentée par une variable binaire x_i telle que dans la solution, $x_i = 1$ si $\langle \Phi_i, \alpha_i \rangle$ doit être retirée et $x_i = 0$ sinon.
- L'objectif est de minimiser la fonction objective suivante :

$$\sum_{i=1}^n F(\alpha_i) \cdot x_i$$

1. \ln représente la fonction de logarithme népérien

- Chaque conflit \mathcal{C}_j est représenté par une contrainte C_j qui stipule que le conflit en question doit être éliminé en supprimant au moins une des formules qu'il contient, i.e. au moins une formule $\langle \Phi_i, \alpha_i \rangle \in \mathcal{C}_j$ est telle que sa variable représentative x_i est égale à 1 :

$$C_j : \sum_{\langle \Phi_i, \alpha_i \rangle \in \mathcal{C}_j} x_i \geq 1$$

Notons que la construction de $LP_{\mathcal{K}}$ est conditionnée dans la Définition 6.3 par le fait qu'il n'existe pas de conflit $\mathcal{C} \in \zeta$ tel que $\Gamma \cap \mathcal{C} = \emptyset$ puisque dans ce cas, le degré d'inconsistance $Inc(\mathcal{K})$ peut être calculé sans construire $LP_{\mathcal{K}}$. En effet, Le conflit \mathcal{C} ne contient dans ce cas que des formules certaines. Cela veut dire que \mathcal{C} ne peut pas être éliminé car les formules certaines ne peuvent pas être supprimées. Ainsi, chaque interprétation falsifie au moins une formule de \mathcal{C} . Ceci signifie simplement que $Inc(\mathcal{K}) = 1$.

La proposition suivante montre comment le degré d'inconsistance $Inc(\mathcal{K})$ est calculé en utilisant le programme linéaire en 0-1 $LP_{\mathcal{K}}$.

Proposition 6.1 *Soient \mathcal{K} une base de connaissances Pb- π -DL-Lite et $LP_{\mathcal{K}}$ le programme linéaire en 0-1 correspondant donné par la Définition 6.3. $Inc(\mathcal{K}) = \lambda$ si et seulement si $F(\lambda)$ est la valeur de la fonction objective d'une solution optimale de $LP_{\mathcal{K}}$.*

Preuve 6.1 *Nous considérons que les formules (axiomes et assertions) de la base \mathcal{K} sont indexées à partir de 1 à $n = |\mathcal{T} \cup \mathcal{A}|$.*

La formule $\langle \Phi_i, \alpha_i \rangle$ dans \mathcal{K} correspond à la variable d'indice i dans $LP_{\mathcal{K}}$.

- **La 1^{er} partie** $Inc(\mathcal{K}) = \lambda$

Nous supposons que $Inc(\mathcal{K}) = \lambda$. En utilisant la définition 3.5, $\lambda = 1 - \max_{I \in \Omega} (\pi_{\mathcal{K}}(I))$.

Soit I_1 une interprétation telle que $\pi_{\mathcal{K}}(I_1) = \max_{I \in \Omega} (\pi_{\mathcal{K}}(I))$.

Soit J l'ensemble des indices des formules falsifiées par I_1 , c'est-à-dire $J = \{i \mid I_1 \not\models \langle \Phi_i, \alpha_i \rangle\}$. D'après la définition 6.2, nous avons : $\pi_{\mathcal{K}}(I_1) = \prod_{i \in J} (1 - \alpha_i)$ et donc $\lambda = 1 - \pi_{\mathcal{K}}(I_1)$.

Soit $Y = (y_1, \dots, y_n)$ un vecteur binaire tel que :

- $y_i = 1$ si $i \in J$ et
- $y_i = 0$ sinon.

Nous montrons que Y est une solution optimale de $LP_{\mathcal{K}}$ dont la fonction objective est $F(\lambda)$. Puisque pour tout $i \notin J$, $I_1 \models \langle \Phi_i, \alpha_i \rangle$, il est clair que l'ensemble $S = \{\langle \Phi_i, \alpha_i \rangle \in \mathcal{T} \cup \mathcal{A} \mid i \notin J\}$ est consistant et ne contient aucun conflit. Cela signifie que pour chaque conflit \mathcal{C}_j il existe une formule $\langle \Phi_i, \alpha_i \rangle \in \mathcal{C}_j$ tels que $i \in J$, c'est-à-dire $y_i = 1$. Il s'ensuit que la contrainte correspondante C_j est satisfaite par Y .

La fonction objective de Y est :

$$\begin{aligned} \sum_{i=1}^n F(\alpha_i) \cdot y_i &= \sum_{i \in J} F(\alpha_i) \\ &= \sum_{i \in J} -\ln(1 - \alpha_i) \\ &= -\ln(\prod_{i \in J} (1 - \alpha_i)) \\ &= F(1 - \prod_{i \in J} (1 - \alpha_i)) \\ &= F(1 - \pi_{\mathcal{K}}(I_1)) = F(\lambda). \end{aligned}$$

Pour montrer que Y est une solution optimale, nous supposons qu'il existe une solution $Z = (z_1, \dots, z_n)$ de $LP_{\mathcal{K}}$ tel que $\sum_{i=1}^n F(\alpha_i) \cdot z_i < \sum_{i=1}^n F(\alpha_i) \cdot y_i$.

Soit J' l'ensemble des indices de variables mises à 1 dans Z : $J' = \{i \mid z_i = 1\}$ et I_2 une interprétation qui satisfait toutes les formules $\langle \Phi_i, \alpha_i \rangle$ où $i \notin J'$. Une telle interprétation existe parce que l'ensemble des formules $H = \{\langle \Phi_i, \alpha_i \rangle \mid i \notin J'\}$ est consistant.

En effet, si nous supposons que ce n'est pas le cas, il s'ensuit qu'il existe un conflit $C_j \subseteq H$, i.e., pour tout $\langle \Phi_i, \alpha_i \rangle \in C_j$, nous avons $z_i = 0$. Cela signifie que la contrainte C_j n'est pas satisfaite par Z qui contredit le fait que Z soit une solution de $LP_{\mathcal{K}}$. Le degré de possibilité de l'interprétation I_2 est donné par : $\pi_{\mathcal{K}}(I_2) = \prod_{i \in J'} (1 - \alpha_i)$ et nous avons :

$$\begin{aligned} \sum_{i=1}^n F(\alpha_i).z_i < \sum_{i=1}^n F(\alpha_i).y_i &\Leftrightarrow \sum_{i \in J'} F(\alpha_i) < \sum_{i \in J} F(\alpha_i) \\ &\Leftrightarrow \sum_{i \in J'} -\ln(1 - \alpha_i) < \sum_{i \in J} -\ln(1 - \alpha_i) \\ &\Leftrightarrow -\ln(\prod_{i \in J'} (1 - \alpha_i)) < -\ln(\prod_{i \in J} (1 - \alpha_i)) \\ &\Leftrightarrow \ln(\prod_{i \in J'} (1 - \alpha_i)) > \ln(\prod_{i \in J} (1 - \alpha_i)) \\ &\Leftrightarrow \prod_{i \in J'} (1 - \alpha_i) > \prod_{i \in J} (1 - \alpha_i) \\ &\Leftrightarrow \pi_{\mathcal{K}}(I_2) > \pi_{\mathcal{K}}(I_1). \end{aligned}$$

Cela contredit le fait que λ soit le degré d'inconsistance de \mathcal{K} .

• **La 2^{ème} partie Y est une solution optimale de $LP_{\mathcal{K}}$**

Soit $Y = (y_1, \dots, y_n)$ une solution optimale de $LP_{\mathcal{K}}$ dont la valeur de la fonction objective est $F(\lambda) = \sum_{i=1}^n F(\alpha_i).y_i$ et $J = \{i \mid y_i = 1\}$. Il est facile de vérifier que : $\lambda = 1 - \prod_{i \in J} (1 - \alpha_i)$. Soit I_1 une interprétation qui satisfait toutes les formules $\langle \Phi_i, \alpha_i \rangle$ où $i \notin J$. Une telle interprétation existe parce que l'ensemble des formules $H = \{\langle \Phi_i, \alpha_i \rangle \mid i \notin J\}$ est consistant. En effet, si nous supposons que ce n'est pas le cas, il s'ensuit qu'il existe un conflit $C_j \subseteq H$, i.e., pour tout $\langle \Phi_i, \alpha_i \rangle \in C_j$ nous avons $y_i = 0$.

Cela signifie que la contrainte C_j n'est pas satisfait par Y ce qui contredit le fait que Y est une solution de $LP_{\mathcal{K}}$. La distribution de possibilité de I_1 est donnée par $\pi_{\mathcal{K}}(I_1) = \prod_{i \in J} (1 - \alpha_i)$ et donc, $\lambda = 1 - \pi_{\mathcal{K}}(I_1)$.

Pour montrer que λ est le degré d'inconsistance de \mathcal{K} il suffit de montrer que :

$$\pi_{\mathcal{K}}(I_1) = \max_{I \in \Omega} (\pi_{\mathcal{K}}(I)).$$

Supposons, pour chercher une contradiction, qu'il y ait une interprétation I_2 telle que $\pi_{\mathcal{K}}(I_2) > \pi_{\mathcal{K}}(I_1)$. Soit J' l'ensemble des indices de formules falsifiées par I_2 : $J' = \{i \mid I_2 \not\models \langle \Phi_i, \alpha_i \rangle\}$ et $Z = (z_1, \dots, z_n)$ un vecteur binaire tel que $z_i = 1$ si $i \in J'$ et $z_i = 0$ sinon.

Montrons que Z satisfait toutes les contraintes de $LP_{\mathcal{K}}$.

Puisque pour tout $i \notin J'$, $I_2 \models \langle \Phi_i, \alpha_i \rangle$, il est clair que l'ensemble $S = \{\langle \Phi_i, \alpha_i \rangle \in \mathcal{T} \cup \mathcal{A} \mid i \notin J'\}$ est consistant (ne contient aucun conflit). Cela signifie que pour chaque conflit C_j , il existe une formule $\langle \Phi_i, \alpha_i \rangle \in C_j$ telle que $i \in J$, c'est à dire, $z_i = 1$. il s'ensuit que la contrainte correspondante C_j est satisfaite par Y et il est vrai que : Mais cela contredit le

$$\begin{aligned} \pi_{\mathcal{K}}(I_2) > \pi_{\mathcal{K}}(I_1) &\Leftrightarrow \prod_{i \in J'} (1 - \alpha_i) > \prod_{i \in J} (1 - \alpha_i) \\ &\Leftrightarrow \ln(\prod_{i \in J'} (1 - \alpha_i)) > \ln(\prod_{i \in J} (1 - \alpha_i)) \\ &\Leftrightarrow \sum_{i \in J'} \ln(1 - \alpha_i) > \sum_{i \in J} \ln(1 - \alpha_i) \\ &\Leftrightarrow -\sum_{i \in J'} \ln(1 - \alpha_i) < -\sum_{i \in J} \ln(1 - \alpha_i) \\ &\Leftrightarrow \sum_{i \in J'} -\ln(1 - \alpha_i) < \sum_{i \in J} -\ln(1 - \alpha_i) \\ &\Leftrightarrow \sum_{i \in J'} F(\alpha_i) < \sum_{i \in J} F(\alpha_i) \\ &\Leftrightarrow \sum_{i=1}^n F(\alpha_i).z_i < \sum_{i=1}^n F(\alpha_i).y_i. \end{aligned}$$

fait que Y est une solution optimale de $LP_{\mathcal{K}}$. ■

Selon la Définition 6.3 et la Proposition 6.1, le degré d'inconsistance $Inc(\mathcal{K})$ est calculé étant donné une base de connaissances Pb- π -DL-Lite \mathcal{K} comme suit : Tout d'abord, l'ensemble ζ des conflits est calculé. Ensuite, cet ensemble de conflits est transformé en un

programme linéaire en 0-1 en utilisant la définition 6.3. Enfin, le degré d'inconsistance est calculé par un appel à un outil de programmation linéaire. Dans cette thèse, nous avons utilisé l'outil de programmation linéaire commercial Cplex (IBM) [70] qui est un logiciel d'optimisation développé et vendu par ILOG, Inc.

Exemple 6.4 *Considérons à nouveau l'Exemple précédent 6.2.*

Le codage en programmation linéaire en 0-1 obtenu par la Définition 6.3 est le suivant :

$$LP_{\mathcal{K}} = \begin{cases} \min z & = \sum_{i=1}^8 F(\alpha_i).x_i & \text{fonction objective} \\ C_1 & : x_1 + x_2 + x_3 \geq 1 \\ C_2 & : x_1 + x_4 + x_5 \geq 1 \\ C_3 & : x_6 + x_2 + x_7 \geq 1 \\ C_4 & : x_6 + x_4 + x_8 \geq 1 \\ x_i \in \{0, 1\} \ (1 \leq i \leq 9) \end{cases} \quad (6.4)$$

où :

- x_1 correspond à $\langle \text{Professeur}(\text{John}), 0.95 \rangle$,
- x_2 correspond à $\langle \text{Enseigne}(\text{Jean}, \text{John}), 0.98 \rangle$,
- x_3 correspond à $\langle \exists \text{anseigne}^- \sqsubseteq \text{Etudiant}, 0.79 \rangle$,
- x_4 correspond à $\langle \text{aTuteur}(\text{John}, \text{Jean}), 0.5 \rangle$,
- x_5 correspond à $\langle \exists \text{aTuteur} \sqsubseteq \text{Etudiant}, 0.77 \rangle$,
- x_6 correspond à $\langle \text{Etudiant}(\text{Jean}), 0.9 \rangle$,
- x_7 correspond à $\langle \exists \text{Enseigne} \sqsubseteq \text{Professeur}, 0.79 \rangle$,
- x_8 correspond à $\langle \exists \exists \text{aTuteur}^- \sqsubseteq \text{Professeur}, 0.75 \rangle$.

Sémantiquement, le degré d'inconsistance de la base \mathcal{K} est :

$$\text{Inc}(\mathcal{K}) = 1 - \max_{I \in \Omega} (\pi_{\mathcal{K}}(I)) = 0.983.$$

La solution de notre $LP_{\mathcal{K}}$ consiste à mettre les variable x_3, x_4 et x_7 à 1 et le reste des variables à 0. Nous obtenons alors : $\text{Inc}(\mathcal{K}) = F^{-1}(F(\alpha_{x_3}) + F(\alpha_{x_4}) + F(\alpha_{x_7})) = 0.983$.

Le degré d'inconsistance $\text{Inc}(\mathcal{K})$ est calculé en se basant sur la Définition 6.3 et la Proposition 6.1. L'Algorithme 4 décrit la procédure du calcul le degré d'inconsistance

Algorithm 4 calcul le degré d'inconsistance

Require: $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$: une base de connaissances Pb- π -DL-Lite.

Ensure: Inc : le degré d'inconsistance.

```

Pb_Cln( $\mathcal{T}$ ) = Calcul_Fer_Neg_Etendue( $\mathcal{T}, \mathcal{A}$ )
 $\zeta$          = Calcul_Conflits(Pb_Cln( $\mathcal{T}$ ),  $\mathcal{A}$ )
LP $_{\mathcal{K}}$       = Construction_LP(Pb_Cln( $\mathcal{T}$ ),  $\mathcal{A}$ )
Inc( $\mathcal{K}$ )     = Outil_LP(LP $_{\mathcal{K}}$ )
retourne  ( $\text{Inc}$ )

```

6.5 Étude expérimentale

Dans cette section, nous exposons l'étude expérimentale que nous avons effectuée afin de pouvoir évaluer les performances de notre algorithme basé sur la programmation linéaire

en 0-1 et utilisant l'outil Cplex (IBM) [70] pour le calcul des degrés d'inconsistance de bases de connaissances Pb- π -DL-Lite. Pour l'implémentation des algorithmes, nous avons aussi utilisé le langage de programmation JAVA.

L'objectif de la première expérience est de comparer les performances des deux algorithmes exacts de calcul du degré d'inconsistance, à savoir :

- L'algorithme de calcul du degré d'inconsistance en utilisant l'outil Cplex.
- L'algorithme de calcul du degré d'inconsistance en utilisant l'outil W-Max-SAT.

Cette expérimentation a été élaborée selon le processus suivant :

- La génération aléatoire de 588 bases de connaissances Pb- π -DL-Lite où le nombre de conflits varie de 1 à 2365 ;
- Pour chaque base de connaissances \mathcal{K} , nous avons considéré le temps nécessaire pour calculer $Inc(\mathcal{K})$ en utilisant :
 - l'algorithme basé sur l'outil Cplex
 - l'algorithme basé sur l'outil W-Max-SAT.

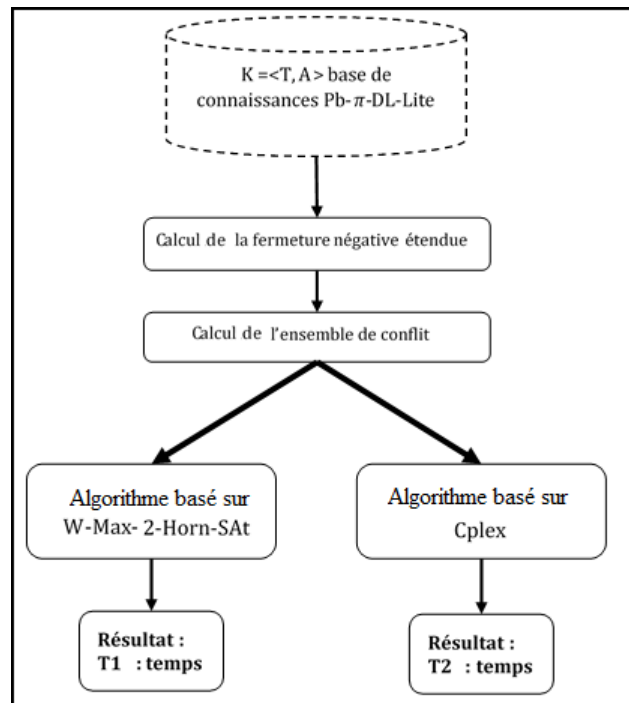


FIGURE 6.1 – Schéma récapitulatif du déroulement de l'expérimentation

La Figure 6.1 illustre les résultats obtenus en comparant les temps de réponse des deux algorithmes en fonction du nombre de conflits présents dans la base.

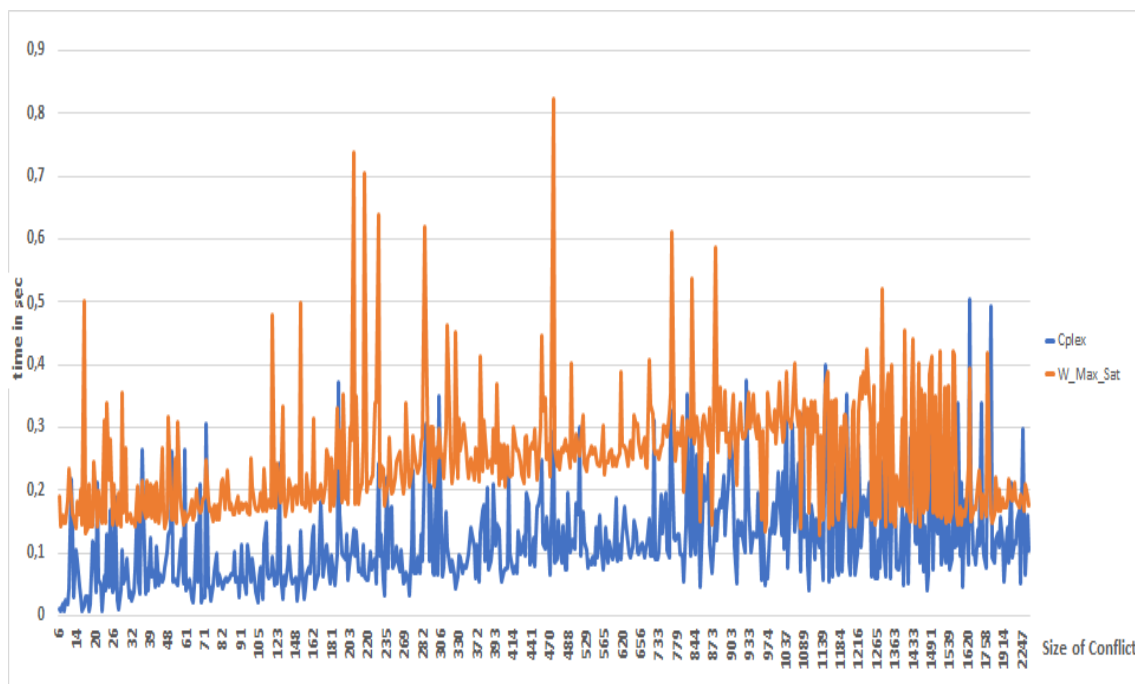


FIGURE 6.2 – Comparaison entre les temps de réponse des deux outil Cplex et W-Max-SAT

Nous remarquons que l'algorithme basé sur l'outil Cplex l'emporte et que la différence entre les temps de réponse est importante. Quelque soit le nombre de conflits dans la base, l'algorithme basé sur Cplex permet le calcul en un temps largement plus court. Ces résultats expérimentaux sont très encourageants. Ils permettent ainsi d'affirmer que "l'algorithme basé sur l'outil Cplex est le plus performant".

Comme nous l'avons déjà dit, la complexité théorique de l'algorithme basé sur l'outil Cplex est NP-Complet. Comme nous avons proposé un algorithme glouton approximatif de complexité polynomiale basé sur le problème de couverture par ensembles pondérée pour calculer le degré d'inconsistance, nous avons alors décidé d'effectuer des tests sur cet algorithme approximatif en utilisant les mêmes bases de connaissances que nous avons utilisé lors de la comparaison précédente.

Pour cela, nous avons, dans une deuxième expérience, calculé le taux de réussite de l'algorithme approximatif, pour le calcul du degré d'inconsistance, sur les bases générées dans l'expérience précédente. Nous avons obtenu une valeur du taux de réussite égale à **81,26 %** mais avec un écart moyen par rapport aux valeurs exactes (dans les cas d'erreur) trop petit (égale à **0,0023**). Ce résultat est très encourageant et nous permet de conclure que les valeurs approximatives trouvées sont de très bonne qualité.

Enfin dans une troisième expérience, nous avons comparé le temps d'exécution nécessaire pour calculer le degré d'inconsistance par l'algorithme exacte basé sur Cplex et l'algorithme glouton approximatif en fonction du nombre de conflits et en utilisant les mêmes bases générée dans la première expérience. Les résultats sont illustrés dans la Figure 6.3.

À partir de ces résultats, nous pouvons constater que les deux algorithmes, exact utilisant Cplex et approximatif, sont très compétitifs et leurs performances sont très proches : le temps d'exécution moyen pour l'ensemble des bases de connaissances générées est :

- **0.119 sec** pour Cplex et

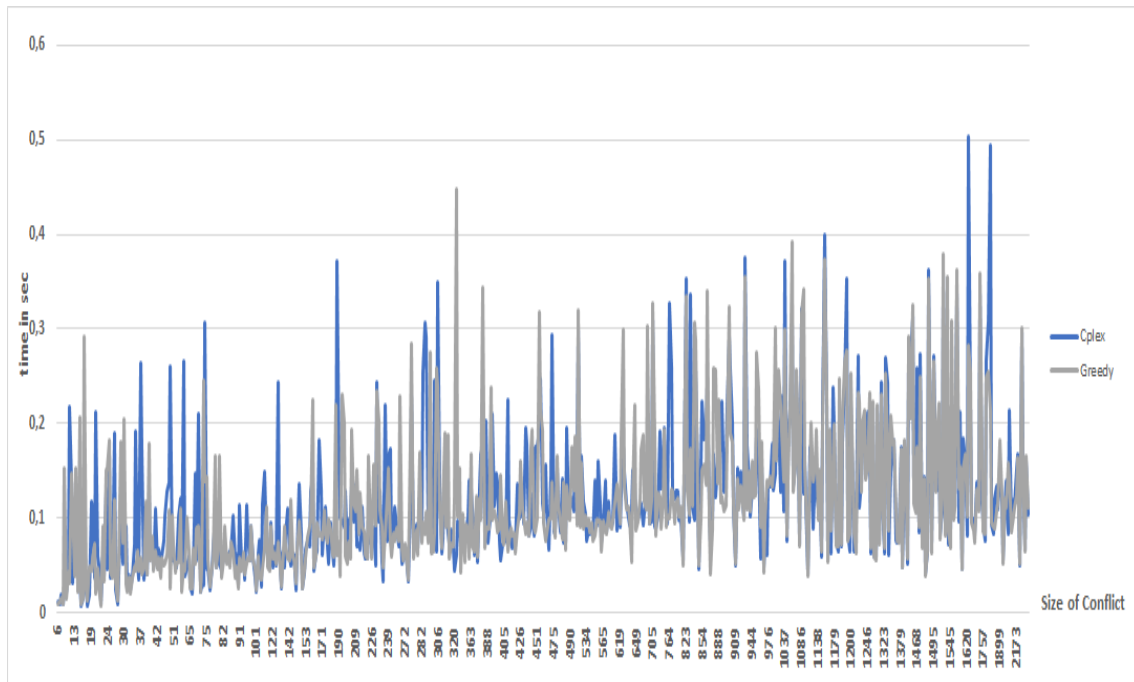


FIGURE 6.3 – Comparaison entre Cplex et l’algorithme approximatif

- **0.117** pour l’algorithme approximatif.

Ceci est très intéressant car il montre que, en plus du fait que Cplex donne des valeurs exactes, il réalise en pratique une performance comparable à un algorithme polynomial même si sa complexité théorique dans le pire des cas reste exponentielle. Cela fait de Cplex un très bon choix pour calculer des degrés d’inconsistance de bases de connaissances Pb- π -DL-Lite et par conséquent pour répondre efficacement à des requêtes.

6.6 Conclusion

Dans ce chapitre, nous avons étudié le cas général où la ABox et la TBox peuvent être incertaines. Après un rappel sur la programmation linéaire, le chapitre propose une modélisation du problème du calcul de degré d’inconsistance d’une base Pb- π -DL-Lite avec TBox incertaine en termes d’un programme linéaire en 0-1 et l’utilisation de l’outil Cplex comme outil pratique pour calculer le degré d’inconsistance. Enfin, nous avons montré expérimentalement qu’en termes de temps d’exécution, Cplex est meilleur que W-Max-SAT et est très proche de l’algorithme glouton pour le problème de couverture par ensembles pondérée qui ne donne que des solutions approximatives.

Troisième partie

Modèles graphiques pour la prise de la décision

Introduction de la troisième partie

Dans cette partie, nous explorons le domaine de prise de décision sous incertitude en utilisant des modèles graphiques possibilistes.

Nous présentons deux modèles graphiques pour le calcul des décisions optimales optimistes : les réseaux possibilistes et les diagrammes d'influence. Nous avons implémenté l'algorithme de calcul des décisions séquentielles optimales optimistes et nous avons fait une étude comparative qui a montré la prédominance du nouveau modèle par rapport à l'algorithme d'évaluation d'un diagramme d'influence en termes de performance.

Nous avons proposé une approche approximative, de complexité polynomiale, pour le calcul de la décision optimale optimiste basée sur la fusion des réseaux possibilistes sans passer par la construction de l'arbre de jonction.

Chapitre 7

Les approches graphiques pour la prise de décision dans l'incertitude

7.1 Introduction

Un agent (humain ou artificiel) met en œuvre un comportement "intelligent" lorsque, en fonction de ses objectifs (buts, préférences, intentions), il raisonne sur les informations dont il dispose pour décider, agir au mieux en exploitant les actions dont il dispose. Le problème de décision est inhérent à la discipline de l'intelligence artificielle. Ce domaine a enregistré des progrès importants, ces dernières décennies, dans la compréhension et la modélisation des processus de prise de décisions et a produit de nombreux systèmes et modèles permettant la prise de décision rationnelle.

La modélisation de la décision ne se ramène pas à une unique fonction à optimiser : la présence de plusieurs critères sur lesquels évaluer la situation, l'incertitude existante sur les événements futurs sont autant de cas où la prise de décision devient un processus très complexe.

Un problème de décision consiste en un choix entre une liste d'alternatives possibles des connaissances d'un agent (des connaissances qui sont parfois entachées d'incertitudes), en considérant les résultats possibles de sa décision exprimée par un ensemble d'utilités. Plusieurs outils de prise de décision ont été développés pour aider les décideurs dans leurs tâches [8, 41, 17, 30, 19, 54, 105].

Dans cette thèse, nous nous focalisons sur les modèles de décision graphiques qui fournissent des outils de décision efficaces en permettant une représentation compacte des problèmes de décision sous incertitude. Le modèle de décision possibiliste qualitative propose deux critères qualitatifs d'utilité pour l'approche de la décision sous incertitude : le critère de décision pessimiste et le critère de décision optimiste.

Le calcul de décisions optimistes revient à calculer le degré de normalisation de l'arbre de jonction correspondant à la fusion du réseau possibiliste codifiant les connaissances de l'agent et le réseau possibiliste codifiant ses préférences. Dans [17], une décomposition des diagrammes d'influence vers deux réseaux possibilistes a été proposée afin de calculer la décision séquentielle. Les approches exactes pour calculer la décision basées sur des réseaux possibilistes sont limitées par la taille des distributions possibles. Une étape importante lors le calcul de la décision est la transformation du DAG en une structure secondaire, connue sous le nom d'arbres de jonction. Cette transformation est connue pour être coûteuse et représente un problème difficile (NP-Hard). Dans [30], une approche approximative en un temps polynomiale a été développée afin de calculer la décision optimale optimiste.

Cette approche est basée sur la transformation du graphe résultant de la fusion des réseaux représentant les croyances et les préférences de l'agent en une structure non-dirigée équivalente, nommée pour des raisons de simplicité, graphe moral.

Dans ce chapitre, nous présentons deux modèles graphiques pour la prise de décision. La section 2 présente les concepts de base d'une théorie de la décision. La section 3 présente d'abord le problème du calcul de décisions séquentielles en se basant sur la décomposition des diagrammes d'influence vers deux réseaux possibilistes. Nous avons présenté une étude comparative afin de valider l'algorithme de calcul de décision optimale optimiste en utilisant deux réseaux possibilistes. Enfin, dans la section 4 une approche approximative a été proposée pour le calcul de la décision sous incertitude dans les réseaux possibilistes sans construire l'arbre de jonction.

7.2 Concepts de base de la théorie de la décision

Les théories de décision proposent des critères quantitatifs et qualitatifs qui combinent l'incertitude et les préférences d'un agent afin déduire la décision optimale [47]. En d'autres termes, les options alternatives désignent l'éventail des choix qui s'offrent au décideur. L'information recouvre l'ensemble des événements pertinents à considérer et leur probabilité d'apparition, et les préférences mesurent la désirabilité relative des différents résultats aux yeux du décideur.

7.2.1 Les choix

Le processus de choix consiste à l'évaluation des alternatives. Formellement, soit A un ensemble fini d'alternatives. Un agent décideur peut effectuer un choix parmi un sous-ensemble d'alternatives de A qui lui sont proposées [57].

7.2.2 Les préférences

Pour qu'un agent puisse exprimer ses préférences, une relation d'ordre doit être définie. Une relation de préférence \succ est définie sur un ensemble de paires d'éléments (a_1, a_2) de $A \times A$ vérifiant, $a_1 \succ a_2$. La relation $a_1 \succ a_2$ est interprétée par : "l'agent préfère strictement a_1 à a_2 ".

Deux idées simples et importantes relient les choix et les préférences. Si un agent possède des préférences, il doit y avoir une implication en termes de choix.

Nous allons illustrer dans l'exemple suivant les notions d'états du monde, d'action et de conséquence [53] :

Exemple 7.1 (*Investissement immobilier*)

Faut-il investir dans une résidence, un immeuble, des appartements ou ne faire aucun investissement ? Cela va dépendre de l'état du marché immobilier : Fort, Moyen, Faible. Soient les quatre actions disponibles suivantes :

- *Investir dans une résidence*
- *Investir dans un immeuble*
- *Investir dans un appartement*
- *Ne pas investir*

Les états du monde sont :

- *Marché immobilier Fort*

- *Marché immobilier Moyen*
- *Marché immobilier Faible*

Décisions(D)/états (S)	Fort	Moyen	Faible
Résidence	550	110	-310
Immeuble	300	129	-100
Appartements	200	100	- 32
Aucun	0	0	0

TABLE 7.1 – Résumé du problème de l’investissement immobilier

Le problème de décision revient à choisir quelle est l’action à appliquer parmi les quatre actions disponibles. Le choix de l’agent est guidé par une relation de préférence sur les conséquences éventuelles des actions applicables et une relations d’incertitude entre les états possibles du monde. En effet, une action peut donner soit de meilleurs résultats dans certains états du monde et soit des résultats médiocres dans d’autres états. Ce qui implique que les plausibilités relatives à des états du monde possibles doivent être prises en considération pour évaluer les actions.

7.3 Le calcul de la décision séquentielle

Dans cette section, nous allons aborder le problème du calcul de décision séquentielle. Ensuite nous allons présenter l’algorithme du calcul de la décision séquentielle.

7.3.1 La démarche du calcul de la décision séquentielle

La résolution des problèmes de décision séquentielle revient à trouver les stratégies les plus optimales. Chaque stratégie peut être décomposée en sous-stratégies. C’est la programmation dynamique qui est utilisée pour trouver les meilleures stratégies. En effet les algorithmes de propagation basés sur la programmation dynamique peuvent être utilisés pour évaluer les stratégies tout en commençant par la dernière décision sachant que les décisions doivent être prises dans l’ordre temporel initialement donné par le décideur [101].

Chaque étape i correspondra au calcul de la meilleure instantiation de la décision D_i en maximisant l’utilité optimiste dans le réseau possibiliste ΠG_{\oplus} tenant compte des valeurs des décisions déjà prises lors des étapes précédentes.

Ce processus peut se faire à travers quatre étapes qui sont :

✓ **La décomposition d’un diagramme d’influence :**

Cette étape permet de décomposer un diagramme d’influence en deux réseaux possibilistes : un réseau possibiliste représentant les connaissances de l’agent et un réseau possibiliste représentant les préférences de l’agent [20]. Deux méthodes ont été élaborées pour spécifier le réseau de préférences : réduire les utilités ou bien garder les utilité telles quelles.

✓ **La fusion des deux reseaux :**

cette étape, comme son nom l’indique, consiste en la fusion de deux réseaux possibilistes ΠK_{min} et ΠP_{min} [119]. Au niveau du problème de la fusion, deux cas de figures se présentent : le cas où les deux réseaux ont la même structure et le cas contraire.

1. **Fusion des réseaux possibilistes de structures identiques** Soient $N1 = (\pi_{N1}, G_{N1})$ et $N2 = (\pi_{N2}, G_{N2})$ deux réseaux possibilistes, tels que $G_{N1} = G_{N2}$. La fusion de $N1$ et $N2$ est un réseau possibiliste noté $N_{\oplus} = (\pi_{\oplus}, G_{\oplus})$ où :
 - $G_{\oplus} = G_{N1} = G_{N2}$.
 - Pour chaque variable X , $\pi_{\oplus}(X|Par(X)) = \min(\pi_{N1}(X|Par(X)), \pi_{N2}(X|Par(X)))$ ¹.

2. **Fusion des réseaux possibilistes de structures différentes**

Deux cas sont alors distingués :

L'union des graphes est acyclique : le graphe résultant ne contient pas de cycles, dit $G1$ et $G2$ sont alors des réseaux U-acycliques où :

- L'ensemble de ses variables est l'union des ensembles des variables appartenant à $G1$ et à $G2$.
- Pour chaque variable A , ses parents sont ceux de $G1$ et de $G2$.

L'union des graphes est cyclique : Deux approches sont proposées dans [119]. La première méthode consiste à renommer toutes les variables du graphe considéré afin d'éviter toute confusion. La seconde méthode met le point sur la minimisation des variables dupliquées. En effet, cette méthode est basée sur le processus de décomposition d'un des graphes à fusionner de telle façon que le processus de fusion appliqué n'entraîne plus de cycles.

- ✓ **La construction de l'arbre de jonction :** Un algorithme d'arbre de jonction est appliqué. L'objectif de cet algorithme est de compiler le graphe original en une structure secondaire [?] :
 - **Moralisation du graphe initial :** Cette étape permet de créer un graphe non orienté à partir du graphe initial en rajoutant des liens entre les parents de chaque variable.
 - **Triangulation du graphe moral :** Cette étape a pour but d'identifier des ensembles de variables qui peuvent être regroupées sous forme de cliques notées C_i . Notons qu'il est important de trouver une triangulation optimale qui va minimiser la taille des cliques afin de rendre les calculs locaux possibles. Ce problème est connu comme étant NP-complet et plusieurs heuristiques ont été proposées pour le résoudre.
 - **Construction de l'arbre de jonction JT :** Cette étape consiste à connecter les cliques identifiées lors de l'étape précédente, avec la condition que toutes les cliques se trouvant dans un chemin entre cliques C_i et C_j doivent contenir $C_i \cap C_j$. Une fois les cliques adjacentes identifiées, un séparateur, noté S_{ij} , est inséré entre chaque paire de cliques C_i et C_j , contenant leurs variables communes.
- ✓ **L'initialisation :** l'étape d'initialisation est répétée pour chaque décision d_i dans le cadre de la quantification de l'arbre de jonction.

Pour chaque clique C_i (resp. séparateur S_{ij}), un potentiel est affecté $\pi_{C_i}^t$ (resp. $\pi_{S_{ij}}^t$) où t est relatif à l'étape de propagation. En particulier, $t = I$ (resp. $t = C$) correspond à l'étape d'initialisation (resp. de cohérence globale).

Ces potentiels permettent d'associer à l'arbre de jonction JT une distribution de possibilité jointe, unique, notée π_{JT}^t définie par (7.1) :

$$\pi_{JT}^t(A_1, \dots, A_N) = \min_{i=1, \dots, n} \pi_{C_i}^t \quad (7.1)$$

1. $Par(X)$, est l'ensemble des parents de X

où n est le nombre de cliques dans JT .

Les grandes lignes de la procédure d'initialisation peuvent être résumées comme suit :

- Pour chaque clique C_i , affecter une distribution uniforme : $\pi_{C_i}^I \leftarrow 1$.
où 1 signifie la distribution de possibilité où tous les éléments sont égaux à 1.
- Pour chaque séparateur S_{ij} , affecter une distribution uniforme : $\pi_{S_{ij}}^I \leftarrow 1$.
- Pour chaque variable A , choisir une clique C_i contenant $A_i \cup Par_{A_i}$ et mettre à jour ses distributions de possibilité locales :

$$\pi_{C_i}^t : \pi_{C_i}^t \leftarrow \min(\pi_{C_i}^t, \pi_{\oplus}(A_i | Par_{A_i}))$$

- Coder l'évidence $D = d_i$ par la distribution $\Lambda_D(d)$ suivante :

$$\Lambda_D(d) = \begin{cases} 1 & \text{si D est instanciée à d} \\ 0 & \text{si D est instanciée à une valeur différente de d.} \end{cases} \quad (7.2)$$

- Choisir une clique C_i contenant D, et mettre à jour son potentiel comme suit :

$$\pi_{C_i}^t \leftarrow \min(\pi_{C_i}^t, \Lambda_D). \quad (7.3)$$

- ✓ **La propagation globale** : cette étape permet d'assurer la cohérence globale de l'arbre de jonction. La propagation globale induit à chaque cluster de passer des messages à l'ensemble de ses clusters adjacents. Ces messages sont ordonnés afin que chaque passage de message préserve la consistance introduite par le passage de message précédent. Une fois la propagation globale est achevée, chaque paire (cluster, séparateur) est consistante et l'arbre de jonction est localement consistant.

À partir d'un arbre de jonction de n clusters, l'algorithme de propagation globale sélectionne un cluster pivot arbitraire et effectue $2^*(n-1)$ passages de messages en deux phases :

- ✓ **Collecte de l'évidence** : Chaque cluster passe un message à ses clusters adjacents dans la direction du pivot, en commençant par les clusters les plus éloignés de celui-ci.
- ✓ **Distribution de l'évidence** : Chaque cluster envoie des messages à ses clusters adjacents, en s'éloignant de la direction du pivot et ceci en commençant par le pivot lui-même jusqu'à atteindre les feuilles.

Le schéma suivant donne un aperçu un peu plus détaillé sur la démarche de calcul de la décision optimiste :

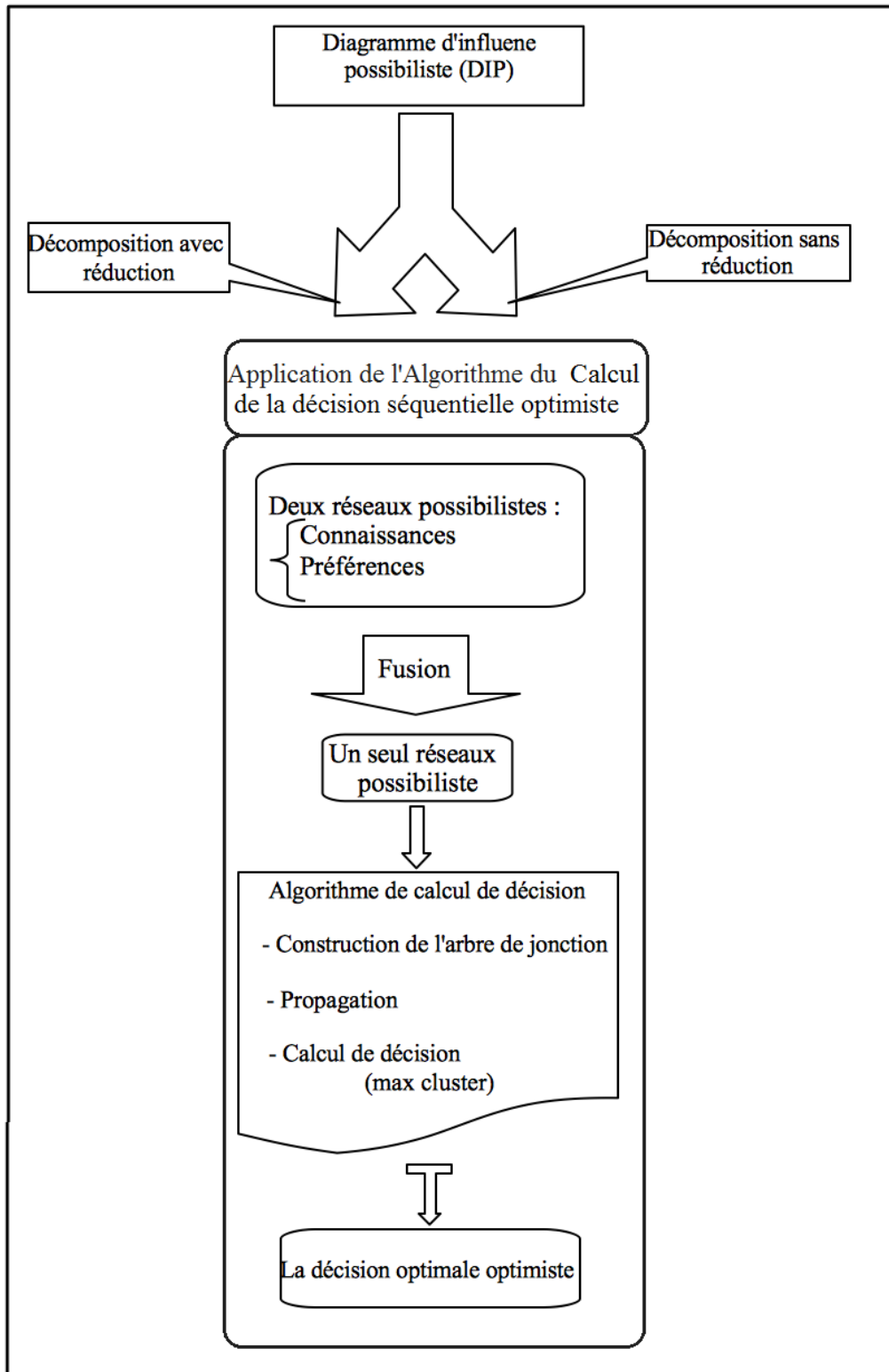


FIGURE 7.1 – Le calcul de décision en se basant sur la décomposition d'un diagramme d'influence

7.3.2 Algorithme du calcul de la décision séquentielle

L'algorithme qui calcule la décision séquentielle optimiste accepte comme entrée un diagramme d'influence et un ensemble des décisions est $D = \{d_1, \dots, d_p\}$ retourne la décision

séquentielle est le suivant [127] :

Algorithm 5 L'algorithme de calcul de la décision séquentielle

Require: $\Pi ID_{min} = (\pi_{ID}, \mu_{ID})$: un diagramme d'influence possibiliste

Require: $D = \{d_1, \dots, d_p\}$: un ensemble des décisions

Ensure: D : Ensemble de décisions

$i = 1, \text{norm} = 0, u^* = 0$

$\text{Decision} = \emptyset, \text{Decision_Seq} = \emptyset$

$[\Pi K_{min}, \Pi K_{min}] = \text{Decomposition_Diagramme_Influence}(\Pi ID_{min})$

$\Pi G_{\oplus} = \text{Fusion}(\Pi K, \Pi P)$

$JT = \text{Construction_Arbre_de_Jonction}(\Pi G_{\oplus})$

for $i = 1$ à p **do**

for $\text{inst} = 1$ à m **do**

$D_j \leftarrow \text{Decision_Seq} \cup D_{inst}$

Initialisation_Arbre_de_Jonction(JT, D_i)

 Norme = **Propagation**(JT, D_i)

if Norm $> u^*$ **then**

$u^* \leftarrow \text{Norm}$

$\text{Decision_Seq} \leftarrow D_j$

else

if Norm == u^* **then**

$\text{Decision_Seq} \leftarrow \text{Decision_Seq} \cup D_j$

else

$\text{Decision} \leftarrow \text{Decision_Seq}$

end if

end if

end for

end for

Afin de mieux comprendre le déroulement du processus de la décomposition diagramme d'influence et le processus calcul de décision, nous considérons l'exemple concret comment choisir la décision optimale optimiste.

Exemple 7.2 Dans le cadre de la préparation d'une omelette. Cinq œufs ont été déjà cassés dans un bol, le sixième-tenu dans la main-semble suspect : est-il pourri ou sain ?

Soient les trois actions disponibles suivantes :

— Casser l'œuf dans l'omelette

— Le jeter directement à la poubelle

— Sortir une tasse pour y casser l'œuf et examiner son état de fraîcheur, avant éventuellement l'incorporer à l'omelette.

Les conséquences possibles sont : avoir une grande omelette, perdre les 5 œufs, avoir une assiette inutilement salie ou bien avoir une petite omelette.

Avant de présenter le diagramme d'influence possibiliste, nous allons tout d'abord présenter l'ensemble des nœuds utilisés ainsi que leurs significations.

— F : indique que l'œufs est soit bon/pourri.

— OF : état qui indique est ce que nous avons déjà observé l'œuf ou non.

— BIO : indique que nous avons cassé l'œuf dans l'omelette.

— BAC : indique que nous avons cassé l'œuf dans le bol.

— Rc, Ro et Rs : indique les gains.

Les Figures 7.2 et 7.2 représentent le diagramme d'influence avec ses distributions de possibilités que nous allons utiliser dans nos calculs.

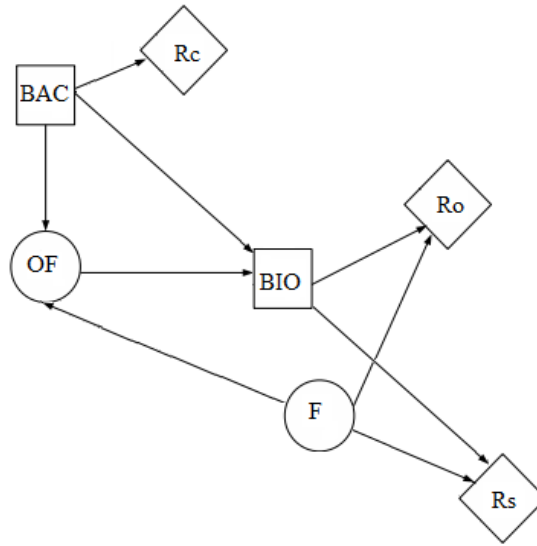


FIGURE 7.2 – Le Diagramme d'influence

BAC	$\mu(Rc)$
T	1
F	0

F	$\pi(F)$
T	1
F	1

F	BAC	$\mu(Rs)$
T	T	1
T	F	0.2
F	T	1
F	F	1

F	BIO	$\mu(Ro)$
T	T	1
T	F	0.8
F	T	0
F	F	1

F	BAC	$\pi(OF=T F, BAC)$	$\pi(OF=F F, BAC)$
T	T	1	0
T	F	0	0
F	T	0	1
F	F	1	1

TABLE 7.2 – Les distributions de possibilités

Application du modèle basé sur la fusion de deux réseaux possibilistes sans réduction

L'Algorithme 7.3.2 prend en entrée un diagramme d'influence et un ensemble des décisions $D = \{BAC, BIO\}$.

La première étape est la décomposition de diagramme d'influence en deux réseaux possibilistes, un représente le réseau de connaissances et l'autre représente le réseau de préférences. Les réseaux obtenus sont schématisés par les Figures 7.3 et 7.4.

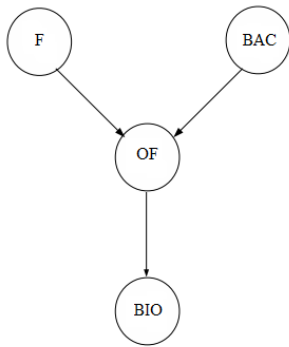


FIGURE 7.3 – Le réseau de connaissances

Le réseau obtenu de la fusion des deux réseaux est schématisé par la Figure 7.5.

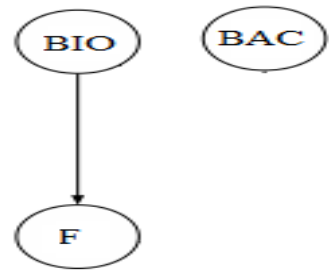


FIGURE 7.4 – Le réseau de préférences

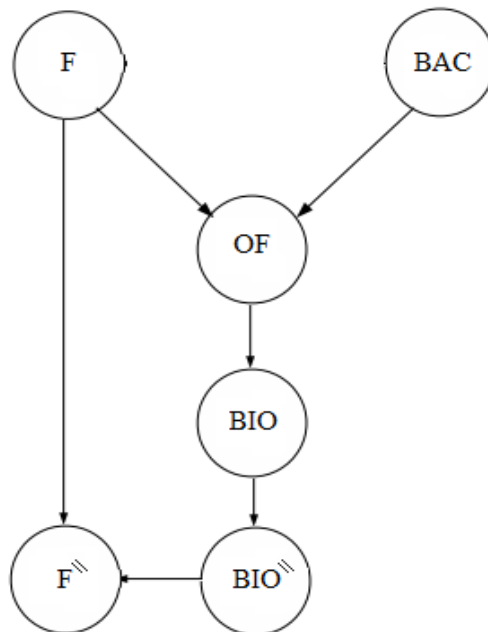


FIGURE 7.5 – Le réseau résultant de la fusion des deux réseaux (7.3 et 7.4)

La troisième étape, consiste à construire de l'arbre de jonction (voir la Figure 7.6) :

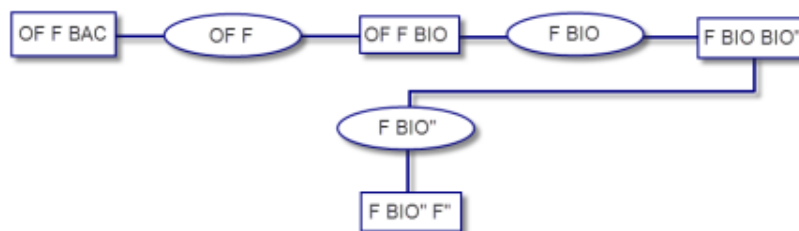


FIGURE 7.6 – L'arbre de jonction

La quatrième étape, consiste à calculer la décision optimale optimiste. Le calcul se déroule en plusieurs étapes :

$Max = 0;$

$Décision = \emptyset;$

1. étape 1 : (i=1)

— $dec = (BIO : Vrai);$

— Le calcul du degré de normalisation : **CalculDecision** (JT, dec)=1;

— $degré = 1;$

— $degré > Max$

— $Max = degré;$

— $Décision = (BIO : true);$

— $dec = (BIO : Faux);$

— Le calcul du degré de normalisation : **CalculDecision** (JT, dec)=0.8;

— $degré = 0.8;$

— $degré < Max$

$Décision = (BIO : vrai);$

2. étape 2 : (i=2)

$Max = 0;$

— $dec = (BIO : true, BAC : vrai);$

— Le calcul du degré de normalisation donne : **CalculDecision** (JT, dec)=0;

— $degré = 0;$

— $Max < degré$

— $Max = degré;$

— $Décision = (BIO : true, BAC : vrai);$

— $dec = (BIO : vrai, BAC : Faux);$

— Le calcul du degré d'inconsistance donne : **CalculDecision** (JT, dec)=1;

— $degré = 1;$

— $Max < degré$

— $Max = degré;$

— $Décision = (BIO : vrai, BAC : Faux);$

Conclusion : La décision optimiste optimale est "BIO" : "Casser l'œuf dans l'omelette".

7.3.3 Etude expérimentale

Dans cette partie, nous présentons les résultats obtenus à l'issue de l'application de notre système de calcul de décision. Ces résultats sont répartis en deux parties. La première partie comporte les résultats obtenus en utilisant l'algorithme de calcul de décision optimale optimiste en se basant sur les diagrammes d'influence [59] et l'algorithme de calcul de décision optimale optimiste en se basant sur la décomposition d'un diagramme d'influence [20, 17]. La seconde partie est dédiée aux résultats obtenus en utilisant l'algorithme de calcul de décision optimale optimiste en se basant sur la décomposition d'un diagramme d'influence en deux réseaux possibilites avec et sans réduction des nœuds d'utilité.

Pour l'implémentation des algorithmes, nous avons opté pour le langage *MATLABTM* qui est à la fois un langage de programmation et un environnement d'exécution. *Matlab(MATrixLABoratory)*

est un logiciel pour le calcul scientifique, orienté vers les vecteurs et les listes de données. L'implémentation est basée sur l'outil "PNT" qui est un package (open-source) de Matlab destiné aux modèles graphiques possibilistes.

Nous avons décidé de tester notre application sur quatre types de diagrammes d'influence possibilistes, que nous allons détailler dans ce qui suit. Pour chaque DIP de ces quatre types, nous avons généré un ensemble d'échantillons, en variant le nombre de nœuds tout en respectant la définition de chacun de ces DIPs [28]. Les schémas suivants donnent un aperçu des DIPs de type 1, 2, 3 et 4 sur lesquels nous avons travaillé.

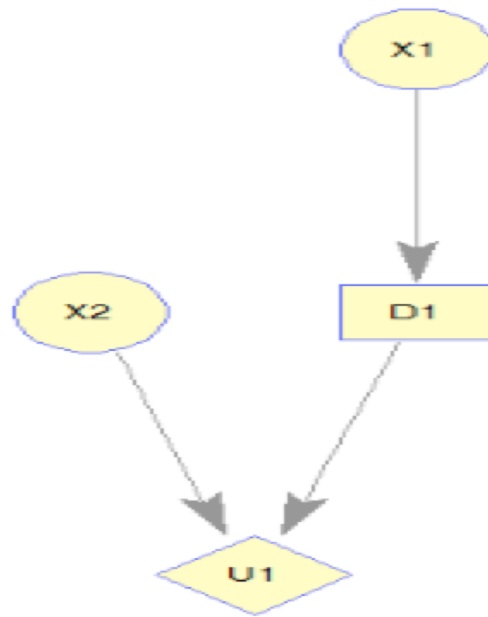


FIGURE 7.7 – Exemple d'un diagramme possibiliste de type 1

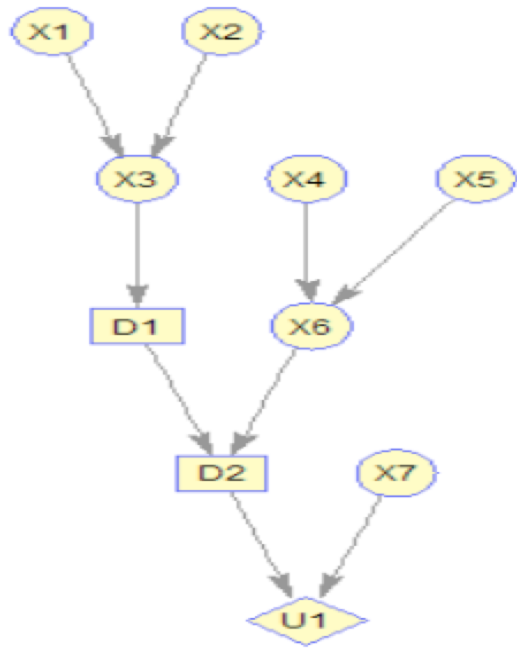


FIGURE 7.8 – Exemple d'un diagramme possibiliste de type 2

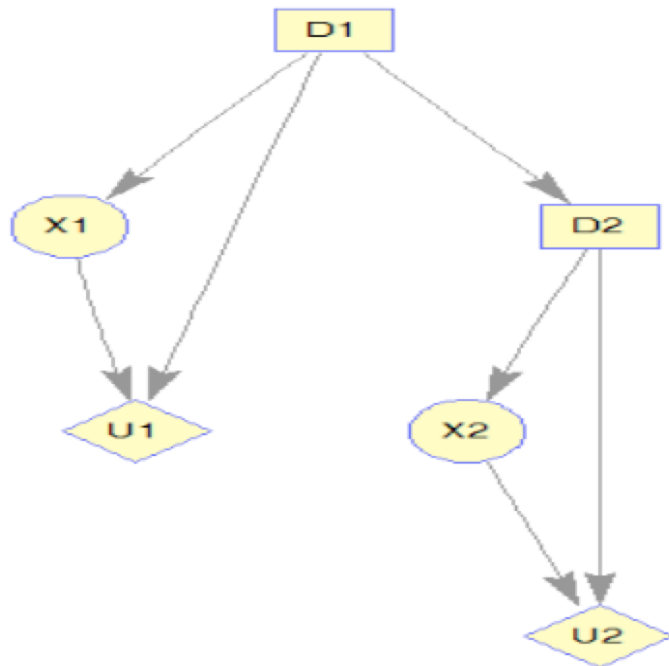


FIGURE 7.9 – Exemple d'un diagramme possibiliste de type 3

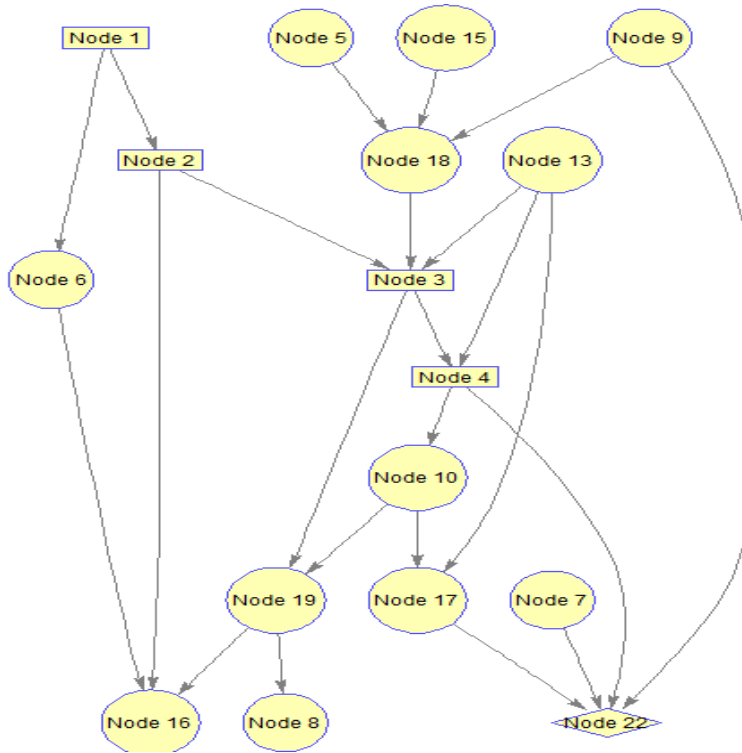


FIGURE 7.10 – Exemple d’un diagramme possibiliste de type 4

- **Diagramme d’influence possibiliste de type 1** : Il contient un seul nœud d’utilité et un seul nœud de décision (c’est le modèle le plus simple).
- **Diagramme d’influence possibiliste de type 2** : ce modèle représente un diagramme d’influence contenant un seul nœud d’utilité mais plusieurs nœuds de décision.
- **Diagramme d’influence possibiliste de type 3** : ce modèle représente un diagramme d’influence d’une complexité moyenne, qui contient plusieurs nœuds d’utilité et plusieurs nœuds de décision.
- **Diagramme d’influence possibiliste de type 4** : ce modèle représente un diagramme d’influence plus complexe que le précédent diagramme. Sa structure est aléatoire.

L’objectif de la première partie est de comparer les performances des algorithmes de raisonnement à savoir :

- *Le premier modèle* : l’algorithme de calcul de décision optimale optimiste en se basant sur la fusion des réseaux possibilistes résultant de la décomposition d’un diagramme d’influence [20, 17].
- *Le deuxième modèle* : l’algorithme de calcul de décision optimale optimiste en se basant sur les diagrammes d’influence [59].

Cette expérimentation a été élaborée comme suit :

- Pour chacun des quatre types de diagramme d’influence que nous avons défini, nous avons effectué les traitements suivants :
 - La génération d’un ensemble d’échantillons de DIP de ce type.
 - L’application de l’algorithme de calcul de décision optimale optimiste en se basant sur le diagramme d’influence [59].

- La décomposition de chaque diagramme d'influence généré en deux réseaux possibilistes qualitatifs, un représente le réseau de connaissances et l'autre représente le réseau de préférences [20, 17].
- L'application de l'algorithme de calcul de décision optimale optimiste en se basant sur la fusion des réseaux possibilistes [128].

Les Figures 7.11, 7.12, 7.13, 7.14 et 7.15 présentent les résultats obtenus (temps d'exécution) pour la recherche de la décision pour les deux modèles en fonction du nombre de nœuds utilisés.

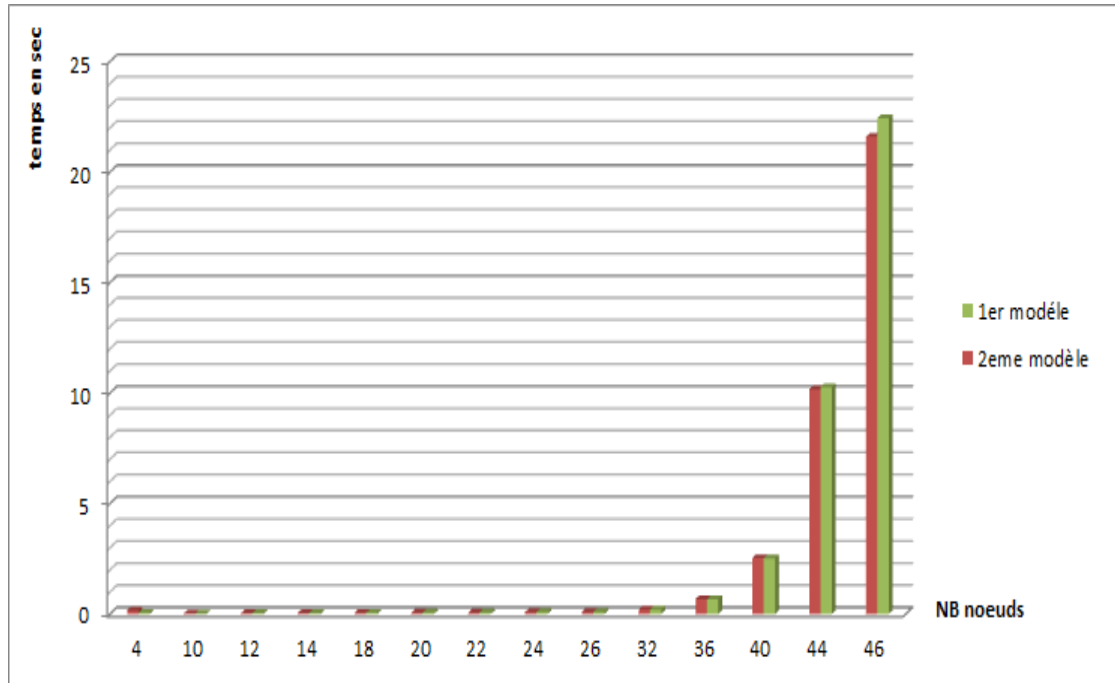


FIGURE 7.11 – Le temps de calcul de la décision optimiste pour les DIPs de type1

Nous remarquons que pour les diagrammes d'influence avec un seul nœud utilité et un seul nœud de décision, d'une taille inférieure à 40 nœuds, le temps de réponse reste stable. Mais dès que le nombre de nœuds dépasse un seuil, à partir de 44 nœuds pour notre expérimentation, c'est le deuxième modèle qui permet le calcul en un temps optimal. Bien que ces résultats ne confirment pas notre objectif de validation, ils n'influencent pas trop sur la qualité des résultats du premier modèle.

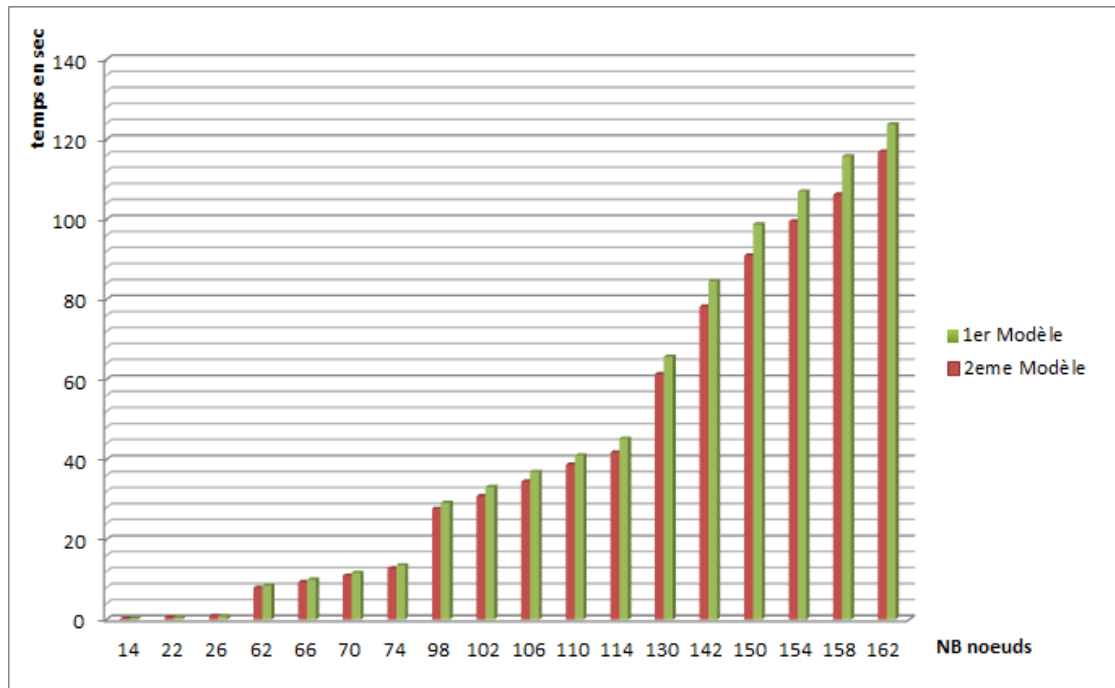


FIGURE 7.12 – Le temps de calcul de la décision optimiste pour les DIPs de type2

Nous remarquons que pour les diagrammes d'influence avec un seul nœud utilité et plusieurs nœuds de décisions, que quelque soit sa taille c'est le deuxième modèle qui se comporte le mieux. Nous pouvons voir aussi, que la différence du temps de réponse reste stable.

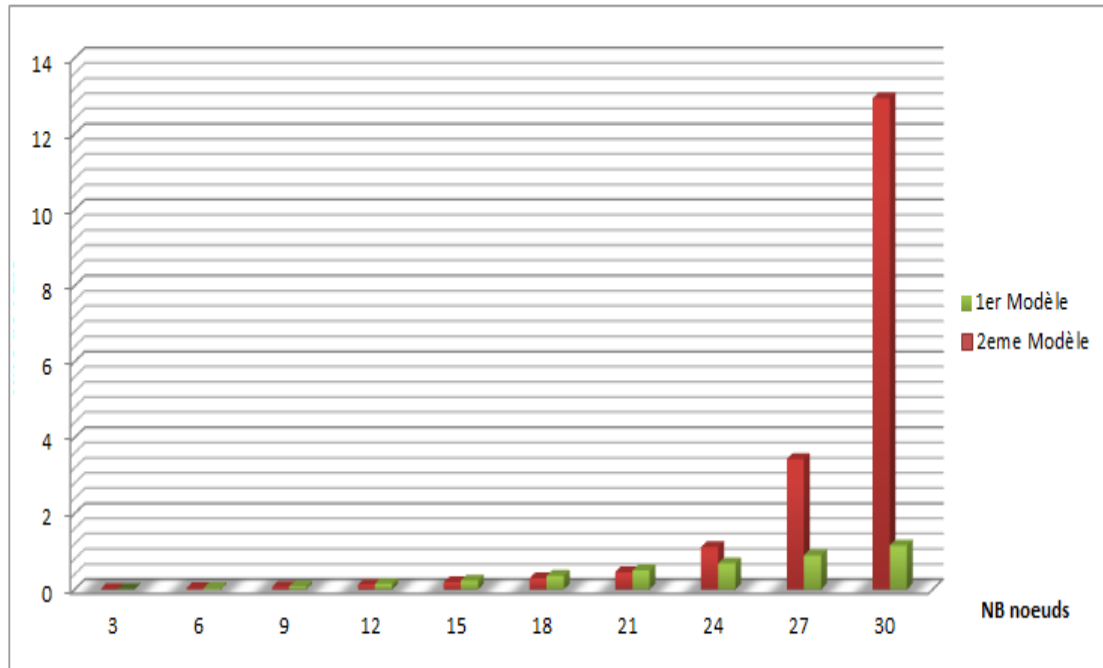


FIGURE 7.13 – Le temps de calcul de la décision optimiste pour les DIPs de type3

Nous remarquons que pour les diagrammes d'influence avec plusieurs utilité et plusieurs nœuds de décisions, la différence entre les deux modèles est négligeable bien que c'est le premier modèle qui l'emporte. Mais après avoir utilité 24 nœuds et plus, le premier modèle se détache largement du deuxième modèle et la différence des temps de réponse devient importante.

Quelque soit la taille du diagramme d'influence, c'est le premier modèle qui se comporte le mieux.

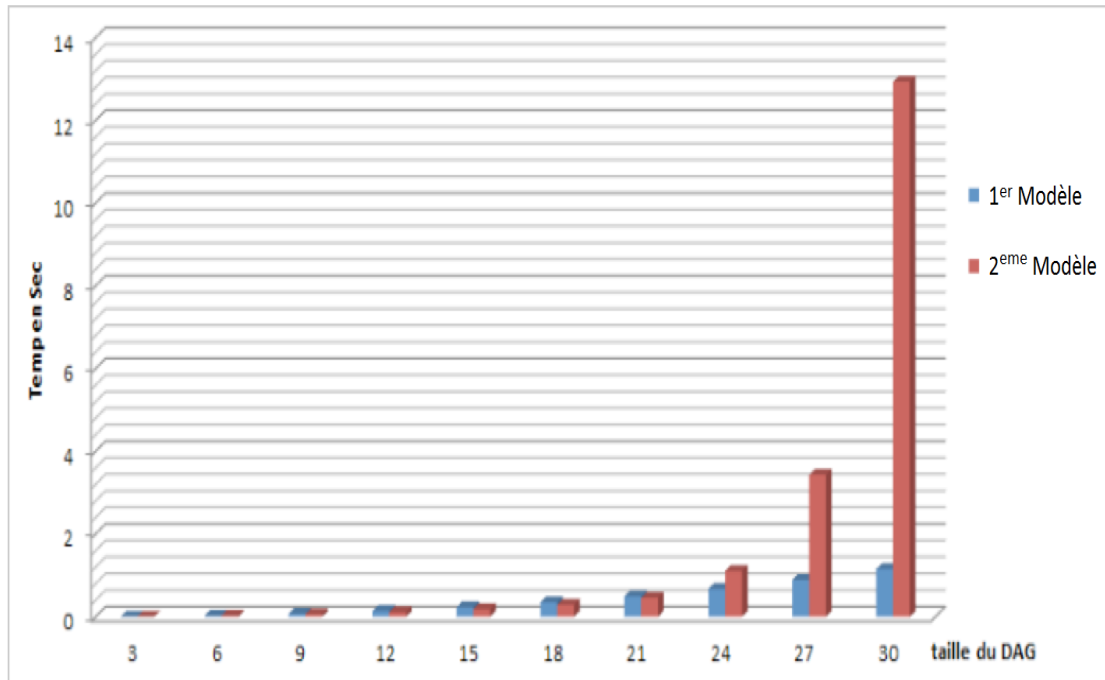


FIGURE 7.14 – Le temps de calcul de la décision optimiste pour les DIPs de type4

Nous remarquons que pour les diagrammes d’influence aléatoires, comme pour les diagrammes avec nœuds utilité et plusieurs nœuds de décisions, c’est le premier modèle qui l’emporte. La différence entre les temps de réponse est importante.

À partir des résultats des tests précédents, nous remarquons que quelque soit la taille du diagramme d’influence en fonction du nombre du nœuds, c’est le premier modèle qui se comporte le mieux en induisant un temps de calcul inférieur à celui du deuxième modèle. La performance du premier modèle par rapport au deuxième modèle est expliquée par le fait que le premier modèle est plus adapté au calcul plus complexe étant donné que les diagrammes d’influence de type 1 et 2 sont des cas spéciaux du diagrammes d’influence de type 3.

Les résultats obtenus, en ne considérant que le nombre de nœuds indépendamment du type du diagramme, confirment l’objectif de la validation et sont vraiment encourageants. Nous pouvons conclure que le premier modèle *est le plus performant* en temps de calcul de décision optimale optimiste.

Comme nous l’avons déjà souligné, le premier modèle permet de faire des calculs avec et sans réduction de nœuds d’utilité. Nous avons décidé d’effectuer des tests de calcul de décision en se basant sur les mêmes échantillons que nous avons utilisé au niveau de la comparaison précédente. Cette expérimentation a été élaborée comme suit : pour chaque échantillon DIP, le DIP a été décomposé en deux réseaux possibilistes qualitatifs avec réduction de nœuds d’utilité, un représente le réseau de connaissances et l’autre représente le réseau de préférences.

Ensuite, nous avons appliqué l’algorithme de calcul de décision optimale optimiste en se basant sur la fusion des réseaux possibilistes et nous avons calculé le temps d’exécution nécessaire. Pour le même DIP, nous avons décomposé ce dernier en deux réseaux possibilistes qualitatifs sans réduction de nœuds d’utilité et nous avons calculé le temps d’exécution. La dernière étape est la comparaison entre les deux temps. Les résultats obtenus sont donnés

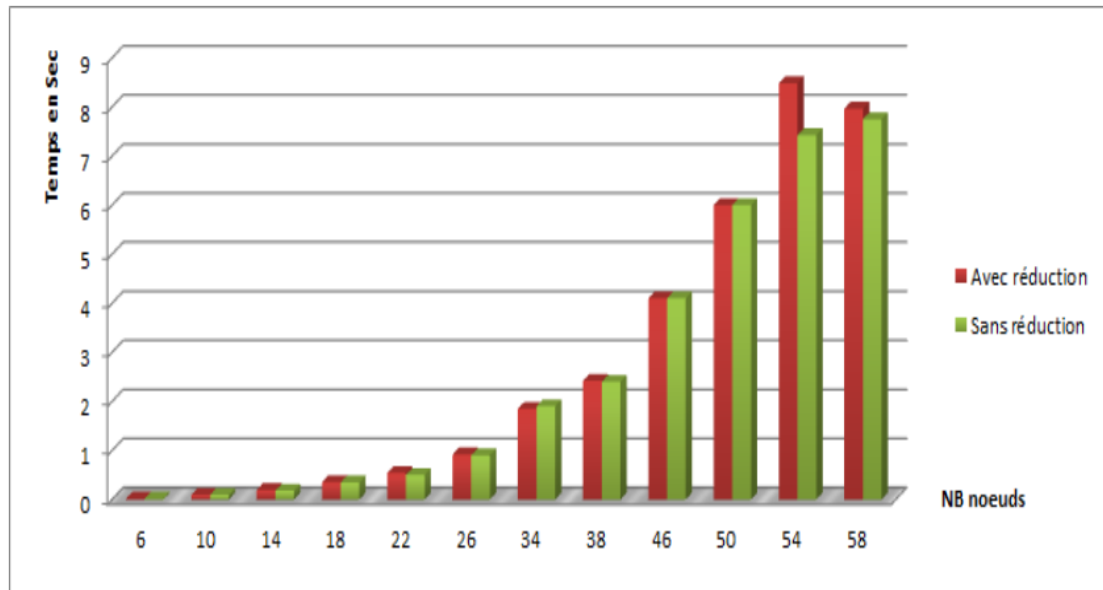


FIGURE 7.15 – Comparaison entre le temps de réponse du premier modèle avec et sans réduction

par la Figure suivante :

A partir de l’histogramme précédent, nous pouvons voir que le modèle de calcul sans réduction est plus performant que le modèle avec réduction. Ceci est expliqué par le fait que la réduction du nombre de nœuds d’utilité ne fait qu’augmenter la complexité du graphe ainsi que le calcul des distributions de possibilités jointes.

7.4 L’approche approximative pour la prise de la décision exclusive

Les réseaux possibilistes basés sur le min sont des outils efficaces pour une représentation compacte des problèmes de décision sous incertitude. Les approches exactes pour calculer la décision basée sur des réseaux possibilistes sont limitées par la taille des distributions de possibilité. Généralement, ces approches sont basées sur des algorithmes de propagation [19, 18]. Une étape importante dans le calcul de la décision est la transformation du DAG en une structure secondaire, connue sous le nom d’arbres de jonction [11]. Cette transformation est connue pour être coûteuse et représente un problème difficile.

Dans cette section, nous présentons une nouvelle approche approximative pour le calcul de la décision possibiliste qualitative dans un contexte graphique. Cette approche exploite les procédures de fusion des réseaux possibilistes développées dans [119]. Elle permet d’éviter la transformation du graphe initial en un arbre de jonction qui est connu comme un problème difficile [31].

7.4.1 La démarche du calcul de la décision approximative

Ce processus peut se faire à travers cinq étapes :

- ✓ La fusion des deux réseaux
- ✓ La construction du graphe moral

- ✓ **L'initialisation**
- ✓ **Procédure de stabilité simple**
- ✓ **Procédure de stabilité multiple**

7.4.1.1 Fusion des réseaux possibilistes

Un des objectifs de la fusion d'informations incertaines est d'exploiter la complémentarité entre les sources d'informations afin d'avoir un point de vue global plus complet et plus précis sur un problème donné. Étant donné un ensemble de distributions de possibilité π_i , le mode de combinaison de base, dans le cadre de la théorie des possibilités, est la conjonction idempotente (c.à.d., le minimum) des π_i , définie par :

$$\forall \omega \pi_{\oplus}(\omega) = \min_{i=1,n} \pi_i(\omega) \quad (7.4)$$

Dans [119], les auteurs proposent deux classes principales pour fusionner des réseaux possibilistes :

1. Fusion des réseaux possibilistes de structures identiques

Le résultat de la fusion de $N1$ et $N2$ est un réseau possibiliste dont DAG est celui de $N1$ et $N2$, et dont les distributions de possibilités locales sont le minimum des distributions locales de $N1$ et $N2$ [119].

- $G_{\oplus} = G_{N1} = G_{N2}$.
- Pour chaque variable X , $\pi_{\oplus}(X|Par(X)) = \min(\pi_{N1}(X|Par(X)), \pi_{N2}(X|Par(X)))$ ².

2. Fusion des réseaux possibilistes ayant des structures différentes

Dans le cas où les réseaux possibilistes ont des structures différentes, deux cas sont alors distingués :

L'union des graphes est acyclique : le graphe résultant ne contient pas de cycles.

$G1$ et $G2$ sont alors des réseaux U-acycliques où :

- L'ensemble de ses variables est l'union des ensembles des variables appartenant à $G1$ et à $G2$.
- Pour chaque variable A , ses parents sont ceux de $G1$ et de $G2$.

L'union des graphes est cyclique : Dans le cas où le graphe est avec cycles [119], deux approches sont proposées. La première méthode consiste à renommer toutes les variables du graphe considéré afin d'éviter toute confusion. La seconde méthode met le point sur la minimisation des variables dupliquées. En effet, cette méthode est basée sur le processus de décomposition d'un des graphes à fusionner de telle façon que le processus de fusion appliqué n'entraîne plus de cycles.

7.4.1.2 La construction du graphe moral

La première étape consiste à transformer le réseau initial en une structure secondaire équivalente, appelée graphe moral et noté MG [10] en regroupant chaque nœud avec ses parents dans une même clique. Chaque arc dans MG est étiqueté par l'intersection de ses cliques adjacentes C_i et C_j , dans un séparateur noté S_{ij} .

2. $Par(X)$, est l'ensemble des parents de X

7.4.1.3 L'initialisation pour chaque décision d_i

Une fois le graphe moral construit, cette étape permet de le quantifier en transformant les distributions de possibilité conditionnelles initiales en des distributions jointes locales rattachées aux cliques et aux séparateurs. Plus spécifiquement, pour chaque clique C_i (resp. Séparateur S_{ij}), un potentiel $\pi_{C_i}^t$ (resp. $\pi_{S_{ij}}^t$) est affecté où t est relatif à l'étape de propagation. Les grandes lignes de la procédure d'initialisation peuvent être résumées comme suit :

- Pour chaque clique C_i , affecter une distribution uniforme : $\pi_{C_i}^I \leftarrow 1$.
où 1 signifie la distribution de possibilité où tous les éléments sont égaux à 1.
- Pour chaque séparateur S_{ij} , affecter une distribution uniforme : $\pi_{S_{ij}}^I \leftarrow 1$.
- Pour chaque variable A , choisir une clique C_i contenant $A_i \cup Par_{A_i}$ et mettre à jour ses distributions de possibilité locales :

$$\pi_{C_i}^t : \pi_{C_i}^t \leftarrow \min(\pi_{C_i}^t, \pi_{\oplus}(A_i | Par_{A_i}))$$

- Coder l'évidence $D = d_i$ par la distribution $\Lambda_D(d)$ suivante :

$$\Lambda_D(d) = \begin{cases} 1 & \text{si } D \text{ est instanciée à } d \\ 0 & \text{si } D \text{ est instanciée à une valeur différente de } d. \end{cases} \quad (7.5)$$

- Choisir une clique C_i contenant D , et mettre à jour son potentiel comme suit (7.5) :

$$\pi_{C_i}^t \leftarrow \min(\pi_{C_i}^t, \Lambda_D). \quad (7.6)$$

7.4.1.4 Procédure de stabilité simple

La procédure de stabilisation simple garantit que le potentiel de chaque clique est en accord avec celui de ses cliques adjacentes. Elle est définie comme suit :

Définition 7.1 Soient C_i et C_j deux cliques adjacentes dans un graphe moral MG et soit S_{ij} leur séparateur. Le séparateur S_{ij} est dit stable si :

$$\max_{C_i} \max_{S_{ij}} \pi_{C_i}^I = \max_{C_j} \max_{S_{ij}} \pi_{C_j}^I. \quad (7.7)$$

où $\max_{C_i} \max_{S_{ij}}$ (resp. $\max_{C_j} \max_{S_{ij}}$) est la distribution marginale de S_{ij} définie à partir de $\pi_{C_i}^I$ (resp. $\pi_{C_j}^I$).

Un graphe moral MG est stable si tous ses séparateurs sont stables.

7.4.1.5 Procédure de stabilité multiple

Dans [11], les auteurs ont prouvé que la procédure de stabilité simple ne garantit pas toujours des marginaux précis. Il faut stabiliser chaque clique par rapport à toutes ses cliques adjacentes, mais cela peut s'avérer très coûteux en termes de calcul si le nombre de cliques est important.

Pour cela, dans [11], l'auteur a proposé plusieurs méthodes pour stabiliser le graphe moral. Parmi ces méthodes, nous considérons le cas de la stabilisation de deux parents.

Le principe de cette procédure est de garantir pour chaque clique, avec au moins deux parents, sa stabilisation sur chaque paire de parents [10].

Une fois la stabilité atteinte, le calcul de l'utilité qualitative sur une décision d sera obtenu.

Proposition 7.1 Soient $\Pi K_{\min} = (G_K, \pi_K)$ un réseau possibiliste représente les connaissances de l'agent et $\Pi P_{\min} = (G_P, \mu)$ un réseau possibiliste représente les préférences de l'agent.

ΠG_{\oplus} est le résultat de la fusion de ΠK_{\min} et ΠP_{\min} . Soit MG un graphe moral ΠG_{\oplus} . Le calcul de décision optimiste revient à calculer le degré de normalisation de MG :

$$U^*(d) = \max_{C_i} \pi_{C_i}. \quad (7.8)$$

7.4.2 Algorithme approximatif

Le calcul de la décision optimiste est obtenu en utilisant l'algorithme suivant [31] :

Algorithm 6 Méthode approximative pour calculer la décision

Require: $\Pi K_{\min} = (G_K, \pi_K)$: réseau de connaissances

Require: $\Pi P_{\min} = (G_P, \mu)$: réseau de préférences

Require: $D = \{d_1, \dots, d_p\}$: un ensemble des décisions

Ensure: D : Ensemble de décisions

Decision $\leftarrow \emptyset$

$\Pi G_{\oplus} = \text{Fusion}(G_{\oplus}, \pi_{\oplus})$

$MG = \text{Construction_Graphe_Morale}(\Pi G_{\oplus})$

for $i = 1$ à n **do**

Initialisation(MG, D_i)

 Norme \leftarrow **Procédure_stabilité_simple**(MG, D_i)

 Norme \leftarrow **Procédure_stabilité_multiple**(MG, D_i)

if Norme $> u^*$ **then**

Decision $\leftarrow D_i$

$u^* \leftarrow$ Norme

else

if Norme $== u^*$ **then**

Decision \leftarrow *Decision* $\cup D_i$

end if

end if

end for

Exemple 7.3 Nous considérons le problème de décider si nous devrions ou ne pas prendre un parapluie, sachant qu'il pleut. Nous présentons d'abord l'ensemble des nœuds utilisés dans les réseaux et leurs significations.

- R : Il pleut
- W : L'herbe est humide.
- UM : Prenez le parapluie.
- C : Atmosphère nuageuse.

Les deux réseaux possibilistes représentant les connaissances et les préférences de l'agent sont désignés par ΠK_{\min} et ΠP_{\min} respectivement.

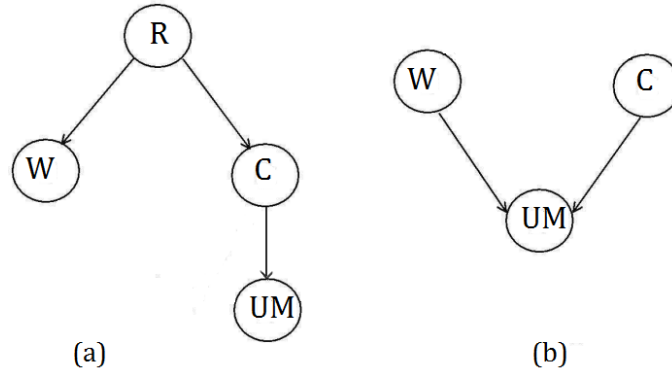


FIGURE 7.16 – Les réseaux possibiliste de connaissances et de préférences de l’agent

— Réseau de connaissances : La composante graphique G_K est donnée par la Figure 7.3(a). Les distributions de possibilité initiales sont données par les Tables 7.3 et 7.4.

R	$\Pi_K(R)$
$r1$	0.9
$r2$	1.0

TABLE 7.3 – Distribution de possibilité initiale $\Pi_{K_{\min}}$

W	R	$\Pi_K(W R)$	C	R	$\Pi_K(C R)$	UM	C	$\Pi_K(UM C)$
$w1$	$r1$	0.4	$c1$	$r1$	1.0	$um1$	$c1$	1.0
$w1$	$r2$	1.0	$c1$	$r2$	0.2	$um1$	$c1$	1.0
$w2$	$r1$	1.0	$c2$	$r1$	0.3	$um2$	$c1$	1.0
$w2$	$r2$	0.0	$c2$	$r2$	1.0	$um2$	$c1$	1.0

TABLE 7.4 – Distribution de possibilité initiale $\Pi_{K_{\min}}$

— Réseau de préférences : la composante graphique G_P est donnée par la Figure 7.3(b). Les distributions de possibilités initiales sont données par les Tables 7.5 et 7.6.

W	$\Pi_P(W)$	C	$\Pi_P(C)$
$w1$	1.0	$c1$	1.0
$w2$	1.0	$c2$	1.0

TABLE 7.5 – Distribution de possibilité initiale $\Pi_{P_{\min}}$

UM	C	W	$\Pi_P(W R)$	UM	C	W	$\Pi_P(W R)$
$um1$	$c1$	$w1$	1.0	$um2$	$c1$	$w1$	1.0
$um1$	$c1$	$w2$	0	$um1$	$c1$	$w2$	1.0
$um1$	$c2$	$w1$	0.8	$um1$	$c2$	$w1$	1.0
$um1$	$c2$	$w2$	1.0	$um1$	$c2$	$w2$	1.0

TABLE 7.6 – Distribution de possibilité initiale $\Pi_{P_{\min}}$

Le résultat de la fusion de ΠK_{\min} et ΠP_{\min} est le réseau possibiliste $\Pi G_{\oplus} = (G_{\oplus}, p_{\oplus})$ est donné à la Figure 7.17. Les distributions de possibilités initiales sont données par les Tables

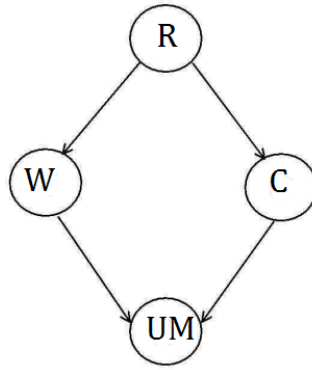


FIGURE 7.17 – Le DAG G_{\oplus}

7.7 et 7.8.

R	$\Pi_{\oplus}(R)$	W	R	$\Pi_K(W R)$	C	R	$\Pi_K(C R)$
$r1$	0.9	$w1$	$r1$	0.4	$c1$	$r1$	1.0
$r2$	1.0	$w1$	$r2$	1.0	$c1$	$r2$	0.2
		$w2$	$r1$	1.0	$c2$	$r1$	0.3
		$w2$	$r2$	0.0	$c2$	$r2$	1.0

TABLE 7.7 – Distribution de possibilité initiale ΠG_{\oplus}

UM	W	C	$\Pi_{\oplus}(UM WC)$	UM	W	C	$\Pi_{\oplus}(UM WC)$
$um1$	$w1$	$c1$	1.0	$um2$	$w1$	$c1$	1.0
$um1$	$w1$	$c2$	0.0	$um2$	$w1$	$c2$	1.0
$um1$	$w2$	$c1$	0.8	$um2$	$w2$	$c1$	1.0
$um1$	$w2$	$c2$	1.0	$um2$	$w2$	$c2$	1.0

TABLE 7.8 – Distribution de possibilité initiale ΠG_{\oplus}

Pour calculer la décision optimiste optimale $UM = \{um1, um2\}$, nous commençons par construire le graphe moral associé au graphe G_{\oplus} représentant la fusion de ΠK_{\min} et ΠP_{\min} . Le graphe moral résultant contient trois cliques $C_1 = \{R, W\}$, $C_3 = \{R, W, UM\}$ et leur séparateur $S_{12} = \{R, W\}$. Pour chaque valeur de décision dans $UM = \{um1, um2\}$, nous allons exécuter l'algorithme afin de calculer le degré de normalisation associé au graphe moral.

Etape 1 : $UM = um1$

$$\Lambda_{UM}(um1) = \begin{cases} 1 & UM \text{ is instanciated as } um1 \\ 0 & UM \text{ is instanciated as } um2. \end{cases} \quad (7.9)$$

La table 7.10 représente la distribution jointe codée par MG après la procédure d'initialisation.

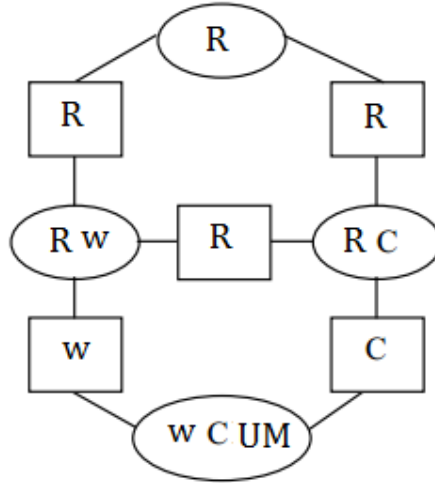


FIGURE 7.18 – Le graphe moral MG

R	W	C	UM	Π_{MG}	R	W	C	UM	Π_{MG}
r1	w1	c1	um1	0.3	r2	w1	c1	um1	0.0
r1	w1	c1	um2	0.3	r2	w1	c2	um2	0.0
r1	w1	c2	um1	0.0	r2	w1	c1	um1	0.0
r1	w1	c2	um2	1.0	r2	w1	c2	um2	0.0
r1	w2	c1	um1	0.3	r2	w2	c1	um1	0.8
r1	w2	c1	um2	0.3	r2	w2	c2	um2	0.9
r1	w2	c2	um1	0.4	r2	w2	c1	um1	0.2
r1	w2	c2	um2	0.4	r2	w2	c2	um2	0.2

TABLE 7.9 – Distribution jointes Π_{MG}

Une fois que le graphe moral est quantifié, la procédure de stabilisation simple nous permet de calculer le degré de normalisation du graphe moral :

$$\max_{\pi_R} = \max_{\pi_{RW}} = \max_{\pi_{RWUM}} = 0.9 \quad (7.10)$$

D'après la Table 7.10, nous pouvons vérifier que : $(h(\Pi_{MG}) \doteq 0.8 \neq 0.9)$, ce qui signifie que le graphe moral n'est pas cohérent. Nous devons donc rétablir le graphe moral en utilisant la procédure de stabilité multiple. En utilisant cette procédure, nous obtenons :

$$\max_{\pi_R} = \max_{\pi_{RW}} = \max_{\pi_{RWUM}} = 0.8 \quad (7.11)$$

le degré de normalisation du graphe moral est : $U^*(um1) = 0.8$

Etape 2 : $UM = um2$ Nous répétons la même procédure décrite dans l'étape précédente, avec :

$$\Lambda_{UM}(um2) = \begin{cases} 1 & UM \text{ is instantiated as } um2 \\ 0 & UM \text{ is instantiated as } um1. \end{cases} \quad (7.12)$$

nous obtenons :

R	W	C	UM	Π_{MG}	R	W	C	UM	Π_{MG}
r1	w1	c1	um1	0.3	r2	w1	c1	um1	0.0
r1	w1	c1	um2	0.0	r2	w1	c2	um2	0.0
r1	w1	c2	um1	0.0	r2	w1	c1	um1	0.0
r1	w1	c2	um2	0.0	r2	w1	c2	um2	0.0
r1	w2	c1	um1	0.3	r2	w2	c1	um1	0.8
r1	w2	c1	um2	0.0	r2	w2	c2	um2	0.0
r1	w2	c2	um1	0.4	r2	w2	c1	um1	0.2
r1	w2	c2	um2	0.0	r2	w2	c2	um2	0.0

TABLE 7.10 – Distributions jointes Π_{MG} après la procédure d'initialisation

$$U^*(um2) = \max_{\pi_R} = \max_{\pi_{RW}} = \max_{\pi_{RWUM}} = 1.0 \quad (7.13)$$

Nous pouvons conclure que la décision optimale est $UM = um2$ avec l'utilité qualitative maximale égale 1.0.

7.5 Conclusion

Dans ce chapitre, nous avons présenté brièvement les concepts de base des théories de la décision qui ont toutes les caractéristiques communes de faire intervenir des relations de préférences sur un ensemble de conséquences et de plausibilité sur un ensemble d'états du monde. Puis, nous avons présenté l'algorithme exact pour le calcul de décision séquentielle en se basant sur la décomposition d'un diagramme d'influence en deux réseaux possibilistes. Nous avons aussi présenté des expérimentations afin de valider l'approche proposée. Enfin, nous avons présenté une approche approximative intéressante pour calculer la décision. Elle est efficace lorsque la génération des distributions de possibilités locales par l'algorithme standard est impossible ou lorsque le temps de réponse est trop long.

Conclusion générale et Perspectives

Dans cette thèse, nous avons étudié des modèles logiques et graphiques pour la gestion d'informations incohérentes et incertaines.

Dans la première partie, nous avons étudié le problème de la gestion de l'incertitude dans les logiques DL-Lite à basse complexité. Nous avons proposé une extension des ontologies légères, encodées dans le langage DL-Lite, au cadre de la théorie des possibilités basée sur le produit. A notre connaissance, il s'agit du premier travail qui utilise la théorie des possibilités quantitative comme moyen de représenter l'incertitude dans le cadre des logiques de description.

Après la présentation de la syntaxe et la sémantique de DL-Lite possibiliste basé sur le produit (Pb- π -DL-Lite), nous avons abordé le problème de calcul du degré d'inconsistance d'une base de connaissances Pb- π -DL-Lite et son rôle crucial dans les tâches d'inférence et de réponse aux requêtes. Nous nous sommes concentrés sur le problème de vérification d'instance. Nous avons montré, en particulier, que le problème de calcul du degré d'inconsistance dans le cadre de Pb- π -DL-Lite est un problème difficile, contrairement au cas de DL-Lite possibiliste basée sur le min, où ce problème reste de complexité polynomiale.

Lorsque l'incertitude est considérée seulement au niveau des assertions de la ABox, nous avons montré que, comme dans le cas de DL-Lite standard, les conflits potentiels restent binaires et assertionnels et ils sont obtenus en appliquant les mêmes règles dites de fermeture négative. Nous avons fourni également des transformations équivalentes entre le problème du calcul du degré d'inconsistance d'une base de connaissances Pb- π -DL-Lite et le problème Max-2-Horn-SAT pondéré, ce qui nous a permis de proposer un algorithme exact, utilisant un solveur W-Max-SAT, pour le calcul du degré d'inconsistance et la réponse à des requêtes de vérification d'instance. De plus, nous avons proposé un codage de notre problème en termes de couverture par ensembles pondérés afin d'utiliser un algorithme glouton polynomial conçu pour ce problème comme moyen de calcul de valeurs approximatives du degré d'inconsistance et de répondre à des requêtes de vérification d'instance.

Nous avons aussi considéré le cadre général où la TBox peut également être incertaine. Contrairement au cas où la TBox est certaine, un conflit dans ce cadre général peut impliquer des axiomes incertains de la TBox. Nous avons alors proposé une version étendue des règles de la fermeture négative qui permettent de garder trace de tous les axiomes de la TBox participant à un conflit. Pour le problème du calcul du degré d'inconsistance, nous avons proposé une modélisation en termes de la programmation linéaire en nombres entiers 0-1. Ceci nous a permis de proposer un autre algorithme exact qui fait appel au solveur CPLEX comme outil pratique pour le calcul du degré d'inconsistance.

Aussi bien pour le cas d'une TBox certaine que pour le cas général, nous avons mené des études expérimentales où les différentes solutions proposées ont été évaluées et/ou comparées

en termes de la qualité des résultats trouvés et d'efficacité (temps de calcul). Nous avons mis en évidence en particulier l'efficacité de l'algorithme approximatif proposé mais aussi l'intérêt de l'approche basée sur la programmation linéaire en nombres entiers 0-1 qui est, en termes d'efficacité, clairement meilleure que l'approche basée sur W-Max-SAT et très proche de celle basée sur l'algorithme glouton approximatif polynomial, tout en ayant l'avantage d'assurer une solution exacte.

Dans la deuxième partie de la thèse, nous avons étudié le problème de la prise de décision qui est l'une des activités les plus couramment pratiquées dans la vie quotidienne. Nous avons implémenté une version améliorée du procédé de décomposition d'un diagramme d'influence possibiliste en deux réseaux possibilistes sans réduction, permettant de réduire la complexité des réseaux résultants et offrant ainsi une amélioration du processus de calcul des décisions optimales optimistes. Afin d'évaluer l'efficacité de l'algorithme du calcul de décision optimiste en utilisant la fusion de deux réseaux possibilistes, nous avons mené une expérimentation dans l'objectif de comparaison des résultats de ce dernier avec un autre modèle basé sur un algorithme de calcul de décision optimiste en utilisant les diagrammes d'influence. De plus, nous avons proposé une approche approximative pour le calcul de la décision possibiliste qualitative en incorporant les techniques de fusion des réseaux possibilistes. Cette approche est efficace et elle est en particulier utile lorsque la génération des distributions de possibilités locales par l'algorithme standard est impossible ou prend un temps de réponse trop long.

Ce travail nous a ouvert plusieurs perspectives pour des travaux futurs. Nous citons en particulier les points suivants :

- Étudier d'autres formes de mesure de degrés d'inconsistance qui sont encore intéressantes sur le plan informatique et dans le contexte des ontologies incertaines. Notre objectif est de définir de nouvelles mesures d'inconsistance qui, sans augmenter la complexité du calcul, permettent de discriminer les bases de connaissances DL-lite ayant le même degré d'inconsistance mais pas les mêmes distributions de possibilités induites.
- Proposer un modèle de gestion de l'incertitude dans les bases de connaissances DL-Lite où l'incertitude est exprimée, de façon plus flexible, dans le cadre de la logique possibiliste à intervalles. Là aussi, toute la démarche théorique allant de la définition de la syntaxe et la sémantique du langage jusqu'aux méthodes de réponses à des requêtes conjonctives doit être adaptée au nouveau cadre.
- Combiner les modèles graphiques possibilistes avec les logiques DL-Lite.
- Intégrer de l'algorithme de calcul de décision optimale pessimiste en se basant sur la fusion de deux réseaux possibilistes.
- Traiter le cas où les préférences et les connaissances proviennent de plusieurs sources.
- Implémenter de la rétro-transformation vers d'autres modèles tels que les diagrammes d'influences possibilistes et les arbres de décision.

Bibliographie

- [1] Emile HL Aarts, Frans MJ de Bont, JHA Habers, and Peter JM van Laarhoven. A parallel statistical cooling algorithm. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 87–97. Springer, 1986.
- [2] Ernest Wilcox Adams. *The logic of conditionals : An application of probability to deductive logic*. Number 86. Springer Science and Business Media, 1975.
- [3] Sergio Alejandro Gomez, Carlos Ivan Chesnevar, and Guillermo Ricardo Simari. Reasoning with inconsistent ontologies through argumentation. *Applied Artificial Intelligence*, 24(1-2) :102–148, 2010.
- [4] Alessandro Antonucci, Cassio Polpo De Campos, and Marco Zaffalon. Probabilistic graphical models. 2014.
- [5] Alessandro Artale, Diego Calvanese, Roman Kontchakov, and Michael Zakharyashev. The DL-Lite family and relations. *Journal of artificial intelligence research*, 36(1) :1–69, 2009.
- [6] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the EL envelope. In *IJCAI*, volume 5, pages 364–369, 2005.
- [7] Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter Patel-Schneider. *The description logic handbook : theory, implementations and applications*. Cambridge U. Press, Cambridge, England, 2003.
- [8] Micheline Belanger and Jean Marc Martel. Explanations for a Decision Support System based on MCDA. *Computing and Informatics*, 25(2-3) :195–221, 2012.
- [9] Nahla Ben Amor, Salem Benferhat, and Khaled Mellouli. Anytime possibilistic propagation algorithm. In *Soft-Ware 2002 : Computing in an Imperfect World*, pages 263–279. Springer, 2002.
- [10] Nahla Ben Amor, Salem Benferhat, and Khaled Mellouli. Anytime propagation algorithm for min-based possibilistic graphs. *Soft Computing*, 8(2) :150–161, 2003.
- [11] Nahla Benamor. *Qualitative possibilistic graphical models from independance to propagation algorithmes*. PhD thesis, Centre de Recherche en Informatique de Lens, 2002.
- [12] Salem Benferhat. *Evolution des croyances et raisonnement plausible-Modèles qualitatifs*. Thèse d’habilitation-dirigée les recherches, Université Paul Sabatier Toulouse, 2000.
- [13] Salem Benferhat and Zied Bouraoui. Min-based possibilistic DL-Lite. *Journal of Logic and Computation*, 27(1) :261–297, 2015.
- [14] Salem Benferhat, Zied Bouraoui, Sylvain Lagrue, and Julien Rossit. Min-based assertional merging approach for prioritized DL-Lite knowledge bases. In *International Conference on Scalable Uncertainty Management*, pages 8–21. Springer, 2014.

- [15] Salem Benferhat, Khaoula Boutouhami, Faiza Khellaf, and Farid Nouioua. Algorithms for Quantitative-Based Possibilistic Lightweight Ontologies. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 364–372. Springer, 2016.
- [16] Salem Benferhat, Khaoula Boutouhami, Faiza Khellaf, and Farid Nouioua. Representing Lightweight Ontologies in a Product-Based Possibility Theory Framework. In *Soft Methods for Data Science*, pages 45–52. Springer, 2017.
- [17] Salem Benferhat, Khaoula Boutouhami, Hadja Faiza Khellaf-Haned, and Ismahane Zeddigha. Qualitative Possibilistic Decisions : Decomposition and Sequential Decisions Making. In *International Conference on Agents and Artificial Intelligence*, pages 171–188. Springer, 2016.
- [18] Salem Benferhat, Faiza Khellaf, and Ismahane Zeddigha. A possibilistic graphical model for handling decision problems under uncertainty. In *8th conference of the European Society for Fuzzy Logic and Technology (EUSFLAT-13)*. Atlantis Press, 2013.
- [19] Salem Benferhat, Faiza Khellaf-Haned, and Ismahane Zeddigha. Computing optimal optimistic decisions using min-based possibilistic networks. In *Fuzzy Information Processing Society (NAFIPS), 2012 Annual Meeting of the North American*, pages 1–6, 2012.
- [20] Salem Benferhat, Hadja Faiza Khellaf-Haned, and Ismahane Zeddigha. On the Decomposition of Min-based Possibilistic Influence Diagrams. In *Proceedings of the 8th International Conference on Agents and Artificial Intelligence , Volume 2, Rome, Italy*, pages 117–128, 2016.
- [21] Leopoldo Bertossi. Database repairing and consistent query answering. *Synthesis Lectures on Data Management*, 3(5) :1–121, 2011.
- [22] Paul-Antoine Bisgambiglia, Kevin Guelton, Jimmy Lauber, and Jean-Luc Marichal. La logique floue et ses applications, 2016.
- [23] Fernando Bobillo, Miguel Delgado, and Juan Gomez-Romero. A crisp representation for fuzzycal SHOIN with fuzzy nominals and general concept inclusions. In *Uncertainty Reasoning for the Semantic Web I*, pages 174–188. Springer, 2008.
- [24] Fernando Bobillo, Miguel Delgado, and Juan Gomez-Romero. Crisp representations and reasoning for fuzzy ontologies. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 17(04) :501–530, 2009.
- [25] Fernando Bobillo, Miguel Delgado, Juan Gómez-Romero, and Umberto Straccia. Fuzzy description logics under Gödel semantics. *International Journal of Approximate Reasoning*, 50(3) :494–514, 2009.
- [26] Fernando Bobillo and Umberto Straccia. Towards a crisp representation of fuzzy description logics under Lukasiewicz semantics. In *Lecture Notes in Computer Science*, volume 4994, pages 309–318. Springer, 2008.
- [27] Christian Borgelt, Jorg Gebhardt, and Rudolf Kruse. Possibilistic graphical models. In *Computational Intelligence in Data Mining*, pages 51–67. Springer, 2000.
- [28] Khaoula Boutouhami and Abdallah Arioua. Une approche graphique pour le calcul de la décision possibiliste qualitative, 2013.
- [29] Khaoula Boutouhami, Salem Benferhat, Faiza Khellaf, and Farid Nouioua. Uncertain lightweight ontologies in a product-based possibility theory framework. volume 88, pages 237–258. Elsevier, 2017.

- [30] Khaoula Boutouhami and Faiza Khellaf. An approximate possibilistic graphical model for computing optimistic qualitative decision. In *International Conference on Artificial Intelligence and Applications*, volume 5, pages 183–1960, 2015.
- [31] Khaoula Boutouhami and Faiza Khellaf. Optimistic decision making using an approximate graphical model. *International Journal of Artificial Intelligence and Applications*, 6(2) :1, 2015.
- [32] Jay Buckley and Thomas J Dudley. How Gerber Used a Decision Tree in Strategic Decision-Making. *Graziadio Business Review*, Volume 2 Issue 3, 1999.
- [33] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, Mariano Rodriguez-Muro, and Riccardo Rosati. Ontologies and databases : The DL-Lite approach. In *Reasoning Web. Semantic Technologies for Information Systems*, pages 255–356. Springer, 2009.
- [34] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. DL-Lite : Tractable description logics for ontologies. In *AAAI*, volume 5, pages 602–607, 2005.
- [35] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable reasoning and efficient query answering in description logics : The DL-Lite family. *Journal of Automated Reasoning*, 39(3) :385–429, 2007.
- [36] Laurence Cholvy. Applying theory of evidence in multisensor data fusion : a logical interpretation. In *Information Fusion, 2000. FUSION 2000. Proceedings of the Third International Conference on*, volume 1, pages TUB4–17. IEEE, 2000.
- [37] Gregory F Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial intelligence*, 42(2-3) :393–405, 1990.
- [38] Gregory F Cooper. A method for using belief networks as influence diagrams. *arXiv preprint arXiv :1304.2346*, 2013.
- [39] Arthur P Dempster. Upper and lower probabilities induced by a multivalued mapping. *The annals of mathematical statistics*, pages 325–339, 1967.
- [40] Arthur P Dempster. A generalization of Bayesian inference. *Classic works of the dempster-shafer theory of belief functions*, 219 :73–104, 2008.
- [41] Didier Dubois, Daniel Le Berre, Henri Prade, and Régis Sabbadin. Using possibilistic logic for modeling qualitative decision : ATMS-based algorithms. *Fundamenta Informaticae*, 37(1, 2) :1–30, 1999.
- [42] Didier Dubois, Lang Jérôme, and Henri Prade. Towards Possibilistic Logic Programming. In *ICLP*, pages 581–595, 1991.
- [43] Didier Dubois, Jérôme Lang, and Henri Prade. Possibilistic logic. 1994.
- [44] Didier Dubois, Jérôme Mengin, and Henri Prade. Possibilistic uncertainty and fuzzy features in description logic. A preliminary discussion. *Capturing Intelligence*, 1 :101–113, 2006.
- [45] Didier Dubois and Henri Prade. Necessity measures and the resolution principle. *IEEE transactions on systems, man, and cybernetics*, 17(3) :474–478, 1987.
- [46] Didier Dubois and Henri Prade. Possibilistic logic, preferential models, non-monotonicity and related issues. In *Proc. of IJCAI*, volume 91, pages 419–424, 1991.
- [47] Didier Dubois and Henri Prade. Possibilistic logic in decision. In *Fuzzy Logic and Soft Computing*, pages 3–17. Springer, 1999.

- [48] Didier Dubois and Henri Prade. Possibilistic logic : a retrospective and prospective view. *Fuzzy sets and Systems*, 144(1) :3–23, 2004.
- [49] Didier Dubois and Henri Prade. Possibility theory and its applications : a retrospective and prospective view. In *Decision Theory and Multi-Agent Planning*, pages 89–109. Springer, 2006.
- [50] Didier Dubois and Henri Prade. *Possibility theory : an approach to computerized processing of uncertainty*. Springer Science and Business Media, 2012.
- [51] Didier Dubois, Henri Prade, and Sandra Sandri. On possibility/probability transformations. *Fuzzy logic*, pages 103–112, 1993.
- [52] Michael Durig and Thomas Studer. Probabilistic ABox Reasoning : Preliminary Results. In *Description Logics*, pages 104–111, 2005.
- [53] Hélène Fargier and Didier Dubois. Décision sous incertitude. In *3èmes Journées Doctorales / Journées Nationales MACS*, 2009.
- [54] Laurent Garcia and Régis Sabbadin. Diagrammes d’influence possibilistes. *Revue d’intelligence artificielle*, 21(4) :521–554, 2007.
- [55] Michael R Garey, David S. Johnson, and Larry Stockmeyer. Some simplified NP-complete graph problems. *Theoretical computer science*, 1(3) :237–267, 1976.
- [56] Rosalba Giugno and Thomas Lukasiewicz. P-SHOQ (D) : A probabilistic extension of SHOQ (D) for probabilistic ontologies in the semantic web. In *JELIA*, volume 2, pages 86–97. Springer, 2002.
- [57] Olivier Gossner. Théorie des jeux et théorie de la décision. Un cours de théorie des jeux, chapitre1 : théorie de la décision, 2003.
- [58] Thomas R Gruber. A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2) :199–220, 1993.
- [59] Wided Guezguez. *Possibilistic Decision Theory : From Theoretical Foundations to Influence Diagrams Methodology*. PhD thesis, Université Paul Sabatier, Toulouse, France, 2012.
- [60] Wided Guezguez, Nahla Ben Amor, and Khaled Mellouli. Qualitative possibilistic influence diagrams based on qualitative possibilistic utilities. *European Journal of Operational Research*, 195(1) :223–238, 2009.
- [61] Peter Haase, Frank Van Harmelen, Zhisheng Huang, Heiner Stuckenschmidt, and York Sure. A framework for handling inconsistency in changing ontologies. In *International semantic web conference*, volume 3729, pages 353–367. Springer, 2005.
- [62] Michael Haber and Morton B Brown. Maximum likelihood methods for log-linear models when expected frequencies are subject to linear constraints. *Journal of the American Statistical Association*, 81(394) :477–482, 1986.
- [63] Jochen Heinsohn. Probabilistic description logics. In *Proceedings of the Tenth international conference on Uncertainty in artificial intelligence*, pages 311–318. Morgan Kaufmann Publishers Inc., 1994.
- [64] Ellen Hisdal. Conditional possibilities independence and noninteraction. *Fuzzy Sets and Systems*, 1(4) :283–297, 1978.
- [65] Pascal Hitzler, Markus Krotzsch, and Sebastian Rudolph. *Foundations of semantic web technologies*. CRC press, 2009.

- [66] Bernhard Hollunder. An alternative proof method for possibilistic logic and its application to terminological logics. In *Proceedings of the Tenth international conference on Uncertainty in artificial intelligence*, pages 327–335. Morgan Kaufmann Publishers Inc., 1994.
- [67] Raiffa Howard. *Decision Analysis : Introductory lectures on choices under uncertainty*. 1968.
- [68] Ronald A Howard and James E Matheson. Influence diagrams. *Decision Analysis*, 2(3) :127–143, 2005.
- [69] Sajjad Hussain, Jos De Roo, Ali Daniyal, and Syed Sibte Raza Abidi. Detecting and resolving inconsistencies in ontologies using contradiction derivations. In *Computer Software and Applications Conference (COMPSAC), 2011 IEEE 35th Annual*, pages 556–561. IEEE, 2011.
- [70] ILOG. Cplex 10.0. "<http://www.ilog.com/products/cplex/>".
- [71] Manfred Jaeger et al. Probabilistic Reasoning in Terminological Logics. *KR*, 94 :305–316, 1994.
- [72] Qiu Ji, Peter Haase, Guilin Qi, Pascal Hitzler, and Steffen Stadtmüller. RaDON—repair and diagnosis in ontology networks. *The semantic web : research and applications*, pages 863–867, 2009.
- [73] Souhila Kaci. *Connaissances et Préférences : représentation et fusion en logique possibiliste*. PhD thesis, Université Paul Sabatier, Toulouse, France, 2002.
- [74] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.
- [75] Daniel Kayser. La représentation des connaissances. *Le Courrier du CNRS*, (80) :63–64, 1993.
- [76] Hadja Faiza Khellaf-Haned. *Autour des différents modes de représentation et de raisonnement en théorie des possibilités quantitatave*. PhD thesis, Université des sciences et de la technologie USTHB et Université d’Artois, 2007.
- [77] Jurg Kohlas and Paul-André Monney. *A mathematical theory of hints : An approach to the Dempster-Shafer theory of evidence*, volume 425. Springer Science and Business Media, 2013.
- [78] Amit Konar and Ajit K. Mandal. Uncertainty management in expert systems using fuzzy Petri nets. *IEEE Transactions on Knowledge and Data Engineering*, 8(1) :96–105, 1996.
- [79] Henry E Kyburg. Probabilistic reasoning in intelligent systems : networks of plausible inference by Judea Pearl. *The Journal of Philosophy*, 88(8) :434–437, 1991.
- [80] Philippe Laublet, Chantal Reynaud, and Jean Charlet. Sur quelques aspects du Web sémantique. *Assises du GDR I*, 3 :59–78, 2002.
- [81] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. Inconsistency-tolerant semantics for description logics. In *International Conference on Web Reasoning and Rule Systems*, pages 103–117. Springer, 2010.
- [82] Deyi Li and Yi Du. *Artificial intelligence with uncertainty*. CRC press, 2007.
- [83] Baoding Liu. Uncertainty theory. In *Uncertainty Theory*, pages 205–234. Springer, 2007.

- [84] Thomas Lukasiewicz. Probabilistic deduction with conditional constraints over basic events. *J. Artif. Intell. Res.(JAIR)*, 10 :199–241, 1999.
- [85] Thomas Lukasiewicz. Probabilistic logic programming under inheritance with overriding. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 329–336. Morgan Kaufmann Publishers Inc., 2001.
- [86] Thomas Lukasiewicz. Probabilistic default reasoning with conditional constraints. *Annals of Mathematics and Artificial Intelligence*, 34(1-3) :35–88, 2002.
- [87] Thomas Lukasiewicz. Expressive probabilistic description logics. *Artificial Intelligence*, 172(6) :852–883, 2008.
- [88] Thomas Lukasiewicz, Maria Vanina Martinez, Giorgio Orsi, and Gerardo I Simari. Heuristic Ranking in Tightly Coupled Probabilistic Description Logics. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence, Catalina Island, CA, USA, August 14-18, 2012*, pages 554–563, 2012.
- [89] Thomas Lukasiewicz and Umberto Straccia. Uncertainty and vagueness in description logic programs for the semantic web. Technical report, Citeseer, 2007.
- [90] Richard E Neapolitan. Probabilistic reasoning in expert systems : theory and algorithms. CreateSpace Independent Publishing Platform, 2012.
- [91] Thomas Dyhre Nielsen and Finn Verner Jensen. *Bayesian networks and decision graphs*. Springer Science and Business Media, 2009.
- [92] Nils J Nilsson. Probabilistic logic. *Artificial intelligence*, 28(1) :71–87, 1986.
- [93] Andreas Nolle, Christian Meilicke, Heiner Stuckenschmidt, and German Nemirovski. Efficient federated debugging of lightweight ontologies. In *International Conference on Web Reasoning and Rule Systems*, pages 206–215. Springer, 2014.
- [94] Odile Papini. Cours : Logiques de description. "odile.papini.perso.luminy.univ-amu.fr/sources/WEBSEM/cours-WEBSEM-3.pdf".
- [95] Odile Papini. Cours : Ontologies. "odile.papini.perso.luminy.univ-amu.fr/sources/WEBSEM/cours-WEBSEM-2.pdf".
- [96] Judea Pearl. Causality : models, reasoning and inference. *Econometric Theory*, 19(675-685) :46, 2003.
- [97] David Poole. Average-case analysis of a search algorithm for estimating prior and posterior probabilities in Bayesian networks with extreme probabilities. In *IJCAI*, pages 606–612, 1993.
- [98] Guilin Qi, Qiu Ji, Jeff Z Pan, and Jianfeng Du. PossDL—a possibilistic DL reasoner for uncertainty reasoning and inconsistency handling. In *Extended Semantic Web Conference*, pages 416–420. Springer, 2010.
- [99] Guilin Qi, Jeff Z Pan, and Qiu Ji. Extending description logics with uncertainty reasoning in possibilistic logic. In *ECSQARU*, volume 7, pages 828–839. Springer, 2007.
- [100] Riccardo Rosati, Marco Ruzzi, Mirko Graziosi, and Giulia Masotti. Evaluation of techniques for inconsistency handling in OWL 2 QL ontologies. In *International Semantic Web Conference*, pages 337–349. Springer, 2012.
- [101] Régis Sabbadin. A possibilistic model for qualitative sequential decision problems under uncertainty in partially observable environments. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 567–574. Morgan Kaufmann Publishers Inc., 1999.

- [102] Thomas Scharrenbach, Rolf Grutter, Bettina Waldvogel, and Abraham Bernstein. Structure preserving TBox repair using defaults. In *23rd International Workshop on Description Logics DL2010*, page 384, 2010.
- [103] Glenn Shafer et al. *A mathematical theory of evidence*, volume 1. Princeton university press Princeton, 1976.
- [104] Prakash P Shenoy. Valuation based systems : A framework for managing uncertainty in expert systems. In *Fuzzy Logic for the Management of Uncertainty*, pages 83–104. L. A. Zadeh and J. Kacprzyk, Eds. John Wiley and Sons, New York, NY, 1992.
- [105] Prakash P Shenoy. A comparison of graphical techniques for decision analysis. *European Journal of Operational Research*, 78(1) :1–21, 1994.
- [106] Philippe Smets. The combination of evidence in the transferable belief model. *IEEE Transactions on pattern analysis and machine intelligence*, 12(5) :447–458, 1990.
- [107] Philippe Smets. Belief functions : the disjunctive rule of combination and the generalized Bayesian theorem. *International Journal of approximate reasoning*, 9(1) :1–35, 1993.
- [108] Philippe Smets et al. What is Dempster-Shafer’s model. *Advances in the Dempster-Shafer theory of evidence*, pages 5–34, 1994.
- [109] Wolfgang Spohn. Ordinal conditional functions : A dynamic theory of epistemic states. In *Causation in decision, belief change, and statistics*, pages 105–134. Springer, 1988.
- [110] Wolfgang Spohn. *The laws of belief : Ranking theory and its philosophical applications*. Oxford University Press, 2012.
- [111] Giorgos Stoilos, Giorgos Stamou, and Jeff Z Pan. Handling imprecise knowledge with fuzzy description logic. In *Proc. 2006 Internat. Workshop on Description Logics (DL 2006)*, pages 119–126, 2006.
- [112] Giorgos Stoilos and Giorgos B Stamou. Extending Fuzzy Description Logics for the Semantic Web. In *OWLED*, volume 258, 2007.
- [113] Giorgos Stoilos, Giorgos B Stamou, Jeff Z Pan, Vassilis Tzouvaras, and Ian Horrocks. Reasoning with very expressive fuzzy description logics. *J. Artif. Intell. Res.(JAIR)*, 30 :273–320, 2007.
- [114] Giorgos Stoilos, Giorgos B Stamou, Vassilis Tzouvaras, Jeff Z Pan, and Ian Horrocks. Fuzzy OWL : Uncertainty and the Semantic Web. In *OWLED*, 2005.
- [115] Umberto Straccia. A fuzzy description logic. In *AAAI/IAAI*, pages 594–599, 1998.
- [116] Umberto Straccia. Reasoning within fuzzy description logics. *J. Artif. Intell. Res.(JAIR)*, 14 :137–166, 2001.
- [117] Umberto Straccia. Transforming fuzzy description logics into classical description logics. In *JELIA*, pages 385–399. Springer, 2004.
- [118] Umberto Straccia. A fuzzy description logic for the semantic web. *Capturing Intelligence*, 1 :73–90, 2006.
- [119] Faiza Titouna. *Fusion de réseaux causaux possibilistes*. PhD thesis, Artois, 2009.
- [120] Larry Wasserman. Statistical Reasoning With Imprecise Probabilities. *Journal of the American Statistical Association*, 88(422) :700–703, 1993.
- [121] Ronald R Yager. On the specificity of a possibility distribution. *Fuzzy Sets and Systems*, 50(3) :279–292, 1992.

- [122] John Yen et al. Generalizing Term Subsumption Languages to Fuzzy Logic. In *IJCAI*, volume 91, pages 472–477, 1991.
- [123] Neal E Young. Greedy set-cover algorithms. In *Encyclopedia of algorithms*, pages 1–99. Springer, 2008.
- [124] Lotfi Zadeh. Fuzzy sets. *Information and control*, 8(3) :338–353, 1965.
- [125] Lotfi Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy sets and systems*, 100 :9–34, 1999.
- [126] Marco Zaffalon and Gert De Cooman. Imprecise probability perspectives on artificial intelligence. *Annals of mathematics and artificial intelligence*, 45(1) :1–4, 2005.
- [127] Ismahane Zeddigha. *Autour de la décision qualitative*. PhD thesis, Université des sciences et de la technologie USTHB et Université d’Artois, 2015.
- [128] Ismahane Zeddigha, Salem Benferhat, and Faiza Khellaf. New Graphical Model for Computing Optimistic Decisions in Possibility Theory Framework. *Computing and Informatics*, 34(5) :1038–1064, 2016.