

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
**Université Mohamed Khider de BISKRA**  
Faculté des sciences exactes, des sciences de la nature et de la vie



Département d'Informatique

*Thèse de Doctorat*

En Informatique

Option

Intelligence artificielle et systèmes distribués

Thème

---

**Raisonnement sur les ontologies légères**

---

Présentée par

**Abdelmoutia TELLI**

Devant le jury composé de :

Président	DJEDI Nouredine	Professeur U. Biskra, Algérie
Rapporteur	BOURAHLA Mustapha	Professeur U. M'Sila, Algérie
Rapporteur	BENFERHAT Salem	Professeur U. Artois, France
Examineur	KHELLAF Faiza Haned	Professeur U. USTHB, Algérie
Examineur	TABIA Karim	MCF-HDR U. Artois, France
Examineur	CROITORU Madalina	MCF-HDR U. Montpellier, France

**Résumé :**

*DL-Lite* est l'une des logiques de description légère les plus importantes. Cette thèse étudie différentes approches pour traiter des bases de connaissances *DL-Lite* dans le cas où la base d'assertions est prioritaire et inconsistante avec la base terminologique. Le problème d'inconsistance se produit souvent lorsque les assertions sont fournies par multiples sources conflictuelles ayant des différents niveaux de fiabilité. Nous examinons d'abord comment raisonner à partir d'une base de connaissances standard *DL-Lite*, avec un ABox multiple, qui peut être issu de plusieurs sources d'informations. Nous proposons ensuite différentes stratégies d'inférence basées sur la sélection d'une base d'assertions consistante, appelée réparation préférée. Pour chaque stratégie, un algorithme polynomial pour calculer la réparation préférée unique associée est proposé. La sélection d'une réparation unique est importante car elle permet une gestion efficace des requêtes. Nous fournissons des études expérimentales montrant (du point de vue du calcul) les avantages de sélectionner une réparation lors du raisonnement sous les bases de connaissances légères inconsistantes.

**Mots clés :** Inconsistance, Bases de connaissances prioritaires, Les ontologies légères.

**Abstract :**

*DL-Lite* is one of the most important lightweight description logics. This thesis investigates different approaches for handling inconsistent *DL-Lite* knowledge bases in the case where the assertional base is prioritized and inconsistent with the terminological base. The inconsistency problem often happens when the assertions are provided by multiple conflicting sources having different reliability levels. We first give how to reason from a flat *DL-Lite* knowledge base, with a multiple ABox, which can be either issued from multiple information sources. We then propose different inference strategies based on the selection of one consistent assertional base, called a preferred repair. For each strategy, a polynomial algorithm for computing the associated single preferred repair is proposed. Selecting a unique repair is important since it allows an efficient handling of queries. We provide experimental studies showing (from a computational point of view) the benefits of selecting one repair when reasoning under inconsistency in lightweight knowledge bases.

**Keywords :** Inconsistency, Prioritized knowledge bases, Lightweight Ontologies.

## Dédicaces

*À mes parents, mes frères et mes sœurs.*

*À ma femme : Nour.*

*À mes enfants : Moutia & Danial.*

# Remerciements

*Soyons reconnaissants aux personnes qui nous  
donnent du bonheur ; elles sont les charmants  
jardiniers par qui nos âmes sont fleuries.  
Marcel Proust (1871-1922).*

**J**E tiens à remercier mon Dieu.

Mes très vifs remerciements vont à :

L'encontre de mes directeurs de thèse, Monsieur **Mustapha Bourahla**, professeur à l'université Mohamed BOUDIAF de M'Sila, Monsieur **Salem Benferhat** professeur à l'université d'Artois pour avoir acceptés de m'encadrer, de m'orienter, de m'avoir conseillés, de m'encourager et de me soutenir tout au long de la thèse avec patience et disponibilité, et pour la confiance qu'ils m'ont accordée.

Je tiens à remercier :

Monsieur **Djedi Nour eddine** qui m'a fait l'honneur de présider le jury de ma soutenance.

Je remercie :

Mes deux rapporteurs de thèse ; Mes dame **Faiza Haned Khellaf** et **Madalina Croitoru** pour l'intérêt qu'ils ont porté à ce travail en acceptant de le juger et pour le temps qu'ils ont consacré pour la lecture de cette thèse.

Je tiens à remercier également :

Monsieur **Karime Tabia** de m'avoir honoré en acceptant d'examiner ma thèse.

Je souhaite également remercier :

Tous les membres du CRIL (Centre de Recherche en Informatique de Lens), doctorants et permanents, de m'avoir accepté au sein du centre et soutenu mes travaux de recherche.

Un grand merci s'adresse à :

Ma femme pour sa patience et son soutien indéfectible qui m'ont été plus qu'indispensables.

# Table des matières

## Table des matières

Liste des figures	iii
Liste des tableaux	iv
Introduction générale	1
<b>1 Ontologies et logique de description</b>	<b>7</b>
1.1 Introduction . . . . .	7
1.2 Les Ontologies . . . . .	8
1.2.1 Définitions . . . . .	8
1.2.2 Structure d'une ontologie . . . . .	9
1.2.3 Caractéristiques . . . . .	10
1.2.4 Classification . . . . .	11
1.2.5 Langages de représentation . . . . .	11
1.3 Formalismes de l'ontologie . . . . .	15
1.4 Logique de description . . . . .	18
1.4.1 Représentation des connaissances dans les LDs . . . . .	18
1.4.2 Syntaxe de la logique de description . . . . .	19
1.4.3 Sémantique de $\mathcal{ALC}$ . . . . .	20
1.4.4 Raisonnement sur $\mathcal{ALC}$ . . . . .	21
1.4.4.1 Inférence au niveau Assertionnel . . . . .	21
1.4.4.2 Inférence au niveau terminologique . . . . .	22
1.4.5 Extensions de $\mathcal{ALC}$ . . . . .	22
1.4.6 Moteurs d'inférence . . . . .	23
1.4.7 Complexité dans Les LDs . . . . .	24
1.4.8 La famille $DL-Lite$ . . . . .	26
1.4.8.1 Syntaxe et sémantique . . . . .	26
1.4.8.2 Extension de la famille $DL-Lite$ . . . . .	29
1.5 Conclusion . . . . .	32

<b>2</b>	<b>Débogage d'ontologie : Traitement sur les bases de connaissances</b>	
	<i>DL-Lite</i>	<b>33</b>
2.1	Introduction . . . . .	33
2.2	Approche OBDA . . . . .	34
2.2.1	Avantages de l'approche OBDA . . . . .	35
2.2.2	Description logique pour OBDA avec bases de données . . . . .	36
2.3	Caractéristiques d'une base de connaissances <i>DL-Lite</i> . . . . .	37
2.3.1	Incohérence et inconsistance . . . . .	37
2.3.2	Réductibilité de FOL . . . . .	37
2.3.3	Fermeture négative . . . . .	38
2.3.4	Calcul des conflits . . . . .	39
2.3.5	Traitement d'inconsistance sur <i>DL-Lite</i> . . . . .	40
2.4	Bases de connaissances <i>DL-Lite</i> MBox . . . . .	41
2.4.1	Fermeture déductive sur MBox . . . . .	42
2.4.2	Stratégies d'inférence sur MBox . . . . .	42
2.4.2.1	Inférence universelle . . . . .	42
2.4.2.2	Inférence existentielle . . . . .	43
2.4.2.3	Inférence Saine . . . . .	43
2.4.2.4	Inférence basée sur la majorité . . . . .	44
2.4.2.5	Inférence basée sur la proportionnalité . . . . .	45
2.4.2.6	Inférence sans objection . . . . .	45
2.4.2.7	Comparaison entre les stratégies d'inférence sur MBox . . . . .	46
2.5	Autres Travaux . . . . .	48
2.6	Conclusion . . . . .	49
<b>3</b>	<b>Gestion d'inconsistance dans <i>DL-Lite</i> avec priorités</b>	<b>50</b>
3.1	Introduction . . . . .	50
3.2	Base de connaissances <i>DL-Lite</i> avec priorité . . . . .	51
3.2.1	Approches de tolérance aux inconsistances dans <i>DL-Lite</i> avec priorité . . . . .	52
3.2.2	Le concept de réparations . . . . .	52
3.2.3	L'ensemble de Conflit et l'ensemble libre . . . . .	52
3.2.4	Notion de clôture prioritaire . . . . .	54
3.2.5	Analyse de complexité . . . . .	55
3.3	Stratégies de calcul des réparations dans <i>DL-Lite</i> avec priorité . . . . .	59
3.3.1	Réparation à base de la théorie des possibilités . . . . .	59
3.3.2	Réparation à base linéaire . . . . .	61
3.3.3	Réparation à base d'assertions libres prioritaires . . . . .	62
3.3.4	Réparation à base de clôture déductive d'assertions libres prioritaires . . . . .	64
3.3.5	Réparation à base linéaire d'assertions libres prioritaires . . . . .	66
3.4	Conclusion . . . . .	67

<b>4</b>	<b>Résultats expérimentaux et études comparatives</b>	<b>68</b>
4.1	Introduction . . . . .	68
4.2	Environnement de développement . . . . .	69
4.2.1	Environnement software . . . . .	69
4.2.2	Environnement hardware . . . . .	69
4.3	Paramètres Expérimentaux . . . . .	70
4.3.1	Bases de tests utilisées . . . . .	70
4.3.2	Métriques d'évaluation . . . . .	71
4.3.3	Tests et Résultats . . . . .	71
4.4	Étude comparatives . . . . .	75
4.5	Application de réparation d'assertions libres prioritaires à traitement vidéo	84
4.5.1	Contraintes de description logique pour les vidéos . . . . .	84
4.5.2	Construire d'une ABox et TBox pour l'ontologie danse . . . . .	85
4.6	Conclusion . . . . .	88
<b>5</b>	<b>AniAge : Projet d'ontologie de danses traditionnelles</b>	<b>89</b>
5.1	Introduction . . . . .	89
5.2	AniAge : Projet d'ontologie pour représenter les danses traditionnelles .	90
5.3	Approche proposée pour développer une ontologie de danse Vietnamiennne	91
5.3.1	Rassembler et analyser des connaissances d'experts . . . . .	92
5.3.2	Analyser des mouvements dans les danses vietnamiennes . . . . .	94
5.3.3	Analyse sémantique d'un mouvement de la danse Mõ . . . . .	96
5.3.3.1	Outils d'implémentation . . . . .	96
5.3.3.2	Ontologie de danse . . . . .	98
5.3.3.3	Concepts et propriétés . . . . .	102
5.3.3.4	Relations et interprétations . . . . .	103
5.4	Conclusion . . . . .	104
	<b>Conclusion Générale et Perspectives</b>	<b>105</b>
	<b>Annexe A</b>	<b>107</b>
	<b>Bibliographie</b>	<b>113</b>

# LISTE DES FIGURES

1.1	Les familles du langage OWL . . . . .	14
1.2	Vue générale de système OBDA . . . . .	16
1.3	Structure générale de la logique de description . . . . .	18
2.1	Architecture de système OBDA . . . . .	35
2.2	Comparaison entre les stratégies d'inférence sur MBox. . . . .	48
3.1	Transformation d'une base <i>DL-Lite</i> vers un graphe non orienté. . . . .	55
3.2	Transformation d'un graphe non orienté vers une base <i>DL-Lite</i> . . . . .	56
4.1	Comparaison entre $\pi(\mathcal{A})$ , $\ell(\mathcal{A})$ , $nd(\mathcal{A})$ , $clnd(\mathcal{A})$ , et $lnd(\mathcal{A})$ . . . . .	83
5.1	Construction d'une ontologie pour danse folklorique vietnamienne . . . . .	91
5.2	Groupes ethniques de danses folkloriques Vietnamiennes . . . . .	93
5.3	Les orientations des parties du corps. . . . .	94
5.4	Interface graphique de protégé 2000. . . . .	97
5.5	Ontologie de domaine de danse traditionnelle. . . . .	98
5.6	Position initiale de danse Mõ. . . . .	99
5.7	Position 1 de la jambe. . . . .	99
5.8	Position 2 de la jambe. . . . .	99
5.9	Position 3 de la jambe. . . . .	100
5.10	Position 4 de la jambe. . . . .	100
5.11	Position initiale des mains. . . . .	101
5.12	Position 01 des mains. . . . .	101
5.13	Position 02 des mains. . . . .	101
5.14	Classes et sous classes des mouvements de base pour la danse Mõ. . . . .	102
2	Interface principale . . . . .	109
3	Détection d'incohérence de TBox . . . . .	111
4	Détection d'inconsistance dans les ABoxs . . . . .	112
5	Le temps et le taux de réponse . . . . .	112

# LISTE DES TABLEAUX

1.1	Dictionnaire de concepts pour l'ontologie d'université. . . . .	17
1.2	Des relations binaires pour l'ontologie d'université. . . . .	17
1.3	Extensions de $\mathcal{ALC}$ . . . . .	23
1.4	Complexité de calcul pour la vérification de la consistance en fonction de l'expressivité dans certains LD. . . . .	25
1.5	Les axiomes d'inclusion des concepts et des assertions dans $DL-Lite_\alpha$ . . . . .	29
1.6	Extension de la famille $DL-Lite$ . . . . .	30
2.1	Équivalence entre les axiomes $DL-Lite$ et les formules FOL. . . . .	38
3.1	Notations de réparations utilisées. . . . .	51
4.1	Information d'ontologie LUBM. . . . .	70
4.2	Pourcentage de faits supprimés et le temps (en secondes) pris pour calculer les conflits, $\pi$ , $\ell$ , $nd$ , $\ell nd$ , et $\ell nd$ en variant le nombre de conflits et le nombre de strates. . . . .	72
4.3	Impact du nombre de strates sur la productivité et sur le temps de calcul. . . . .	73
4.4	Impact du nombre des éléments de conflits sur la productivité et sur le temps de calcul des réparations. . . . .	74
4.5	Les Concepts d'ontologie danse. . . . .	85
4.6	Les relations d'ontologie danse. . . . .	85
4.7	La base terminologique d'ontologie danse. . . . .	86
5.1	Relations entre les classe de la danse $M\ddot{o}$ . . . . .	103

# INTRODUCTION GÉNÉRALE

*L'ordinateur parfait a été inventé :  
on entre un problème et il n'en ressort jamais !  
Al Goodman*

## Contexte et motivations

**D**E nos jours, il existe une utilisation croissante des ontologies dans de nombreux domaines d'application. Ces ontologies répondent aux problèmes de représentation et de manipulation des connaissances. Elles ont été très utilisées dans le domaine du Web sémantique. Par ailleurs, les logiques de description (LDs en abrégé), sont reconnues comme des cadres formels puissants pour la représentation et le raisonnement sur les ontologies. Elles ont été largement utilisées au cours des dernières années en tant que langages formels pour les ontologies, grâce à leurs capacités puissantes de modélisation et du raisonnement [1] [2].

Les LDs sont particulièrement adaptées à la gestion de grandes quantités de données qui est une utilisation difficile dans l'intégration des données basée sur les ontologies. Ce problème a récemment reçu une attention croissante dans la communauté du Web sémantique [3]. En effet, l'intégration des données dans le contexte du Web sémantique signifie principalement l'accès, la collecte et l'échange de données réparties sur le web à travers des schémas donnés en termes d'ontologies. Pour ces raisons, il peut souvent arriver que les données contredisent les connaissances intentionnelles fournies par l'ontologie, en particulier dans les cas où l'ontologie fournit une vue conceptuelle d'un certain nombre de sources de données autonomes, hétérogènes et largement distribuées.

Les logiques de description ont été introduites pour représenter et raisonner sur des connaissances structurées [2]. Une base de connaissances de la LDs se compose de deux composantes distinctes : une base terminologique (appelée TBox), qui représente des connaissances structurelles génériques sur un domaine d'application, et une base assertionnelle (appelée ABox), qui contient les faits assertionnels (les individus ou les instances) représentant des instanciations de la connaissance générique. Dans le contexte du Web sémantique, les LD fournissent la base logique du langage OWL (Web Ontology Language), normalisé par le W3C.

Au cours des dernières années, une famille particulière de la LD, appelée *Les logiques de descriptions légères* de type *DL-Lite* est apparue [4] [3]. *DL-Lite* est une famille traitable de LD spécialement dédiée aux applications qui utilisent de gros volumes de données. Elle fournit un cadre puissant de représentation flexible des connaissances assortie d'une complexité de calcul faible des processus de raisonnement.

*DL-Lite* tient une place importante dans des domaines d'application variés tels que l'accès aux données basées sur des ontologies et le Web sémantique. Elle offre le fondement logique du langage OWL2-QL dédié spécifiquement pour capturer l'ontologie basique et les langages conceptuels de modélisation des données, tout en conservant la faible complexité du raisonnement.

Dans cette famille de LDs, toutes les tâches de raisonnement (en particulier, répondre à des requêtes complexes comme les requêtes conjonctives) sont des calculs polynomiaux par rapport à la taille de la base des assertions [3]. D'autre part, la complexité du raisonnement est calculée en LogSpace par rapport la taille des données [5].

Le contexte OBDA (Ontology-Based Data Access) étudie comment interroger un ensemble de sources de données en utilisant une vue générique (ontologique) unifiée. Dans ce contexte, la base terminologique agit comme un schéma utilisé pour reformuler les requêtes afin d'offrir un meilleur accès à l'ensemble des données stockées dans plusieurs bases assertionnelles [6]. Le problème important qui se pose dans le paramètre OBDA est de savoir comment gérer efficacement les multiples sources de données. Ce fait rend *DL-Lite* particulièrement adapté au contexte OBDA, dans lequel les assertions sont souvent fournies par plusieurs sources potentiellement conflictuelles ayant des niveaux de fiabilité différents. Le regroupement de ces ensembles d'assertions donne une base d'assertion prioritaire ou stratifiée.

La gestion de l'inconsistance est un problème largement étudié dans le cadre de la logique propositionnelle [7]. D'ailleurs, les paramètres de *DL-Lite* et OBDA, définissent l'inconsistance en fonction de certaines assertions qui se contredisent avec la base de la terminologie.

Plusieurs approches ont été proposées pour gérer le problème de l'inconsistance dans les bases de connaissances de la logique propositionnelle. Ces approches sont basées sur la restauration de la consistance [8] [9], en utilisant des logiques [10], ou en analysant et en mesurant l'inconsistance [11], soit en employant un cadre d'argumentation [12]. Dans le même esprit, plusieurs travaux [13], [17], [14] et [15] ont été proposés pour gérer l'inconsistance et l'incohérence dans les ontologies (débogage et réparation de l'ontologie).

En ce qui concerne le *DL-Lite* et dans le contexte OBDA, les travaux existants (comme [16], [18] et [19]), sont essentiellement inspirés par les approches proposées dans le domaine de base de données. Ils traitent l'inconsistance dans *DL-Lite* en proposant une adaptation de plusieurs méthodes d'inférence tolérantes aux inconsistances. Toutes ces approches proposées sont basées sur la notion de réparation qui est étroitement liée à la notion de réparation de base de données afin de répondre aux requêtes posées aux des bases de données inconsistantes.

Une réparation d'une base de données en contradiction avec des contraintes d'intégrité est une base de données obtenue en appliquant un ensemble de modifications afin de restaurer la consistance. Cette notion a été étendue au langage *DL-Lite* [16]. Une réparation d'une base des connaissances *DL-Lite* est une sous-base assertionnelle maximale consistante avec une TBox. Ces travaux sont liés aux travaux de restauration de la consistance, proposés pour gérer l'inconsistance dans les bases de connaissances de la logique propositionnelle.

Dans cette thèse, nous étudions la réparation de l'inconsistance dans les bases de connaissances *DL-Lite*. Nous détaillons en particulier la gestion de l'inconsistance dans le cas où la base de connaissances *DL-Lite* est prioritaire.

## Problématique

Dans une même ontologie, on peut avoir plusieurs sources d'informations différentes. En fait, dans de nombreux contextes, les informations et les connaissances disponibles peuvent être incertaines ou prioritaires nécessitant un cadre pour gérer l'inconsistance et les priorités. L'objectif principal de notre travail concerne le problème de la gestion de l'inconsistance dans *DL-Lite*. L'idée générale est de développer des algorithmes polynomiaux afin de calculer la réparation des ontologies dont leurs bases de connaissances sont représentées et distribuées sur différentes sources d'informations.

## Contributions

Notre contribution consiste à étudier comment raisonner à partir d'une base de connaissances *DL-Lite* avec des ABox multiples prioritaire, appelées base de connaissances ABox *DL-Lite* prioritaire.

Une base de connaissances prioritaire est un multi-ensemble d'ABoxes qui peut être délivré à partir des sources d'informations multiples prioritaire. Nous utilisons d'abord le concept de ABox pour fournir une feuille de route pour la gestion des bases de connaissances standard *DL-Lite* inconsistantes. Ensuite, nous analysons des stratégies d'inférences sur ABox *DL-Lite* prioritaire. Nous proposons par la suite, des stratégies de sélection d'une réparation préférée à partir des bases de connaissances *DL-Lite* lorsque les assertions dans les ABoxes sont prioritaires.

En premier lieu, nous examinons les approches existantes pour sélectionner les réparations préférées. Ensuite, nous nous concentrons sur des stratégies appropriées pour gérer l'inconsistance dans *DL-Lite*. Nous proposons, en particulier, de nouvelles approches de sélection d'une réparation basée sur la notion d'assertions libres prioritaires (*Non-defeated*) suivie par des critères supplémentaires comme la fermeture déductive et la linéarité. Enfin, nous fournissons une analyse comparative sur les études expérimentales des différentes approches proposées .

## Organisation de la thèse

Notre thèse est constituée, en plus de l'introduction et la conclusion, de cinq chapitres. Les deux premiers chapitres sont des chapitres d'état de l'art, alors que les trois autres chapitres sont des contributions sur la gestion de l'inconsistance dans *DL-Lite* avec priorités. Le dernier chapitre est une étude de cas sur un domaine particulier. La thèse est donc organisée comme suit :

Nous présentons dans le chapitre 1 un rappel sur les ontologies et sur les logiques des descriptions, dans lequel, nous détaillons les principales notions sur les ontologies : ontologies pour la représentation des connaissances, les composants d'une ontologie et les langages de la représentations des ontologies. Nous focalisons sur la famille *DL-Lite* et ses extensions.

Le chapitre 2 est consacré au traitements des bases de connaissances *DL-Lite*. Il présente les caractéristiques d'une base de connaissances *DL-Lite*. Nous analysons dans ce chapitre les stratégies d'inférence sur les bases de connaissances *DL-Lite*.

Le troisième chapitre, décrit en détail les stratégies de calcul des réparations dans *DL-Lite* avec priorité. Ce chapitre propose des algorithmes polynomiaux pour calculer les réparations dans *DL-Lite*.

Dans le chapitre 4, l'évaluation et l'expérimentation des algorithmes proposés sont présentées. Cette expérimentation est menée en exploitant des bases de test Benchmark. Elle est effectuée par l'intermédiaire d'un ensemble de métriques d'évaluation, qui permettent d'apprécier la qualité des réparations obtenues. Le chapitre présente aussi une étude comparative entre les différentes stratégies de calcul des réparations.

Une description globale du projet AniAge (High Dimensional Heterogeneous Data based Animation Techniques for Southeast Asian Intangible Cultural Heritage Digital Content) a été présentée dans le chapitre 5,.

Enfin, cette thèse se termine par une conclusion générale suivie de quelques perspectives et une vue sur des travaux futurs. Nous proposons également une annexe supplémentaire qui décrit notre outil développé pour la représentation et le raisonnement dans le cadre *DL-Lite* avec priorité.

## Publications atteint dans cette thèse

1. Telli Abdelmoutia, Mustapha Bourahla, Salem Benferhat , Zied Bouraoui, and Karim Tabia : *Polynomial Algorithms for Computing a Single Preferred Assertional Based Repair*. *Künstliche Intelligenz*, 31(1) : 15-30, 2017.
2. Salem Benferhat, Zied Bouraoui, Huma Chaudhry, Karim Tabia, Mohd Shafry Bin Mohd Rahim Fc and Telli Abdelmoutia *Characterizing Non-Defeated Repairs in Inconsistent Lightweight Ontologies*. Proceedings of the 11th International Conference on Signal-Image Technology and Internet-Based Systems (SITIS 2016).
3. Ce travail a bénéficié du soutien du projet européen H2020 *Marie Skłodowska-Curie Actions* (MSCA), et *Research and Innovation Staff Exchange* (RISE) : AniAge (*High Dimensional Heterogeneous Data based Animation Techniques for Southeast Asian Intangible Cultural Heritage Digital Content*), numéro de projet 691215.

# Chapitre 1

## Ontologies et logique de description

*"Au fond d'un trou ou d'un puits,  
il arrive qu'on aperçoive les étoiles"  
Aristote*

### 1.1 Introduction

DANS de nombreuses applications, la gestion des connaissances complexes est l'un des défis les plus importants. Les informations doivent être manipulées fidèlement, tout en évitant la confusion, l'incohérence, et les contradictions entre les éléments représentant le domaine étudié. Les recherches dans le domaine de l'Intelligence Artificielle (IA), en particulier, l'ingénierie des connaissances sont concentrées sur les formalismes de représentation des connaissances [20]. Une attention particulière a été accordée à l'utilisation des représentations sémantiques des connaissances, susceptibles d'être manipulées par les machines. Cette sémantique peut être exprimée à l'aide des ontologies. La logique de description, est une famille de langages d'ontologies, basée principalement sur des logiques de premier ordre, qui permet un encodage et un raisonnement efficace sur les ontologies.

Ce chapitre présente un état de l'art sur les ontologies. Il résume les principaux concepts des logiques de description. Enfin, il se concentre sur la famille *DL-Lite* qui est dédié aux applications qui utilisent des bases de données très volumineuses.

## 1.2 Les Ontologies

### 1.2.1 Définitions

Le mot Ontologie est employé dans des contextes différents tels que la linguistique, la philosophie et l'Intelligence Artificielle :

**En Linguistique :** Le mot tiré du grec ; *Ontos* est le participe présent du verbe être et *Logos* qui veut dire discours. C'est l'étude des propriétés générales de ce qui existe.

**En Philosophique :** Le mot *ontologie* est un terme philosophique introduit au *XIX<sup>eme</sup>* siècle pour caractériser l'étude des êtres dans notre univers.

**Aristote** a décrit l'ontologie comme *la science de l'être en tant qu'être*. Selon cette théorie, l'ontologie est la science de l'être ou l'étude des êtres en tant qu'ils existent. Plus précisément, l'ontologie se préoccupe de déterminer quelles catégories de l'être sont fondamentales [21].

**En Informatique :** Depuis les années quatre vingt dix du dernier siècle l'ontologie est devenue l'un des domaines les plus importants de la recherche scientifique dans le domaine des sciences de l'information. Il est devenu un terme plus spécifique et plus détaillé.

En Intelligence Artificielle, le terme ontologie désigne une organisation (formalisation) des connaissances d'un domaine. Sur cette idée, plusieurs définitions ont été proposées :

**Gruber** [22] a donné la définition la plus référencée dans le domaine d'Intelligence Artificielle ; *An ontology is a formal, explicit specification of a shared conceptualization of a domain of interest ; une ontologie est une spécification explicite et formelle d'une conceptualisation d'un domaine de connaissances.*

Autre définition est donnée par **Neches** [23] ; *An ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary.*

**SOWA** [25] définit l'ontologie comme la méthode d'extraction d'un catalogue des entités existant dans un domaine selon la perception d'une personne qui décrit ce catalogue par un langage.

## 1.2.2 Structure d'une ontologie

Une ontologie se compose d'un ensemble de concepts, d'axiomes, et de relations qui décrivent un domaine d'intérêt [24]. Les composants primaires d'une ontologie qui est considérée en tant qu'objet informatique sont :

**Les concepts :** Un concept selon [26] est la description abstraite d'un ou plusieurs objets similaires. Il peut être divisé en trois parties :

- *Un terme (ou plusieurs)* : est un élément lexical qui permet d'exprimer le concept en langue naturelle, il peut admettre des synonymes.
- *La notion* : également appelée intention du concept, contient la sémantique du concept, exprimée en termes de propriétés, d'attributs, et de contraintes.
- *L'objet* : appelé extension du concept, regroupe les objets manipulés à travers le concept ; ces objets sont appelés instances du concept.

Un concept peut avoir les propriétés suivantes [27] :

- *La généralité* : s'il n'admet pas d'extension (par exemple : la vérité est un concept générique).
- *L'identité* : s'il permet de conclure l'identité de deux instances de ce concept.
- *La rigidité* : Un concept est rigide si toute instance de concept reste instance dans tous les mondes possibles.
- *Un concept est anti-rigide* si toute instance de ce concept est essentiellement définie par son appartenance à l'extension d'un autre concept.
- *Deux concepts sont équivalents* s'ils ont la même extension.
- *Deux concepts sont disjoints* si leurs extensions sont disjointes.
- *Deux concepts sont dépendants* si pour toute instance de l'un des deux concepts, il existe une instance de l'autre qui ne soit ni partie ni constituant de l'instance du premier concept.

**Les Relations :** Les concepts (respectivement, les instances) peuvent être reliés entre eux par des relations au sein d'une ontologie. Elles représentent les interactions entre les concepts dans une ontologie [28].

Les relations peuvent être spécifiées par des propriétés [29] :

- *Une relation est algébrique* : elle peut avoir les propriétés de la symétrie, la réflexivité et la transitivité.
- *Une cardinale de relation* : est le nombre possible de relations de ce type entre les mêmes concepts (ou instances de concept).
- *Relations incompatibles* : Ne peuvent lier les mêmes instances de concepts.
- *Deux relations sont inverses l'une de l'autre* : si quand l'une lie deux instances  $I_1$  et  $I_2$ , l'autre lie  $I_2$  et  $I_1$ .

- *Deux relations sont dites exclusives* : si quand l'une lie des instances de concepts, l'autre ne lie pas ces instances, et vice-versa. L'exclusivité entraîne l'incompatibilité.
- Un cas particulier s'appelle **fonctions** de relations dans lesquelles le  $N^{ieme}$  élément de la relation est défini de manière unique à partir des  $n-1$  premiers.

**Propriété** : Appelée aussi attribut ou propriété de données, elle est utilisée pour qualifier un élément de l'ontologie (un concept ou une relation). Une propriété peut être atomique (par exemple, un entier) ou structurée (par exemple, une liste).

**Axiome** : Appelé aussi assertion, c'est une formule logique utilisée pour exprimer des contraintes ou des règles qui doivent être satisfaites par les éléments spécifiés au niveau extensionnel.

**Instance** : Appelée aussi fait, représente un individu ou un objet qui appartient à un concept.

### 1.2.3 Caractéristiques

Une ontologie est une bibliothèque de concepts bien définie. Cette bibliothèque décrit la structure de l'information pour un domaine particulier. Les ontologies possèdent les caractéristiques fondamentales suivantes [30] :

**La clarté et l'objectivité** : elles doivent fournir le sens des termes définis en offrant des définitions objectives en langage naturel.

**La formalisation** : Les concepts et les relations sont exprimés dans une syntaxe claire et cohérente sur une base mathématique qui peut être traitée par des programmes informatiques.

**La lisibilité** : Les concepts et les relations doivent être lisibles par les programmeurs, les communautés d'experts de domaine et aussi les utilisateurs potentiels.

**L'exhaustivité** : Toutes les définitions des concepts ou des relations doivent être exprimées par des conditions nécessaires et suffisantes.

**Les ontologies sont partageables** : Ils sont utilisables à travers de multiples domaines d'application pour faciliter la combinaison entre les ontologies développées.

## 1.2.4 Classification

Plusieurs dimensions ont été proposées pour la classification des ontologies [31]. Nous sommes intéressés par la classification basée sur les objets modélisés qui sont proposés par [32] et [33]. Cette classification comprend les catégories suivantes :

**Les ontologies de haut niveau (Top-level ontologies) :** sont des ontologies générales qui décrivent les concepts très généraux et le plus haut niveau d'abstraction dans le monde ; tel que : l'espace, le temps, la matière, les événements, les actions, etc. Elles sont indépendantes d'un problème ou d'un domaine d'application particulier [34].

**Les ontologies de domaine (Domaine ontologies) :** décrivent le vocabulaire lié à un domaine d'application. Elles sont considérées comme un méta description d'une représentation des connaissances. Noter que, la plupart des ontologies existantes sont des ontologies du domaine [35].

**Les ontologies de tâche (Task ontologies) :** toutes les tâches des résolutions de problèmes (diagnostic, planification,...) dans un système ont été décrites par des ontologies de tâche. Elle régit un ensemble de vocabulaires et de concepts qui décrit une structure de résolution des problèmes. L'ontologie de tâche caractérise l'architecture computationnelle d'un système à base de connaissances qui réalise une tâche [36].

**Les ontologies d'application (Application ontologies) :** Ces ontologies sont les plus spécifiques des ontologies de domaine et des ontologies de tâche. Elles décrivent des concepts correspondant à l'exécution d'une tâche particulière dans leur domaine d'application [37].

## 1.2.5 Langages de représentation

Plusieurs langages de représentation d'ontologie ont été développés afin de mettre en œuvre des raisonnements sur une ontologie. Parmi ces langages développés au niveau conceptuel pour la modélisation d'ontologie, trois grands modèles sont distingués :

- Les langages **basés sur les graphes** tel que : les réseaux sémantiques et les graphes conceptuels. D'ailleurs, les diagrammes de classe UML (Unified Modeling Language) utilisé comme un langage pour modéliser les ontologies [38] [39].

- Les langages **basés sur les frames** comme : OKBC<sup>1</sup> (Open Knowledge Base Connectivity) et KM<sup>2</sup> (Knowledge Machine).
- Les langages **basés sur la logique** tel que KIF (Knowledge Interchange Format)<sup>3</sup>, CycL<sup>4</sup> qui sont fondés sur la logique de premier ordre, et sur la logique de description [40].

La logique de description est l'une des logiques d'ontologies les plus importantes recommandées par W3C<sup>5</sup> (World Wide Web Consortium) pour le Web sémantique. Elle permet aux machines de chercher, de combiner et de traiter intelligemment les contenus du web en fonction de leurs significations [41].

Nous nous intéressons aux ontologies décrites en langage de représentation des connaissances OWL (Web Ontology Language). Ce langage est utilisé pour obtenir une représentation sémantiquement riche d'un domaine d'intérêt, puis utiliser ce dernier pour effectuer des tâches de raisonnement. Les langages d'ontologie largement utilisés et basés sur la logique de description sont :

**eXtensible Markup Language (XML)** : [42] La syntaxe de XML issu de SGML (Standard Generalized Markup Language) et défini par le consortium web. Le langage XML est un langage de balisage ; il va permettre de structurer les informations contenues dans un document grâce à leurs balises. Contrairement à HTML, XML offre la possibilité de créer ses propres balises. Effectivement le but de XML n'est pas le développement d'ontologie (il ne permet pas d'imposer des contraintes sémantiques) mais il est considéré comme un langage de description et d'échange de documents structurés. En d'autres termes, XML traite la structure syntaxique des documents sans aucune spécification sémantique de cette structuration. Le langage RDF (Resource Description Framework) est un langage qui a été défini pour cette raison.

**RDF (Resource Description Framework)** : RDF<sup>6</sup> est un standard utilisé pour l'échange des ressources sur le web. Le W3C a adopté le langage RDF comme une recommandation en 1999. Les ressources dans le RDF sont identifiées par les URIs (Universal Resources Identifiers). Un document RDF est à la base exprimé sous la forme d'un graphe orienté, mais on peut le décrire par la syntaxe XML. Il fournit une sémantique simple pour des données sans faire des

---

1. <http://www.ai.sri.com/okbc/>
2. <http://www.cs.utexas.edu/users/mfkb/RKF/km.html>
3. <http://www.ksl.stanford.edu/knowledge-sharing/kif/>
4. <http://www.cyc.com/documentation/syntax-cycl>
5. <https://www.w3.org/>
6. <http://www.w3.org/RDF/>

contraintes sur la structure du document [43]. Cependant, Il existe certaines limites pour RDF ; par exemple, nous avons besoin de connaître des informations sur les ressources identifiées. C'est pour cela que le W3C a associé à RDF le standard RDF Schéma (RDFS).

**RDFs (Resource Description Framework Schema :)** RDFS<sup>7</sup> Construit par le W3C en 2000 comme une extension sémantique et fournit un vocabulaire de modélisation RDF. RDFS permet de définir des mécanismes d'héritage pour la description des groupes de ressources similaires (classes) et des relations entre ces ressources (propriétés). Comme son nom l'indique, le RDFS est un langage permettant de définir des propriétés sémantiques pour les ressources par un schéma. Dans un schéma on peut définir de nouvelles ressources comme des spécialisations d'autres ressources. Les schémas contraignent aussi le contexte d'utilisation des ressources.

Avec RDFS, des nouvelles notions sémantiques apparaissent. Cependant, les primitives offertes par RDFS sont insuffisantes pour la modélisation dans le Web sémantique ; il est impossible de raisonner sur ces représentations car la sémantique (hors la subsomption) reste très limitée.

**DAML+OIL : DARPA Agent Markup Language + Ontology Interchange Language ;**<sup>8</sup> Jim Hendler a annoncé le 11 janvier 2001 la première version du langage DAML appelé DAML-ONT. Les langages DAML et OIL se regroupent et forment le langage DAML+OIL qui remplace donc DAML-ONT.

Il est possible de faire des inférences compatibles avec les logiques de description [44]. Il utilise la syntaxe des standards XML, RDF et RDFS.

DAML+OIL est considéré comme le premier langage à fournir des mécanismes d'inférence sur les concepts d'ontologies. Cependant, Les limites suivantes sont présentées par DAML+OIL ; il ne permet pas la composition et la fermeture transitive. Il prend en compte uniquement des relations unaires et binaires. Il ne possède pas des déclarations de variables.

**OWL (Ontology Web Language) :** OWL<sup>9</sup> est une extension du langage RDF et RDFS, inspiré de DAML+OIL. Il définit un vocabulaire riche pour la description d'ontologies complexes. Le World Wide Web Consortium (W3C) a recommandé le langage OWL qui est conçu pour la représentation des ontologies pour le Web sémantique basé sur les logiques de description.

---

7. <http://www.w3.org/TR/rdf-schema/>

8. <http://www.daml.org/about.html>

9. <http://www.w3.org/TR/owl-ref/> : Web Ontology Language Reference Version 1.0. W3C Working Draft Mars 2003.

Le langage d'ontologies OWL se décompose en trois sous-langages : OWL Lite, OWL DL et OWL Full d'après la complexité croissante et les différentes utilisations : [45]

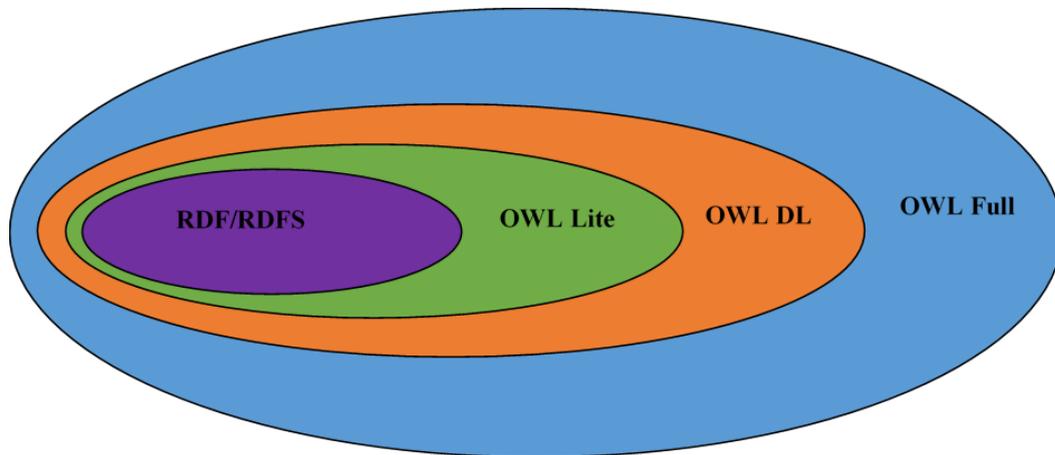


FIGURE 1.1 – Les familles du langage OWL

**OWL-Lite (Ontology Web Language Lite)** : Il est recommandé pour la représentation des thésaurus, des taxonomies ou hiérarchie de classes avec des contraintes simples. Il reprend tous les constructeurs de RDF/XML.

**OWL-DL (Ontology Web Language-Description Logic)** correspond à la variante de la logique de description *SHOIN*. Il est approprié pour la représentation des ontologies ayant besoin de la puissance d'expressivité tout en gardant la calculabilité [46].

**OWL-Full** : est conçu pour les développeurs, les implémenteurs et les utilisateurs qui ont besoin de l'expressivité maximale, de la liberté syntaxique de RDF mais sans se soucier d'une garantie de calculabilité.

Les trois sous-langages d'OWL présentent une hiérarchie sur la validité des ontologies suivantes : les ontologies OWL-Lite sont également des ontologies OWL-DL. Et les ontologies OWL-DL sont également des ontologies OWL-Full. (Voir Figure 1.1)

## 1.3 Formalismes de l'ontologie

Au cours des dernières années, les ontologies sont largement utilisées dans divers domaines d'application où de nouveaux défis ont émergé. Ces défis consistent principalement à installer des ontologies avec de nouvelles capacités de raisonnement (par exemple, évolution, fusion, gestion de l'inconsistance, etc.) ou avec une expressivité supplémentaire (par exemple, gestion de l'incertitude) afin de traiter des nouvelles exigences. Actuellement, de nombreux axes de recherche sur les formalismes de l'ontologie :

1. **L'alignement d'ontologies** (*Ontology matching* [47], [48]) : L'alignement de deux ontologies hétérogènes consiste à produire une ontologie unifiée associée à des liens qui explicitent les différentes correspondances entre les vocabulaires utilisés dans les deux ontologies hétérogènes.
2. **Translation d'ontologies** (*Ontology translation* [49], [50], [51]) : Elle consiste à traduire (les axiomes et les vocabulaires) d'une ontologie exprimée à l'aide d'un langage  $\mathcal{L}_1$ , vers une ontologie équivalente utilisant un autre langage de représentation  $\mathcal{L}_2$ .
3. **Intégration des ontologies** (*Ontology integration* [52], [53]) : L'intégration consiste à combiner un ensemble d'ontologies qui représente des connaissances sur un domaine similaire afin d'obtenir plus de connaissances en unifiant le domaine.
4. **Modularité des ontologies** (*Ontology modularity* [54], [55]) : Ce formalisme consiste à extraire les plus petits sous-ensembles indépendants d'une ontologie, appelés *modules*, dans le but de les réutiliser plus tard dans d'autres applications.
5. **Evolution des ontologies** (*Ontology evolution* [56], [57]) : Elle consiste à modifier une ontologie en fonction d'un ensemble d'opérations de changement qui peuvent concerner les connaissances sur le domaine ou la structure de l'ontologie.
6. **Fusion de l'ontologie** (*Ontology merging* [58], [59], [60]) : La fusion des ontologies consiste à produire à partir de deux ontologies représentant des connaissances sur le même domaine, une ontologie unique qui représente un point de vue global.
7. **Gestion de l'incertitude dans des ontologies** [61], [62], [63] : Elle consiste à traiter l'imprécision ou l'incertitude qui peut être attachée aux éléments du vocabulaire (Par exemple, des concepts reflétant des termes imprécis) ou des axiomes (par exemple, l'incertitude attachée à un axiome).
8. **Systèmes d'ontologie peer-to-peer** [64] : Elle consiste à offrir un accès aux informations détenues par un ensemble de pairs liés les uns aux autres.

9. **Débogage d'ontologies** Cette formalise comprend le diagnostic ontologique qui consiste à restaurer la cohérence du niveau intentionnel d'une ontologie [65], [66], et la réparation de l'ontologie qui consiste à restaurer la consistance de l'ontologie [67], [18].
10. **Accès aux données basées sur les ontologies** (*Ontology-based Data Access* [68], [69]) : OBDA est une nouvelle génération de systèmes d'information. Cela consiste à uniformiser l'accès à des ensembles indépendants de sources de données en utilisant une vue ontologique unifiée à laquelle les sources de données sont liées (Voir Figure 1.2).

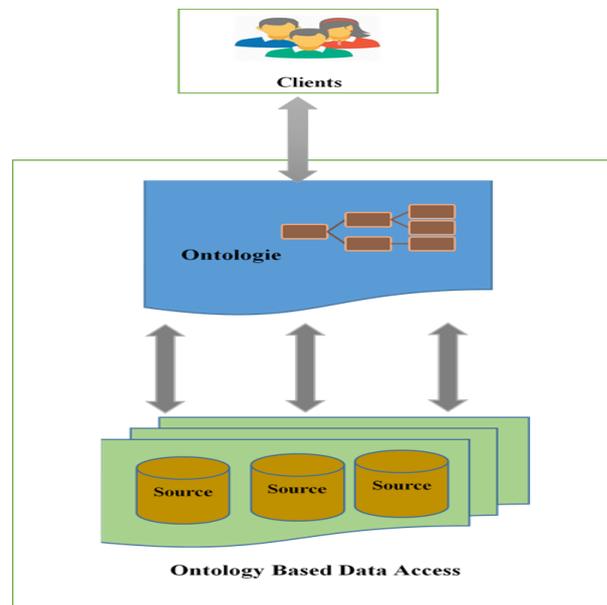


FIGURE 1.2 – Vue générale de système OBDA

Dans le reste de notre thèse, nous nous concentrerons sur la formalise de débogage d'ontologie en particulier la gestion des inconsistances dans le cadre OBDA. Ces tâches sont étroitement liées aux travaux existants effectués sur l'incertitude et la gestion des inconsistance dans le cadre de la logique propositionnelle. (Voir 2.2)

**Exemple 1.** Une ontologie appelé *Université (University Ontology)* est une ontologie de domaine de l'éducation qui définit les éléments pour décrire les universités et les activités qui s'y déroulent. Les concepts de cette ontologie sont illustrés dans le Tableau 1.1.

Concept	Description (Terme)	Attributs	Instances
Teacher	Personne qui donne des cours aux étudiants	Nom Prénom Grade	John Mary Joe
Student	Personne qui apprend des cours par des enseignants	Nom Prénom Numéro	Moutia Bill Anne
Administrator	Enseignant chargé des services administratifs	Nom Prénom Poste	Jone Joe
Course	Sujet enseigné aux étudiants par l'enseignant du département	Titre Code Crédit	Web 3.0 Math
Department	Chacune des spécialités d'étude possède son département (Locale physique d'enseignement)	Nom Adresse	Chemistry Physical

TABLE 1.1 – Dictionnaire de concepts pour l'ontologie d'université.

Pour des raisons de compatibilité, les relations atomiques entre les concepts utilisés dans cette ontologie sont définis dans le Tableau 1.2.

Nom de la relation	Description	Concept source	Concept cible
TeacherTo	Personne qui donne des cours aux étudiants	Teacher(1,n)	Student(1,n)
TeacherOf	Département où Enseignant donne les cours	Teacher(1,n)	Department(1,n)
Takes Course	chaque cours est appris par les étudiants	Student(1,n)	Course(1,n)
Given Course	Chaque Enseignant donne des cours	Teacher(1,n)	Course(1,n)

TABLE 1.2 – Des relations binaires pour l'ontologie d'université.

Une grande variété des contraintes utiles sont supportées par les langages d'ontologie. Par exemple :

- Des disjonctions entre les concepts : Student et Teacher (l'étudiant ne peut pas appartenir à l'enseignant).
- Des inclusions entre les concepts : Administrator et Teacher (un administrateur est un enseignant).
- Chaque Département possède un Administrateur.

Cette ontologie manque de taxonomie détaillée. Ils n'incluent pas tous les concepts (et les rôles) et informations nécessaires pour faciliter la tâche de gestion d'une université. Mais elle démontre les concepts et les relations de base de ce domaine. De plus, quelques propriétés des données sont mises en évidence et élaborées en détail.

Cet exemple, sera utilisé pour expliquer toutes les définitions et tous les algorithmes que nous nous présenterons. ■

Nous présentons dans la prochaine partie de ce chapitre la logique de description qui forme la base théorique des langages d'ontologie comme OWL.

## 1.4 Logique de description

Les logiques de descriptions (LDs) ont été introduites par Brachman [40]. C'est une famille de formalismes conçus pour représenter la connaissance d'un domaine particulier, puis la raison en dérivant de nouvelles connaissances [70].

Le but de la logique de description dans l'Intelligence Artificielle est de développer des langages pour représenter formellement les connaissances d'un domaine et de rendre cette représentation disponible pour le raisonnement [71].

Comme son nom l'indique, les LDs ont une sémantique formelle; la description de concepts utilisée pour décrire un domaine, et la sémantique basée sur la logique qui peut être donnée par une transcription en logique des prédicats ou en logique du premier ordre.

### 1.4.1 Représentation des connaissances dans les LDs

La logique de description utilise uniquement des prédicats unaires et binaires, appelés : concepts et rôles. Les concepts représentent l'ensemble des individus, et les rôles représentent les relations binaires entre les individus (ou entre les concepts). Les concepts et les rôles peuvent être définis ou primitifs. Un concept (ou un rôle) est défini s'il est structuré à partir des constructeurs. Un concept (ou un rôle) est primitif s'il est comparable à des atomes et serve de base à la construction des concepts (ou des rôles) définis. Une LD est composée de deux niveaux (Voir Figure 1.3) :

- *Un niveau terminologique TBox* : dédié à la description de concepts et de rôles.
- *Un niveau assertionnel ABox* : décrit la description de faits [72].

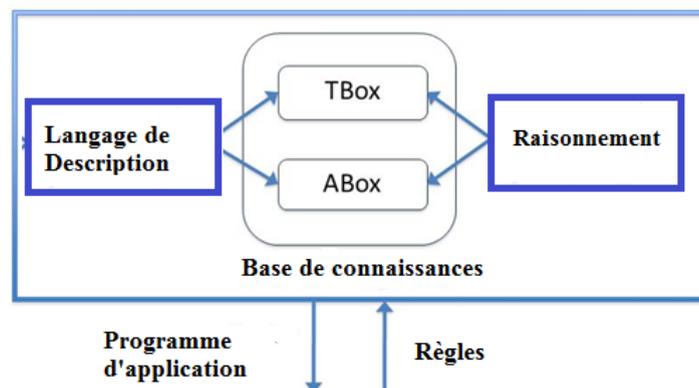


FIGURE 1.3 – Structure générale de la logique de description

Une base de connaissances LD, dénotée par  $\mathcal{K}=\langle\mathcal{T}, \mathcal{A}\rangle$ , tel que  $\mathcal{T}$  représente le niveau terminologique TBox (les connaissances génériques sur le domaine), et  $\mathcal{A}$  représente le niveau extensional ABox, qui contient les faits assertifs (les individus).

## 1.4.2 Syntaxe de la logique de description

Un vocabulaire  $\mathcal{V}$  de la logique de description est un triple  $(N_C, N_R, N_I)$  tel que :  $N_C$  l'ensemble des concepts atomiques,  $N_R$  l'ensemble des rôles atomiques, et  $N_I$  l'ensemble des individus.

Un langage de description  $\mathcal{L}$  est caractérisé par un ensemble de constructions utilisées pour former des concepts et des rôles complexes à partir de structures atomiques ( $N_C$  et  $N_R$ ). Il est utilisé pour structurer un domaine d'intérêt à travers un ensemble de formules logiques appelées les constructeurs. De nombreux constructeurs permettent de former toute une famille de logique de description, suivant les constructeurs retenus.

Nous nous intéressons par l'une des logique de description les plus élémentaires dénotée par  $\mathcal{ALC}$  (*Attributive Language with Complement*).  $\mathcal{ALC}$  est une extension de la logique  $\mathcal{AL}$  qui constitue le cadre de base de LD [70]. La famille  $\mathcal{ALC}$ , contient les constructeurs de conjonction suivants :

- $C, D \rightarrow \top$  Le plus général.
- |  $\perp$  Le plus spécifique.
- |  $P$  Concept atomique.
- |  $C \cap D$  Conjonction de concepts.
- |  $C \cup D$  Disjonction de concepts.
- |  $\neg C$  Négation.
- |  $\forall R.C$  Restriction universelle.
- |  $\exists R.C$  Restriction existentielle.

Dans une base de connaissances  $\mathcal{ALC}$ . Les axiomes contenus dans la TBox sont des axiomes terminologiques de la forme :

- $C \sqsubseteq D$  :  $C$  est un sous-concept de  $D$ .
- $C \equiv D$  :  $C$  est équivalent à  $D$  ( $C \sqsubseteq D$  et  $D \sqsubseteq C$ ).

D'autre part, Les assertions contenues dans la ABox sont des assertions de la forme :

- $C(a)$  :  $a$  est une instance de  $C$ .
- $R(a, b)$  :  $a$  et  $b$  sont en relation par  $R$ . ( $\{C, D\} \in N_C$ .  $R \in N_R$ .  $a$  et  $b \in N_I$ ).

### 1.4.3 Sémantique de $\mathcal{ALC}$

La sémantique de  $\mathcal{ALC}$  est inspirée de la sémantique de logique de premier ordre. Elle est donnée en termes d'interprétations. Une interprétation  $\mathcal{I}=(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$  consiste d'un domaine non vide, noté  $\Delta^{\mathcal{I}}$ , et une fonction d'interprétation, notée  $\cdot^{\mathcal{I}}$  définie de  $N_I$  vers  $\Delta^{\mathcal{I}}$ . La fonction  $\cdot^{\mathcal{I}}$  associe à chaque individu  $a$  un élément  $a^{\mathcal{I}}$  de  $\Delta^{\mathcal{I}}$ , à chaque concept atomique  $A$  un sous-ensemble  $A^{\mathcal{I}}$  de  $\Delta^{\mathcal{I}}$ , et à chaque rôle atomique  $R$  une relation binaire  $R^{\mathcal{I}}$  sur  $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ .

Par ailleurs, la fonction d'interprétation  $\cdot^{\mathcal{I}}$  est étendue de façon simple pour les concepts et les rôles de  $\mathcal{ALC}$  comme suit :

$$\begin{aligned}
 A^{\mathcal{I}} &\subseteq \Delta^{\mathcal{I}} \\
 R^{\mathcal{I}} &\subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \\
 (\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
 (C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
 (C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
 (\exists R.C)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}} \text{ tel que } (x, y) \in R^{\mathcal{I}} \text{ et } y \in C^{\mathcal{I}}\} \\
 (\forall R.C)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} \mid \text{si } (x, y) \in R^{\mathcal{I}} \text{ alors } y \in C^{\mathcal{I}}\}
 \end{aligned}$$

Une interprétation  $\mathcal{I}$  est dite satisfaire une base de connaissances  $\mathcal{K}=\langle \mathcal{T}, \mathcal{A} \rangle$  si et seulement si  $\mathcal{I}$  satisfait tous les axiomes de  $\mathcal{T}$  et tous les axiomes de  $\mathcal{A}$ . Cette interprétation est dite un modèle de  $\mathcal{K}$  [73], [74].

Notons que, la sémantique de  $\mathcal{ALC}$  peut être étendue pour traiter différents types de constructions comme ceux décrits dans [75].

La satisfaction des axiomes de  $\mathcal{K}$  par rapport à une interprétation  $\mathcal{I}$  est définie comme suit [74] :

- $\mathcal{I} \models C \sqsubseteq D$  si et seulement si  $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ .
- $\mathcal{I} \models C \equiv D$  si et seulement si  $C^{\mathcal{I}} = D^{\mathcal{I}}$ .
- $\mathcal{I} \models C(a)$  si et seulement si  $a^{\mathcal{I}} \in C^{\mathcal{I}}$ .
- $\mathcal{I} \models R(a, b)$  si et seulement si  $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$ .

### 1.4.4 Raisonnement sur $\mathcal{ALC}$

La LD dispose d'une sémantique claire et d'une description puissante de concepts complexes et offre des mécanismes d'inférences efficaces [76]. Elle représente le type de formalisme retenu par le projet Web sémantique pour représenter les ontologies [77].

Le problème de satisfaction ou de cohérence de la base de connaissances est la principale tâche de raisonnement dans les LDs. Il permet de vérifier si les connaissances codées dans le TBox et l'ABox ne sont pas contradictoires.

Il existe plusieurs approches de raisonnement. La plus largement utilisée est l'algorithme de tableau [73]. Un algorithme de tableau utilise les notions de réfutation pour réduire la subsomption à un problème de satisfaction. Il vérifie si une base de connaissances contient des contradictions ou non en vérifiant l'existence d'une interprétation qui est un modèle de la base de connaissances en construisant sa représentation finie. Une telle technique décompose les axiomes de la base de connaissances à l'aide d'un ensemble de règles de transformation qui préservent la cohérence (selon les constructeurs utilisés dans la LD), appelées règles d'achèvement (completion rules). Cette décomposition conduit à présenter des éléments contradictoires de la ABox.

Un système de LD offre des services d'inférence. Le but principal du raisonnement dans une LD est de découvrir des connaissances implicites à partir des connaissances explicites par l'inférence. Cette inférence s'effectue à deux niveaux d'ontologie : Terminologique et Assertionnel :

#### 1.4.4.1 Inférence au niveau Assertionnel

D'après [73], le niveau assertionnel ABox de LD comprend les principaux problèmes d'inférence suivants :

**Instanciation** : permet de vérifier qu'un individu  $a$  est instance d'un concept  $C$ , si seulement si  $a^{\mathcal{I}} \in C^{\mathcal{I}}$ .

D'autre part, les individus  $a$  et  $b$  sont liés par un rôle  $R$ , si pour toute interprétation  $\mathcal{I}$  alors  $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$ .

**Contrôle de cohérence** : Une ABox  $\mathcal{A}$  est dite cohérente par rapport à une TBox  $\mathcal{T}$ , s'il existe une interprétation  $\mathcal{I}$  tel que  $\mathcal{I} \models \mathcal{T}$  and  $\mathcal{I} \models \mathcal{A}$ .

**Problème de récupération** : Permet de trouver le concept le plus spécifique d'une ABox pour tout modèle  $\mathcal{I}$ .

#### 1.4.4.2 Inférence au niveau terminologique

[73] a présenté les principaux problèmes d'inférence au niveau terminologique TBox :

**Satisfiabilité** : Permet de vérifier qu'un concept admet des instances. Un concept  $C$  d'une terminologie  $\mathcal{T}$  est satisfiable si et seulement s'il existe un modèle  $\mathcal{I}$  de  $\mathcal{T}$  tel que  $C^{\mathcal{I}} \neq \emptyset$ .

**Subsommation** : Permet de vérifier qu'un concept est subsumé par d'autre concept. Un concept  $C$  est subsumé par un concept  $D$  pour une terminologie  $\mathcal{T}$  si et seulement si  $C^{\mathcal{I}} \sqsubseteq D^{\mathcal{I}}$  pour tout modèle  $\mathcal{I}$  de  $\mathcal{T}$ .

**Équivalence** : Un concept  $C$  est équivalent à un concept  $D$  pour une terminologie  $\mathcal{T}$  si et seulement si  $C^{\mathcal{I}} \equiv D^{\mathcal{I}}$  pour chaque modèle  $\mathcal{I}$  de  $\mathcal{T}$ .

**Disjonction** : Des concepts  $C$  et  $D$  sont disjoints par rapport à la terminologie  $\mathcal{T}$  si et seulement si  $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$  pour chaque modèle de  $\mathcal{I}$  de  $\mathcal{T}$ .

Dans les LDs, la principale tâche de raisonnement est le problème de satisfiabilité ou de cohérence de la base de connaissances (TBox et ABox ne sont pas en contradiction). Selon [78] et [73] toutes les tâches de raisonnement ci-dessus(1.4.4.1 et 1.4.4.2) peuvent être réduites au test de subsommation et test de satisfiabilité, du contrôle d'instance à la cohérence de base de connaissances. Cette propriété implique, que les moteurs d'inférence des LDs ne nécessitent souvent qu'un seul algorithme d'inférence pour raisonner au niveau terminologique. Nous citons des exemples de ces réductions :

- $C$  est insatisfiable si et seulement si  $\mathcal{T} \models C \sqsubseteq \perp$ .
- $\mathcal{T} \models C \sqsubseteq D$  si et seulement si  $C \sqcap \neg D$  Est insatisfiable.
- $C$  est satisfiable si et seulement si  $\{C(a)\}$  est cohérente par rapport à  $\mathcal{T}$ .
- $A \models C(a)$  si et seulement si  $A \cup \{C(a)\}$  est incohérente par rapport à  $\mathcal{T}$ .
- $C \equiv D$  si et seulement si  $\mathcal{T} \models C \sqsubseteq D$  et  $\mathcal{T} \models D \sqsubseteq C$ .
- $C \sqcap D$  si et seulement si  $\mathcal{T} \models C \sqcap D \sqsubseteq \perp$ .

#### 1.4.5 Extensions de $\mathcal{ALC}$

L'ensemble des constructeurs dans  $\mathcal{ALC}$  a été enrichi de différentes extensions avec plusieurs constructeurs, afin d'avoir des concepts complexes composés de concepts atomiques ou des rôles composés de rôles atomiques. Le Tableau 1.3 résume les extensions les plus utilisées de  $\mathcal{ALC}$  [73].

	Description	Syntaxe	Sémantique
$\mathcal{R}$	Conjonction des rôles	$R_1 \sqcap R_2$	$R_1^{\mathcal{I}} \sqcap R_2^{\mathcal{I}}$
$\mathcal{S}$	Transitivité des rôles	$R^+$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$ et $(b^{\mathcal{I}}, c^{\mathcal{I}}) \in R^{\mathcal{I}}$ Implique $(a^{\mathcal{I}}, c^{\mathcal{I}}) \in R^{\mathcal{I}}$
$\mathcal{H}$	Hiérarchie des rôles	$R_1 \sqsubseteq R_2$	$R_1^{\mathcal{I}} \subseteq R_2^{\mathcal{I}}$
$\mathcal{I}$	Rôles inverses	$R^-$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \mid (b^{\mathcal{I}}, a^{\mathcal{I}}) \in R^{\mathcal{I}}$
$\mathcal{F}$	Rôles fonctionnels	$\text{funct}(R)$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$ et $\nexists c^{\mathcal{I}} \neq b^{\mathcal{I}} : (a^{\mathcal{I}}, c^{\mathcal{I}}) \in R^{\mathcal{I}}$
$\mathcal{N}$	Restriction de cardinalité	$\bowtie nR$	$\{a \in \Delta^{\mathcal{I}} \mid \text{card} \mid \{b \in \Delta^{\mathcal{I}} \mid (a, b) \in R\} \bowtie n\}$
$\mathcal{Q}$	Restriction de cardinalité qualifiée	$\bowtie nR.C$	$\{a \in \Delta^{\mathcal{I}} \mid \text{card} \mid \{b \in \Delta^{\mathcal{I}} \mid (a, b) \in R \wedge b \in C^{\mathcal{I}}\} \bowtie n\}$
$\mathcal{O}$	Nominal	$\{a_1 \cdots a_n\}$	$\{a_1^{\mathcal{I}} \cdots a_n^{\mathcal{I}}\}$

 TABLE 1.3 – Extensions de  $\mathcal{ALC}$ 

Une extension de langage LD est définie par une chaîne de lettres majuscules se référant aux constructeurs utilisés. Par convention, on utilise  $\mathcal{S}$  pour désigner  $\mathcal{ALCR}^+$ . Le langage OWL est basé sur plusieurs extensions de LD ; dans sa première version OWL1 où OWL-Lite est basé sur  $\mathcal{SHIF}$  et OWL-DL est basé sur  $\mathcal{SHON}$ , cependant dans sa deuxième version OWL2 où OWL2-DL est basé sur  $\mathcal{SROIQ}$ , etc.

### 1.4.6 Moteurs d'inférence

Les moteurs d'inférence suivants sont conçus pour raisonner sur les logiques de description, ils permettent le raisonnement sur la ABox et la TBox en exploitant des ontologies possédant un niveau d'expressivité en logique de description et en OWL :

**Le moteur Pellet** [79] est un raisonneur open source développé en Java, il offre des évolutions fréquentes. Il supporte des ontologies décrites en RDF ou OWL2-DL et accepte les requêtes avec RDQL et SPARQL sur la ABox et la TBox.

Les principales caractéristiques de Pellet est qu'il intègre des optimisations pour l'utilisation des nominales (l'incohérence dans l'ontologie), la réponse à des requêtes conjonctives et le raisonnement incrémental. En tant que technique de raisonnement, Pellet utilise un algorithme basé sur la sémantique de tableau.

**Le moteur Racer** (Renamed Abox and Concept Expression Reasoner) [80] est un moteur de raisonnement commercialisé par Racer Systems GmbH & Co. KG,

fondé en 2004 par Volker Haarslev, Kay Hidde, Ralf Moller et Michael Wessel (il n'existe pas de version libre d'utilisation mais il est possible d'obtenir une licence gratuite dans le cadre de la recherche scientifique). Racer accepte des ontologies décrites en RDF ou OWL. Il possède son propre langage de requête nRQL (new Racerpro query Language) pour interroger les ontologies sur la ABox et la TBox.

**HermiT**<sup>10</sup> est un moteur d'inférence open source pour la logique de description *SROIQ* avec des types de données simples OWL2-DL qui prend en charge les descriptions graphiques. HermiT met en œuvre un raisonnement basé sur un nouveau calcul «hypertable» qui fournit un raisonnement beaucoup plus efficace. Il est le premier raisonneur capable de classer un certain nombre d'ontologies trop complexes pour un système disponible.

Ces moteurs de raisonnement fournissent une interface de programmation (API) afin de faciliter l'accessibilité par un programme Java. Pour cette raison, une bibliothèque de classes Java appelée **JENA**<sup>11</sup> a été développé pour faciliter la manipulation des applications pour le Web sémantique. JENA permet d'accepter le stockage en mémoire et les déclarations RDF, lecture et écriture des données RDF/XML, et utilise du langage RDFS pour l'interrogation d'une base RDF.

### 1.4.7 Complexité dans Les LDs

Une logique de description est expressive; une LD trop peu expressive ne permet pas de représenter des connaissances de domaines complexes. Jusqu'à maintenant, les logiques que nous avons présenté sont toutes décidables, mais de complexité différente [81]. La complexité est calculée en fonction des extensions choisies; cette complexité est polynomiale pour *ALC*, c'est-à-dire il est décidable par un algorithme déterministe en espace polynomial par rapport à la taille de son instance.

Dans [82] des classes de complexité proviennent de la théorie de la complexité informatique ont été défini :

- **La classe P** : est la classe des problèmes qui peuvent être résolus en temps polynomial par une machine déterministe.
- **La classe NP** : est la classe des problèmes qui peuvent être résolus en temps polynomial par une machine non déterministe

---

11. <https://jena.apache.org/>

- **La classe ExpTime** : est la classe des problèmes qui peuvent être résolus en temps exponentiel par une machine déterministe
- **La classe NExpTime** : est la classe des problèmes qui peuvent être résolus en temps exponentiel par une machine non déterministe.

Deux paramètres effectuent pour explique la complexité de tâche de raisonnement sur le LD est :

**La complexité combinée** : Elle considère toutes les composantes de la base de connaissances  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  comme des entrées. Autres mots, la taille du problème est égal à  $|\mathcal{T}| + |\mathcal{A}|$ .

**La complexité des données** : Elle prend comme entrée la taille de la ABox =  $|\mathcal{A}|$ .

Un problème NP-complet (c'est-à-dire un problème complet pour la classe NP) est un problème de décision capable de vérifier une solution efficacement (en temps polynomial); la classe des problèmes vérifiant cette propriété est notée NP.

Un problème NP-hard est un problème de la classe NP se ramène à celui-ci via une réduction polynomiale, et donc peut être dans une classe de problème plus large et donc plus difficile que la classe NP. Le Tableau 1.4 résume la complexité de calcul pour la vérification de la consistance en fonction de l'expressivité dans certaines LD [86], [85].

Famille LD	Complexité des données	Complexité combinée
<i>ALC</i>	NP-complete	EXPTIME-complete
<i>SHIF</i>	NP-complete	EXPTIME-complete
<i>SHOIN</i>	NP-hard	NEXPTIME-complete
<i>SROIQ</i>	NP-hard	N2EXPTIME-complete

TABLE 1.4 – Complexité de calcul pour la vérification de la consistance en fonction de l'expressivité dans certains LD.

Certaines LDs sont indécidables dans le sens où il n'y a pas d'algorithme efficace pour vérifier la satisfiabilité (c'est-à-dire les calculs en temps polynomial). Pour cette raison, plusieurs fragments légers de LD ont été introduits. L'un de ces fragments est LD légères; la famille *DL-Lite* [5]. Nous nous intéressons à cette famille *DL-Lite* de logique de description. Il convient parfaitement aux paramètres d'accès aux données basés sur l'ontologie.

## 1.4.8 La famille *DL-Lite*

Le consortium W3C a proposé des profils de OWL2 pour offrir des avantages importants dans des scénarios d'application particuliers. Au cours des dernières années, *DL-Lite* (famille de la logique de description légères), a été spécialement conçue pour les applications utilisant d'énormes volumes de données telles que les applications web (Web sémantique) où la réponse aux requêtes est la tâche de raisonnement la plus importante. *DL-Lite* garantis une complexité de calcul efficace du processus de raisonnement en utilisant les techniques de bases de données relationnelles [3].

### 1.4.8.1 Syntaxe et sémantique

Plusieurs fragments légères de LD ont été introduits. Ils offrent un bon compromis entre l'expressivité et la complexité du raisonnement ; *DL-Lite<sub>core</sub>* est le noyau de base de toutes les logiques *DL-Lite*, *DL-lite<sub>F</sub>* et *DL-lite<sub>R</sub>* sont sous famille de profile *OWL2-QL*. Dans notre thèse, nous utilisons la famille *DL-Lite* de logique de description pour faire référence à ces trois fragments.

#### Syntaxe

Le format de représentation des connaissances pour *DL-Lite* est comme suit :  $N_C$  l'ensemble de concepts atomiques,  $N_R$  l'ensemble de rôles atomiques, et  $N_I$  l'ensemble des individus.

Nous définissons aussi :  $A$  et  $C$  sont des concepts de base ( $\{A, C\} \in N_C$ ).  $P$  et  $E$  sont des rôles de base ( $\{P, E\} \in N_R$ ). D'autre part,  $B$  est un concept complexe et  $R$  est un rôle complexe.  $a$  et  $b$  sont des individus ( $\{a, b\} \in N_I$ ).

Le langage *DL-Lite* utilise les connecteurs de syntaxe pour définir  $C$  et  $D$  comme suit :

$$\begin{array}{l}
 R \longrightarrow P \mid P^- \\
 E \longrightarrow R \mid \neg R \\
 B \longrightarrow A \mid \exists R \\
 C \longrightarrow B \mid \neg B
 \end{array}$$

Une base de connaissances *DL-Lite* est une paire  $K = \langle \mathcal{T}, \mathcal{A} \rangle$ .  $\mathcal{T} = \text{TBox}$  est constitué par un ensemble fini d'axiomes d'inclusion entre concepts de la forme :

$$B \sqsubseteq C \quad \text{ou} \quad B \sqsubseteq \neg C.$$

Pour l'ABox, nous suivons la description de *DL-Lite* utilisée dans [3]. Les négations des rôles de cette description apparaissent dans *DL-Lit<sub>core</sub>*.

$\mathcal{A} = \text{ABox}$  contient l'ensemble fini d'assertions (les faits) des concepts et rôles atomiques de la forme :

$$A(a) \quad \text{et} \quad P(a, b).$$

Le langage *DL-Lite<sub>F</sub>* étend *DL-Lit<sub>core</sub>* avec la capacité de spécification fonctionnelle sur les rôles ou sur leurs inverses de la forme :

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

Le langage *DL-Lite<sub>R</sub>* étend *DL-Lit<sub>core</sub>* avec la capacité de spécification d'axiomes d'inclusion entre les rôles dans TBox de la forme :

$$R \sqsubseteq E.$$

Le langage *DL-Lite* n'utilise pas des opérateurs conjonctifs et disjonctifs. Cependant, une transformation logique permet d'obtenir les conjonctions et les disjonctions comme suit :

- Une conjonction de la forme :  $B \sqsubseteq C \cap D$  est équivalente à la paire d'axiomes d'inclusion :  $B \sqsubseteq C$  et  $B \sqsubseteq D$ .
- Une disjonction de la forme :  $C \sqcup D \sqsubseteq B$  est équivalente à la paire d'axiomes d'inclusion :  $C \sqsubseteq B$  et  $D \sqsubseteq B$ .

Noter que, toutes les bases de connaissances *DL-Lite* peuvent être écrites comme une base de connaissances de la logique de premier ordre (FOL : First order logic) [83] (Voir 2.3.2).

## Sémantique

La sémantique d'une base de connaissances *DL-Lite* est donnée en terme d'interprétations. Une interprétation  $\mathcal{I}=(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$  consiste d'un domaine non vide  $\Delta^{\mathcal{I}}$  et une fonction d'interprétation  $\cdot^{\mathcal{I}}$  qui attribue à chaque individu  $a \in N_I$  à un élément  $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ , chaque  $A \in N_C$  à un sous-ensemble  $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ , et chaque  $P \in N_R$  à  $P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ . Par ailleurs, la fonction d'interprétation  $\cdot^{\mathcal{I}}$  est étendue de manière simple pour les concepts et les rôles comme suit :

$$\begin{aligned} A^{\mathcal{I}} &\subseteq \Delta^{\mathcal{I}} \\ P^{\mathcal{I}} &\subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \\ (P^-)^{\mathcal{I}} &= \{(y, x) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} | (x, y) \in P^{\mathcal{I}}\} \\ (\exists R)^{\mathcal{I}} &= \{x \in \Delta^{\mathcal{I}} | \exists y \in \Delta^{\mathcal{I}} \text{ tel que } (x, y) \in R^{\mathcal{I}}\} \\ (\neg B)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus B^{\mathcal{I}} \\ (\neg R)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \setminus R^{\mathcal{I}} \end{aligned}$$

- Une interprétation  $\mathcal{I}$  est un modèle d'un concept d'axiome d'inclusion, noté par  $\mathcal{I} \models B \sqsubseteq C$  si et seulement si  $B^{\mathcal{I}} \subseteq C^{\mathcal{I}}$ .
- Une interprétation  $\mathcal{I}$  est un modèle d'un rôle d'axiome d'inclusion noté par  $\mathcal{I} \models R \sqsubseteq E$  si et seulement si  $R^{\mathcal{I}} \subseteq E^{\mathcal{I}}$ .
- Une interprétation  $\mathcal{I}$  est un modèle d'un membre d'assertion  $A(a)$  noté par  $\mathcal{I} \models A(a)$  si et seulement si  $a^{\mathcal{I}} \in A^{\mathcal{I}}$ .
- Une interprétation  $\mathcal{I}$  est un modèle d'un membre d'assertion  $P(a, b)$  noté par  $\mathcal{I} \models P(a, b)$  si et seulement si  $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in P^{\mathcal{I}}$ .
- Pour le *DL-Lite<sub>F</sub>* une interprétation  $\mathcal{I}$  est un modèle d'axiomes (*functR*) si et seulement si  $(a, b) \in R^{\mathcal{I}}$  et  $(a, c) \in R^{\mathcal{I}}$  implique que  $c = b$ .

Une interprétation  $\mathcal{I}$  est un modèle d'une base de connaissances  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  si et seulement si cette interprétation  $\mathcal{I}$  satisfait la base de connaissances  $\mathcal{K}$  pour toutes les axiomes dans  $\mathcal{T}$  et toutes les axiomes dans  $\mathcal{A}$  [84].

### 1.4.8.2 Extension de la famille *DL-Lite*

La syntaxe de la famille étendue de *DL-Lite* est composée de  $DL-Lite_\alpha^\beta$  logiques tel que :

- $\alpha = \{core, krom, horn, bool\}$ .
- $\beta = \{-, \mathcal{H}, \mathcal{F}, \mathcal{N}, \mathcal{HF}, \mathcal{HN}, (\mathcal{HF}), (\mathcal{HN}), (\mathcal{HF})^+, (\mathcal{HN})^+\}$ .

Noter que :

- $\mathcal{H}$  désigne les hiérarchies de rôles (les axiomes d'inclusion de rôles).
- $\mathcal{F}$  désigne la fonctionnalité de rôle (*Funct R*).
- $\mathcal{N}$  indique la restriction de nombre  $\geq n R$ .

$\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ \diagdown $DL-Lite_\alpha$	$DL-Lite_{bool}$	$DL-Lite_{krom}$	$DL-Lite_{horn}$
TBox	$C \sqsubseteq D$	$C \sqsubseteq D, C \sqsubseteq \neg D, \neg C \sqsubseteq D$	$\sqcap_n C_n \sqsubseteq D$
ABox	$A(a), \neg A(a), P(a, b), \neg P(a, b)$		

TABLE 1.5 – Les axiomes d'inclusion des concepts et des assertions dans  $DL-Lite_\alpha$

Nous avons donné la sémantique de  $\mathcal{H}$ ,  $\mathcal{F}$ , et  $\mathcal{N}$  dans le Tableau 1.3. Cependant, Le tableau 1.5 donne les formes des axiomes d'inclusion de concepts et des assertions qui peuvent être autorisés dans une base de connaissances  $DL-Lite_\alpha$ .

Il existe une relation entre les différentes extensions de  $DL-Lite$  ; la logique  $DL-Lite_{bool}^{HN}$  est considérée comme la logique la plus expressive. Cependant, la logique le plus basique (moins expressive) est la logique  $DL-Lite_{core}$  (Voir 1.4.8.1) [2].

Tandis que  $C \sqsubseteq \neg D$  est équivalent à  $C \sqcap B \sqsubseteq \perp$ ,  $DL-Lite_{core}$  qui est situé à l'intersection de  $DL-Lite_{krom}$  et  $DL-Lite_{horn}$ . De plus, la logique  $DL-Lite_\alpha^{HN}$  est considéré comme l'extension de  $DL-Lite_\alpha^{HF}$  car ce dernier est considéré comme l'extension de  $DL-Lite_\alpha^{HN}$  [5].

Les TBox's de  $DL-Lite_\alpha^{HN}$ ,  $DL-Lite_\alpha^{(HN)^+}$ ,  $DL-Lite_\alpha^{HF}$ , et  $DL-Lite_\alpha^{(HF)^+}$  contient les contraintes des rôles de la forme :

- Disjonction des rôles :  $Dis(R_1, R_2)$ , Asymétrie du rôle :  $Asym(R)$ .
- Symétrie du rôle :  $Sym(R)$ , Irréflexivité du rôle :  $Irr(R)$ .
- Réflexivité du rôle :  $Rfe(R)$ , et Transitivité du rôle :  $Tra(R)$ .

La sémantique de ces nouvelles constructions est définie comme suit pour une interprétation  $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$  :

**Disjonction des rôles** :  $\mathcal{I} \models Dis(R_1, R_2)$  si et seulement si  $R_1^{\mathcal{I}} \cap R_2^{\mathcal{I}} = \emptyset$ .

**Asymétrie du rôle** :  $\mathcal{I} \models Asym(R)$  si et seulement si  $R^{\mathcal{I}} \cap (R^-)^{\mathcal{I}} = \emptyset$ .

**Symétrie du rôle** :  $\mathcal{I} \models Sym(R)$  si et seulement si  $R^{\mathcal{I}} = (R^-)^{\mathcal{I}}$ .

**Irréflexivité du rôle** :  $\mathcal{I} \models Irr(R)$  si et seulement si  $(x, x) \notin R^{\mathcal{I}}$  pour tous  $x \in \Delta^{\mathcal{I}}$ .

**Réflexivité du rôle** :  $\mathcal{I} \models Ref(R)$  si et seulement si  $(x, x) \in R^{\mathcal{I}}$  pour tous  $x \in \Delta^{\mathcal{I}}$ .

**Transitivité du rôle** :  $\mathcal{I} \models Tra(R)$  si et seulement si  $(x, y) \in R^{\mathcal{I}} \wedge (y, z) \in R^{\mathcal{I}}$  implique  $(x, z) \in R^{\mathcal{I}}$ .

$DL-Lite_{\alpha}^{\beta}$ \diagdown Axiomes	Nombre de restriction	Inclusion de rôle	Contraintes en rôles
$DL-Lite_{\alpha}$	$\exists R$	-	-
$DL-Lite_{\alpha}^F$	$\exists R, funct$	-	-
$DL-Lite_{\alpha}^N$	$\geq nR$	-	-
$DL-Lite_{\alpha}^H$	$\exists R$	+	-
$DL-Lite_{\alpha}^{HF}$	$\exists R, funct$	+	-
$DL-Lite_{\alpha}^{HN}$	$\geq nR$	+	-
$DL-Lite_{\alpha}^{(HF)}$	$\exists R.C, funct$	+	Dis, (A)sym et (Ir)ref
$DL-Lite_{\alpha}^{(HN)}$	$\geq nR.C$	+	Dis, (A)sym et (Ir)ref
$DL-Lite_{\alpha}^{(HF)^+}$	$\exists R.D, funct$	+	Dis, (A)sym, (Ir)ref, Tra
$DL-Lite_{\alpha}^{(HN)^+}$	$\geq nR.D$	+	Dis, (A)sym, (Ir)ref, Tra

TABLE 1.6 – Extension de la famille  $DL-Lite$

Le Tableau 1.6 présente les différentes extensions de la famille  $DL-Lite$  (Le symbole + signifie l'existence des propriétés dans la famille, et le symbole - signifie l'absence de ces propriétés). En effet, uniquement  $DL-Lite_{\alpha}^{(HN)^+}$  et  $DL-Lite_{\alpha}^{(HF)^+}$  sont autorisés par la contrainte de transitivité.

Afin d'éviter l'ambiguïté, nous utilisons dans le reste de cette thèse le terme  $DL-Lite$  pour désigner les trois extensions :  $DL-Lite_{core}$ ,  $DL-Lite_F$  et  $DL-Lite_R$ .

**Exemple 2.** Nous prenons l'ontologie décrite dans l'exemple 1 pour donner un exemple de  $DL-Lite_{core}$ . Cette ontologie est encodée par la TBox suivante :

$$\begin{array}{l}
 Teacher \sqsubseteq \neg Student \\
 Teacher \sqsubseteq \exists TeachesTo \\
 \exists TeachesTo^- \sqsubseteq Student \\
 Student \sqsubseteq \exists HasSupervisor \\
 \exists HasSupervisor^- \sqsubseteq Teacher \\
 Researcher \sqsubseteq Teacher
 \end{array}$$

Pour obtenir  $DL-Lite_R$ , on peut étendre la TBox de  $DL-Lite_{core}$  avec l'axiome suivant :

$$HasSupervisor^- \sqsubseteq TeachesTo$$

Pour obtenir  $DL-Lite_F$ , on peut étendre la TBox de  $DL-Lite_{core}$  avec l'axiome suivant :

$$funct\ HasSupervisor$$

L'Abox peut être exprimée comme suit :

$$\begin{array}{l}
 Student(Bill) \quad Student(Bob) \\
 Teacher(Mary) \quad Teacher(Johne) \\
 TeacherTo(Mary, Bill) \\
 TakesCourse(Bob, Web3.0) \\
 HasSupervisor(Bill, Mary)
 \end{array}$$

Nous supposons que  $\Delta^{\mathcal{I}} = \{Bill, John, Mary, Bob, Joe, Anne\}$ . On peut considérer l'interprétation suivantes (des modèles des bases de connaissances  $\mathcal{K}$  :) )

$$\begin{array}{l}
 (Student)^{\mathcal{I}} = \{Bill, Bob\} \\
 (Teacher)^{\mathcal{I}} = \{Mary, Johne\} \\
 (Department)^{\mathcal{I}} = \{Chimie, Informatique\} \\
 (Course)^{\mathcal{I}} = \{Web3.0, Physique\} \\
 (TeachesTo)^{\mathcal{I}} = \{(Mary, Bill)\} \\
 (TeachesOf)^{\mathcal{I}} = \{(Johne, Web3.0)\} \\
 (TakesCourse)^{\mathcal{I}} = \{(Bob, Web3.0)\}
 \end{array}$$

■

## 1.5 Conclusion

Les ontologies sont apparues comme un composant principal dans les applications du Web sémantique. Dans ce chapitre, nous avons rappelé des notions de base sur les ontologies.

Nous avons présenté aussi le formalisme de la logique de description, qui offre un cadre logique pour représenter les ontologies. Un accent particulier a été faite sur trois membres principaux de la famille *DL-Lite* : *DL-Lite<sub>core</sub>*, *DL-Lite<sub>F</sub>*, et *DL-Lite<sub>R</sub>*.

Dans les applications réelles, les connaissances peuvent être issues de différentes sources d'informations. Donc, des problèmes difficiles sont reconnus comme la fusion, l'évolution, l'incohérence et la gestion de l'incertitude dans les ontologies. Le chapitre suivant se concentre sur ces problèmes et donne un aperçu des différentes techniques et outils proposés pour les traiter.

## Chapitre 2

# Débogage d'ontologie : Traitement sur les bases de connaissances *DL-Lite*

*The task of classifying all the words of language, or what's the same thing, all the ideas that seek expression, is the most stupendous of logical tasks.*

*Charles Sanders Peirce, letter to editor B. E. Smith of the Century Dictionary*

### 2.1 Introduction

DANS la communauté de l'Intelligence Artificielle (IA), et notamment dans les applications web, les connaissances peuvent constituer des sources d'information différentes et souvent contradictoires. Cependant, dans le projet OBDA, une base de connaissances ABox de *DL-Lite* peut être émise à partir de sources d'informations multiples. En général, la révision ou la fusion de différentes sources d'information peut entraîner des problèmes d'inconsistance. Plusieurs approches sont proposées pour la restauration de la consistance de la base de connaissances en raison de l'exploiter et d'effectuer des inférences [9].

Ce chapitre fournit une vue généralisée sur les caractéristiques des bases de connaissances *DL-Lite* standard. Il étudie les différentes stratégies d'inférence pour la réponse aux requêtes sur ces bases de connaissances *DL-Lite*.

## 2.2 Approche OBDA

OBDA (Ontology-Based Data Access)<sup>1</sup> est un nouveau paradigme, fondé sur l'utilisation de la représentation des connaissances et des techniques de raisonnement, pour régir les ressources (données, méta-données, services, processus, etc.) des systèmes d'information modernes. L'approche OBDA est le résultat de plus d'une décennie de recherche menée par le groupe DASILab<sup>2</sup> au Département d'informatique, de contrôle et d'ingénierie de gestion Antonio Ruberti à l'Université Sapienza de Rome, sous la direction du Maurizio Lenzerini, et maintenant aussi par les systèmes OBDA. Ces dernières années, cette recherche s'est transformée en travail sur le terrain, en collaboration avec des entreprises dans les secteurs public et privé italien et international.

Dans les grandes organisations, les sources de données sont généralement reformulées pour s'adapter aux exigences de l'application, et les applications sont modifiées pour répondre aux nouvelles exigences. Le résultat est que les données stockées dans différentes sources et les processus d'exploitation sur eux ont tendance à être redondants, mutuellement incompatibles (inconsistantes), et obscure pour les grandes catégories d'utilisateurs.

Par conséquent, l'accès aux données signifie interagir avec les experts informatiques qui savent où sont les données et ce que cela signifie dans différents contextes et peuvent donc traduire le besoin d'informations exprimé par l'utilisateur dans les enquêtes appropriées. Ce processus peut coûter cher et prendre du temps. La qualité des données est souvent citée comme un facteur essentiel dans la prestation de services d'information à haute niveau. Mais comment vérifier la qualité et la fiabilité des données si nous ne comprenons pas clairement la sémantique que les données devraient apporter ?

L'idée principale de OBDA est de fournir aux utilisateurs un accès aux informations contenues dans leurs sources de données grâce à une architecture à trois niveaux, constitué par l'ontologie, les sources, et la correspondance (mapping). Grâce à cette architecture, OBDA fournit une connexion sémantique de bout en bout entre les utilisateurs et les sources de données, permettant aux utilisateurs d'interroger directement les données réparties sur plusieurs sources distribuées, grâce au vocabulaire familier de l'ontologie : l'utilisateur formule des requêtes SPARQL sur l'ontologie qui sont transformées, via la couche de mapping, en requêtes SQL sur les bases de données relationnelles.

---

1. <http://www.obdasystems.com/>

2. <http://www.dis.uniroma1.it/>

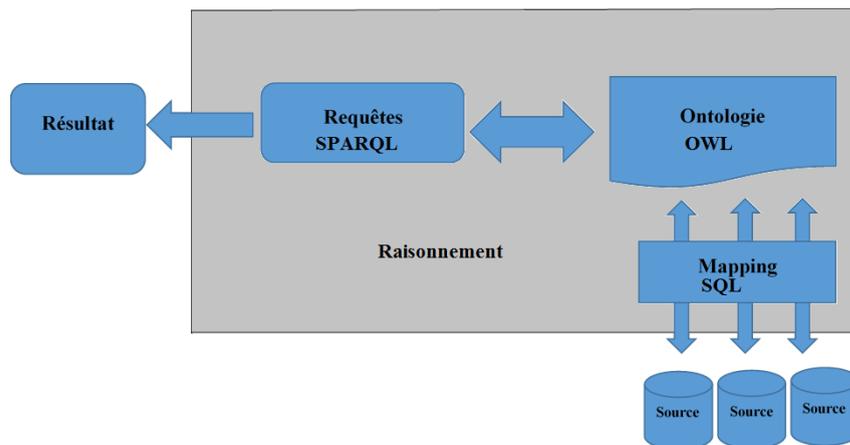


FIGURE 2.1 – Architecture de système OBDA

La Figure 2.2 représente l'architecture à trois couches de système OBDA :

**Couche Ontologie :** dans l'architecture OBDA, l'ontologie est le moyen de poursuivre une approche déclarative de l'intégration de l'information. La base de connaissances de domaine est spécifiée au moyen d'une description formelle et de niveau élevé de ses aspects statiques et dynamiques, représentés par l'ontologie. Par la représentation explicite du domaine, nous obtenons une réutilisation des connaissances acquises, ce qui n'est pas atteint lorsque le schéma global est simplement une description unifiée des sources de données.

**Couche source de données :** La couche source de données est constituée par les sources de données existantes de l'organisation.

**Couche Mapping :** La couche mapping relie la couche d'ontologie avec la couche source de données en définissant les relations entre les concepts de domaine, d'une part, et les sources de données d'autre part.

### 2.2.1 Avantages de l'approche OBDA

Dans les trois niveaux OBDA, l'utilisateur du système d'information peut interagir avec le système au moyen d'une représentation abstraite du domaine. Les utilisateurs peuvent demander des requêtes sur la base des concepts du domaine (n'est sur les structures des sources de données). En prenant en compte l'ontologie et les mappings entre les sources de données, le système OBDA est chargé de traduire la requête d'origine dans une requête pour être évaluée à la source.

OBDA peut être considéré comme une forme d'intégration d'informations, où le schéma global habituel est remplacé par le modèle conceptuel du domaine d'application, formulé comme une ontologie. Avec cette approche, la vue intégrée que le système fournit aux consommateurs d'information n'est pas simplement une structure de données permettant d'accueillir les différentes données sur les sources, mais une description sémantique riche des concepts pertinents dans le domaine d'intérêt, ainsi que les relations entre ces concepts.

L'approche OBDA n'impose pas d'intégrer pleinement les sources de données à la fois. Plutôt, après avoir construit même un squelette approximatif du modèle de domaine, on peut ajouter de nouvelles sources de données ou de nouveaux éléments, lorsqu'ils deviennent disponibles au besoin. Par conséquent, la conception globale peut être considérée comme le processus progressif de compréhension et de représentation du domaine, les sources de données disponibles et les relations entre elles. L'objectif est de soutenir l'évolution de l'ontologie et des mappages de telle sorte que le système continue à fonctionner tout en évoluant.

L'ontologie et les mappages correspondants aux sources de données constituent un terrain d'entente pour la documentation de toutes les données de l'organisation, avec des avantages évidents pour la gouvernance et la gestion du système d'information [87].

### **2.2.2 Description logique pour OBDA avec bases de données**

La notion clé d'OBDA avec les bases de données est la réécriture des requêtes. L'utilisateur formule une requête  $q$  dans le vocabulaire d'une ontologie donnée  $T$ .  $(T, q)$  est appelée une requête à l'ontologie. La tâche d'un système OBDA est de "réécrire"  $q$  et  $T$  dans une nouvelle requête  $q'$  dans le vocabulaire des données de sorte que, pour toute donnée  $A$  possible (dans ce vocabulaire), les réponses à  $q$  plus  $(T, A)$  sont précisément le même que les réponses à  $q'$  sur  $A$ . Ainsi, le problème de l'interrogation des données  $A$  (dont la structure n'est pas connue de l'utilisateur) en termes d'ontologie  $T$  (accessible à l'utilisateur) est réduit au problème de l'interrogation  $A$  directement [87].

## 2.3 Caractéristiques d'une base de connaissances *DL-Lite*

Pour toutes les caractéristiques suivantes, on considère une base de connaissances *DL-Lite*  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ . Avec,  $\mathcal{T} = \{\mathcal{T}_p, \mathcal{T}_n\}$  est la TBox qui se compose d'un ensemble d'axiomes d'inclusion positive  $\mathcal{T}_p$  de la forme  $B_1 \sqsubseteq B_2$  (dépendance d'inclusion), et  $\mathcal{T}_n$  est l'ensemble d'axiomes d'inclusion négative de la forme  $B_1 \sqsubseteq \neg B_2$  (contrainte d'intégrité).

### 2.3.1 Incohérence et inconsistance

Une base de connaissances *DL-Lite*  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  est dite incohérente s'il existe au moins un concept non satisfaisant (aucun individu ne peut appartenir au concept) ;  $\mathcal{T} \models \exists C \in N_C, C^{\mathcal{I}} \neq \emptyset$  [40]. Dans la plupart des travaux actuels sur les ontologies, le problème de l'incohérence est souvent discuté au niveau de la base de terminologie TBox. D'autre part, une base de connaissances *DL-Lite*  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  est inconsistante si et seulement si elle n'admet aucun modèle [88].

L'incohérence et l'inconsistance sont liées entre elles. Autrement dit, l'incohérence est une cause potentielle de l'inconsistance. Par exemple, la base de connaissances  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  avec  $\mathcal{T} = \{B \sqsubseteq \neg C, C \sqsubseteq B\}$  est incohérente mais elle est consistante. Si on suppose que  $B$  est un concept insatisfaisant, et si on ajoute une instance  $a$  à  $B$ , l'ontologie sera incohérente et inconsistante [89].

### 2.3.2 Réductibilité de FOL

Cette propriété réduit les tâches de raisonnement dans la base de connaissances *DL-Lite* à l'évaluation sur la ABox des requêtes FOL (*First Order Logic*) obtenues à partir de  $\mathcal{T}$ . En effet, les tâches de raisonnement sont réalisées en deux étapes : La première consiste à produire des requêtes FOL en utilisant des axiomes de TBox. La deuxième étape consiste à évaluer les requêtes obtenues sur la Abox qui peuvent être stockées dans une base de données relationnelle. La propriété de réductibilité de FOL n'est pas toujours garantie pour tous les fragments de *DL-Lite* indiqués dans 1.4.8.2 [89] [5].

Le Tableau 2.1 donne l'équivalence entre les axiomes *DL-Lite* et les formules FOL ( $A, B$  sont des concepts,  $R, P$  sont des rôles, et  $a, b, c$  sont des individus).

Axiome <i>DL-Lite</i>	Formule FOL
$A \sqsubseteq B / (A \sqsubseteq \neg B)$	$\forall a. A(a) \rightarrow B(a) / (\forall a. A(a) \rightarrow \neg B(a))$
$A \sqsubseteq \exists R / (A \sqsubseteq \neg \exists R)$	$\exists a. A(a) \rightarrow \exists b. R(a, b) / (\exists a. A(a) \rightarrow \neg \exists b. R(a, b))$
$A \sqsubseteq \exists R^- / (A \sqsubseteq \neg \exists R^-)$	$\exists a. A(a) \rightarrow \exists b. R(b, a) / (\exists a. A(a) \rightarrow \neg \exists b. R(b, a))$
$\exists R \sqsubseteq A / (\exists R \sqsubseteq \neg A)$	$\forall a. \exists b. R(a, b) \rightarrow A(a) / (\forall a. \exists b. R(a, b) \rightarrow \neg A(a))$
$\exists R^- \sqsubseteq A / (\exists R^- \sqsubseteq \neg A)$	$\forall a. \exists b. R(b, a) \rightarrow A(a) / (\forall a. \exists b. R(b, a) \rightarrow \neg A(a))$
$\exists R \sqsubseteq \exists P / (\exists R \sqsubseteq \neg \exists P)$	$\forall a. \exists b. R(a, b) \rightarrow \exists c. P(a, c) / (\forall a. \exists b. R(a, b) \rightarrow \neg \exists c. P(a, c))$
$\exists R \sqsubseteq \exists P^- / (\exists R \sqsubseteq \neg \exists P^-)$	$\forall a. \exists b. R(a, b) \rightarrow \exists c. P(c, a) / (\forall a. \exists b. R(a, b) \rightarrow \neg \exists c. P(c, a))$
$\exists R^- \sqsubseteq \exists P / (\exists R^- \sqsubseteq \neg \exists P)$	$\forall a. \exists b. R(b, a) \rightarrow \exists c. P(a, c) / (\forall a. \exists b. R(b, a) \rightarrow \neg \exists c. P(a, c))$
$\exists R^- \sqsubseteq \exists P^- / (\exists R^- \sqsubseteq \neg \exists P^-)$	$\forall a. \exists b. R(b, a) \rightarrow \exists c. P(c, a) / (\forall a. \exists b. R(b, a) \rightarrow \neg \exists c. P(c, a))$
$R \sqsubseteq P / (R \sqsubseteq \neg P)$	$\forall a, b. R(a, b) \rightarrow P(a, b) / (\forall a, b. R(a, b) \rightarrow \neg P(a, b))$
$R^- \sqsubseteq P / (R^- \sqsubseteq \neg P)$	$\forall a, b. R(a, b) \rightarrow P(b, a) / (\forall a, b. R(a, b) \rightarrow \neg P(b, a))$
$R \sqsubseteq P^- / (R \sqsubseteq \neg P^-)$	$\forall a, b. R(a, b) \rightarrow P(b, a) / (\forall a, b. R(a, b) \rightarrow \neg P(b, a))$
$R^- \sqsubseteq P^- / (R^- \sqsubseteq \neg P^-)$	$\forall a, b. R(a, b) \rightarrow P(a, b) / (\forall a, b. R(a, b) \rightarrow \neg P(a, b))$
$(\text{funct}R)$	$\forall a, b, c. R(a, b) \wedge R(a, c) \rightarrow b = c$
$(\text{funct}R^-)$	$\forall a, b, c. R(a, b) \wedge R(c, a) \rightarrow b = c$

TABLE 2.1 – Équivalence entre les axiomes *DL-Lite* et les formules FOL.

### 2.3.3 Fermeture négative

La fermeture négative de  $\mathcal{T}$  notée par  $\text{cln}(\mathcal{T})$ , représente la propagation des axiomes négatifs en utilisant à la fois des axiomes positifs et des axiomes négatifs dans la TBox. Pour obtenir  $\text{cln}(\mathcal{T})$  de *DL-Lite<sub>core</sub>* nous utilisons les règles suivantes à plusieurs reprises jusqu'à atteindre un point fixe [3] :

- Tous les axiomes négatifs de  $\mathcal{T}$  sont dans  $\text{cln}(\mathcal{T})$ .
- Tous les axiomes de fonctionnalité de  $\mathcal{T}$  sont dans  $\text{cln}(\mathcal{T})$ .
- Si  $A \sqsubseteq B$  dans  $\mathcal{T}$  et  $B \sqsubseteq \neg C$  (ou  $C \sqsubseteq \neg B$ ) dans  $\text{cln}(\mathcal{T})$ . Alors,  $A \sqsubseteq \neg C$  est dans  $\text{cln}(\mathcal{T})$ .
- Si  $R \sqsubseteq P$  dans  $\mathcal{T}$  et  $\exists P \sqsubseteq \neg E$  (ou  $E \sqsubseteq \neg \exists P$ ) sont dans  $\text{cln}(\mathcal{T})$ . Alors,  $\exists R \sqsubseteq \neg E$  est dans  $\text{cln}(\mathcal{T})$ .

- Si  $R \sqsubseteq P$  dans  $\mathcal{T}$  et  $\exists P^- \sqsubseteq \neg E$  (ou  $E \sqsubseteq \neg \exists P^-$ ) sont dans  $cln(\mathcal{T})$ . Alors,  $\exists R^- \sqsubseteq \neg E$  est dans  $cln(\mathcal{T})$ .

**Exemple 3.** Nous considérons la TBox donnée dans l'exemple 2, l'ensemble initiale de la fermeture négative de cette TBox est :  $cln(\mathcal{T}) = \{Teacher \sqsubseteq \neg Student\} \cup \{funct \ HasSupervisor\}$ . En suite, nous pouvons déduire l'ensemble finale de la fermeture négative par l'application des règles d'inclusions :

$$\begin{aligned}
 cln(\mathcal{T}) = \{ & Teacher \sqsubseteq \neg Student, \\
 & \exists TeacherTo^- \sqsubseteq Student, \\
 & \exists HasSupervisor^- \sqsubseteq Teacher, \\
 & \exists TeacherTo \sqsubseteq \neg Student, \\
 & HasSupervisor \sqsubseteq \neg Teacher, \\
 & funct \ HasSupervisor \}
 \end{aligned}$$

■

### 2.3.4 Calcul des conflits

Cette propriété suit l'algorithme donné par [3] pour vérifier la consistance d'une base de connaissances *DL-Lite*. Par conséquent, notre travail doit adapter l'algorithme en ajoutant un pas de plus afin d'énumérer toutes les paires d'assertions impliquées dans les conflits.

L'algorithme 1 décrit comment calculer  $\mathcal{C}(\mathcal{K} \cup \{N\})$ ; l'ensemble  $\mathcal{C}(\mathcal{K}')$  stocke les éléments de conflits; pour chaque axiome négatif de NI  $X \sqsubseteq \neg Y$  de  $cln(\mathcal{T})$ . L'algorithme recherche l'existence d'une contradiction dans ABox (vérifier si  $\{X \sqsubseteq \neg Y, (\alpha, \beta)\}$  est consistant ou non).

**Exemple 4.** On considère  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  une base de connaissances avec  $\mathcal{T} = \{A \sqsubseteq \neg B, C \sqsubseteq D\}$ , et  $\mathcal{A} = \{B(a), C(a), A(a)\}$ . Nous avons,  $cln(\mathcal{T}) = \{A \sqsubseteq \neg B\}$ . Par conséquence,  $\{(A \sqsubseteq \neg B), (B(a), C(a))\}$  et  $\{(A \sqsubseteq \neg B), (A(a), C(a))\}$  sont consistants. Mais,  $\{(A \sqsubseteq \neg B), (B(a), A(a))\}$  est inconsistant. Donc,  $\mathcal{C}(\mathcal{K}) = \{(B(a), A(a))\}$ . ■

---

**Algorithme 1** Calcul de l'ensemble de conflits

---

**Input:**  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$

**Output:**  $\mathcal{C}(\mathcal{K})$

```

1:  $\mathcal{K}' = \langle \mathcal{T}', \mathcal{A}' \rangle \leftarrow \mathcal{K} \cup \{N\}$ 
2:  $\mathcal{C}(\mathcal{K}') \leftarrow \emptyset$ 
3: for  $(X \sqsubseteq \neg Y) \in \text{cln}(\mathcal{T}')$  do
4:   for  $(\alpha, \beta) \in \mathcal{A}'$  do
5:     if  $\{(X \sqsubseteq \neg Y), (\alpha, \beta)\}$  est inconsistant then
6:        $\mathcal{C}(\mathcal{K}') \leftarrow \mathcal{C}(\mathcal{K}') \cup (\alpha, \beta)$ 
7: return  $\mathcal{C}(\mathcal{K}')$ 

```

---

Pour calculer  $\mathcal{C}(\mathcal{K} \cup \{N\})$ , il faut d'abord obtenir la fermeture négative  $\text{cln}(\mathcal{T}')$ , par l'utilisation des règles rappelées dans la section 2.3.3. Le calcul des conflits procède à l'évaluation sur  $\mathcal{A}'$  pour chaque axiome négatif NI dans  $\text{cln}(\mathcal{T}')$ , afin de montrer si  $\mathcal{A}'$  contient des paires d'assertions qui contredisent avec les axiomes NI.

Intuitivement, l'évaluation de chaque  $X \sqsubseteq \neg Y$  de  $\text{cln}(\mathcal{T}')$ , tout simplement renvoie tout  $(X(x), Y(y))$ , tel que  $X(x)$  et  $Y(y)$  appartiennent à  $\mathcal{A}'$ .  $X(x)$  et  $Y(y)$  sont des assertions de concept de base ou des assertions de rôle de la forme  $R(x, y)$  si  $X$  et  $Y = \exists R$  ou de la forme  $R(y, x)$  si  $X$  et  $Y = \exists R^-$ . Le résultat de l'évaluation d'un axiome NI est une collection d'ensembles contenant deux éléments (Cette particularité est détaillée dans le prochain chapitre).

### 2.3.5 Traitement d'inconsistance sur DL-Lite

La principale approche de la gestion des inconsistances consiste à éliminer l'inconsistance en calculant d'abord les sous-ensembles consistants qui rétablissent la consistance de la base initiale, puis les utiliser pour effectuer des inférences.

**Définition 1.** Soit  $\mathcal{K}$  une base de connaissances inconsistante. Un sous-ensemble maximal consistant de  $\mathcal{K}$ , désigné par  $\mathcal{K}'$ , est défini comme suit :

- $\mathcal{K}' \subseteq \mathcal{K}$ .
- $\mathcal{K}'$  est consistante.
- $\forall \mathcal{K}''$ , Si  $\mathcal{K}' \subseteq \mathcal{K}'' \subseteq \mathcal{K}$ , alors  $\mathcal{K}''$  est inconsistante.

Le problème de gérer l'inconsistance est important dans le projet OBDA. Dans ce cadre, l'inconsistance est définie par rapport à certaines assertions qui contredisent avec la terminologie TBox. Cette TBox est généralement vérifiée et validé alors que les assertions peuvent être fournies en grandes quantités par des sources diverses et peuvent contredire avec TBox. Par conséquent, il est difficile de vérifier et de valider manuellement toutes les assertions de la ABox.

De nombreuses stratégies d'inférence sont proposées pour sélectionner l'ensemble de sous-bases consistantes de  $\mathcal{K}$  afin d'effectuer des inférences. Comme il sera mentionné dans le chapitre 3, une extension de ces stratégies d'inférence est proposée lorsque la base des connaissances est stratifiée avec priorité [91] [92].

Plusieurs travaux existants [93], [67], [18] et [19] fondamentalement inspirés par les approches proposées dans le domaine de base de données, ont essayé de traiter l'inconsistance dans *DL-Lite* en adoptant plusieurs méthodes d'inférence. Toutes les approches proposées sont basées sur la notion de réparation qui est fortement liée à la notion de réparation de base de données afin de répondre aux requêtes posées sur des bases de données inconsistantes. Cette notion de réparation a été étendue au *DL-Lite* en définissant la base d'assertion réparée qui est simplement une sous base d'assertions maximale consistante avec la terminologie. Cette dernière a été inspirée des sous ensembles maximales cohérentes définies dans la logique propositionnelle.

## 2.4 Bases de connaissances *DL-Lite* MBox

**Définition 2.** *Nous appelons MBox une base de connaissances DL-Lite avec multiple ABox de la forme  $\mathcal{K}_{\mathcal{M}} = \langle \mathcal{T}, \mathcal{M} \rangle$  sachant que  $\mathcal{T}$  soit un standard TBox de DL-Lite, et  $\mathcal{M} = \{\mathcal{A}_1, \dots, \mathcal{A}_n\}$  est un ensemble multiple des assertions, et chaque  $\mathcal{A}_i$  est un standard ABox de DL-Lite.*

Dans la définition 2, il n'y a pas d'hypothèse supplémentaire concernant la consistance pour chaque  $\langle \mathcal{T}, \mathcal{A}_i \rangle$ . Cependant,  $\langle \mathcal{T}, \cup_{\mathcal{A}_i \in \mathcal{M}} \mathcal{A}_i \rangle$  peut être susceptible d'être inconsistant. MBox est considéré comme un moyen pratique de représenter une source multiple d'informations où chaque  $\mathcal{A}_i \in \text{ABox}$  fournit par une source d'informations distincte. Dans ce cas, chaque élément du MBox reflète comme une réparation de la base de connaissances *DL-Lite* inconsistante.

### 2.4.1 Fermeture déductive sur MBox

L'application de la fermeture déductive sur un ABox conduit à produire toutes les assertions possibles dans cette ABox. Dans le cas d'une base de connaissances  $\mathcal{K}_{\mathcal{M}} = \langle \mathcal{T}, \mathcal{M} \rangle$  avec MBox  $\mathcal{M} = \{\mathcal{A}_1 \cdots \mathcal{A}_n\}$ .  $\mathcal{T}_p$  est l'ensemble de tous les axiomes d'inclusion positive de  $\mathcal{T}$ .  $\mathcal{D}_{\mathcal{I}}$  est l'ensemble de tous les individus dans toutes les ABoxes.  $D_C$  est l'ensemble des concepts, et  $D_R$  l'ensemble des rôles dans  $\mathcal{K}_{\mathcal{M}}$  :

— La fermeture déductive d'un ABox  $\mathcal{A}_i$  par rapport à  $\mathcal{T}$  est définie comme suit :

$$Cl_{\mathcal{T}}(\mathcal{A}_i) = \{B(a) : \langle \mathcal{T}_p, \mathcal{A}_i \rangle \models B(a), B \in D_C, a \in D_{\mathcal{I}}\}$$

$$\cup$$

$$R(a, b) : \langle \mathcal{T}_p, \mathcal{A}_i \rangle \models R(a, b), R \in D_R, (a, b) \in D_{\mathcal{I}}\}.$$

— La fermeture déductive de MBox est défini comme suit :

$$ocl(\mathcal{M}) = \{Cl_{\mathcal{T}}(\mathcal{A}_i) : \mathcal{A}_i \in \mathcal{M}\}.$$

**Exemple 5.** Si  $\mathcal{K}_{\mathcal{M}} = \langle \mathcal{T}, \mathcal{M} \rangle$  une base de connaissances avec  $\mathcal{T} = \{B \sqsubseteq C, \exists R \sqsubseteq P\}$ , et  $\mathcal{M} = \{\{B(a), R(a, b)\}, \{C(b)\}\}$ . La fermeture déductive de  $\mathcal{M}$  est :  $ocl(\mathcal{M}) = \{Cl_{\mathcal{T}}(\{B(a), R(a, b)\}) \cup Cl_{\mathcal{T}}(\{C(b)\})\} = \{B(a), C(a), R(a, b), P(a, b), C(b)\}$ . ■

### 2.4.2 Stratégies d'inférence sur MBox

Une stratégie d'inférence prend comme entrée un MBox  $\mathcal{M}$ , un TBox, et une requête. Il vise à trouver s'il existe une réponse pour cette requête sur l'ensemble des ABoxes de la base de connaissances MBox *DL-Lite*. Plusieurs stratégies d'inférence peuvent être appliquées sur une base de connaissances MBox *DL-Lite*  $\mathcal{K}_{\mathcal{M}} = \langle \mathcal{T}, \mathcal{M} \rangle$ . Les inférences les plus connues sont : *Universal inference*, *Existential inference* [67], *Majority inference* [94] [69] [95], *Cardinality inference* [9], et *safe inference* [96].

#### 2.4.2.1 Inférence universelle

**Définition 3.** On considère  $\mathcal{K}_{\mathcal{M}} = \langle \mathcal{T}, \mathcal{M} \rangle$  une base de connaissances avec MBox  $\mathcal{M} = \{\mathcal{A}_1 \cdots \mathcal{A}_n\}$ . Une requête  $q$  est considérée comme une inférence universelle de  $\mathcal{K}_{\mathcal{M}}$ , notée par  $\mathcal{K}_{\mathcal{M}} \models_{\forall} q$ , si et seulement si  $\forall \mathcal{A}_i \in \mathcal{M}, \langle \mathcal{T}, \mathcal{A}_i \rangle \models^3 q$ .

---

3.  $\models$  est la conséquence standard utilisée dans la base de connaissances *DL-Lite* inconsistante [3]

Cette stratégie basée sur l'inférence universelle stipule qu'une conclusion est valable si et seulement si elle peut être obtenue à partir de toutes les ABoxs  $\mathcal{A}_i$  d'un MBox  $\mathcal{M}$  donné. L'inférence universelle est un moyen standard de dériver des conclusions de différentes sources. Elle est appelée aussi : *inférence sceptique*, utilisée par exemple dans le raisonnement par défaut dans laquelle il n'accepte que les conclusions dérivées de chaque source d'information [97].

Autrement dit, l'utilisation de la relation de conséquence universelle, répond à une requête  $q$  à partir d'une base de connaissances DL-Lite MBox  $\mathcal{K}_{\mathcal{M}}$ , est définie comme l'intersection des réponses de  $q$  obtenues à partir de chaque base de connaissances DL-Lite standard  $\mathcal{K}_i$  de  $\mathcal{K}_{\mathcal{M}}$ .

**Exemple 6.** Si  $\mathcal{K}_{\mathcal{M}} = \langle \mathcal{T}, \mathcal{M} \rangle$  une base de connaissances avec  $\mathcal{T} = \{B \sqsubseteq C, C \sqsubseteq \neg D\}$ , et  $\mathcal{M} = \{\{B(a), D(b)\}, \{C(a)\}\}$ . On peut vérifier que  $\mathcal{K}_{\mathcal{M}} \models_{\forall} C(a)$  ; car :

—  $\langle \mathcal{T}, \{B(a), D(b)\} \rangle \models C(a)$ . et  $\langle \mathcal{T}, \{C(a)\} \rangle \models C(a)$ . ■

#### 2.4.2.2 Inférence existentielle

**Définition 4.** On considère  $\mathcal{K}_{\mathcal{M}} = \langle \mathcal{T}, \mathcal{M} \rangle$  une base de connaissances avec MBox  $\mathcal{M} = \{\mathcal{A}_1 \cdots \mathcal{A}_n\}$ . Une requête  $q$  est considérée comme une inférence existentielle de  $\mathcal{K}_{\mathcal{M}}$ , notée par  $\mathcal{K}_{\mathcal{M}} \models_{\exists} q$ , si et seulement si  $\exists \mathcal{A}_i \in \mathcal{M}, \langle \mathcal{T}, \mathcal{A}_i \rangle \models q$ .

La stratégie d'inférence existentielle est une stratégie d'inférence qui ne vérifie que si une conclusion existe dans au moins un ABox d'un MBox donné.

L'inférence existentielle, appelée aussi : *inférence crédule*, est une relation d'inférence très aventureuse, qui peut conduire à un ensemble de conclusions inconsistantes.

**Exemple 7.** On considère  $\mathcal{K}_{\mathcal{M}} = \langle \mathcal{T}, \mathcal{M} \rangle$  une base de connaissances avec  $\mathcal{T} = \{B \sqsubseteq C, C \sqsubseteq D\}$ , et  $\mathcal{M} = \{\{B(a), D(b)\}, \{C(a), D(a)\}\}$ .

On peut vérifier que  $\mathcal{K}_{\mathcal{M}} \models_{\exists} D(b)$  ; car :  $\langle \mathcal{T}, \{B(a), D(b)\} \rangle \models D(b)$  ■

#### 2.4.2.3 Inférence Saine

**Définition 5.** On considère  $\mathcal{K}_{\mathcal{M}} = \langle \mathcal{T}, \mathcal{M} \rangle$  une base de connaissances avec MBox  $\mathcal{M} = \{\mathcal{A}_1 \cdots \mathcal{A}_n\}$ . Une requête  $q$  est considérée comme une saine inférence de  $\mathcal{K}_{\mathcal{M}}$ , notée par  $\mathcal{K}_{\mathcal{M}} \models_{\cap} q$ , si et seulement si  $\langle \mathcal{T}, \bigcap_{\mathcal{A}_i \in \mathcal{M}} \mathcal{A}_i \rangle \models q$ .

Cette stratégie d'inférence considère les conclusions valides celles qui sont dérivées à l'intersection de toutes les ABoxes. Évidemment, l'inférence saine est une relation d'inférence très conservatrice puisqu'elle considère uniquement les assertions communes entre les différents ABoxes pour effectuer des inférences.

**Exemple 8.** On considère  $\mathcal{K}_{\mathcal{M}} = \langle \mathcal{T}, \mathcal{M} \rangle$  une base de connaissances avec  $\mathcal{T} = \{B \sqsubseteq C, D \sqsubseteq B\}$ , et  $\mathcal{M} = \{\{B(a), D(b)\}, \{C(a), D(a)\}, \{B(a)\}\}$ .

On a :  $\bigcap_{\mathcal{A}_i \in \mathcal{M}} \mathcal{A}_i = B(a)$ .

Donc,  $\mathcal{K}_{\mathcal{M}} \models_{\cap} B(a)$ . ■

D'autres stratégies d'inférence supplémentaires offrent un bon compromis entre les inférences universelles, existentielles et les inférences saine. Ces relations d'inférence appelées : inférence basée sur la majorité (Majority inference), inférence basée sur la proportionnalité (Cardinality inference), et inférence sans objection (non objection inference).

#### 2.4.2.4 Inférence basée sur la majorité

**Définition 6.** On considère  $\mathcal{K}_{\mathcal{M}} = \langle \mathcal{T}, \mathcal{M} \rangle$  une base de connaissances avec  $MBox \mathcal{M} = \{\mathcal{A}_1 \cdots \mathcal{A}_n\}$ . Une requête  $q$  est considérée comme une inférence basée sur la majorité de  $\mathcal{K}_{\mathcal{M}}$ , notée par  $\mathcal{K}_{\mathcal{M}} \models_{maj} q$ , si et seulement si :

$$\frac{|\mathcal{A}_i : \mathcal{A}_i \in \mathcal{M}, \langle \mathcal{T}, \mathcal{A}_i \rangle \models q|}{|\mathcal{M}|} > \frac{1}{2}$$

L'inférence basée sur la majorité (*maj* pour abrégé) considère une conclusion valide si elle est confirmée par la majorité des ABoxes. C'est à dire, elle déduit de plus de la moitié des ABoxes dans  $\mathcal{M}$ .

**Exemple 9.** On considère  $\mathcal{K}_{\mathcal{M}} = \langle \mathcal{T}, \mathcal{M} \rangle$  une base de connaissances avec  $\mathcal{T} = \{B \sqsubseteq C, B \sqsubseteq \neg D\}$ , et  $\mathcal{M} = \{\{B(a), D(b)\}, \{C(a), D(b)\}, \{B(b)\}\}$ .

On a :  $|\mathcal{M}| = 3$ , et  $\langle \mathcal{T}, \{B(a), D(b)\} \rangle \models C(a)$ ,  $\langle \mathcal{T}, \{C(a), D(b)\} \rangle \models C(a)$ ,

$\langle \mathcal{T}, \{B(b)\} \rangle \not\models C(a)$ . Donc,  $\mathcal{K}_{\mathcal{M}} \models_{maj} C(a)$ . ■

### 2.4.2.5 Inférence basée sur la proportionnalité

**Définition 7.** On considère  $\mathcal{K}_{\mathcal{M}} = \langle \mathcal{T}, \mathcal{M} \rangle$  une base de connaissances avec MBox  $\mathcal{M} = \{\mathcal{A}_1 \cdots \mathcal{A}_n\}$ , et  $0 \leq \alpha \leq 1$ . Une requête  $q$  est considérée comme une inférence  $\alpha$ -proportionnelle de  $\mathcal{K}_{\mathcal{M}}$ , notée par  $\mathcal{K}_{\mathcal{M}} \models_{\alpha} q$ , si et seulement si :

$$\frac{|\mathcal{A}_i : \mathcal{A}_i \in \mathcal{M}, \langle \mathcal{T}, \mathcal{A}_i \rangle \models q|}{|\mathcal{M}|} > \alpha$$

L'inférence basée sur la proportionnalité (*prop* pour abrégé) exige qu'une conclusion soit valide si elle peut être dérivée d'une proportion  $\alpha$  de ABox d'une MBox.

Dans cette inférence, trois cas distingués selon le paramètre  $\alpha$  :

- Si  $\alpha = 1$ . Alors, nous récupérons l'inférence universelle.
- Si  $\alpha > \frac{1}{2}$ . Alors, nous récupérons l'inférence basée sur la majorité.
- Si  $\alpha = \text{const.}$  Alors, nous récupérons l'inférence existentielle.

**Exemple 10.** On considère  $\mathcal{K}_{\mathcal{M}} = \langle \mathcal{T}, \mathcal{M} \rangle$  une base de connaissances avec  $\mathcal{T} = \{B \sqsubseteq C, B \sqsubseteq \neg D\}$ , et  $\mathcal{M} = \{\{B(b), D(b)\}, \{C(a), D(b)\}, \{B(b)\}, \{C(b)\}\}$ . Si  $\alpha = 3/4$  On a :  $|\mathcal{M}| = 4$ , et  $\langle \mathcal{T}, \{B(b), D(b)\} \rangle \models C(b)$ ,  $\langle \mathcal{T}, \{B(b)\} \rangle \models C(b)$ , et  $\langle \mathcal{T}, \{C(b)\} \rangle \not\models C(b)$ . Donc,  $\mathcal{K}_{\mathcal{M}} \models_{\alpha=3/4} C(b)$ . ■

### 2.4.2.6 Inférence sans objection

**Définition 8.** On considère  $\mathcal{K}_{\mathcal{M}} = \langle \mathcal{T}, \mathcal{M} \rangle$  une base de connaissances avec MBox  $\mathcal{M} = \{\mathcal{A}_1 \cdots \mathcal{A}_n\}$ . Une requête  $q$  est considérée comme une inférence de sans objection de  $\mathcal{K}_{\mathcal{M}}$ , notée par  $\mathcal{K}_{\mathcal{M}} \models_{\text{obj}} q$ , si et seulement si :

- $\exists \mathcal{A}_i \in \mathcal{M} : \langle \mathcal{T}, \mathcal{A}_i \rangle \models q$ . et
- $\nexists \mathcal{A}_j \in \mathcal{M} : \langle \mathcal{T}, \mathcal{A}_j \rangle \models \neg q$ .

L'inférence sans objection est une stratégie d'inférence où une conclusion  $q$  est acceptée si elle peut être obtenue en utilisant au moins un ABox, mais il n'y a pas un ABox implique  $\neg q$ .

**Exemple 11.** On considère  $\mathcal{K}_{\mathcal{M}} = \langle \mathcal{T}, \mathcal{M} \rangle$  une base de connaissances avec  $\mathcal{T} = \{B \sqsubseteq C, C \sqsubseteq D\}$ , et  $\mathcal{M} = \{\{B(a), D(b)\}, \{C(a), D(b)\}, \{C(b)\}\}$ . On peut vérifier que :  $\mathcal{K}_{\mathcal{M}} \models_{\text{obj}} C(a)$ . ■

### 2.4.2.7 Comparaison entre les stratégies d'inférence sur MBox

Intuitivement, si  $q$  est une formule de premier ordre, l'inférence de sans objection est plus prudente que la relation d'inférence existentielle, mais elle est plus productive que la relation d'inférence universelle.

D'après les Définitions 4 à 8, les propositions suivantes résument les relations entre les différentes stratégies d'inférence sur une base de connaissances *DL-Lite* MBox :

**PROPOSITION 1.** *On considère  $\mathcal{K}_{\mathcal{M}} = \langle \mathcal{T}, \mathcal{M} \rangle$  une base de connaissances avec MBox  $\mathcal{M} = \{\mathcal{A}_1 \cdots \mathcal{A}_n\}$ , et  $q$  une requête, Donc :*

- i) *Si  $\langle \mathcal{T}, \mathcal{M} \rangle \models_{\cap} q$  alors  $\langle \mathcal{T}, \mathcal{M} \rangle \models_{\forall} q$ .*
- ii) *Si  $\langle \mathcal{T}, \mathcal{M} \rangle \models_{\forall} q$  alors  $\langle \mathcal{T}, \mathcal{M} \rangle \models_{maj} q$ .*

La démonstration de cette proposition à été donné dans [123] :

- i) Il a vérifié que  $\mathcal{A}_i \in \mathcal{M}$ . Donc,  $(\bigcap_{\mathcal{A}_i \in \mathcal{M}} \mathcal{A}_i) \subseteq \mathcal{A}_i$ .
- ii) Il a vérifié que l'inférence universelle requiert la requête  $q$  soit valable pour toutes les ABoxes dans  $\mathcal{M}$ . Donc,  $q$  est valable pour au moins la moitié de  $\mathcal{A}_i \in \mathcal{M}$ .

Des contre-exemples de cette proposition 1 sont donnés dans l'exemple 12 :

**Exemple 12.**

- i) *On considère  $\mathcal{K}_{\mathcal{M}} = \langle \mathcal{T}, \mathcal{M} \rangle$  une base de connaissances avec  $\mathcal{T} = \{B \sqsubseteq C, C \sqsubseteq D\}$ , et  $\mathcal{M} = \{\{B(a), D(b)\}, \{D(a)\}, \{B(a)\}\}$ .  
On peut vérifier que :  $\mathcal{K}_{\mathcal{M}} \models_{\forall} D(a)$ , mais  $\mathcal{K}_{\mathcal{M}} \not\models_{\cap} D(a)$ .*
- ii) *On considère  $\mathcal{K}_{\mathcal{M}} = \langle \mathcal{T}, \mathcal{M} \rangle$  une base de connaissances avec  $\mathcal{T} = \{C \sqsubseteq B, C \sqsubseteq \neg D\}$ , et  $\mathcal{M} = \{\{B(a), D(b)\}, \{C(a)\}, \{D(b)\}\}$ .  
On peut vérifier que :  $\mathcal{K}_{\mathcal{M}} \models_{maj} B(a)$ , mais  $\mathcal{K}_{\mathcal{M}} \not\models_{\forall} B(a)$ , car :  
 $\mathcal{K}_{\mathcal{M}} = \langle \mathcal{T}, \{D(b)\} \rangle \not\models B(a)$ . ■*

La Proposition 2 donne les relations entre les différentes stratégies d'inférence à partir de deux MBox où l'un est inclus dans l'autre :

**PROPOSITION 2.** *On considère  $\mathcal{K}_{\mathcal{M}_1} = \langle \mathcal{T}, \mathcal{M}_1 \rangle$  et  $\mathcal{K}_{\mathcal{M}_2} = \langle \mathcal{T}, \mathcal{M}_2 \rangle$  deux bases de connaissances MBox DL-Lite, avec  $\mathcal{M}_1 \subseteq \mathcal{M}_2$ ,  $q$  une requête, Donc :*

- i) *Si  $\langle \mathcal{T}, \mathcal{M}_2 \rangle \models_{\forall} q$  alors  $\langle \mathcal{T}, \mathcal{M}_1 \rangle \models_{\forall} q$ .*
- ii) *Si  $\langle \mathcal{T}, \mathcal{M}_2 \rangle \models_{\cap} q$  alors  $\langle \mathcal{T}, \mathcal{M}_1 \rangle \models_{\cap} q$ .*
- iii)  *$\langle \mathcal{T}, \mathcal{M}_2 \rangle \models_{maj} q$  et  $\langle \mathcal{T}, \mathcal{M}_1 \rangle \models_{maj} q$  sont incomparables.*

La démonstration de cette proposition à été donné dans [123] :

i)  $\mathcal{M}_1 \subseteq \mathcal{M}_2$  signifie que  $\forall \mathcal{A}_i \in \mathcal{M}_1$ , on a  $\mathcal{A}_i \in \mathcal{M}_2$ . Donc, si  $q$  valable avec tous  $\mathcal{A}_i$  de  $\mathcal{M}_1$ , alors il est valable avec tous  $\mathcal{A}_j$  de  $\mathcal{M}_2$ .

ii) Tan que  $\mathcal{M}_1 \subseteq \mathcal{M}_2$ , alors  $\bigcap_{\mathcal{A}_i \in \mathcal{M}_1} \mathcal{A}_i \subseteq \bigcap_{\mathcal{A}_j \in \mathcal{M}_2} \mathcal{A}_j$ .

Des contre-exemples pour cette proposition 2 sont donnés dans l'exemple 13 :

**Exemple 13.**

i) et ii) On considère  $\mathcal{K}_{\mathcal{M}_1} = \langle \mathcal{T}, \mathcal{M}_1 \rangle$ ,  $\mathcal{M}_1 = \{B(a), C(a)\}$ , et  $\mathcal{K}_{\mathcal{M}_2} = \langle \mathcal{T}, \mathcal{M}_2 \rangle$ ,  $\mathcal{M}_2 = \{\{B(b), C(b)\}, \{C(b)\}, \{C(b), C(a)\}\}$ .  $\mathcal{T} = \{B \sqsubseteq C\}$ . Tout d'abord, noter que :  $\mathcal{M}_1 \subseteq \mathcal{M}_2$ . On peut vérifier clairement,  $\langle \mathcal{T}, \mathcal{M}_2 \rangle \models_{\forall} B(b)$ ,  $\langle \mathcal{T}, \mathcal{M}_2 \rangle \models_{\cap} C(b)$ . Mais,  $\langle \mathcal{T}, \mathcal{M}_1 \rangle \not\models_{\forall} B(b)$ , et  $\langle \mathcal{T}, \mathcal{M}_1 \rangle \not\models_{\cap} C(b)$ .

iii) Concernant l'inférence basée sur la majorité, On peut vérifier que :

$\langle \mathcal{T}, \mathcal{M}_1 \rangle \models_{maj} B(a)$ , mais  $\langle \mathcal{T}, \mathcal{M}_2 \rangle \not\models_{maj} B(a)$ . D'autre part,  $\langle \mathcal{T}, \mathcal{M}_2 \rangle \models_{maj} C(b)$ , mais  $\langle \mathcal{T}, \mathcal{M}_1 \rangle \not\models_{maj} C(b)$ . ■

La Proposition 3 donne les relations entre l'inférence sur la base MBox et sa fermeture déductive  $ocl(\mathcal{M})$  :

**PROPOSITION 3.** On considère  $\mathcal{M}_1$  et  $\mathcal{M}_2$  deux bases de connaissances MBox DL-Lite,  $\mathcal{M}_1 \subseteq \mathcal{M}_2$ , sachant que  $\mathcal{M}_2$  la fermeture déductive de  $\mathcal{M}_1$ .  $q$  une requête, Donc :

i)  $\langle \mathcal{T}, \mathcal{M}_1 \rangle \models_{\forall} q$  si et seulement si  $\langle \mathcal{T}, \mathcal{M}_2 \rangle \models_{\forall} q$ .

ii)  $\langle \mathcal{T}, \mathcal{M}_1 \rangle \models_{maj} q$  si et seulement si  $\langle \mathcal{T}, \mathcal{M}_2 \rangle \models_{maj} q$ .

iii)  $\langle \mathcal{T}, \mathcal{M}_1 \rangle \models_{\cap} q$  si et seulement si  $\langle \mathcal{T}, \mathcal{M}_2 \rangle \models_{\cap} q$ .

La démonstration de cette proposition à été donné dans [123] :

i) et ii) La démonstration consiste à suivre le fait que dans DL-Lite standard. Si  $\mathcal{A}$  est une consistante ABox respect  $\mathcal{T}$ , alors  $\langle \mathcal{T}, \mathcal{A} \rangle \models q$  si et seulement si  $\langle \mathcal{T}, Cl_{\mathcal{T}}(\mathcal{A}) \rangle \models q$ .

iii) Suivre le fait que  $\mathcal{A}_i \subseteq Cl_{\mathcal{T}}(\mathcal{A}_i)$  pour tous  $\mathcal{A}_i \in \mathcal{M}_1$ . Donc,

$$\bigcap_{\mathcal{A}_i \in \mathcal{M}_1} \mathcal{A}_i \subseteq \bigcap_{\mathcal{A}_i \in \mathcal{M}_1} Cl_{\mathcal{T}}(\mathcal{A}_i) = \bigcap_{\mathcal{A}_j \in \mathcal{M}_2} \mathcal{A}_j.$$

La Figure 2.2 représente les relations entre les différentes stratégies d'inférence sur les bases de connaissances MBox DL-Lite. L'inférence la plus conservatrice est l'inférence Safe. L'inférence existentielle est l'inférence la plus productive.

Dans la Figure 2.2,  $X \longrightarrow Y$  signifie que chaque conclusion de  $X$  est également une conclusion de  $Y$ .

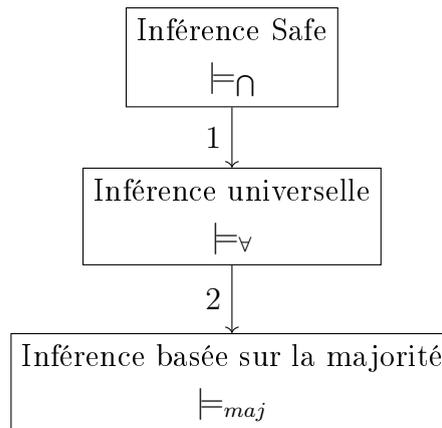


FIGURE 2.2 – Comparaison entre les stratégies d'inférence sur MBox.

Dans la figure 2.2, les preuves des relations d'inclusion 1 et 2 sont données dans la Proposition 1. Des contre-exemples pour l'inverse des relations 1, 2 sont données dans l'exemple 12.

## 2.5 Autres Travaux

L'inconsistance dans la manipulation sur les ontologies est un problème très important qui a reçu une attention particulière (les travaux sur le débogage ou la réparation des connaissances) au cours des dernières années [98], [99], [65].

La sélection d'un sous-ensemble est une conséquence de traitement des bases de connaissances inconsistantes. La base de connaissances résultante ne contiendra pas de formules qui ne sont pas présentées dans la base de connaissances d'origine. En travaillant uniquement avec des informations explicitement fournies, c'est le point de vue de Hansson [100].

Le travail de [18] est orienté vers la gestion des inconsistances dans un contexte de OBDA. Quatre sémantiques tolérantes aux inconsistances appelées : *AR*, *IAR*, *CAR* et *ICAR* ont été proposées. Une sémantique tolérante à l'inconsistance correspond à notre travail est suivi d'une stratégie d'inférence. En plus, une nouvelle sémantique, appelée *ICR*, a été donnée comme une approximation pour la sémantique *AR*.

Dans [19], deux paramètres de tolérance d'incohérence sémantiques, appelées : *k-support* et *k-defeater*, ont été étudiées pour les logiques *DL-Lite<sub>horn</sub>* et *DL-Lite<sub>core</sub>* qui sont montrés pour la vérification de l'instance. En terme de complexité, le calcul de complexité dans le cadre *DL-Lite<sub>core</sub>* est NLSpace, et pour la sémantique *k-support* et pour *k-defeater* est NP.

Une autre famille de paramètres sémantiques, appelée sémantique *k-paresseuse*, a été proposée dans [101]. Cependant, comme il est mentionné dans [19]. Ces sémantiques ne sont pas une approximation de la réponse de requête cohérente.

Dans [102], une approche de raisonnement sémantique à quatre valeurs, a été proposée pour raisonner dans des bases de connaissances *DL-Lite* inconsistantes.

## 2.6 Conclusion

Les paramètres de projet OBDA étudient comment interroger un ensemble des sources de données indépendantes à l'aide d'une vue ontologique unifiée. Nous nous situons dans le contexte de la gestion spécifique lorsque les sources de données soient réunies sont incompatibles.

Ce chapitre produit une feuille de route des différentes caractéristiques techniques d'une base de connaissances *DL-Lite* avec un ABox multiple. De plus, de nombreuses stratégies d'inférence sont discutées.

Le reste de cette thèse étudie ce problème d'inconsistance dans le cas où les bases de connaissances *DL-Lite* seront ordonnées par des priorités.

# Chapitre 3

## Gestion d'inconsistance dans *DL-Lite* avec priorités

*"What a computer is to me is the most remarkable tool that we have ever come up with. It's the equivalent of a bicycle for our minds."*

*Steve Jobs.*

### 3.1 Introduction

Le problème d'inconsistance dans les bases de connaissances *DL-Lite* se produit souvent lorsque les assertions sont fournies par plusieurs sources conflictuelles ayant des niveaux de fiabilité différents. Nous proposons différentes stratégies d'inférence basées sur la sélection d'une base d'assertions consistante, appelée *réparation préférée*. Pour chaque stratégie, un algorithme polynomial pour calculer la réparation préférée unique a été proposé. La sélection d'une réparation unique est importante car elle permet un traitement efficace des requêtes. Il est important de noter que certaines relations d'inférence sont spécifiques à *DL-Lite*, même si elles sont inspirées par d'autres formalismes tel que la logique propositionnelle avec des paramètres.

Ce chapitre étudie les différentes approches pour traiter les bases de connaissances inconsistantes *DL-Lite* dans le cas où la base d'assertions est prioritisée et incompatible avec la base terminologique.

D'abord, nous avons résumé dans le Tableau 3.1 les différentes notations de réparation qui seront utilisées dans ce chapitre.

Abréviation	Définition	Signification
$MAR(\mathcal{A})$	9	Réparation maximale d'inclusion de $\mathcal{A}$
$\mathcal{C}(\mathcal{A})$	10	Ensemble de conflits dans $\mathcal{A}$
$free(\mathcal{A})$	11	Ensemble des éléments libres dans $\mathcal{A}$
$cl(\mathcal{A})$	12	La clôture prioritaire de $\mathcal{A}$
$lcl(\mathcal{A})$	14	La clôture prioritaire local de $\mathcal{A}$
$Consrank(\mathcal{K})$	15	La degré de consistance de $\mathcal{K}$
$\pi(\mathcal{A})$	15	Réparation à base de la théorie des possibilités
$\ell(\mathcal{A})$	16	Réparation à base linéaire
$nd(\mathcal{A})$	17	Réparation à base de d'assertions libres prioritaires
$clnd(\mathcal{A})$	18	Réparation à base de clôture d'assertions libres prioritaires
$lnd(\mathcal{A})$	19	Réparation à base linéaire d'assertions libres prioritaires

TABLE 3.1 – Notations de réparations utilisées.

## 3.2 Base de connaissances *DL-Lite* avec priorité

Une base de connaissances *DL-Lite* avec priorité (la priorité est dans la bases d'assertions ABox) est l'ensemble des assertions noté par :  $\mathcal{A}=(\mathcal{S}_1, \dots, \mathcal{S}_n)$ . Les ensembles  $\mathcal{S}_i$  sont appelés couches ou strates. Chaque couche  $\mathcal{S}_i$  contient l'ensemble des assertions ayant le même niveau de priorité  $i$  et elles sont considérées comme les plus fiables que celles présentes dans une couche  $\mathcal{S}_j$  lorsque  $j > i$ . Par conséquent,  $\mathcal{S}_1$  contient les assertions les plus importantes, tandis que,  $\mathcal{S}_n$  contient les assertions les moins importantes.

Dans le reste de ce chapitre et dans le chapitre suivant et quand il n'y a pas d'ambiguïté, nous utilisons simplement la notation de *DL-Lite*  $\mathcal{K}=\langle \mathcal{T}, \mathcal{A} \rangle$  pour référer une base de connaissances *DL-Lite* avec priorité de la forme :  $\mathcal{A}=(\mathcal{S}_1, \dots, \mathcal{S}_n)$ .

### 3.2.1 Approches de tolérance aux inconsistances dans DL-Lite avec priorité

### 3.2.2 Le concept de réparations

Dans le cadre OBDA (*Ontology Based Data Access*), et lorsque  $\mathcal{T}$  est cohérente, avant de calculer l'inconsistance de  $\mathcal{K}=\langle\mathcal{T}, \mathcal{A}\rangle$ , il faut d'abord calculer les sous-ensembles consistants des assertions (pas nécessairement maximales), appelées *réparations*, avant de les utiliser pour effectuer l'inférence (c'est-à-dire la réponse à la requête). Formellement, une réparation est définie comme suit :

**Définition 9.** *On considère  $\mathcal{K}=\langle\mathcal{T}, \mathcal{A}\rangle$  une base de connaissances inconsistant DL-Lite avec priorité;  $\mathcal{A} = (\mathcal{S}_1, \dots, \mathcal{S}_n)$ .*

*Un sous-ensemble  $\mathcal{R} \subseteq (\mathcal{S}_1 \cup \dots \cup \mathcal{S}_n)$  est dit une réparation si  $\langle\mathcal{T}, \mathcal{R}\rangle$  est consistant.  $\mathcal{R}$  est dit une réparation maximale (pour l'inclusion) de  $\mathcal{K}$ , noté par *MAR*, si  $\langle\mathcal{T}, \mathcal{R}\rangle$  est consistant et  $\forall \mathcal{R}' \subseteq (\mathcal{S}_1 \cup \dots \cup \mathcal{S}_n) : \mathcal{R} \subsetneq \mathcal{R}', \langle\mathcal{T}, \mathcal{R}'\rangle$  est inconsistant.*

Selon la Définition 9, l'ajout d'une assertion  $f$  de  $(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_n) \setminus \mathcal{R}$  à  $\mathcal{R}$  implique l'inconsistance de  $\langle\mathcal{T}, \mathcal{R} \cup \{f\}\rangle$ . La réparation maximale d'inclusion *MAR* est utilisée dans le sens de l'inclusion d'ensemble. Nous désignons par *MAR*( $\mathcal{A}$ ) l'ensemble de *MAR* de  $\mathcal{A}$  par rapport à  $\mathcal{T}$ . La définition de *MAR* coïncide avec la définition de réparation des ABoxes proposée dans [16].

L'utilisation de la notion de réparation, pour calculer l'inconsistance dans les bases de connaissances *DL-Lite* peut être réalisée en appliquant la réponse de requête standard soit en utilisant l'ensemble des réparations (implication universelle) ou *AR*-conséquence [16] ou seulement en utilisant une réparation.

### 3.2.3 L'ensemble de Conflit et l'ensemble libre

Un conflit est un sous-ensemble minimal  $\mathcal{C}$  d'assertions de  $\mathcal{A}$  tel que  $\mathcal{K}=\langle\mathcal{T}, \mathcal{C}\rangle$  est inconsistant.

**Définition 10.** *On considère  $\mathcal{K}=\langle\mathcal{T}, \mathcal{A}\rangle$  une base de connaissances DL-Lite avec priorité. Un sous-ensemble  $\mathcal{C} \subseteq \mathcal{A}$  est dit un conflit d'assertions de  $\mathcal{K}$  si et seulement si  $\langle\mathcal{T}, \mathcal{C}\rangle$  est inconsistant et  $\forall f \in \mathcal{C}, \langle\mathcal{T}, \mathcal{C} \setminus \{f\}\rangle$  est consistant.*

Selon la Définition 10, la suppression d'une instance  $f$  de  $\mathcal{C}$  restaure la consistance de  $\langle \mathcal{T}, \mathcal{C} \rangle$ . Dans *DL-Lite* avec un TBox cohérente, un conflit nécessite exactement deux assertions [16]. Nous notons par  $\mathcal{C}(\mathcal{A})$  l'ensemble de conflits dans  $\mathcal{A}$ . On note aussi par *Conf* la complexité de calcul de l'ensemble des conflits  $\mathcal{C}(\mathcal{A})$  d'un standard *DL-Lite*. Pour vérifier si  $\langle \mathcal{T}, \mathcal{A} \rangle$  est consistant ou non, revient de vérifier si  $\mathcal{C}(\mathcal{A})$  est vide ou non.

Une caractéristique intéressante de *DL-Lite* est que le calcul de l'ensemble des conflits se fait en temps polynomial [103]. La définition 11 donne la condition nécessaire pour qu'une assertion soit libre.

**Définition 11.** *On considère  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  une base de connaissances DL-Lite avec priorité. Une assertion  $f \in \mathcal{A}$  est dite libre si et seulement si  $\forall c \in \mathcal{C}(\mathcal{A}) : f \notin c$ .*

Intuitivement, les assertions libres correspondent à des assertions qui ne sont pas impliquées dans aucun conflit. On note par  $free(\mathcal{A})$  l'ensemble d'assertions libres dans  $\mathcal{A}$ . La notion d'éléments libres  $free(\mathcal{A})$  est proposée à l'origine dans le contexte de la logique propositionnelle [96]. Dans le cadre *DL-Lite*,  $free(\mathcal{A})$  est calculée en temps polynomial, car le calcul de conflits se fait également en temps polynomial.

La formulation de la Définition 11 avec l'utilisation d'ensemble des réparations maximales d'inclusion est comme suite :

$$free(\mathcal{A}) = \bigcap_{X \in MAR(\mathcal{A})} X.$$

**Exemple 14.** *On considère  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  une base de connaissance DL-Lite avec priorité. Tel que :  $\mathcal{T} = \{A \sqsubseteq \neg B, A \sqsubseteq D\}$  et  $\mathcal{A} = \{A(b), A(a), D(a), B(a)\}$ .*

*Nous avons,  $MAR_1 = \{A(b), A(a), D(a)\}$ , et  $MAR_2 = \{A(b), B(a), D(a)\}$ .*

*Selon la Définition 11 :  $\mathcal{C}(\mathcal{A}) = \{A(a), B(a)\}$ .*

*Par conséquence,  $MAR_1 \cap MAR_2 = \{A(b), D(a)\} = free(\mathcal{A})$ . ■*

Pour les bases de connaissances *DL-Lite*, le *libre-conséquence* (implication basée sur les assertions libres) équivaut à *IAR-conséquence* proposée dans [67]. En fait, dans le contexte de la logique propositionnelle, le concept d'un sous-ensemble maximal consistant a été introduit avant le concept des réparations [104]. Par ailleurs, le concept de *libre-conséquence* a été introduit dans [96].

L'algorithme 2 représente la fonction qui calcule l'ensemble des éléments libres  $free(\mathcal{A})$  dans un ensemble d'assertions  $X$  (elle donne un ensemble de conflits  $\mathcal{C}$ ). Cette fonction sera utilisée par les algorithmes développés dans la suite.

---

**Algorithme 2** Ensemble des éléments libres

---

1: **function** FREE( $X, \mathcal{C}$ )

**Input:**  $X$  : Ensemble des assertions

**Input:**  $\mathcal{C}$  : Liste des conflits

**Output:** Liste des éléments libres dans  $X$

2:     **return** ( $X \setminus \{f : f \in X, \exists g \in X \text{ tel que } (f, g) \in \mathcal{C}\}$ )

---

### 3.2.4 Notion de clôture prioritaire

**Définition 12.** On considère  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  une base de connaissances DL-Lite standard.  $\mathcal{T}_p$  l'ensemble de tous les axiomes d'inclusion positifs de  $\mathcal{T}$  ( sous forme  $A \sqsubseteq B$ ). Nous définissons la clôture déductive de  $\mathcal{A}$  comme suit :

$cl(\mathcal{A}) = \{B(a) : \langle \mathcal{T}_p, \mathcal{A} \rangle \models B(a) \text{ tel que, } B \text{ est un concept de } \mathcal{T} \text{ et } a \text{ est un individu de } \mathcal{A}\} \cup \{R(a, b) : \langle \mathcal{T}_p, \mathcal{A} \rangle \models R(a, b), \text{ tel que } R \text{ est un rôle de } \mathcal{T} \text{ et } a, b \text{ sont des individus de } \mathcal{A}\}.$

La définition suivante étend la Définition 12 au cas de ABox avec priorité.

**Définition 13.** On considère  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  une base de connaissances DL-Lite avec priorité. Nous définissons la clôture prioritaire de  $\mathcal{A}$  noté par  $cl(\mathcal{A})$ , comme suit :

$$cl(\mathcal{A}) = (\mathcal{S}'_1, \dots, \mathcal{S}'_n) \text{ telque : } \forall i = 1, \dots, n : \mathcal{S}'_i = cl(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_{i-1} \cup \mathcal{S}_i).$$

La première motivation de la Définition 13 est que si une assertion  $f$  dérive de  $\langle \mathcal{K}, \mathcal{S}_1 \cup \dots \cup \mathcal{S}_n \rangle$  alors  $f$  il doit appartenir à  $cl(\mathcal{A})$ . La deuxième motivation est si  $f$  est dérivé de  $\langle \mathcal{K}, \mathcal{S}_1 \cup \dots \cup \mathcal{S}_i \rangle$  alors,  $\forall j \geq i \quad f \in \mathcal{S}'_j$ .

Il existe une autre alternative a la Définition 13 qui évite la duplication d'ensemble des conclusions dérivées. L'idée est qu'une assertion dérivée  $f$  est ajoutée au rang  $i$  si et seulement si elle peut être obtenue uniquement à partir de  $\mathcal{S}_i$ .

**Définition 14.** On considère  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  une base de connaissances DL-Lite avec priorité. Nous définissons la clôture prioritaire locale de  $\mathcal{A}$  noté par  $lcl(\mathcal{A}) = (\mathcal{S}'_1, \dots, \mathcal{S}'_n)$ , comme suite :

$$\forall i = 1, \dots, n, \quad \mathcal{S}'_i = cl(\mathcal{S}_i).$$

### 3.2.5 Analyse de complexité

L'analyse de complexité consiste à utiliser les résultats de complexité qui sont connus dans la théorie des graphes concernant le problème de l'ensemble indépendant maximal *MIS* (*Maximum Independent Set*). Nous rappelons que la notion de  $k$ -MIS de problème de décision d'un graphe  $G$ , est un ensemble indépendant de taille (au moins)  $k$ . D'après [90], la complexité computationnelle de  $k$ -MIS est connue d'être NP-complet.

**PROPOSITION 4.** *On considère  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  une base de connaissances DL-Lite inconsistante. Si  $R \subseteq \mathcal{A}$  un sous-ensemble de  $\mathcal{A}$  et  $G_R$  l'ensemble des nœuds associés à  $R$ . Alors,  $R$  est un sous-ensemble maximal consistant de  $\mathcal{A}$  si et seulement si  $G_R$  est un ensemble indépendant maximal de  $G$ .*

[123] a donné la démonstration suivante : Si  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  une base de connaissances DL-Lite inconsistante. Avec  $\mathcal{T}$  est cohérente.  $\mathcal{C}(\mathcal{A})$  est l'ensemble de tous les conflits dans  $\mathcal{A}$ .  $R$  une sous-base maximale de  $\mathcal{A}$ , mais  $G_R$  n'est pas un ensemble indépendant maximal de  $G$ . Cela signifie qu'il existe un nœud  $f$  (une assertion de  $\mathcal{A}$ ) tel que  $f \notin G_R$  et  $\forall g \in G_R$ , il n'y a pas d'arc entre  $f$  et  $g$ . Autrement dit, il existe une assertion  $f \in \mathcal{A}$  telle que  $f \notin R$  et  $\forall g \in R$ , il n'y a pas de conflit de la forme  $(f, g)$ . Cela signifie que  $R \cup \{g\}$  soit consistant et cela contredit avec la sous-base  $R$ . Par ailleurs, le graphe  $G$  associé à  $\mathcal{K}$  est défini comme suit :

- L'ensemble des nœuds de  $G$  est simplement l'ensemble des assertions dans  $\mathcal{A}$  (une assertion = un nœud différent).
- Pour tout  $(f, g) \in \mathcal{C}$  (un conflit de  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ ) un arc non orienté est tracé à partir de  $f$  vers  $g$ .

**Exemple 15.** *On considère  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  une base de connaissances avec  $\mathcal{T} = \{B \sqsubseteq C, B \sqsubseteq \neg D\}$ , et  $\mathcal{A} = \{B(a), R(a, b), C(a), D(a), C(b)\}$ .*

*La Figure 3.1 représente le graphe non orienté associé à  $\mathcal{K}$  :*

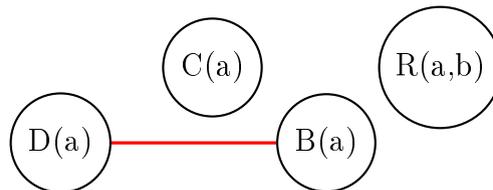


FIGURE 3.1 – Transformation d'une base DL-Lite vers un graphe non orienté. ■

La transformation inverse d'un graphe non orienté  $G$  vers une base de connaissances  $DL-Lite$  inconsistante est définie comme suit :

1. Nous associons à chaque nœud  $e$  un concept également désigné par  $E$  (deux nœuds différents ayant deux concepts associés distincts).
2. Nous utilisons  $a$  comme un individu unique utilisé dans  $\mathcal{A}$ .
3. Pour chaque arc non orienté  $e \longrightarrow f$ , nous ajoutons un axiome négatif ( $E \sqsubseteq \neg F$ ) dans  $\mathcal{T}$ . (TBox associé à  $G$  est défini par :  $\mathcal{T} = \{E \sqsubseteq \neg F : e \longrightarrow f \text{ est un arc de } G\}$ ).
4. La ABox est simplement l'ensemble des nœuds avec le même individu  $a$  ; tel que  $\mathcal{A} = \{E(a) : a \text{ est un individu et } E \text{ est un nœud de } G\}$ .
5. La base de connaissances  $DL-Lite$  associée à un graphe implique seulement un individu ; Il ne contient pas des axiomes positifs. Alors,  $\forall E(a) \in \mathcal{A}$  et  $F(a) \in \mathcal{A}$ ,  $(E(a), F(a)) \in \mathcal{C}(\mathcal{A})$  si et seulement s'il existe un arc non orienté entre  $E$  et  $F$ .

**Exemple 16.** Soit  $G$  un graphe non orienté illustré par la Figure 3.2 :

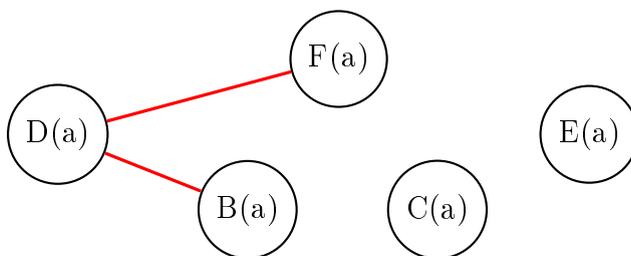


FIGURE 3.2 – Transformation d'un graphe non orienté vers une base  $DL-Lite$ .

La base de connaissances  $DL-Lite$  associée à  $G$  est :  $\mathcal{T} = \{B \sqsubseteq \neg D, D \sqsubseteq \neg F\}$   
 Tel que :  $\mathcal{A} = \{B(a), D(a), C(a), F(a), E(a)\}$ . ■

Par l'utilisation des deux transformations ci-dessus, la Proposition 5 donne la complexité du calcul de la cardinalité de la plus grande sous base maximale de  $\mathcal{A}$ .

**PROPOSITION 5.** On considère  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  une base de connaissances  $DL-Lite$  inconsistante. La complexité du calcul de la cardinalité du plus grand sous-ensemble maximal consistant de  $\mathcal{K}$  est :  $\mathcal{O}(\log_2(|\mathcal{A}|) * (k-MIS))$ .

Pour démontrer cette proposition, il suffit d'appliquer une recherche par dichotomie entre 1 et  $|\mathcal{A}|$ . Et pour chaque valeur  $1 \leq k \leq |\mathcal{A}|$  nous appelons un problème de  $k$ -MIS.

**Exemple 17.** Nous considérons la TBox présenté dans l'exemple 2. Nous supposons qu'il y a une certaine incertitude concernant l'état et les rôles entre les individus. Les informations assertionnelles disponibles sont codées par les ABoxes suivants, qui sont fournies par des sources distinctes  $\mathcal{A} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4, \mathcal{S}_5)$  tel que :

$$\begin{aligned} \mathcal{S}_1 &= \{Student(Bill), Teacher(John)\}, \\ \mathcal{S}_2 &= \{Teacher(Mary), Student(Bob)\}, \\ \mathcal{S}_3 &= \{HasSupervisor(Bob, Bill), Researcher(Joe)\}, \\ \mathcal{S}_4 &= \{Student(Anne)\}, \text{ et} \\ \mathcal{S}_5 &= \{Teacher(Bill)\}. \end{aligned}$$

■

Dans cet exemple,  $\mathcal{S}_1$  contient les assertions les plus fiables, tandis que  $\mathcal{S}_5$  contient les assertions les moins fiables. Il est facile de vérifier que la base de connaissances  $\mathcal{K}$  est inconsistante. Par exemple, dans TBox l'axiome  $\exists HasSupervisor^- \sqsubseteq Teacher$  peut conclure le fait suivant :  $Teacher(Bill)$ . Mais dans ABox, on a les assertions  $Student(Bill)$  et  $HasSupervisor(Bob, Bill)$  cela contredit l'axiome négatif  $Teacher \sqsubseteq \neg Student$ .

Pour traiter cette inconsistance, nous essayerons de sélectionner des sous-ensembles consistants (réparations) à partir de cette base de connaissances inconsistante afin de répondre de manière significative aux requêtes. Pour cette raison, deux sous-bases de réparations maximales des assertions ont été calculées :

$$\begin{aligned} \mathcal{R}_1 &= \{Student(Bill), Teacher(John), Teacher(Mary), \\ &\quad Student(Anne), Student(Bob), Researcher(Joe)\} \\ \mathcal{R}_2 &= \{Teacher(Bill), Teacher(John), Teacher(Mary), \\ &\quad Student(Bob), Student(Anne), Researcher(Joe), \\ &\quad HasSupervisor(Bob, Bill)\}. \end{aligned}$$

En effet, soit nous ignorons l'assertion  $Student(Bill)$ , et alors les assertions restantes sont  $\mathcal{R}_2 = \mathcal{A} \setminus \{Student(Bill)\}$  sont consistantes avec  $\mathcal{T}$ . Soit nous gardons l'assertion  $Student(Bill)$  et dans ce cas nous devons supprimer l'assertion  $Teacher(Bill)$  et  $HasSupervisor(Bob, Bill)$  pour restaurer la consistance de  $\langle \mathcal{T}, \mathcal{A} \rangle$ . Ensuite, nous obtenons  $\mathcal{R}_1$ . ■

D'ailleurs, l'ensemble des conflits est :

$$\mathcal{C}(\mathcal{A}) = \left\{ \left\{ \text{Student}(\text{Bill}), \text{Teacher}(\text{Bill}) \right\}, \right. \\ \left. \left\{ \text{HasSupervisor}(\text{Bob}, \text{Bill}), \text{Student}(\text{Bill}) \right\} \right\}$$

L'ensemble des éléments libres est :

$$\text{free}(\mathcal{A}) = \left\{ \text{Teacher}(\text{John}), \text{Teacher}(\text{Mary}), \right. \\ \left. \text{Student}(\text{Anne}), \text{Admin}(\text{Joe}), \text{Student}(\text{Bob}) \right\}.$$

La clôture prioritaire est  $\text{cl}(\mathcal{A}) = \mathcal{S}'_1 \cup \dots \cup \mathcal{S}'_5$  tel que :

$$\begin{aligned} \mathcal{S}'_1 &= \{ \text{Student}(\text{Bill}), \text{Teacher}(\text{John}) \}, \\ \mathcal{S}'_2 &= \{ \text{Student}(\text{Bill}), \text{Teacher}(\text{John}), \\ &\quad \text{Student}(\text{Bob}), \text{Teacher}(\text{Mary}) \}, \\ \mathcal{S}'_3 &= \{ \text{Student}(\text{Bill}), \text{Teacher}(\text{John}), \\ &\quad \text{Student}(\text{Bob}), \text{Teacher}(\text{Mary}), \\ &\quad \text{HasSupervisor}(\text{Bob}, \text{Bill}), \text{Researcher}(\text{Joe}), \\ &\quad \text{Teacher}(\text{Joe}), \text{Teacher}(\text{Bill}) \}, \\ \mathcal{S}'_4 &= \{ \text{Student}(\text{Bill}), \text{Teacher}(\text{John}), \\ &\quad \text{Student}(\text{Bob}), \text{Teacher}(\text{Mary}), \\ &\quad \text{HasSupervisor}(\text{Bob}, \text{Bill}), \text{Researcher}(\text{Joe}), \\ &\quad \text{Teacher}(\text{Joe}), \text{Teacher}(\text{Bill}), \text{Student}(\text{Anne}) \}, \\ \mathcal{S}'_5 &= \{ \text{Student}(\text{Bill}), \text{Teacher}(\text{John}), \\ &\quad \text{Student}(\text{Bob}), \text{Teacher}(\text{Mary}), \\ &\quad \text{HasSupervisor}(\text{Bob}, \text{Bill}), \text{Researcher}(\text{Joe}), \\ &\quad \text{Teacher}(\text{Joe}), \text{Teacher}(\text{Bill}), \text{Student}(\text{Anne}) \}. \end{aligned}$$

Nous remarquons que  $\text{Teacher}(\text{Joe})$  est une nouvelle assertion qui a été engendrée par l'application de l'axiome  $\text{Researcher} \sqsubseteq \text{Teacher}$  sur l'assertion  $\text{Researcher}(\text{Joe})$ .

■

### 3.3 Stratégies de calcul des réparations dans *DL-Lite* avec priorité

Les approches traitant l'inconsistance dans les bases de connaissances *DL-Lite* avec priorité ont été proposées dans le cadre de la logique propositionnelle, mais ils ont besoin d'une adaptation légère pour être adapté à *DL-Lite*. Nous proposons différentes approches pour traiter le problème de l'inconsistance dans les *DL-Lite*. L'entrée de ces approches est une base de connaissances *DL-Lite* avec priorité  $\mathcal{K}=\langle\mathcal{T},\mathcal{A}\rangle$  tel que  $\mathcal{A}=(\mathcal{S}_1,\dots,\mathcal{S}_n)$ . La sortie de nos approches est une base de connaissances *DL-Lite* standard  $\mathcal{K}'=\langle\mathcal{T},\mathcal{R}\rangle$ , tel que  $\mathcal{R}$  n'est pas prioritaire (juste un ensemble des assertions).  $\mathcal{K}$  et  $\mathcal{K}'$  ont la même base terminologique TBox.  $\mathcal{R}$  sera appelée une réparation préférée. Par conséquence, une requête  $q$  est déduite de  $\mathcal{K}$  si elle peut être dérivée, en utilisant l'inférence standard de *DL-Lite* de  $\mathcal{K}'$ .

#### 3.3.1 Réparation à base de la théorie des possibilités

La théorie des possibilités [105] et la logique possibiliste [106] sont des cadres naturels et intuitifs pour représenter des informations incertaines, incomplètes, qualitatives et prioritaires. L'un des aspects intéressants de la théorie des possibilités est sa capacité de raisonnement avec une connaissance partiellement inconsistante [107]. La réparation à base de la théorie des possibilités est basée sur la sélection d'une sous-base de  $\mathcal{K}$  consistante, mais pas nécessairement maximale [108]. Cette sous-base est induite par un niveau de priorité appelé *le degré de consistance* de  $\mathcal{K}$ .

**Définition 15.** On considère  $\mathcal{K}=\langle\mathcal{T},\mathcal{A}\rangle$  une base de connaissances *DL-Lite* avec priorité. Le degré de consistance de  $\mathcal{K}$ , noté par :  $Consrank(\mathcal{K})$ , est défini comme suit :

$$Consrank(\mathcal{K}) = \begin{cases} 0 & \text{si } \langle\mathcal{T},\mathcal{S}_1\rangle \text{ est inconsistent.} \\ \max \{i : \langle\mathcal{T},(\mathcal{S}_1,\dots,\mathcal{S}_i)\rangle \text{ est consistant}\} & \text{Sinon.} \end{cases}$$

La notion de degré de consistance donnée dans la Définition 15 est liée à la notion de degré de consistance utilisé dans la théorie des possibilités (où les degrés sont encodés en utilisant l'intervalle unitaire  $[0,1]$  au lieu d'une stratification utilisant des nombres entiers positifs). La sous-base  $\pi(\mathcal{A})$  est constituée d'assertions ayant des niveaux de priorité inférieurs ou égaux à  $Consrank(\mathcal{K})$ . Si  $\mathcal{K}$  est consistant alors nous avons considéré que  $\pi(\mathcal{A}) = \mathcal{S}_1 \cup \dots \cup \mathcal{S}_n$ .

---

**Algorithme 3** Réparation à base de la théorie des possibilités

---

**Input:**  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  avec  $\mathcal{A} = (S_1, \dots, S_n)$

**Output:** Une base d'assertions standard  $\pi(\mathcal{A})$

```

1: if  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  est consistant then
2:   return  $\pi(\mathcal{A}) = S_1 \cup \dots \cup S_n$ 
3: else
4:   if  $\mathcal{K} = \langle \mathcal{T}, \mathcal{S}_1 \rangle$  est inconsistant then
5:     return  $\pi(\mathcal{A}) = \emptyset$ 
6:   else
7:      $\ell \leftarrow 1$ 
8:      $u \leftarrow n$ 
9:     while  $(\ell < u)$  do
10:       $\alpha \leftarrow \lfloor \frac{\ell+u}{2} \rfloor$ 
11:      if  $\langle \mathcal{T}, S_1 \cup \dots \cup S_\alpha \rangle$  est consistant then
12:         $\ell \leftarrow \alpha + 1$ 
13:      else
14:         $u \leftarrow \alpha$ 
15:   return  $\pi(\mathcal{A}) = S_1 \cup \dots \cup S_{\alpha-1}$ 

```

---

L'algorithme 3 implémente la Définition 15 et renvoie une sous-base de réparation à base de la théorie des possibilités. C'est l'équivalent de l'algorithme proposé par [106] dans le cadre de la logique propositionnelle proposée par [109] dans le cadre la logique de description.

Comme dans la logique propositionnelle, l'algorithme 3 a besoin de  $\log_2(n)$  tests de consistance, où  $n$  est le nombre de strates dans  $\mathcal{A}$ . Noter que, le calcul des réparations dans la logique propositionnelle standard est une tâche difficile alors qu'il est polynomial dans *DL-Lite* [108]. Par conséquent, l'algorithme 3 renvoie la réparation basée sur la théorie des possibilités en un temps polynomial.

**Exemple 18.** *Nous continuons sur l'exemple 17. Selon l'algorithme 3, on a :*

$\begin{aligned} \pi(\mathcal{A}) &= \mathcal{S}_1 \cup \mathcal{S}_2 \\ &= \{Etudiant(Bill), Enseignant(John), \\ &\quad Enseignant(Mary), Etudiant(Bob)\}. \end{aligned}$
---

*En effet, on peut vérifier que  $\langle \mathcal{T}, \mathcal{S}_1 \cup \mathcal{S}_2 \rangle$  est consistant, mais  $\langle \mathcal{T}, \mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_3 \rangle$  est inconsistant. ■*

La réparation basée sur la théorie des possibilités peut être considérée comme un moyen *saine* pour traiter l'inconsistance, car l'algorithme de réparation s'arrête à la première couche où l'inconsistance est introduite. Par conséquent, seules les assertions ayant un degré inférieur ou égal à celui du degré de consistance sont prises en compte pour tirer des conclusions. Cependant, les assertions ayant des niveaux de priorité strictement supérieurs au degré de consistance sont simplement inhibées même si elles ne sont pas impliquées dans aucun conflit [110].

Pour surmonter cette limitation et fournir des réparations plus productives ou plus grandes, une approche linéaire peut être utilisée pour calculer la réparation dans *DL-Lite* inconsistant.

### 3.3.2 Réparation à base linéaire

Une façon de récupérer les assertions inhibées par la réparation à base de la théorie des possibilités consiste d'utiliser la réparation linéaire à partir de  $\mathcal{A}$ . La conséquence linéaire a été utilisé dans le cadre logique propositionnelle [111]. Il a été appliqué pour la logique de description [109]. La Définition 16 introduit la notion de sous-ensemble linéaire.

**Définition 16.** On considère  $\mathcal{K}=\langle\mathcal{T}, \mathcal{A}\rangle$  une base de connaissances *DL-Lite* avec priorité. La réparation à base linéaire de  $\mathcal{A}$ , noté par :  $\ell(\mathcal{A})= \mathcal{S}'_1 \cup \dots \cup \mathcal{S}'_n$ , est défini comme suit :

- i) Si  $i=1$  :
 
$$\mathcal{S}'_1 = \begin{cases} \mathcal{S}_1 & \text{Si } \langle\mathcal{T}, \mathcal{S}_1\rangle \text{ est consistant} \\ \emptyset & \text{Sinon} \end{cases}$$
- ii) Pour  $i = 2, \dots, n$ 

$$\mathcal{S}'_i = \begin{cases} \mathcal{S}_i & \text{Si } \langle\mathcal{T}, \mathcal{S}'_1 \cup \dots \cup \mathcal{S}'_{i-1} \cup \mathcal{S}_i\rangle \text{ est consistant.} \\ \emptyset & \text{Sinon.} \end{cases}$$

Clairement,  $\ell(\mathcal{A})$  est obtenu en rejetant une couche  $\mathcal{S}_i$  lorsque ses faits entrent en conflit avec les couches précédentes.

L'algorithme 4 implémente la Définition 16. La sous-base  $\ell(\mathcal{A})$  est unique et elle est consistante avec  $\mathcal{T}$ . Le temps de calcul de complexité de  $\ell(\mathcal{A})$  est en P. En effet, d'après l'algorithme 4, le calcul de complexité de  $\ell(\mathcal{A})$  a besoin de  $n$  unités de temps de calcul pour vérifier la consistance de *DL-Lite* standard.

---

**Algorithme 4** Réparation à base linéaire

---

**Input:**  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  avec  $\mathcal{A} = (\mathcal{S}_1, \dots, \mathcal{S}_n)$

**Output:** Une base d'assertions standard  $\ell(\mathcal{A})$

- 1:  $\ell(\mathcal{A}) \leftarrow \emptyset$
  - 2: **for**  $i = 1$  à  $n$  **do**
  - 3:     **if**  $\langle \mathcal{T}, \ell(\mathcal{A}) \cup \mathcal{S}_i \rangle$  est consistante **then**
  - 4:          $\ell(\mathcal{A}) \leftarrow \ell(\mathcal{A}) \cup \mathcal{S}_i$
  - 5: **return**  $\ell(\mathcal{A})$
- 

**Exemple 19.** (Exemple 17 continué). Selon la Définition 16, On a :

$$\begin{aligned} \mathcal{S}'_1 &= \{Student(Bill), Teacher(John)\}, \\ \mathcal{S}'_2 &= \{Teacher(Mary), Student(Bob)\}, \\ \mathcal{S}'_3 &= \emptyset \\ \mathcal{S}'_4 &= \{Student(Anne)\}, \text{ et} \\ \mathcal{S}'_5 &= \emptyset. \\ \text{Donc, } \ell(\mathcal{A}) &= \mathcal{S}'_1 \cup \mathcal{S}'_2 \cup \mathcal{S}'_3 \cup \mathcal{S}'_4 \cup \mathcal{S}'_5. \\ &= \mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_4. \end{aligned}$$

Effectivement, les sous-bases  $\langle \mathcal{T}, \mathcal{S}_1 \cup \mathcal{S}_2 \rangle$  et  $\langle \mathcal{T}, \mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_4 \rangle$  sont consistantes. Cependant, les sous-bases  $\langle \mathcal{T}, \mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_3 \rangle$  et  $\langle \mathcal{T}, \mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_4 \cup \mathcal{S}_5 \rangle$  sont inconsistantes. De plus,  $\ell$ -conséquence est plus productive que  $\pi$ -conséquence comme nous le remarquons dans l'exemple 18. ■

Tandis que,  $\ell$ -conséquence et  $\pi$ -conséquence ne garantissent pas une réparation maximale. Nous présenterons dans la suite trois nouvelles stratégies qui sélectionnent uniquement une réparation préférée. Ces stratégies sont basées sur la notion d'assertions libres prioritaires en ajoutant différents critères.

### 3.3.3 Réparation à base d'assertions libres prioritaires

Une autre façon d'obtenir une réparation préférée est de récupérer interactivement, couche par couche, l'ensemble des éléments libres  $free(\mathcal{A})$ . La Définition 17 donne le calcul d'une réparation à base d'assertions libres prioritaires (*Non-Defeated*).

**Définition 17.** On considère  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  une base de connaissances DL-Lite avec priorité. Nous définissons la réparation à base d'assertions libres prioritaires, notée par  $nd(\mathcal{A})$  comme suite :  $nd(\mathcal{A}) = \mathcal{S}'_1 \cup \dots \cup \mathcal{S}'_n$ , tel que :

$$\forall i = 1, \dots, n \quad \mathcal{S}'_i = free(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_i).$$

Par conséquence :

$$nd(\mathcal{A}) = free(\mathcal{S}_1) \cup free(\mathcal{S}_1 \cup \mathcal{S}_2) \cup \dots \cup free(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_n).$$

Sachant que :  $free(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_i)$  désigne l'ensemble des faits libres en  $(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_i)$ .

La réparation à base d'assertions libres prioritaires est une extension de la base d'assertions libres  $free(\mathcal{A})$  lorsque  $\mathcal{A}$  est prioritaire. La définition d'assertions libres prioritaires est une adaptation de la définition proposée par [112] dans le cadre de la logique propositionnelle. Clairement,  $nd(\mathcal{A})$  ne contient aucun conflit ; elle est consistante. En ce qui concerne la complexité de calcul, le calcul des conflits se fait dans un temps polynomial. Car l'ensemble des assertions  $free$  peut être obtenu en un temps linéaire par rapport à l'ensemble des conflits.

L'algorithme 5 donne la méthode de calcul de la réparation à base d'assertions libres prioritaires. Elle calcule d'abord l'ensemble des conflits (étape 1). L'étape 2 initialise simplement  $nd(\mathcal{A})$  sur un ensemble vide. L'étape 4 calcule l'ensemble des éléments conflictuels dans  $\mathcal{S}_1 \cup \dots \cup \mathcal{S}_i$  puis ajoute ce résultat à  $nd(\mathcal{A})$ .

---

**Algorithme 5** Réparation à base d'assertions libres prioritaires

---

**Input:**  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  avec  $\mathcal{A} = (\mathcal{S}_1, \dots, \mathcal{S}_n)$

**Output:** Une base d'assertions standard  $nd(\mathcal{A})$

- 1:  $\mathcal{C} \leftarrow \mathcal{C}(\mathcal{A})$  { Liste de conflits }
  - 2:  $nd(\mathcal{A}) \leftarrow \emptyset$
  - 3: **for**  $i = 1$  à  $n$  **do**
  - 4:      $nd(\mathcal{A}) \leftarrow nd(\mathcal{A}) \cup FREE(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_i, \mathcal{C})$
  - 5: **return**  $nd(\mathcal{A})$
- 

Effectivement, L'algorithme 5 implique directement la Définition 17. L'ensemble des conflits est calculé une fois dans cet algorithme.

Concernant la complexité de cet algorithme ; elle est égale  $Conf$  (pour l'étape 1) plus  $\mathcal{O}(n)$  (pour les étapes 2-5), où  $n$  est le nombre de strates dans la base de connaissances DL-Lite  $\mathcal{K}$  (c'est à dire  $Conf + \mathcal{O}(n)$ ).

**Exemple 20.** Nous continuons avec l'exemple 17. Suivant l'algorithme 5, on a :

$$\begin{aligned}
 free(\mathcal{S}_1) &= \{Student(Bill), Teacher(John)\}, \\
 free(\mathcal{S}_1 \cup \mathcal{S}_2) &= \{Student(Bill), Teacher(John), Student(Bob), \\
 &\quad Teacher(Mary)\}, \\
 free(\mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_3) &= \{Student(Bob), Teacher(John), Teacher(Mary), \\
 &\quad Researcher(Joe)\}, \\
 free(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_4) &= \{Student(Bob), Teacher(John), Teacher(Mary), \\
 &\quad Student(Anne), Researcher(Joe)\}, \text{ et} \\
 free(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_5) &= \{Teacher(John), Teacher(Mary), Student(Anne), \\
 &\quad Researcher(Joe), Student(Bob)\}.
 \end{aligned}$$

Donc :  $nd(\mathcal{A}) = \{Student(Bill), Teacher(John), Researcher(Joe), Student(Bob), Teacher(Mary), Student(Anne)\}$ . ■

Deux stratégies, basées sur la notion d'assertions libres prioritaires, seront décrites dans les sections suivantes. Elles sont obtenu en ajoutant les critères de la clôture déductive, et de la consistance.

### 3.3.4 Réparation à base de clôture déductive d'assertions libres prioritaires

Quand une réparation d'assertions libres prioritaires est définie sur  $\mathcal{A}$ , elle est *saine* car elle utilise uniquement des éléments de  $\mathcal{A}$  qui ne sont pas impliqués dans un conflit. Cependant, et contrairement au langage propositionnel, la réparation d'assertions libres prioritaires peut être appliquée sur  $\mathcal{A}$  ou sur sa clôture déductive  $cl(\mathcal{A})$  qui conduit à deux manières différentes de sélectionner une seule réparation préférée. Nous définissons la réparation à base de clôture déductive d'assertions libres prioritaires, comme suit :

**Définition 18.** On considère  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  une bases de connaissances DL-Lite avec priorité. La réparation à base de clôture déductive d'assertions libres prioritaires, notée par :  $clnd(\mathcal{A}) = \mathcal{S}'_1 \cup \dots \cup \mathcal{S}'_n$ , tel que :

$$\forall i = 1, \dots, n : \mathcal{S}'_i = free(cl(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_i)).$$

Par conséquence :

$$clnd(\mathcal{A}) = free(cl(\mathcal{S}_1)) \cup free(cl(\mathcal{S}_1 \cup \mathcal{S}_2)) \cup \dots \cup free(cl(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_n)).$$

Nous désignons par  $free(cl(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_i))$  l'ensemble des faits libres dans  $(cl(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_i))$ .

Si on réfère à la Définition 9,  $clnd(\mathcal{A})$  n'est pas une réparation de  $\langle \mathcal{T}, \mathcal{A} \rangle$ . Effectivement,  $clnd(\mathcal{A})$  pas inclus dans  $\mathcal{S}_1 \cup \dots \cup \mathcal{S}_n$  car il peut contenir des éléments qui ne sont pas explicitement indiqués dans  $\mathcal{A}$ . Malgré ça, nous préférons continuer à utiliser le terme de réparation pour signifier  $clnd(\mathcal{A})$  qui est une réparation de  $\langle \mathcal{T}, cl(\mathcal{A}) \rangle$ .

L'algorithme 6 donne le résultat de rajouter la clôture déductive à la réparation à base d'assertions libres prioritaires qui est obtenue à partir de l'algorithme 5 en remplaçant  $\mathcal{A}$  par sa clôture déductive  $cl(\mathcal{A})$ .

---

**Algorithme 6** Réparation à de clôture déductive d'assertions libres prioritaires

---

**Input:**  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  avec  $\mathcal{A} = (\mathcal{S}_1, \dots, \mathcal{S}_n)$

**Output:** Une base d'assertions standard  $clnd(\mathcal{A})$

- 1:  $\mathcal{C} = \mathcal{C}(cl(\mathcal{A}))$  { L'ensemble de conflits dans  $\langle \mathcal{T}, cl(\mathcal{A}) \rangle$ }
  - 2:  $clnd(\mathcal{A}) \leftarrow \emptyset$
  - 3: **for**  $i = 1$  à  $n$  **do**
  - 4:      $clnd(\mathcal{A}) \leftarrow clnd(\mathcal{A}) \cup FREE(cl(\mathcal{S}_i \cup \dots \cup \mathcal{S}_i), \mathcal{C})$
  - 5: **return**  $clnd(\mathcal{A})$
- 

**Exemple 21.** *Permettez-nous d'illustrer cette nouvelle stratégie sur notre exemple 17.*

$free(cl(\mathcal{S}_1)) = \{Student(Bill), Teacher(John)\},$ $free(cl(\mathcal{S}_1 \cup \mathcal{S}_2)) = \{Student(Bill), Teacher(John), Student(Bob),$ $Teacher(Mary)\},$ $free(cl(\mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_3)) = \{Teacher(Mary), Teacher(John), Student(Bob),$ $Researcher(Joe), Teacher(Joe)\},$ $free(cl(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_4)) = \{Teacher(Mary), Teacher(John), Student(Bob),$ $Researcher(Joe), Teacher(Joe), Student(Anne)\},$ $et\ free(cl(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_5)) = \{Teacher(Mary), Teacher(John), Student(Bob),$ $Researcher(Joe), Teacher(Joe), Student(Anne)\}$
---

Alors,  $clnd(\mathcal{A}) = \{Student(Bill), Teacher(John), Student(Bob), Teacher(Mary),$   
 $Researcher(Joe), Teacher(Joe), Student(Anne)\}.$

On peut voir que  $Teacher(Joe)$  est une nouvelle assertion qui n'appartient pas à la partie de la réparation d'assertions libres prioritaires donnée dans l'exemple 20. ■

### 3.3.5 Réparation à base linéaire d'assertions libres prioritaires

Une autre façon de sélectionner une seule réparation préférée consiste à améliorer légèrement la réparation à base linéaire et la réparation à base d'assertions libres prioritaires. L'idée est que dans la réparation linéaire au lieu d'ignorer une strate entière en cas d'inconsistance, on peut seulement ignorer les éléments contradictoires.

La réparation à base linéaire d'assertions libres prioritaires  $\ell nd(\mathcal{A})$  est donnée par la Définition 19.

**Définition 19.** On considère  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  une base de connaissances DL-Lite avec priorité. Nous définissons la réparation à base linéaire d'assertions libres prioritaires, notée par :

$\ell nd(\mathcal{A}) = \mathcal{S}'_1 \cup \dots \cup \mathcal{S}'_n$  comme suit :

$$\mathcal{S}'_1 = \text{free}(\mathcal{S}_1),$$

$$\text{Pour } i = 2, \dots, n \quad \mathcal{S}'_i = \text{free}(\mathcal{S}'_1 \cup \dots \cup \mathcal{S}'_{i-1} \cup \mathcal{S}_i).$$

L'algorithme 7 implique directement la Définition 19.

---

**Algorithme 7** Réparation à base Linear d'assertions libres prioritaires

---

**Input:**  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  avec  $\mathcal{A} = (\mathcal{S}_1, \dots, \mathcal{S}_n)$

**Output:** Une base d'assertions standard  $\ell nd(\mathcal{A})$

- 1:  $\mathcal{C} \leftarrow \mathcal{C}(\mathcal{A})$  { L'ensemble de conflits }
  - 2:  $\ell nd(\mathcal{A}) \leftarrow \emptyset$
  - 3: **for**  $i = 1$  à  $n$  **do**
  - 4:      $\ell nd(\mathcal{A}) \leftarrow \ell nd(\mathcal{A}) \cup \text{FREE}(\mathcal{S}_i \cup \ell nd(\mathcal{A}), \mathcal{C})$
  - 5: **return**  $\ell nd(\mathcal{A})$
- 

La fonctionnalité intéressante de  $\ell nd$ -conséquence est qu'une extension de  $\ell$ -conséquence et  $nd$ -conséquence se déroulent sans coût de calcul supplémentaire. C'est à dire, le calcul  $\ell nd(\mathcal{A})$  est en temps polynomial.

Clairement,  $\ell nd(\mathcal{A})$  entend  $nd(\mathcal{A})$  en se concentrant uniquement sur les éléments de  $\mathcal{S}_i$  qui sont consistants avec  $\mathcal{S}'_1 \cup \dots \cup \mathcal{S}'_{i-1}$  (plutôt qu'avec  $\mathcal{S}_1 \cup \dots \cup \mathcal{S}_{i-1}$ ).

**Exemple 22.** Sur le même exemple 17, nous avons :

$$\begin{aligned}
 \mathcal{S}'_1 &= \text{free}(\mathcal{S}_1) = \{Student(Bill), Teacher(John)\} \\
 \mathcal{S}'_2 &= \text{free}(\mathcal{S}'_1 \cup \mathcal{S}_2) \\
 &= \{Student(Bill), Teacher(John), Teacher(Mary), Student(Bob)\} \\
 \mathcal{S}'_3 &= \text{free}(\mathcal{S}'_1 \cup \mathcal{S}'_2 \cup \mathcal{S}_3) \\
 &= \{Teacher(John), Teacher(Mary), Researcher(Joe)\} \\
 \mathcal{S}'_4 &= \text{free}(\mathcal{S}'_1 \cup \mathcal{S}'_2 \cup \mathcal{S}'_3 \cup \mathcal{S}_4) \\
 &= \{Teacher(John), Teacher(Mary), Researcher(Joe), Student(Anne)\} \\
 \mathcal{S}'_5 &= \text{free}(\mathcal{S}'_1 \cup \mathcal{S}'_2 \cup \mathcal{S}'_3 \cup \mathcal{S}'_4 \cup \mathcal{S}_5) \\
 &= \{Teacher(John), Teacher(Mary), Researcher(Joe), Student(Anne)\}
 \end{aligned}$$

Donc :  $\text{end}(\mathcal{A}) = \{Student(Bill), Teacher(John), Teacher(Mary), Researcher(Joe), Student(Anne), Student(Bob)\}$ . ■

### 3.4 Conclusion

Ce chapitre s'est focalisé sur les méthodes de sélectionner une seule réparation préférée à partir d'une base de connaissances *DL-Lite* inconsistante avec priorité. La sélection d'une seule réparation est importante car elle permet de répondre efficacement à une requête une fois que la répartition préférée est calculée. Nous avons d'abord examiné les deux approches bien connues pour la sélection d'une réparation préférée : la réparation à base de la théorie des possibilités et la réparation linéaire. Ensuite, et en plus de la réparation à base d'assertions libres prioritaires, nous avons présenté deux autres stratégies pour sélectionner une réparation préférée en fonction de la réparation à base d'assertions libres prioritaires en rajoutant le critère de clôture déductive et le critère de consistance.

La caractéristique commune de toutes ces réparations est qu'elles produisent autant de conclusions saines que possible comme il sera montré dans nos études expérimentales dans le chapitre suivant.

# Chapitre 4

## Résultats expérimentaux et études comparatives

*"If art interprets our dreams, the computer executes them in the guise of programs!"*

*Alan J. Perlis.*

### 4.1 Introduction

Plusieurs travaux [114], [115] ont étudié la notion de priorité lors de l'interrogation de bases de données inconsistantes. Dans [116], une réparation maximale d'inclusion a été introduite pour répondre aux requêtes d'une base de connaissances *SHIQ* inconsistante. Dans *DL-Lite*, le seul travail qui existe de [117] a été proposé afin de traiter le raisonnement sous l'inconsistance dans les bases de connaissances en utilisant les priorités d'assertions.

Dans ce chapitre, l'accent est mis sur l'évaluation et l'expérimentation des stratégies de calcul des réparations dans *DL-Lite* avec priorités. Ces expérimentations permettent d'évaluer les résultats obtenus. Cette évaluation est réalisée en exploitant un ensemble de tests. Les tests permettent d'estimer la qualité de l'alignement obtenu et de comparer l'efficacité de ces algorithmes. L'expérimentation est réalisée sur une base d'ontologies de référence mise à la disposition de la communauté. Nous fournissons aussi des études comparatives montrant les avantages d'une réparation pour le raisonnement en *DL-Lite*.

## 4.2 Environnement de développement

### 4.2.1 Environnement software

Nous avons implémenté nos algorithmes pour calculer une seule réparation préférée en langage de programmation Java 8 sur la plateforme NetBeans IDE 8.2<sup>1</sup> (IDE officiel pour Java 8). NetBeans est une plateforme de développement gratuit et open source que possède une grande communauté d'utilisateurs et de développeurs à travers le monde. En plus de Java, NetBeans permet la prise en charge de divers langages tels le C, le C++, le JavaScript, le XML, et le HTML. Il offre toutes les facilités d'un IDE moderne (éditeur en couleurs, projets multi-langages, refactoring, éditeur graphique d'interfaces et de pages Web). NetBeans est disponible sous Windows, Linux, et Mac OS X. Un environnement Java Development Kit JDK (JDK 1.8 utilisé) est requis pour les développements en Java. La plate-forme NetBeans est un cadre générique pour les applications Swing (une bibliothèque graphique pour le langage de programmation Java).

Pour la manipulation de base de données relationnelle, nous avons choisi SQLite<sup>2</sup> (Version 3.8). SQLite est une bibliothèque écrite en C qui propose un moteur de base de données relationnel accessible par le langage SQL. SQLite implémente en grande partie le standard SQL-92 et des propriétés ACID (Atomique, Cohérente, Isolée et Durable).

### 4.2.2 Environnement hardware

Toutes les expériences sont effectuées sur un ordinateur portable ASUS SonicMaster Introduction. Modèle X556QUK. Un processeur Intel(R) Core(TM) (i5) 7200 CPU @ 2.50 GHz 2,71 GHz. Une mémoire vivante RAM DDR3 4 Go. Cette configuration matérielle est installée dans un système d'exploitation 64 bits, processeur x64 (Windows 10 Famille).

- 
1. <https://netbeans.org/>
  2. <https://www.sqlite.org/>

## 4.3 Paramètres Expérimentaux

### 4.3.1 Bases de tests utilisées

Nous avons utilisé le Benchmark<sup>3</sup> comme une source d'ontologie LUBM<sup>3</sup>20 (the Lehigh University Benchmark) qui correspond à la *DL-Lite<sub>R</sub>* (pour la TBox) [3]. L'ontologie LUBM<sup>3</sup>20 fournit plusieurs outils pour l'accès aux données basées sur l'ontologie (OBDA) dans le profil OWL2 QL. Nous avons utilisé l'outil EUDG (Extended University Data Generator) afin de générer les assertions des ABoxs.

Les fondements théoriques de ce travail se trouvent dans le document [3]. Le document explique également comment exactement nous avons modifié le générateur de données LUBM d'origine et l'ontologie. L'ontologie LUBM<sup>3</sup>20 contient 208 axiomes d'inclusion positifs. Nous avons utilisé un ensemble des axiomes d'inclusion négative afin de permettre l'inconsistance de la base de connaissance. (Voir Table 4.1).

Métriques	Nombre	Exemple
Classe	129	FullProfessor, Faculty
ObjectProperty	28	HasFaculty, isPartOfUniversity
DataProperty	7	DataPropertyDomain(age Person)
SubClassOf axioms	153	SubClassOf(Professor Faculty)
DisjointClassOf axioms	643	DisjointClassOf (Techer, Student)
SubObjectPropertyOf axioms	5	SubObjectPropertyOf(headOf worksFor)
InverseObjectProperties axioms	3	InverseObjectProperties(HasSupervisor)
DisjointObjectProperties axioms	227	DisjointObjectPropertyOf(headOf advisor)
ObjectPropertyDomain axioms	25	ObjectPropertyDomain(advisor Person)
ObjectPropertyRange axioms	22	ObjectPropertyRange(advisor Professor)
DataPropertyDomain axioms	4	DataPropertyDomain(age Person)

TABLE 4.1 – Information d'ontologie LUBM.

3. Disponible à :<https://code.google.com/p/combo-obda/>

### 4.3.2 Métriques d'évaluation

Dans l'objectif de mesurer les performances des méthodes de sélection d'une réparation préférée dans *DL-Lite* avec priorité, un ensemble de métriques est nécessaire. Ces mesures permettent l'évaluation de la qualité des alignements produits. Ils incluent les métriques de vérification de la consistance, calcul des éléments de conflits, et calcul de la clôture déductive. Pour calculer efficacement ces métriques, nous évaluons les requêtes sur ABox (stockées en BD relationnelle) exprimées à partir de la fermeture négative du TBox pour montrer si la ABox contient des éléments contradictoires. La fermeture négative d'un ABox est faite de la liste de tous les axiomes négatifs de la forme  $A \sqsubseteq \neg B$  (Voir 2.3.3) qui peut être dérivé de  $\mathcal{T}$  en appliquant des règles positives sur les règles négatives [3]. La fermeture négative peut être pré-calculée et conservée pendant toutes les expérimentations. Lorsque tous ces ingrédients sont disponibles, nous procédons au calcul de la réparation unique préférée en utilisant différents algorithmes proposés dans cette thèse.

### 4.3.3 Tests et Résultats

La première phase dans notre évaluation consiste à calculer la fermeture négative de cette ontologie. Ce temps nécessaire pour calculer la fermeture négative ne sera pas pris en compte comme paramètre dans nos études expérimentales puisqu'il est calculé une seule fois puis stocké dans la mémoire. Une fois que la fermeture négative est calculée, nous utilisons un moteur SQLite pour calculer les éléments de conflits et pour vérifier les inconsistances. Cela permet une gestion efficace de l'inconsistance. Nous considérons trois cas différents, de sorte que, dans tous les cas, nous avons divisé uniformément la ABox et nous avons exécuté les algorithmes proposés pour le calcul des réparations :

## Cas 1 : Le temps de calcul de réparation

Nous avons généré (en utilisant EUDG) quatre ABoxes et nous les divisons respectivement en : 3 strates, 5 strates et 7 strates. Ces ABoxes contiennent respectivement 2532, 2832, 3432 et 4432 assertions avec respectivement des ensembles de conflits de 50, 200, 500 et 1000. Les résultats de cette expérimentation sont présentés dans le Tableau 4.2.

Conflits	strata	$\pi$		$\ell$		$nd$		$\ell nd$		$clnd$	
		$\mathcal{O}(\log_2(nbs)) * Conf$		$Conf * nbs$		$\mathcal{O}(nbs) + Conf$		$\mathcal{O}(nbs) + Conf$		$\mathcal{O}(nbs) + Conf$	
		temps	%	temps	%	temps	%	temps	%	temps	%
50	3	21.47	100	22.19	100	94.91	2.98	159.47	2.98	458.73	45.04
	5	21.92	79.44	65.31	78.63	156.35	0.48	257.61	0.44	907.65	36.31
	7	32.72	85.06	65.84	67.82	188.52	0.48	340.98	2.17	1284.90	36.63
200	3	24.04	100	56.35	100	91.81	14.07	185.15	15.13	478.24	44.74
	5	24.94	81.39	66.19	81.39	134.86	14.05	272.25	14.05	956.20	36.19
	7	33.07	59.40	77.07	51.04	193.99	15.04	414.73	11.43	1202.55	36.68
500	3	27.72	86.94	28.83	86.94	101.37	28.40	293.60	28.40	497.08	52.65
	5	36.24	86.74	59.97	86.74	186.72	26.29	488.98	26.29	1116.06	57.05
	7	39.78	88.07	86.83	76.75	198.86	726.87	669.14	26.66	1424.63	56.95
1000	3	57.26	100	57.66	87.15	126.60	45.53	722.23	45.53	1222.88	25.98
	5	59.10	91.58	84.43	82.87	159.41	45.65	1246.23	45.65	1511.58	31.22
	7	112.11	96.36	171.47	96.36	221.15	40.20	1897.57	40.23	1881.15	43.86

TABLE 4.2 – Pourcentage de faits supprimés et le temps (en secondes) pris pour calculer les conflits,  $\pi$ ,  $\ell$ ,  $nd$ ,  $\ell nd$ , et  $clnd$  en variant le nombre de conflits et le nombre de strates.

**Dans le Tableau 1 :** Du point de vue de productivité, la réparation à base de la théorie des possibilités est très prudente par rapport aux autres stratégies. Cela garantit par conséquent une faible productivité en termes de taille de réparation sélectionnée. Pour une ABox donnée et un nombre donné de strates, la réparation basée sur la théorie des possibilités possède le plus grand nombre d'éléments abandonnés.

Par exemple, lorsqu'un conflit est détecté dans la première couche, la ABox complète est éliminé (par conséquent, 100% des assertions seront supprimées comme indiqué dans la première ligne du Tableau 4.2). Cela tient également à la réparation linéaire lorsqu'il existe au moins un conflit dans chaque strate. Comme le montrent les résultats du Tableau 4.2. Le calcul de la réparation à base de la théorie des possibilités nécessite, dans la plupart de nos expériences moins de temps que pour calculer les autres.

En ce qui concerne le calcul de la réparation à base d'assertions libres prioritaires (ou la réparation à base de la clôture d'assertions libres prioritaires), cela dépend du nombre de conflits dans la ABox. Plus précisément, le temps nécessaire pour calculer la réparation à base d'assertions libres prioritaires (ou la réparation à base de la clôture d'assertions libres prioritaires) augmente avec la taille des conflits dans les ABox.

Enfin, pour la réparation à base linéaire d'assertions libres prioritaires, deux paramètres peuvent influencer sur le temps pris pour le calcul ; à savoir, le nombre de couches et le nombre de conflits dans chaque couche. La proportion d'assertions abandonnées concernant les différentes réparations confirme les relations entre les réparations  $\pi$ ,  $\ell$ ,  $nd$ ,  $\ell nd$ , et  $clnd$  comme indiqué dans les sections suivantes.

Le Tableau 4.2 résume la complexité temporelle des différentes approches.  $Conf$  est la complexité du temps pour calculer les conflits.  $nbs$  est le nombre de strates dans *DL-Lite* avec priorité.

Dans Tableau 4.2, il est évident que le nombre de strates et le nombre d'éléments conflictuels dans les bases assertionnelles ABox sont les deux paramètres principaux qui influencent sur la productivité des réparations et sur le temps nécessaire pour les calculs. Le Tableau 4.3 et le Tableau 4.4 analysent séparément ces deux paramètres.

## Cas 2 : Impact du nombre de strates

Dans ce cas, nous avons généré au hasard une ABox et nous avons varié le nombre de strates de 1 à 15. Pour chaque nombre de strates, nous avons réparti les ABoxes uniformément et nous avons exécuté les algorithmes proposés pour le calcul des réparations. Les résultats de cette expérimentation sont présentés dans le Tableau 4.3.

# strata	$\pi$		$\ell$		$nd$		$\ell nd$		$clnd$	
	temps	%	temps	%	temps	%	temps	%	temps	%
1	28.67	100	28.77	100	48.20	72.29	98.55	39.74	128.65	35.12
3	22.74	100	24.69	100	96.27	39.65	161.52	19.67	399.98	35.04
5	24.01	88.66	30.35	77.01	147.36	37.35	243.82	26.66	915.95	27.85
7	23.11	89.80	51.26	80.12	189.95	37.29	314.54	27.02	1287.85	27.80
9	33.72	92.01	58.68	91.84	255.03	37.20	555.74	28.50	2328.47	27.68
11	21.67	100	61.07	79.57	271.81	36.45	1335.72	28.42	2448.05	27.68
13	23.90	100	61.85	78.72	314.18	36.42	1608.26	28.42	2979.21	27.61
15	30.89	100	64.18	77.85	355.38	35.89	1842.48	28.40	3846.21	20.36

TABLE 4.3 – Impact du nombre de strates sur la productivité et sur le temps de calcul.

Le Tableau 4.3 montre l'impact du nombre de strates sur la productivité (en termes de pourcentage des assertions supprimées lors du calcul des réparations) et sur le calcul des réparations. Pour chaque ABox (utilisé dans le cas 1), nous avons varié le nombre de couches, on peut observer que la productivité de  $clnd(\mathcal{A})$ ,  $\ell nd(\mathcal{A})$ , et  $nd(\mathcal{A})$  augmente proportionnellement avec le nombre de strates. Comme effet négatif, le temps de fonctionnement augmente également proportionnellement par rapport au nombre de strates. Le temps de fonctionnement de  $\ell(\mathcal{A})$  est meilleur que ceux de  $clnd(\mathcal{A})$ ,  $\ell nd(\mathcal{A})$ , et  $nd(\mathcal{A})$ .

### Cas 3 : Impact du nombre des éléments de conflits

Nous avons considéré les ABox utilisées dans le cas 2 et nous avons fixé le nombre de strates à 7 strates. Ensuite, nous avons varié le pourcentage d'éléments de conflits de 0% à 100% par rapport à la taille de la ABox. Les résultats de ce cas sont donnés dans le Tableau 4.4.

Conflits	$\pi$		$\ell$		$nd$		$\ell nd$		$clnd$	
	temps	%	temps	%	temps	%	temps	%	temps	%
0%	86.92	0	111.08	0	131.72	0	370.25	0	597.12	0
10%	18.96	69.27	27.86	7.63	108.90	3.63	422.68	3.63	682.24	8.46
20%	20.19	71.83	31.56	9.27	116.84	7.00	540.77	7.00	817.59	6.75
30%	21.87	74.00	35.23	57.30	157.66	12.84	653.57	12.84	820.25	16.30
40%	21.14	75.85	51.53	50.64	142.94	15.07	836.99	15.07	1318.68	28.77
50%	30.97	100	45.49	53.53	199.97	19.40	840.44	19.40	1495.83	31.19
60%	31.59	100	48.97	56.5	178.30	23.18	1070.42	23.18	1453.53	32.65
70%	32.32	100	54.08	54.05	262.86	24.17	1071.20	24.17	1341.92	41.32
80%	34.48	100	58.77	90.77	249.04	26.77	1379.35	26.77	1501.09	42.11
90%	41.41	100	60.89	97.31	281.91	28.31	1513.00	25.68	1604.95	46.52
100%	45.46	100	64.14	100	342.45	29.10	1754.43	25.95	2156.46	46.91

TABLE 4.4 – Impact du nombre des éléments de conflits sur la productivité et sur le temps de calcul des réparations.

Contrairement au Tableau 4.3, le Tableau 4.4 analyse l'impact du nombre de conflits sur le temps pris pour calculer les réparations et sur le pourcentage de faits supprimés. Pour un nombre fixe de strates (7 strates), nous avons varié le nombre de conflits entre 0% et 100% de la taille d'ABox. On peut observer que pour toutes les stratégies, le temps de fonctionnement et l'ensemble des faits supprimés augmentent avec le pourcentage des éléments contradictoires dans les ABoxes.

Le Tableau 4.4 confirme que  $clnd(\mathcal{A})$ ,  $\ell nd(\mathcal{A})$ , et  $nd(\mathcal{A})$  sont plus productifs que  $\ell(\mathcal{A})$  et  $\pi(\mathcal{A})$  car ils suppriment moins de faits.

Enfin, il est important de noter que le calcul des conflits assertionnels peuvent se faire progressivement (il suffit de calculer uniquement les nouveaux conflits soulevés par l'arrivée d'une nouvelle assertion ou d'un nouvel axiome). Les calculs de manière incrémentale des conflits ont un impact sur nos algorithmes afin de mettre à jour régulièrement l'ensemble des assertions *libres* ou de mettre à jour le degré de consistance (ce qui ne peut que diminuer lorsque de nouvelles assertions sont ajoutées).

D'autre part, la plupart des réparations présentées sont définies à partir de la couche 1 jusqu'à la couche  $n$ . Par conséquent, si de nouvelles assertions sont ajoutées à une couche  $i$ , alors tous les calculs déjà effectués sur des couches  $1, \dots, i - 1$  restent valides.

Dans la suite, Nous présentons des propositions pour montrer les relations d'inclusion ou l'incomparabilité entre les différents algorithmes développés.

## 4.4 Étude comparatives

D'après les exemples précédents 18, 19, 20,21 et 22, et à travers les résultats obtenus par les études expérimentales. On peut vérifier les propriétés d'inclusion suivantes entre les différentes réparations :

**PROPOSITION 6.** *On considère  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  une base de connaissance DL-Lite avec priorité. Alors :*

- i)  $\pi(\mathcal{A}) \subseteq \ell(\mathcal{A})$ .
- ii)  $\pi(\mathcal{A}) \subseteq nd(\mathcal{A})$ .
- iii)  $\pi(\mathcal{A}) \subseteq lnd(\mathcal{A})$ .
- iv)  $nd(\mathcal{A}) \subseteq lnd(\mathcal{A})$ .

*Démonstration.*

Si  $\langle \mathcal{T}, \mathcal{A} \rangle$  est consistant, et par l'utilisation des définitions de la réparation à base de la théorie des possibilités, à base linéaire et la réparation à base d'assertions libres prioritaires, nous avons :

$$\pi(\mathcal{A}) = \ell(\mathcal{A}) = nd(\mathcal{A}) = lnd(\mathcal{A}) = \mathcal{S}_1 \cup \dots \cup \mathcal{S}_i.$$

- i) Si  $\langle \mathcal{T}, \mathcal{A} \rangle$  est inconsistant. Cela signifie qu'il existe un rang  $i$  tel que  $\mathcal{S}_1 \cup \dots \cup \mathcal{S}_i$  est consistant, mais  $\mathcal{S}_1 \cup \dots \cup \mathcal{S}_{i+1}$  est inconsistant ( $Consrank(\mathcal{K}) = i$ ). Alors, par la définition de réparation à base de la théorie des possibilités, nous avons :

$$\pi(\mathcal{A}) = \mathcal{S}_1 \cup \dots \cup \mathcal{S}_i.$$

Il est clair que, par l'utilisation de définition de réparation à base linéaire, nous avons aussi :

$$\pi(\mathcal{A}) \subseteq \ell(\mathcal{A}).$$

ii) De même, tant que  $\mathcal{S}_1 \cup \dots \cup \mathcal{S}_i$  est consistant, alors :

$$\text{free}(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_i) = \mathcal{S}_1 \cup \dots \cup \mathcal{S}_i = \pi(\mathcal{A}).$$

D'autre part,  $nd(\mathcal{A})$  est défini par :

$$nd(\mathcal{A}) = \text{free}(\mathcal{S}_1) \cup \text{free}(\mathcal{S}_1 \cup \mathcal{S}_2) \cup \dots \cup \text{free}(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_n).$$

Alors, nous avons trivialement :

$$\pi(\mathcal{A}) \subseteq nd(\mathcal{A}).$$

iii) Si  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  est inconsistant. Rappelons que  $lnd(\mathcal{A}) = \mathcal{S}'_1 \cup \dots \cup \mathcal{S}'_n$  tel que :

$$\mathcal{S}'_1 = \text{free}(\mathcal{S}_1) \text{ et pour } i = 2, \dots, n \text{ } \mathcal{S}'_i = \text{free}(\mathcal{S}'_1 \cup \dots \cup \mathcal{S}'_{i-1} \cup \mathcal{S}_i).$$

On considère  $i$  un entier tel que  $\pi(\mathcal{A}) = \mathcal{S}_1 \cup \dots \cup \mathcal{S}_i$  est consistant, mais  $\mathcal{S}_1 \cup \dots \cup \mathcal{S}_i \cup \mathcal{S}_{i+1}$  est inconsistant. Nous avons :

$$\mathcal{S}'_1 = \text{free}(\mathcal{S}_1) = \mathcal{S}_1,$$

$$\mathcal{S}'_2 = \text{free}(\mathcal{S}_2 \cup \mathcal{S}'_1) = \mathcal{S}_1 \cup \mathcal{S}_2, \text{ et}$$

$$\mathcal{S}'_i = \text{free}(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_i \cup \mathcal{S}'_{i-1}) = \mathcal{S}_1 \cup \dots \cup \mathcal{S}_i. \text{ Alors :}$$

$$\pi(\mathcal{A}) \subseteq lnd(\mathcal{A}).$$

iv) Rappelons que  $nd(\mathcal{A}) = \mathcal{S}''_1 \cup \dots \cup \mathcal{S}''_n$  tel que :  $\forall i = 1, \dots, n \mathcal{S}''_i = \text{free}(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_i)$ .

Il est clair que,  $\forall i = 1, \dots, n \quad \mathcal{S}''_i \subseteq \mathcal{S}'_i$ . Alors :

$$nd(\mathcal{A}) \subseteq lnd(\mathcal{A}).$$

□

L'exemple suivant montre la situation où  $\ell(\mathcal{A})$ ,  $nd(\mathcal{A})$  et  $lnd(\mathcal{A})$  ne sont pas inclus dans  $\pi(\mathcal{A})$ .

**Exemple 23.** On considère  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  une base de connaissances DL-Lite avec priorité.

- $\mathcal{T} = \{A \sqsubseteq \neg B\}$  et
- $\mathcal{A} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$ ,

tel que :

- $\mathcal{S}_1 = \{A(a)\}$ ,
- $\mathcal{S}_2 = \{B(a)\}$ , et
- $\mathcal{S}_3 = \{C(a)\}$ .

En utilisant les définitions de réparation à base de la théorie des possibilités, réparation à base linear, réparation à base d'assertions libres prioritaires, et réparation à base linear d'assertions libres prioritaires, nous avons :

$$\pi(\mathcal{A}) = \{A(a)\}, \quad nd(\mathcal{A}) = \ell(\mathcal{A}) = lnd(\mathcal{A}) = \{A(a), C(a)\}.$$

Il est clair,  $C(a) \in \ell(\mathcal{A})$ ,  $C(a) \in nd(\mathcal{A})$ ,  $C(a) \in lnd(\mathcal{A})$ , mais  $C(a) \notin \pi(\mathcal{A})$ . ■

Une caractéristique importante des réparations à base de la théorie des possibilités et à base linear c'est qu'elles sont insensibles à la clôture déductive prioritaire locale ; c'est à dire, si on utilise d'abord la réparation à base de la théorie des possibilités (ou à base linear) puis on applique la clôture déductive prioritaire locale, elle donnera le même résultat que si on applique la clôture déductive prioritaire locale puis on utilise la réparation à base de la théorie des possibilités (ou à base linear).

La situation est différente de la Définition 13. Dans ce cas, la réparation à base de la théorie des possibilités est encore insensible à la clôture déductive, ce qui n'est pas le cas avec la réparation à base linear. Cette insensibilité est résumée par la proposition suivante :

**PROPOSITION 7.** On considère  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  une base de connaissances DL-Lite avec priorité :

- i)  $\langle \mathcal{T}, cl(\pi(\mathcal{A})) \rangle = \langle \mathcal{T}, \pi(cl(\mathcal{A})) \rangle$
- ii)  $\langle \mathcal{T}, \ell(cl(\mathcal{A})) \rangle \subseteq \langle \mathcal{T}, cl(\ell(\mathcal{A})) \rangle$ .
- iii)  $\langle \mathcal{T}, \ell(lcl(\mathcal{A})) \rangle = \langle \mathcal{T}, lcl(\ell(\mathcal{A})) \rangle$ .

*Démonstration.* :

Rappelons tout d'abord, dans le DL-Lite standard,  $\langle \mathcal{T}, \mathcal{A} \rangle$  est consistant si et seulement si  $\langle \mathcal{T}, cl(\mathcal{A}) \rangle$  et  $\langle \mathcal{T}, lcl(\mathcal{A}) \rangle$  est consistant.

i) Si  $\langle \mathcal{T}, \mathcal{A} \rangle$  est consistant, alors :  $\pi(\mathcal{A}) = \mathcal{S}_1 \cup \dots \cup \mathcal{S}_n$  et  $\pi(\text{cl}(\mathcal{A})) = \text{cl}(\mathcal{A})$ . Donc :

$$\boxed{\pi(\text{cl}(\mathcal{A})) = \text{cl}(\pi(\mathcal{A}))}.$$

Si  $\langle \mathcal{T}, \mathcal{A} \rangle$  est inconsistant. Cela signifie qu'il existe un nombre entier  $i$  tel que :  $\pi(\mathcal{A}) = \mathcal{S}_1 \cup \dots \cup \mathcal{S}_i$  est consistant, mais :  $\pi(\mathcal{A}) \cup \mathcal{S}_{i+1}$  est inconsistant. Rappelons que  $\text{cl}(\mathcal{A}) = (\mathcal{S}'_1, \dots, \mathcal{S}'_n)$  tel que :

$$\mathcal{S}'_1 = \text{cl}(\mathcal{S}_1),$$

$$\text{et } \forall j = 2, \dots, n \mathcal{S}'_j = \text{cl}(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_{j-1} \cup \mathcal{S}_j).$$

Tandis que, si  $\mathcal{S}_1 \cup \dots \cup \mathcal{S}_i$  est consistant et  $\mathcal{S}_1 \cup \dots \cup \mathcal{S}_{i+1}$  est inconsistant, alors :  $\mathcal{S}'_1 \cup \dots \cup \mathcal{S}'_i$  est consistant et  $\mathcal{S}'_1 \cup \dots \cup \mathcal{S}'_{i+1}$  est inconsistant.

Par conséquence :

$$\pi(\text{cl}(\mathcal{A})) = \mathcal{S}'_1 \cup \dots \cup \mathcal{S}'_i.$$

D'ailleurs, il est facile de vérifier que :

$$\mathcal{S}'_1 \cup \dots \cup \mathcal{S}'_i = \text{cl}(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_i) = \text{cl}(\pi(\mathcal{A})). \text{ Donc :}$$

$$\boxed{\text{cl}(\pi(\mathcal{A})) = \pi(\text{cl}(\mathcal{A}))}.$$

ii) Pour avoir la preuve, il suffit de montrer que :

$$\ell(\text{cl}(\mathcal{A})) = \pi(\text{cl}(\mathcal{A})).$$

Si  $\langle \mathcal{T}, \mathcal{A} \rangle$  est consistant, alors :  $\ell(\mathcal{A}) = \mathcal{S}_1 \cup \dots \cup \mathcal{S}_n$  et  $\ell(\text{cl}(\mathcal{A})) = \text{cl}(\mathcal{A})$ . Trivialement, nous avons :

$$\ell(\text{cl}(\mathcal{A})) = \text{cl}(\ell(\mathcal{A})) = \text{cl}(\pi(\mathcal{A})) = \pi(\text{cl}(\mathcal{A})).$$

On suppose que  $\langle \mathcal{T}, \mathcal{A} \rangle$  est inconsistant. Et encore on considère un entier  $i$  tel que :  $\mathcal{S}_1 \cup \dots \cup \mathcal{S}_i$  est consistant, mais  $\mathcal{S}_1 \cup \dots \cup \mathcal{S}_{i+1}$  est inconsistant. Cela signifie que :

$$\pi(\text{cl}(\mathcal{A})) = \text{cl}(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_i) = (\mathcal{S}'_1 \cup \dots \cup \mathcal{S}'_i).$$

Rappelons que  $\text{cl}(\mathcal{A}) = (\mathcal{S}'_1, \dots, \mathcal{S}'_n)$  tel que :  $\mathcal{S}'_1 = \text{cl}(\mathcal{S}_1)$ ,

$$\text{et } \forall j = 2, \dots, n \mathcal{S}'_j = \text{cl}(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_{j-1} \cup \mathcal{S}_j).$$

Tandis que, si  $\mathcal{S}_1 \cup \dots \cup \mathcal{S}_{i+1}$  est inconsistant, alors :  $\forall j > i$ ,  $\mathcal{S}'_1 \cup \dots \cup \mathcal{S}'_j$  aussi est inconsistant. Donc :

$$\boxed{l(\text{cl}(\mathcal{A})) = \pi(\text{cl}(\mathcal{A}))}.$$

iii) Rappelons que  $l\text{cl}(\mathcal{A}) = \mathcal{S}'_1 \cup \dots \cup \mathcal{S}'_i$  tel que :  $\forall i = 1, \dots, \mathcal{S}'_i = \text{cl}(\mathcal{S}_i)$ .

Si  $\langle \mathcal{T}, \mathcal{A} \rangle$  est consistant alors  $\pi(\mathcal{A}) = \mathcal{S}_1 \cup \dots \cup \mathcal{S}_n$ . Car  $\mathcal{S}_1 \cup \dots \cup \mathcal{S}_n$  est consistant.

Alors :  $\text{cl}(\mathcal{S}_1) \cup \dots \cup \text{cl}(\mathcal{S}_n) = \mathcal{S}'_1 \cup \dots \cup \mathcal{S}'_n$  est aussi consistant. Donc :

$$\boxed{\pi(l\text{cl}(\mathcal{A})) = \mathcal{S}'_1 \cup \dots \cup \mathcal{S}'_n = l\text{cl}(\pi(\mathcal{A}))}.$$

Supposons que  $\langle \mathcal{T}, \mathcal{A} \rangle$  est inconsistant. On considère que  $i$  un entier tel que  $\mathcal{S}_1 \cup \dots \cup \mathcal{S}_i$  est consistant mais  $\mathcal{S}_1 \cup \dots \cup \mathcal{S}_{i+1}$  est inconsistant.

Cela signifie que  $\pi(\mathcal{A}) = \mathcal{S}_1 \cup \dots \cup \mathcal{S}_i$ . Cela aussi signifie que  $\mathcal{S}'_1 \cup \dots \cup \mathcal{S}'_i$  est consistant, mais  $\mathcal{S}'_1 \cup \dots \cup \mathcal{S}'_{i+1}$  est inconsistant. C'est-à-dire :

$$\boxed{\pi(l\text{cl}(\mathcal{A})) = \mathcal{S}'_1 \cup \dots \cup \mathcal{S}'_i = l\text{cl}(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_i) = l\text{cl}(\pi(\mathcal{A}))}.$$

□

Un contre-exemple sur l'inverse de la propriété (ii) de la Proposition 7.

**Exemple 24.** On considère  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  une base de connaissances DL-Lite avec priorité, tel que :

- $\mathcal{T} = \{A \sqsubseteq \neg B, D \sqsubseteq E\}$ ,
- $\mathcal{A} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$ .

Avec :

- $\mathcal{S}_1 = \{A(a)\}$ ,
- $\mathcal{S}_2 = \{B(a), C(a)\}$ , et
- $\mathcal{S}_3 = \{D(a)\}$ .

La clôture prioritaire de  $\mathcal{A}$  est :  $\text{cl}(\mathcal{A}) = (\mathcal{S}'_1, \mathcal{S}'_2, \mathcal{S}'_3)$  tel que :

- $\mathcal{S}'_1 = \{A(a)\}$ ,
- $\mathcal{S}'_2 = \{B(a), C(a), A(a)\}$ , et
- $\mathcal{S}'_3 = \{D(a), E(a), B(a), C(a), A(a)\}$ .

La réparation à base Linéar de  $\langle \mathcal{T}, \mathcal{A} \rangle$  est :  $\ell(\mathcal{A}) = \{A(a), D(a)\}$ ,  $\ell(\text{cl}(\mathcal{A})) = \{A(a)\}$ . En outre, la clôture de  $\ell(\mathcal{A})$  est :  $\text{cl}(\ell(\mathcal{A})) = \{A(a), D(a), E(a)\}$ .

Clairement,  $E(a)$  Peut être obtenu à partir de  $\text{cl}(\ell(\mathcal{A}))$ . Alors, il ne peut pas être obtenu à partir de  $\ell(\text{cl}(\mathcal{A}))$ . ■

Contrairement à la réparation à base de la théorie des possibilités et la réparation à base linéaire, la réparation à base d'assertions libres prioritaires est sensible à l'utilisation de la clôture déductive. La proposition suivante montre la sensibilité de la réparation d'assertions libres prioritaires à la clôture déductive prioritaire. D'autre part,  $\text{lnd}(\mathcal{A})$  est incomparable avec les autres approches présentées ( $\ell(\mathcal{A})$  et  $\text{clnd}(\mathcal{A})$ ). Les propriétés de la réparation linéaire d'assertions libres prioritaires sont résumées dans la Proposition 8.

La propriété (iii) indique que l'utilisation de la réparation linéaire d'assertions libres prioritaires ( $\text{lnd}$ ) sur la clôture prioritaire  $\text{cl}(\mathcal{A})$  conduit simplement à la définition de la réparation de clôture d'assertions libres prioritaires.

**PROPOSITION 8.** *On considère  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  une base de connaissances DL-Lite avec priorité.*

- i)  $\text{nd}(\mathcal{A}) \subseteq \text{clnd}(\mathcal{A})$ . L'inverse est faux.
- ii)  $\text{cl}(\text{nd}(\mathcal{A})) \subseteq \text{clnd}(\mathcal{A})$ . L'inverse est faux.
- iii)  $\text{lnd}(\text{cl}(\mathcal{A})) = \text{clnd}(\mathcal{A})$ .

*Démonstration.*

i) Remarquons que :  $\forall i = 1, \dots, n$  nous avons :

$$\text{free}(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_i) \subseteq \text{free}(\text{cl}(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_i)).$$

C'est-à-dire :

$$\boxed{\text{nd}(\mathcal{A}) \subseteq \text{clnd}(\mathcal{A})}.$$

ii) Remarquons que :  $\forall i = 1, \dots, n$  nous avons :

$$\text{cl}(\text{free}(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_i)) \subseteq \text{free}(\text{cl}(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_i)).$$

C'est-à-dire :

$$\boxed{cl(nd(\mathcal{A})) \subseteq clnd(\mathcal{A}).}$$

L'exemple 25 montre l'inverse de cette sensibilité à la clôture déductive.

**Exemple 25.** On considère la base de connaissances  $\mathcal{K}$  tel que :

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

$$\mathcal{A} = \mathcal{S}_1 = \{A(a), B(a)\}.$$

On peut vérifier que  $nd(\mathcal{A}) = \emptyset$ ,  $cl(nd(\mathcal{A})) = \emptyset$ .

Mais  $nd(cl(\mathcal{A})) = clnd(\mathcal{A}) = \{C(a)\}$ . ■

iii) Pour montrer cette propriété; on considère que  $cl(\mathcal{A}) = (\mathcal{S}'_1, \dots, \mathcal{S}'_n)$  la clôture prioritaire de  $\mathcal{A}$ . Par la définition de la clôture prioritaire, nous avons :

$$\forall i = 1, \dots, n, \mathcal{S}'_i = cl(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_i).$$

On considère aussi que  $clnd(\mathcal{A}) = \mathcal{S}''_1 \cup \dots \cup \mathcal{S}''_n$  la réparation à base de clôture d'assertions libres prioritaires définie par :  $\mathcal{S}''_1 = free(cl(\mathcal{S}_1)) = free(\mathcal{S}'_1)$ , et

$$\forall i = 2, \dots, n, \mathcal{S}''_i = free(cl(\mathcal{S}_1 \cup \dots \cup \mathcal{S}_i)) = free(\mathcal{S}'_i).$$

On considère que  $lnd(cl(\mathcal{A})) = \mathcal{S}'''_1 \cup \dots \cup \mathcal{S}'''_n$  la réparation obtenue par l'application de la réparation à base linear d'assertions libres prioritaires en  $cl(\mathcal{A})$ . Par définition de  $lnd$  réparation, nous avons :  $\mathcal{S}'''_1 = free(\mathcal{S}'_1) = \mathcal{S}''_1$ , et  $\forall i = 2, \dots, n, \mathcal{S}'''_i = free(\mathcal{S}'''_1 \cup \dots \cup \mathcal{S}'''_{i-1} \cup \mathcal{S}'_i) = free(\mathcal{S}'_i) = \mathcal{S}''_i$ . Donc :

$$\boxed{lnd(cl(\mathcal{A})) = clnd(\mathcal{A}).}$$

□

Les deux exemples suivants montrent que la réparation  $lnd$  reste incomparable avec la réparation  $\ell$  et la réparation  $clnd$ .

**Exemple 26.** On considère la base de connaissances  $\mathcal{K}$  tel que :

$$\mathcal{T} = \{A \sqsubseteq \neg B, B \sqsubseteq \neg C, A \sqsubseteq \neg D, D \sqsubseteq E\}$$

$\mathcal{A} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$  Avec :

$$- \mathcal{S}_1 = \{A(a)\},$$

$$- \mathcal{S}_2 = \{B(a), D(a)\}, \text{ et } \mathcal{S}_3 = \{C(a)\}.$$

La  $lnd$ -réparation  $lnd(\mathcal{A}) = \mathcal{S}'_1 \cup \mathcal{S}'_2 \cup \mathcal{S}'_3$  tel que :

$$- \mathcal{S}'_1 = free(\mathcal{S}_1) = \{A(a)\}$$

- $\mathcal{S}'_2 = \text{free}(\mathcal{S}'_1 \cup \mathcal{S}_2) = \emptyset$
- $\mathcal{S}'_3 = \text{free}(\mathcal{S}_3 \cup \mathcal{S}'_1 \cup \mathcal{S}'_2) = \{A(a), C(a)\}$ .

La *clnd*-réparation  $\text{clnd}(\mathcal{A}) = \mathcal{S}''_1 \cup \mathcal{S}''_2 \cup \mathcal{S}''_3$  tel que :

- $\mathcal{S}''_1 = \text{free}(\text{cl}(\mathcal{S}_1)) = \{A(a)\}$ .
- $\mathcal{S}''_2 = \text{free}(\text{cl}(\mathcal{S}_1 \cup \mathcal{S}_2))$
- $\quad = \text{free}(\{A(a), B(a), D(a), E(a)\}) = \{E(a)\}$
- $\mathcal{S}''_3 = \text{free}(\text{cl}(\mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_3)) = \{E(a)\}$

Clairement, nous avons :

$$C(a) \in \text{clnd}(\mathcal{A}) \quad \text{mais} \quad E(a) \notin \text{clnd}(\mathcal{A}).$$

De même,  $E(a) \in \text{clnd}(\mathcal{A})$  mais  $C(a) \notin \text{clnd}(\mathcal{A})$ . Alors, la réparation *lnd* est incomparable avec la réparation *clnd*. ■

**Exemple 27.** On considère la base de connaissances  $\mathcal{K}$  tel que :

- $\mathcal{T} = \{A \sqsubseteq \neg B, D \sqsubseteq \neg C\}$ , et
- $\mathcal{A} = (\mathcal{S}_1, \mathcal{S}_2)$ ,

Avec :

- $\mathcal{S}_1 = \{A(a), B(a), E(a), D(a)\}$ , et
- $\mathcal{S}_2 = \{C(a)\}$ .

nous avons :  $\ell(\mathcal{A}) = \{C(a)\}$ . D'ailleurs :

- $\text{free}(\mathcal{S}_1) = \{E(a), D(a)\}$  et
- $\text{free}(\mathcal{S}_2 \cup \text{free}(\mathcal{S}_1)) = \{E(a)\}$ .

Clairement,  $\text{lnd}(\mathcal{A}) = \{E(a), D(a)\}$ . Alors, la réparation  $\ell(\mathcal{A})$  et la réparation  $\text{lnd}(\mathcal{A})$  sont incomparables. ■

Les réparations préférées peuvent être définies en  $\langle \mathcal{T}, \mathcal{A} \rangle$  ou en  $\langle \mathcal{T}, \text{cl}(\mathcal{A}) \rangle$  tel que  $\text{cl}(\mathcal{A})$  désigne la clôture déductive d'un ensemble d'assertions  $\mathcal{A}$ . Les réparations à base de la théorie des possibilités, à base linear et à base d'assertions libres prioritaires sont sensibles à l'utilisation de la clôture déductive.

La figure suivante résume les différentes relations d'inclusion entre les différentes approches présentées dans cette thèse :

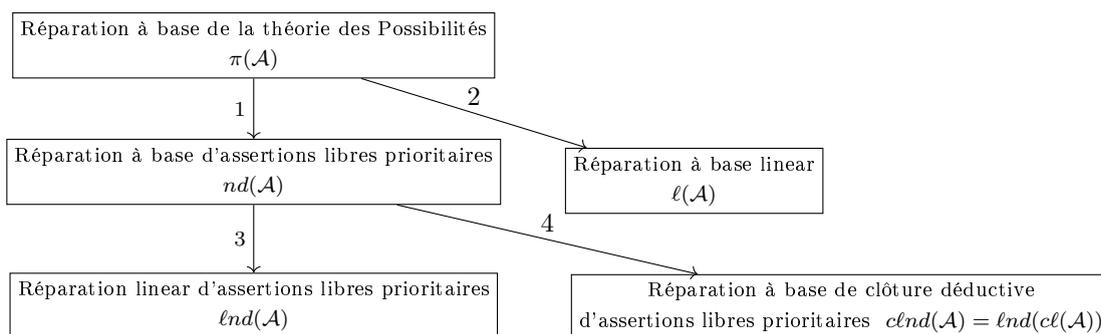


FIGURE 4.1 – Comparaison entre  $\pi(\mathcal{A})$ ,  $\ell(\mathcal{A})$ ,  $nd(\mathcal{A})$ ,  $clnd(\mathcal{A})$ , et  $\ellnd(\mathcal{A})$ .

Dans la Figure 4.1, l'arc  $X \rightarrow Y$  signifie que la réparation  $X$  est inclus dans la réparation  $Y$ .

La preuve des relations d'inclusion 1, 2 et 3 sont données dans la Proposition 6. La preuve de relation d'inclusion 4, est donnée dans la Proposition 8. Un contre-exemple pour l'inverse des relations 1, 2 et 3 est donné dans l'exemple 23. Enfin, la relation d'incomparabilité entre  $\ellnd(\mathcal{A})$  et  $clnd(\mathcal{A})$  est illustrée par l'exemple 26. D'autre part, la relation d'incomparabilité entre  $\ell(\mathcal{A})$  et  $\ellnd(\mathcal{A})$  est illustrée par l'exemple 27 de la preuve de Proposition 8.

Comme il a été indiqué précédemment,  $clnd(\mathcal{A})$  n'est pas une réparation de  $\langle \mathcal{T}, \mathcal{A} \rangle$  mais une réparation de  $\langle \mathcal{T}, cl(\mathcal{A}) \rangle$ . Par conséquent, la productivité de  $clnd(\mathcal{A})$  est considérée plus grande que les autres réparations.

La section suivante contient la description d'une application potentielle de notre approche pour classer et résoudre les conflits dans une collection de vidéos de danse provenant de différentes sources.

## 4.5 Application de réparation d'assertions libres prioritaires à traitement vidéo

Le mouvement en vidéo transporte des informations importantes qui sont de nature multiple. Il doit parfois être interprété pour comprendre ou anticiper toute réaction immédiate à la motion. Par conséquent, l'importance de la perception du mouvement dans une vidéo est perçue dans plusieurs systèmes actuels et il reste à être un domaine de recherche actif. Comme déjà motionné, les mouvements dans les danses, en particulier les danses traditionnelles, englobent différents types d'informations ; rituels sacrés, dialogue social ou expressions culturelles et plus encore. Cela nécessite de décrire et de traiter complètement et précisément les informations des mouvements et de les rendre disponibles numériquement pour un traitement ultérieur.

### 4.5.1 Contraintes de description logique pour les vidéos

Plusieurs applications peuvent être atteintes par des annotations automatiques de vidéos de danse, telles que la récupération, la classification et l'indexation de bases de données vidéo ou l'animation de danses à l'aide de modèles d'informations stockées. Nous sommes en mesure de découvrir les caractéristiques essentielles de la danse comme une action associée au mouvement [130] :

- Une vidéo est une séquence d'images qui s'étend dans le temps.
- Une fois la vidéo segmentée, les données d'annotation peuvent être extraites de la vidéo et de ses segments.
- Un ensemble de Frames peut être considéré comme un segment de danse qu'un être humain termine un pas de danse dans une danse composée.
- Une séquence d'actions affiche un mouvement isolé et constitue un vocabulaire dansant peut être considéré comme un pas de danse.
- Une improvisation dans une danse se réfère à des mouvements simultanés qui ne sont pas préalablement composés.
- Les mouvements de danse doivent être classés comme des expressions telles que les Time Steps : *Cha-Cha-Cha* ou Passo Basico de Samba dance<sup>4</sup> [129].
- Les données vidéos sont codées numériquement et la façon dont elles sont vécues par un utilisateur humain [131].

---

4. <http://www.dancadesalao.com/agenda/ingles.php>

## 4.5.2 Construire d'une ABox et TBox pour l'ontologie danse

En présence d'une collection de segments de danse, issus de différentes sources (caméras par exemple), le problème des assertions de données contradictoires peut survenir par exemple à des positions de caméra ou de, occlusions, etc. Pour illustrer cette situation, nous considérons un exemple où nous supposons que nous avons seulement les concepts suivants :

Forwardmove	définis l'ensemble des segments de danse qui représentent un mouvement vers l'avant.
Backgroundmove	définis l'ensemble des segments de danse qui représentent un mouvement vers l'arrière.
FastForwardmove	définis l'ensemble des segments de danse qui représentent un mouvement rapide vers l'avant.
SlowForwardmove	définis l'ensemble des segments de danse qui représentent un mouvement lent vers l'avant.
FastBackgroundmove	définis l'ensemble des segments de danse qui représentent un mouvement rapide vers l'arrière.
SlowBackgroundmove	définis l'ensemble des segments de danse qui représentent un mouvement lent vers l'arrière.
StartForwardmove	définis l'ensemble des segments de danse qui représentent un mouvement de commencer vers l'avant.
StartBackgroundmove	définis l'ensemble des segments de danse qui représentent un mouvement de commencer vers l'arrière.

TABLE 4.5 – Les Concepts d'ontologie danse.

Nous supposons également que nous avons deux relations :

<b>DanceSegment</b> :	donne pour chaque vidéo de danse la liste des segments de danse.
<b>DanceMessage</b> :	donne pour chaque danse ses messages transmettant.

TABLE 4.6 – Les relations d'ontologie danse.

D'autre part, la base terminologique TBox est exprimée par les axiomes suivants :

- 1) FastForwardmove  $\sqsubseteq$  Forwardmove
- 2) SlowForwardmove  $\sqsubseteq$  Forwardmove
- 3) StartForwardmove  $\sqsubseteq$  Forwardmove
- 4) FastBackgroundmove  $\sqsubseteq$  Backgroundmove
- 5) SlowBackgroundmove  $\sqsubseteq$  Backgroundmove
- 6) StartBackgroundmove  $\sqsubseteq$  Backgroundmove
- 7) StartForwardmove  $\sqsubseteq$  SlowForwardmove
- 8) StartBackgroundmove  $\sqsubseteq$  SlowBackgroundmove
- 9) Forwardmove  $\sqsubseteq \exists$  DanceSegment<sup>-</sup>
- 10) Backgroundmove  $\sqsubseteq \exists$  DanceSegment<sup>-</sup>
- 11) Forwardmove  $\sqsubseteq \neg$  Backgroundmove

TABLE 4.7 – La base terminologique d'ontologie danse.

L'axiome 1 exprime qu'un mouvement d'avance rapide est un mouvement vers l'avant.

L'axiome 2 exprime qu'un mouvement d'avance lent est un mouvement vers l'avant.

L'axiome 3 exprime qu'un mouvement de départ en avant est un mouvement vers l'avant.

L'axiome 4 exprime qu'un mouvement d'arrière rapide est un mouvement vers l'arrière.

L'axiome 5 exprime qu'un mouvement d'arrière lent est un mouvement vers l'arrière.

L'axiome 6 exprime qu'un mouvement de départ en arrière est un mouvement vers l'arrière.

l'axiome 7 exprime qu'un mouvement de départ en avant doit être un mouvement lent vers l'avant .

l'axiome 8 exprime qu'un mouvement de départ en arrière doit être un mouvement lent vers l'arrière.

l'axiome 9 exprime que chaque mouvement vers l'avant est contenu dans au moins un segment de danse.

l'axiome 10 exprime que chaque mouvement vers l'arrière est contenu dans au moins un segment de danse.

Le dernier axiome exprime qu'un mouvement vers l'avant n'est pas un mouvement vers l'arrière.

Supposons maintenant que nous avons quatre individus :

- i) Deux segments de danse  $s_1$  et  $s_2$ , et
- ii) Deux vidéos  $v_1$  et  $v_2$ .

Supposons aussi, que nous avons quatre faits d'assertions donnés par l'ABox  $\mathcal{A} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$  avec :  $\mathcal{S}_1 = \{ \text{Backgroundmove}(s_1), \text{DanceSegment}(v_1, s_1) \}$ ,  $\mathcal{S}_2 = \{ \text{StartForwardmove}(s_1), \text{DanceSegment}(v_1, s_2) \}$ , et  $\mathcal{S}_3 = \{ \text{SlowBackgroundmove}(s_1), \text{FastBackgroundmove}(s_2) \}$ .

Cette base d'assertion est supposée être fournie par différentes sources (différentes caméras par exemple) ayant des niveaux de fiabilité différents. Cette base de connaissances est clairement inconsistante. En effet, en utilisant l'assertion  $\text{StartForwardmove}(s_1)$  et l'axiome d'inclusion positive 3 on peut dériver  $\text{Forwardmove}(s_1)$ . Ce fait dérivé avec l'assertion  $\text{Backgroundmove}(s_1)$  contredit l'axiome négatif 11.

L'ensemble d'assertions libres (A savoir, seules les assertions qui ne sont pas concernées par le conflit sont préservées)  $free(\mathcal{A})$  :

$$\{ \text{DanceSegment}(v_1, s_1), \text{DanceSegment}(v_1, s_2), \text{FastBackgroundmove}(s_2) \}.$$

Lorsqu'il y a une relation prioritaire entre les assertions, on peut aller plus loin dans la résolution des conflits basée sur la caractérisation de la réparation à base d'assertions libres prioritaires proposée dans 3.3.3 en utilisant la notion d'assertions libres prioritaires. Dans notre exemple,  $\text{Backgroundmove}(s_1)$  est supposé provenir d'une source plus fiable que celle qui fournit l'assertion  $\text{StartForwardmove}(s_1)$ . Par conséquent,  $\text{Backgroundmove}(s_1)$  est une assertion libre prioritaire. Donc, l'ensemble d'assertions libres prioritaires  $nd(\mathcal{A})$  :

$$\{ \text{Backgroundmove}(s_1), \text{DanceSegment}(v_1, s_1), \\ \text{DanceSegment}(v_1, s_2), \text{FastBackgroundmove}(s_2) \}$$

. Noter que,  $nd(\mathcal{A})$  est plus grand que  $free(\mathcal{A})$  puisqu'il permet de récupérer  $\text{Backgroundmove}(s_1)$ , en raison de la présence de priorité dans la base d'assertions. La caractéristique intéressante de notre approche est qu'une fois que  $nd(\mathcal{A})$  est calculé, interroger la base de connaissances inconsistante initiale peut être effectuée efficacement.

En fin, nous considérons cet exemple comme une application potentielle de notre approche de classification des vidéos de danse en présence d'informations contradictoires multi-sources. Un travail futur consiste à appliquer notre approche pour interroger les vidéos de danse traditionnelle vietnamienne en présence de conflits possibles.

## 4.6 Conclusion

Dans ce chapitre de résultats, des tests de performance ont été réalisés sur les algorithmes proposés qui sont utilisés pour sélectionner une seule réparation préférée à partir d'une base de connaissances *DL-Lite* inconsistante avec priorité. Nous avons évalué la productivité, la complexité de calcul et le temps nécessaire pour chaque approche.

Dans le contexte d'ontologies légères inconsistantes, les algorithmes de réparation basée sur les assertions libres prioritaires (*nd*, *lnd*, *clnd*) n'ont pas été proposés auparavant. En particulier, le concept de clôture déductive qui est utilisée dans ce document, pour définir de nouvelles approches tolérantes aux inconsistances, n'a pas été envisagé auparavant.

# Chapitre 5

## AniAge : Projet d'ontologie de danses traditionnelles

*Computers are magnificent tools for the realization of our dreams, but no machine can replace the human spark of spirit, compassion, love, and understanding.*

*Louis V. Gerstner, Jr.*

### 5.1 Introduction

Les technologies de l'animation par ordinateur ont connu un essor considérable et elles ont été largement utilisées pour les films et les jeux vidéo. Ces technologies requièrent cependant beaucoup d'efforts et de travail manuel et elles sont très coûteuses. Il est primordial pour pouvoir utiliser ces technologies dans des applications (comme l'archivage et la simulation ou la reproduction de contenus du patrimoine culturel immatériel) et de proposer des solutions d'animation par ordinateur efficaces et moins coûteuses.

Dans ce chapitre, nous présentons un processus de création d'une ontologie pour les danses traditionnelles vietnamiennes dans le cadre de projet AniAge.

## 5.2 AniAge : Projet d'ontologie pour représenter les danses traditionnelles

Le projet AniAge (High Dimensional Heterogeneous Data based Animation Techniques for Southeast Asian Intangible Cultural Heritage Digital Content) est un projet pluridisciplinaire qui implique des chercheurs en Intelligence Artificielle et des chercheurs en animation par ordinateur.

Le projet AniAge est basé sur trois objectifs ; le premier objectif consiste à intégrer des outils de l'Intelligence Artificielle dans l'animation par ordinateur. En particulier, de proposer des solutions originales pour le traitement de données et d'informations hétérogènes, la classification à partir de données incertaines, l'analyse de données volumineuses et le raisonnement avec des ontologies.

Le deuxième objectif du projet est d'appliquer les résultats théoriques au domaine du patrimoine culturel immatériel d'Asie du Sud-Est qui est l'une des régions les plus dynamiques dans le monde avec un patrimoine culturel très riche. Une partie de ce patrimoine est présenté dans les musées, les temples et sites archéologiques, mais une partie importante est présente dans la vie quotidienne. En effet, des formes diverses du patrimoine culturel se transmettent des ancêtres aux nouvelles générations. Ce patrimoine est visuellement et culturellement riche, mais il est, malheureusement, en voie de disparition en raison de la modernisation et de la mondialisation.

Le dernier objectif d'AniAge est de promouvoir l'échange de connaissances entre l'UE (Union Européenne) et ses partenaires d'Asie du Sud-Est en vue de favoriser la mobilité et l'échange des chercheurs et la diffusion des connaissances. Cet objectif sera atteint en mobilisant des chercheurs des deux côtés autour des projets de recherche d'excellence, d'ateliers, de séminaires et de conférences.

Le consortium du projet AniAge est composé de six partenaires : NCCA (National Centre for Computer Animation, Bournemouth University, UK)<sup>1</sup>, coordinateur du projet, CRIL (Centre de Recherche en Informatique de Lens, CNRS UMR 8188 et Université d'Artois, France), HMI (Human Machine Interaction Lab at Vietnam National University, Vietnam)<sup>2</sup>, CAMT (College of Arts Media and Technology at Chiang Mai

---

1. <https://ncca.bournemouth.ac.uk/>

2. <http://www.coltech.vnu.edu.vn/hmi/>

University, Thaïlande)<sup>3</sup>, CICT (College of ICT at Can Tho University, Vietnam)<sup>4</sup> et ViCube (Vision, Virtual and Visualization Lab at University of Technology, Malaysia)<sup>5</sup>.

Pour l'implémentation du projet, quatre work-packages sont prévus. Le premier concerne la capture et le traitement des données qui sont de nature hétérogène et gigantesque en quantité. Le deuxième et le troisième work-packages concernent les développements technologiques à base de l'Intelligence Artificielle et l'animation par ordinateur (évidemment, c'est notre travail). Le dernier work-package est dédié à la diffusion et la dissémination des résultats du projet.<sup>6</sup>

### 5.3 Approche proposée pour développer une ontologie de danse Vietnamiennne

Une ontologie pour les danses folkloriques vietnamiennes est proposé dans la Figure 5.1. Toutefois, nous appliquons la division de Labanotation [124] qui est basée sur les parties du corps dans l'analyse de la danse. Ces parties enregistrent l'idée générale derrière la structure hiérarchique des mouvements de danse et les connaissances de l'expertise du domaine de la danse folklorique. D'autres parties décrivent précisément les éléments de mouvement : les parties du corps, le temps, la direction et la dynamique.

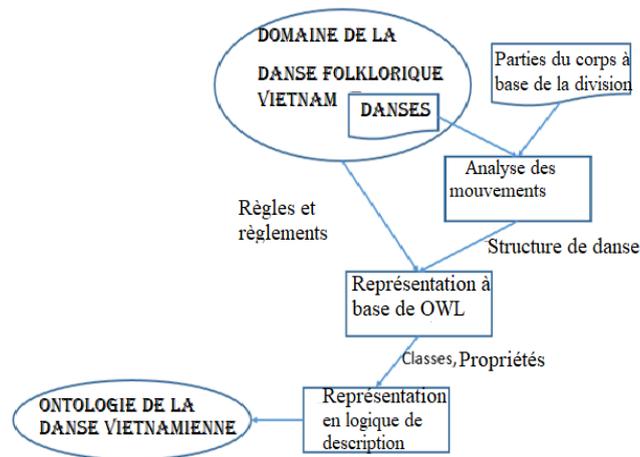


FIGURE 5.1 – Construction d'une ontologie pour danse folklorique vietnamiennne

3. <http://www.camt.cmu.ac.th/en/index.php>
4. <https://ctu.edu.vn/en/>
5. <http://comp.utm.my/vicubelab/>
6. <http://www.cril.univ-artois.fr/aniage/>

Tous considèrent essentiellement le processus de développement d'ontologie identifié par [132] basé sur les étapes suivantes : définition du but de l'ontologie, conceptualisation, et formalisation [134].

**Objectif de l'ontologie** : les ontologies peuvent apparaître comme des moyens utiles pour structurer des descriptions du contenu sémantique vidéo. Ils peuvent prendre en charge des descripteurs sémantiques pour des images, des sons ou d'autres objets. Nous utilisons la solution ontologique afin de réaliser une annotation efficace du contenu vidéo.

**Conceptualisation** : nous commençons en définissant les composants vidéos. Dans ce travail, nous avons segmenté l'entrée vidéo en séquences. Les caractéristiques visuelles seront utilisées pour associer une description à chaque séquence. Chaque séquence consiste en plusieurs poses particulières.

**Formalisation** : notre ontologie a été formalisée en utilisant OWL et Protege.4.3. Il peut être facilement réutilisé et partagé. Pour formaliser dans OWL la composition des mouvements, nous exploitons la méthode définie par [133].

### 5.3.1 Rassembler et analyser des connaissances d'experts

Au Vietnam, 54 groupes ethniques [125] ont leurs propres danses folkloriques, qui expriment la connaissance culturelle, la vie spirituelle, reflétant la créativité et le talent du peuple vietnamien. Les groupes ethniques, se regroupant géographiquement, ont des coutumes similaires. Par conséquent, les danses folkloriques des 54 ethnies vietnamiennes peuvent être classées en 7 groupes principaux [126] :

- Hauts plateaux midland région du nord (Highland-midland northern region).
- Delta de la rivière Rouge (Red river delta).
- Centre-Nord (North Central).
- Région côtière du centre-sud (Coastal South Central).
- Hauts plateaux (Hightlands).
- Sud-est (South east).
- Sud-ouest (South west).



FIGURE 5.2 – Groupes ethniques de danses folkloriques Vietnamiennes

À travers les danses, les gens veulent transmettre une expérience de travail productif; chasser et montrer le comportement des êtres humains, comme la danse de la voile (múa chèo thuyền), et la danse du tissage (múa dệt cửi) [126]. Nous identifions les caractéristiques régionales et les messages transmis dans une danse, basés sur de nombreux aspects tels que la posture et le mouvement de danse, les vêtements, les accessoires de danse et la musique. Dans la première phase, nous nous concentrons uniquement sur la représentation et l'analyse des aspects du mouvement de danse folklorique de type Mỗ qui appartient à la région Red river delta.

Historiquement, Dans la vie rurale de l'ancien vietnamien, il y avait un homme appelé Mỗ. Lors des occasions ou des événements du village, Mỗ battait l'instrument Mỗ et informait les villageois. Les gens mettent Mỗ sur le col de buffle. Lorsque les buffles se déplacent, marchent, deux morceaux de bois frappent régulièrement sur le mur intérieur et émettent un son intéressant. Ceci simule comment les gens battent l'instrument Mỗ pour faire un rythme et un son intéressant.

### 5.3.2 Analyser des mouvements dans les danses vietnamiennes

Une danse folklorique vietnamienne comprend des unités de base [127]. Chaque unité de base est définie comme le plus petit mouvement avec une signification complète. Il existe deux principaux types de mouvements, correspondant à des actions de base : [124]

- Les actions de tout le corps (Mouvements corporels) : changeant la position de tout le corps du danseur dans l'espace ; les danseurs se déplacent sur scène. Ils sont composés de phrases de mouvement ; les phrases de mouvement incluent le déplacement (un point), la translation (une ligne) et la rotation (un arc).
- Les actions de certaines parties du corps (mouvements de la partie du corps) : changeant la position des différentes parties du corps du danseur.

Dans les danses folkloriques du Vietnam, il y a huit orientations de mouvements, définies comme suite (Voir Figure 5.3) :

- Orientation 1 : correspondant à l'avant.
- Orientation 2 : correspondant à la diagonale avant droite.
- Orientation 3 : correspondant à la droite.
- Orientation 4 : correspondant à la diagonale arrière droite.
- Orientation 5 : correspondant à l'arrière.
- Orientation 6 : correspondant à la diagonale arrière gauche.
- Orientation 7 : correspondant à la gauche.
- Orientation 8. correspondant à la diagonale gauche avant.

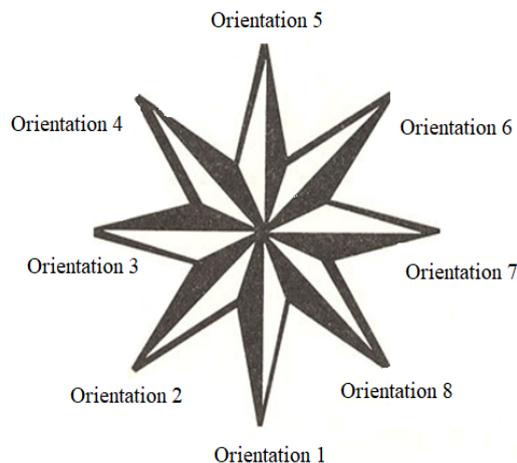


FIGURE 5.3 – Les orientations des parties du corps.

La danse folklorique traditionnelle du Vietnam est différente des danses modernes dans le mouvement du corps. Il n'y a pas des mouvements corporels qui soulèvent les danseurs. Ainsi, les mouvements du corps dans la danse ne sont pas trop compliqués. Pour ce-la, nous définissons les paramètres des mouvements suivants :

- Chaque phrase de mouvement est un simple mouvement corporel, qui a une trajectoire en forme de ligne, un arc, ou un point sur un plan comme un chemin droit, un chemin incurvé et une immobilité comme dans la Labanotation [124].
- Deux types de rotation sont distinguées : dans le sens des aiguilles d'une montre et dans le sens contraire des aiguilles d'une montre. La phrase localisée mobile peut également être dans le tour (avec différents degrés) ou sans virage.
- Parallèlement à la mise en œuvre des phrases corporelles, les danseurs exécutent des mouvements de la partie du corps, appelés *mouvements primitives*.
- Les mouvements primitives : sont des mouvements rapides qui modifient la position des parties du corps. C'est un mouvement entre deux poses de danse.
- Les mouvement primitives comprennent les mouvements de la main, les mouvements des pieds, les mouvements des jambes, les mouvements de la tête et les mouvements bras-jambes.
- Une pose de danse est une position particulière du corps du danseur. Il y a des poses de base pour la tête, des poses de base pour les mains, des poses de bras basiques, des poses de base pour les jambes et des poses de base pour les bras et les jambes combinées.
- Pour décrire les poses élémentaires de la partie du corps, nous utilisons les orientations de danse, des angles entre les bras et jambes avec le torse, et des angles entre les membres des bras ou des jambes.
- Comme mentionné ci-dessus, il existe deux types de mouvements liés aux parties du corps. Par conséquent, il est nécessaire de distinguer l'orientation du corps ou des parties du corps.

Nous distinguons quatre mouvements principaux. Chaque catégorie des mouvements possède ses actions de base. Nous offrons la plus importante :

- **Mouvements de bras** ; contient l'ensemble des actions des bras :
  - *Tissage (Weaving)* : déplacer les bras vers l'intérieur et l'extérieur de 20 cm devant le corps avec la main gauche au-dessus de la main droite ;
  - *Roulement du poignet (Wrist rolling)* : tourner le poignet dans le sens des aiguilles d'une montre.
  - *Doigt roulant (Finger rolling)* : les poignets se déplacent comme Wrist rolling de l'index vers l'auriculaire.
  - *Battement (Flapping)* : déplacer les bras ; la gauche à l'orientation 3 et la droite à l'orientation 4.

- **Mouvements de la jambe** ; contient l'ensemble des actions de la jambe :
  - *Balançoire douce (Soft Swing)* : plier les genoux et soulever le corps par les doigts des pieds.
  - *Déplacement rhombique (Rhombic Moving)* : mettez le talon gauche avant / arrière et éloignez 30 cm les orteils droits sur la droite, puis tendez les orteils gauches au sol doucement.
  
- **Mouvements bras-jambes** ; correspond à l'ensemble des actions suivantes :
  - *Deux-mains-Cueillette* : Les genoux pliés et déplacer le bras gauche vers le haut où l'avant-bras horizontalement jusqu'au genoux droits ; appliquer Wrist rolling.
  - *Pas de rotation horizontale* : déplacer les jambes de l'orientation 1 à 3 ou de l'orientation de 1 à 7 avec genoux fléchis légèrement.
  
- **Mouvements de danse de base** :
  - *MO inviter (Mõ mòi)* : les bras éloignés de 3 cm en face du torse, du biceps et de l'avant-bras sont orthogonaux,  
Répétition : la jambe droite / gauche passe en avant avec le talon touche le sol, le bras droit / gauche aussi tout droit le long de l'orientation de la jambe avec la main ouverte. Le corps peut tourner n'importe quelle orientation, vers l'arrière ou vers l'avant
  - *MO Pied-glisser (Mõ xet chôn)* : déplacer la jambe droite 40 cm vers l'orientation 2, puis glisser la jambe gauche à côté de la jambe droite. De même, la jambe gauche avec orientation 8.
  - *MO Doigt de pied-touchant (Mõ chôn chôn)* : la jambe droite/gauche à l'orientation 1, le genou gauche/droit avec les orteils touche le sol, les bras applaudissent en orientation 8/2, la tête maigre en orientation 8/2.

### 5.3.3 Analyse sémantique d'un mouvement de la danse Mõ

#### 5.3.3.1 Outils d'implémentation

Nous utilisons Protégé<sup>7</sup> pour implémenter notre approche de représentation de danse traditionnelle vietnamienne. Protégé est un outil open source spécialement dédié à OWL. Il a été développé par le centre de recherche en informatique biomédicale à

---

7. <https://protege.stanford.edu/>

l'école de médecine de l'université de Stanford.

Des applications développées avec Protégé sont employées dans la résolution des problèmes et la prise de décision dans un domaine particulier. Protégé est aussi une plate-forme extensible permet de gérer des contenus multimédias, interroger, évaluer et fusionner des ontologies. L'interface graphique de Protégé permet de manipuler aisément tous les éléments d'une ontologie (Voir Figure 5.4).

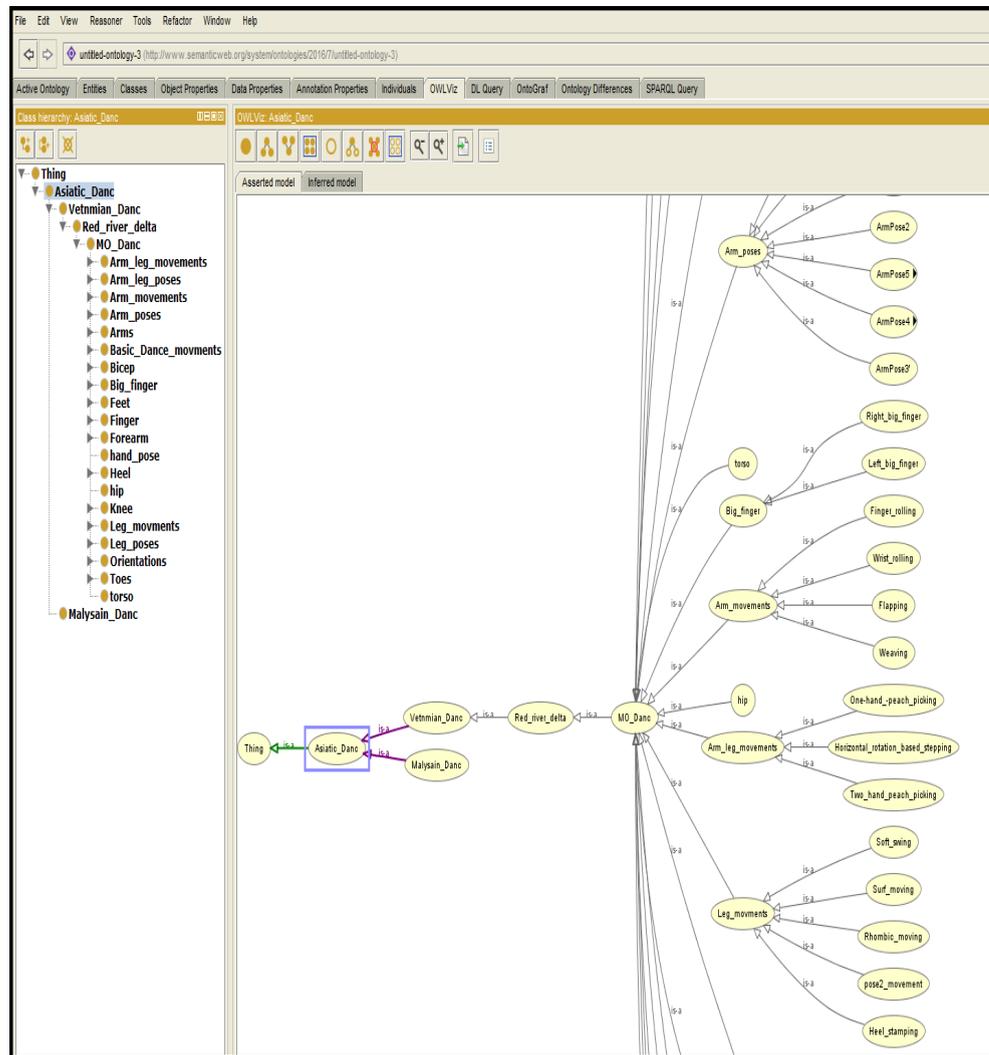


FIGURE 5.4 – Interface graphique de protégé 2000.

Protégé permet aussi de créer ou d'importer des ontologies écrites dans les différents langages d'ontologies tel que : RDFs, OWL, DAML, et OIL. La version Protégé 4.3 offre la possibilité de créer les ontologies utilisées dans notre système avec l'hierarchie des classes, des propriétés et des relations entre les différents éléments de l'ontologie.

### 5.3.3.2 Ontologie de danse

Formellement, une danse est typique du mouvement humain ; le mouvement humain est une connaissance quand nous pouvons utiliser une ontologie pour le modéliser. Nous proposons d'utiliser les technologies de l'ontologie pour représenter et raisonner les chorégraphies de danse en construisant une ontologie de danse par OWL. Cette description logique de danse nous permet d'exprimer des relations complexes et des règles d'inférence pour le domaine du mouvement humain. Cependant, les capacités de raisonnement facilitent l'extraction de nouvelles connaissances à partir des connaissances existantes (Voir Figure 5.5).

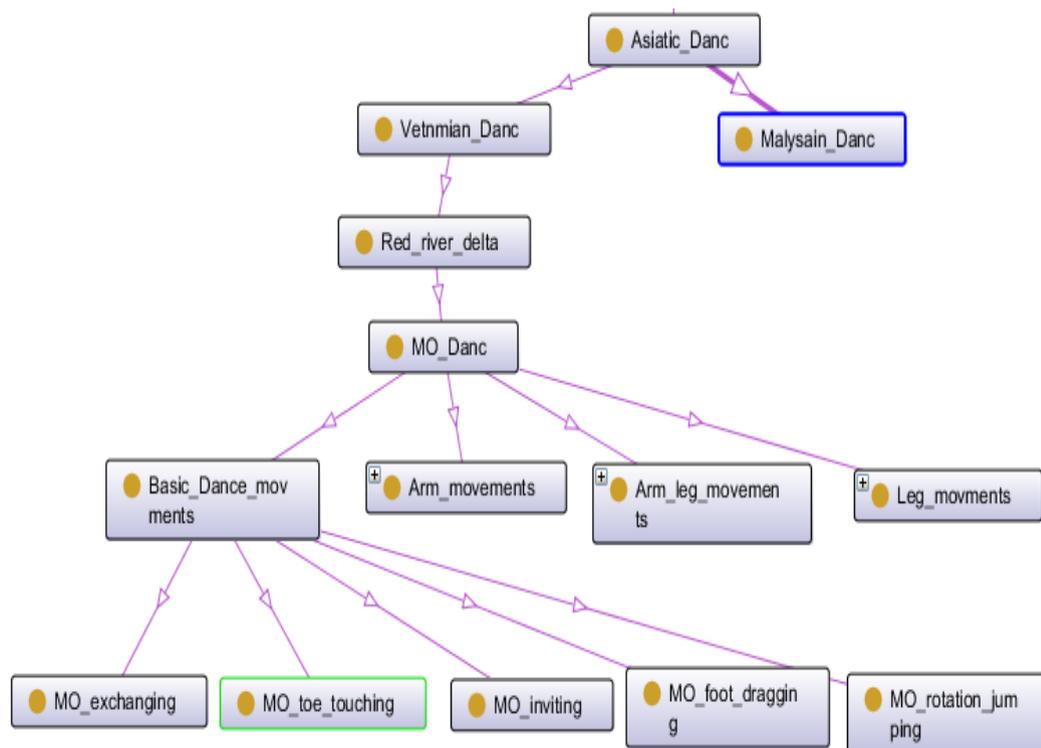


FIGURE 5.5 – Ontologie de domaine de danse traditionnelle.

Notre but est d'explorer comment les différentes étapes de danses traditionnelles vietnamiennes dans une vidéo peuvent être catégorisées et décrites pour extraire les connaissances de la vidéo. Dans la danse Mõ de la région *Red river delta*, nous pouvons décrire l'action *MO foot-dragging* par les images 5.6, 5.7, 5.8, 5.9 et 5.10, qui illustrent les différentes positions des pieds à partir de la position initiale jusqu'à la pose finale (la fin d'action) :

1. Commencer par la position initiale (Voir Figure 5.6).



FIGURE 5.6 – Position initiale de danse Mõ.

2. Bouger la jambe droite 40 cm dans l'orientation 2 : Pied droit dans l'orientation 2; pied gauche en orientation 8 avec talon juste avant les orteils droits (Voir Figure 5.7)



FIGURE 5.7 – Position 1 de la jambe.

3. Glisser la jambe gauche à côté de la jambe droite : pied droit dans l'orientation 2; pied gauche dans l'orientation 8; talon droit est au milieu du pied gauche (Voir Figure 5.8)



FIGURE 5.8 – Position 2 de la jambe.

4. Retour du pied droit derrière le pied gauche : pied gauche dans l'orientation 8, le talon droit est levé dans l'orientation 1 ; les orteils droits sont au talon gauche ; le genou droit est plié (Voir Figure 5.9)



FIGURE 5.9 – Position 3 de la jambe.

5. Les deux pieds dans l'orientation 1 ; le pied gauche devant (20 cm) le côté droit (20cm) du pied droit ; le talon droit est légèrement levé (Voir Figure 5.10)



FIGURE 5.10 – Position 4 de la jambe.

La combinaison en parallèle entre les actions des pieds avec les actions des mains constitue un mouvement corporel complet. Noter que, certaines actions s'exécutent d'une manière répétitive c'est ce que nous appliquons dans l'annotation des mouvements.

Maintenant, nous présentons les séquences des positions des mains pour le mouvement *MO foot-dragging* de danse Mō. Toutes les descriptions données dans les images 5.11, 5.12 et 5.13 sur le mouvement de la main gauche s'appliquent à la main droite :

1. Position initiale : bras gauche dans l'orientation 8 ; bras droit dans l'orientation 2 ; les bras sont levés en forme hexagonale (les bras se croisent devant le torse ; 3 cm du torse) (Voir Figure 5.11).



FIGURE 5.11 – Position initiale des mains.

2. Main ouverte et orthogonal avant-bras ; les doigts de l'index à l'auriculaire sont droits ; gros doigt orthogonal aux autres doigts (Voir la Figure 5.12).



FIGURE 5.12 – Position 01 des mains.

3. Main ouverte et orthogonale avant-bras ; les doigts d sont droits ; le gros doigt orthogonal aux autres doigts qui sont orientées vers la tête (Voir la Figure 5.13).



FIGURE 5.13 – Position 02 des mains.

### 5.3.3.3 Concepts et propriétés

Nous décrivons notre approche par la représentation de la danse traditionnelle vietnamienne Mõ en utilisant le *DL-Lite*. Premièrement, nous considérons que le corps humain est divisé en parties (chaque partie correspondre une classe) : Genou, avant-bras, Biceps, Doigt, Gros doigt, Bras, Orteils, Talon et Pieds. Chaque partie a son côté droite et son côté gauche (sous-classe).

Deuxièmement, le danse Mõ comprend des concepts, des propriétés de concepts et des relations entre les concepts, pour fournir une description formelle d'un domaine. Il procure un vocabulaire commun qui surmonte l'hétérogénéité sémantique de l'information (Voir Figure 5.14).

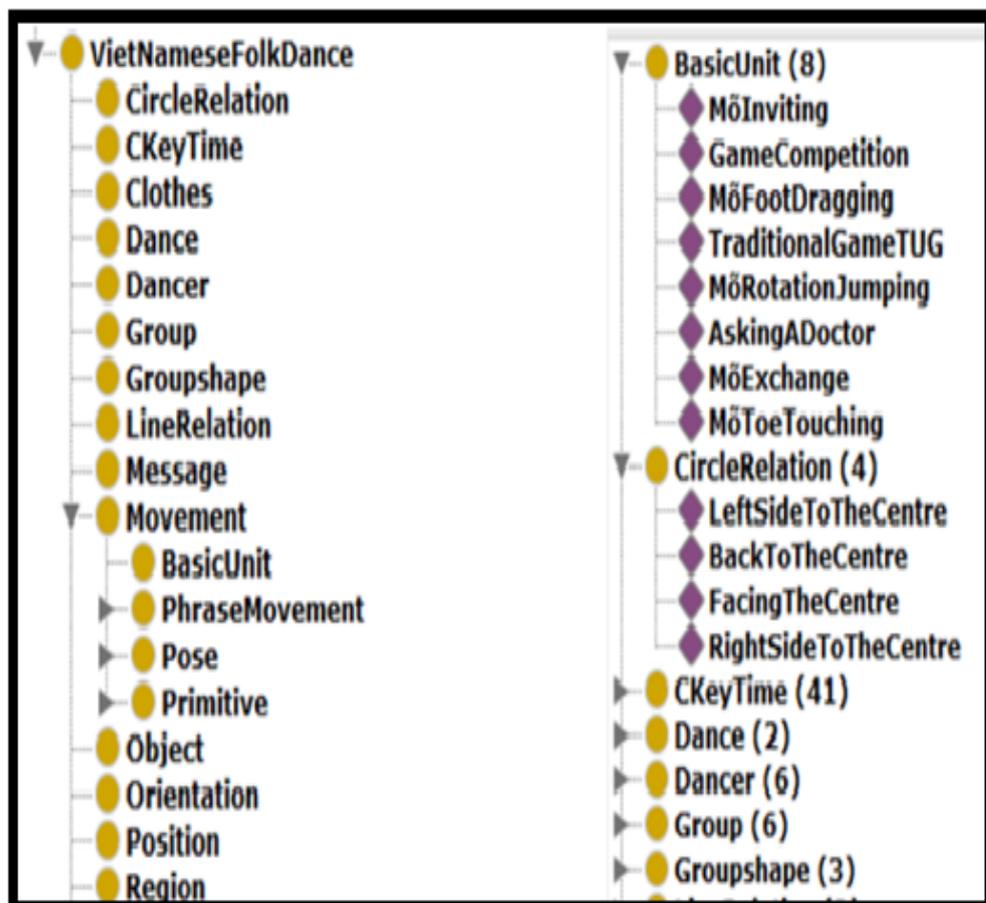


FIGURE 5.14 – Classes et sous classes des mouvements de base pour la danse Mõ.

Nous expliquons certaines des classes les plus éminents d'ontologie de danse :

**Mouvement** : Cette classe peut être considérée comme la racine de l'aspect mouvement (lié au corps et à la partie du corps) de la danse. Il code les informations sur les mouvements de phrases, les primitives et les poses.

**Région, message et vêtements** : Ces classes représentent l'information régionale, le contenu de la danse et les message transmis.

**Group, Groupshape, LineRelation et CircleRelation** : ces classes encodent l'information des groupes de danseurs en danse.

La classe **CircleRelation** contient les quatre direction d'une Cercle de danse : de gauche / droite vers le centre, et de arrière / avant vers la centre.

**CKeyTime, Orientation et Position** : ils représentent la position et l'orientation de corps ou du partie du corps dans à chaque moment.

La sous classe **Unité de base** (*Basic Unit*) de la classe Mouvement contient les actions basiques de chaque mouvement (Voir la section 5.3.2).

#### 5.3.3.4 Relations et interprétations

Il existe certains types de relations entre les classes de danse :

- Le premier concerne la réalisation des poses et de mouvements différents.
- Le second et le troisième concernent les groupes et les régions des danseurs.
- Le quatrième concerne la relation spatiale avec la scène pour les danseurs et les groupes de danseurs.
- Le dernier traite la relation entre le contenu d'une danse folklorique et sa région.

Le tableau 5.1 montre les prédicats typiques et leur explication de ces relations :

Relation	Explication
HasSegment(a,b)	Mouvement (a) appartient au segment (b)
MemberOf(a,b)	Danseur (a) est membre du groupe (b)
BelongToRegion(a,b)	Danse (a) appartient au région (b)
PosDance(a,b)	Danse (a) est à la position (b)
HasMessage(a,b)	Danse (a) transmet le message (b)

TABLE 5.1 – Relations entre les classe de la danse Mõ.

## **5.4 Conclusion**

La création d'une ontologie de danse est une tâche très compliquée. D'une part, elle exige un travail supplémentaire pour le traitement d'images et la segmentation de séquences vidéo. D'autre part, une interprétation culturelle pour chaque partie de la danse doit être représentée.

Dans ce chapitre, nous avons construit une ontologie de danse folklorique vietnamienne en définissant des composants de danse, par la segmentation en des unités de base. Une unité de base consiste en une ou plusieurs phrases de mouvement. La phrase de mouvement contient plusieurs mouvements primitifs. Un mouvement primitif a au moins deux poses de base avec une durée entre la première pose et la dernière pose.

# CONCLUSION GÉNÉRALE

*Failure is success if we learn from it.*

*Malcolm Forbes*

*Success is simple. Do what's right, the right way, at the right time.*

*Arnold H. Glasow*

LES ontologies ont pris une place centrale dans les applications basées sur les connaissances. Elles permettent de capitaliser, de représenter, d'exploiter et de partager sémantiquement des connaissances et des informations. Les travaux qui ont été menés dans le cadre de cette thèse se situent dans le domaine de la représentation des connaissances. D'abord, nous avons présenté les logiques de descriptions comme un formalisme permettant la représentation formelle et le raisonnement sur les ontologies. Nous avons introduit la syntaxe et la sémantique des extensions principales de la famille de *DL-Lite*. Cette famille a un énorme intérêt suscité par la problématique d'accès aux données via une ontologie (OBDA). Ensuite, nous avons étudié le débogage et les traitements sur les bases de connaissances *DL-Lite* avec multiple ABox. Nous avons généralisé les techniques de gestion d'inférence, et étudié les propriétés de la complexité de calcul de conflits dans les bases de connaissances *DL-Lite* avec multiple ABox.

Nous nous sommes intéressés à la résolution de problème de l'inconsistance dans *DL-Lite* stratifié. Pour cette raison, nous avons examiné certaines approches connues qui permettent de sélectionner une base de réparation. Puis, nous avons proposé des stratégies pour sélectionner une seule réparation assertionnelle. Ces stratégies ont pour point de départ la réparation à base d'assertions libres prioritaires. Les critères de fermeture déductive et de consistance sont rajoutées afin d'améliorer l'expressivité de la réparation à base d'assertions libres prioritaires. A la fin de cette thèse, nous avons fourni des résultats de calcul de la complexité et des études expérimentales montrant l'efficacité des stratégies proposées.

## Travaux futurs

D'autres stratégies pour sélectionner une seule réparation qui ont comme point de départ la réparation à base d'assertions libres consistent à ajouter le critère de priorité, ou la cardinalité (inférence lexicographique). Ce critère a été largement utilisé dans le cadre de la logique propositionnelle [110]. Ces stratégies de gestion de l'inconsistance sont propres à la logique *DL-Lite* et elles permettent une manipulation facile des inconsistances.

Comme perspectives à notre travail, nous proposons de l'améliorer par des travaux futurs afin d'obtenir des relations plus productives et de nous fournir que des résultats sûrs dans un temps polynomial. Ces travaux futurs sont comme suit.

**La gestion de l'incertitude dans *DL-Lite* au niveau de base terminologique :** c'est une autre forme d'inconsistance en présence de degrés de certitude par l'utilisation de la théorie des possibilités dans *DL-Lite*. Cela dépend d'autres étapes de la modélisation de la connaissance. Par exemple, dans la théorie des probabilités, l'assertion probabiliste suivante  $\langle B \sqsubseteq C, \alpha \rangle$  signifie que la probabilité conditionnelle de l'axiome  $B \sqsubseteq C$  est égale à  $\alpha$ .

**La gestion de l'inconsistance dans *DL-Lite* avec multiple TBox :** la question est alors de savoir comment étendre les concepts de conflits, associés à des axiomes négatifs pour définir des degrés d'inconsistance dans des bases de terminologie multiple. Cette extension est utile lorsque l'on doit fusionner plusieurs sources d'informations, où l'union du TBox (émis de chaque source) est consistante, mais l'ensemble des TBox est contradictoire.

**La gestion de l'inconsistance dans *DL-Lite* au niveau de requêtes :** toutes les stratégies proposées calculent la réparation avant le lancement des requêtes. Une idée intéressante consiste à calculer la réparation d'une base de connaissances *DL-Lite* en se basant sur les requêtes afin de minimiser la taille de base des connaissances réparées.

# Annexe A

*Anyone who has ever made anything of importance was disciplined.*

*Andrew Hendrixson.*

*Creativity is intelligence having fun.*

*Albert Einstein*

*Sometimes you win, sometimes you learn.*

*John Maxwell.*

## Outil d'ontologie *DL-Lite*

Dans cette annexe, nous présentons un outil de représentation et de raisonnement dans le cadre *DL-Lite*. Pour cela, nous avons utilisé les modules et les classes programmées par [123] dans l'outil de représentation et de raisonnement dans le cadre de possibiliste *DL-Lite*. Cet outil est développé en JAVA et implémente l'algorithme de vérification de l'incohérence de la base de terminologie TBox. Notre addition consiste à calculer la réparation d'une base de connaissance *DL-Lite* en utilisant les algorithmes développés dans le chapitre 3. Dans cet outil, la ABox est gérée à l'aide du moteur de base de données *SQL*. Nous utilisons aussi, la syntaxe fonctionnelle *OWL-QL* comme un langage ontologique.

En fait, cette application n'utilise pas les outils de programmation d'ontologie existants tels que : Jena et les API *OWL* (voir la section 1.4.6). Par conséquent, nous avons développé notre analyseur syntaxique. On peut lire tout types fichier *OWL-QL* standard, édité à l'aide d'un éditeur d'ontologie standard tel que Protégé<sup>8</sup> (voir la section 1.2.5).

Enfin, nous avons adapté ce codage dans le but d'ajouter d'autres capacités de raisonnement en cas de priorité entre les ABoxes.

Dans ce qui suit, nous présentons quelques captures d'écran de cet outil suivies d'une explication de ses fonctionnalités.

## Interface principale

La Figure 5.4 présente l'interface principale de notre éditeur d'ontologie *DL-Lite*. Cette interface est partitionnée en deux panneaux : l'un pour spécifier la base TBox et l'autre pour exprimer les bases ABoxes. Chaque partition de l'interface principale possède de différents boutons fonctionnels. L'utilisation des ces boutons de la barre d'outils permette d'exprimer les axiomes de *DL-Lite<sub>core</sub>*, *DL-Lite<sub>F</sub>* et *DL-Lite<sub>R</sub>*.

---

8. <http://protege.stanford.edu/>

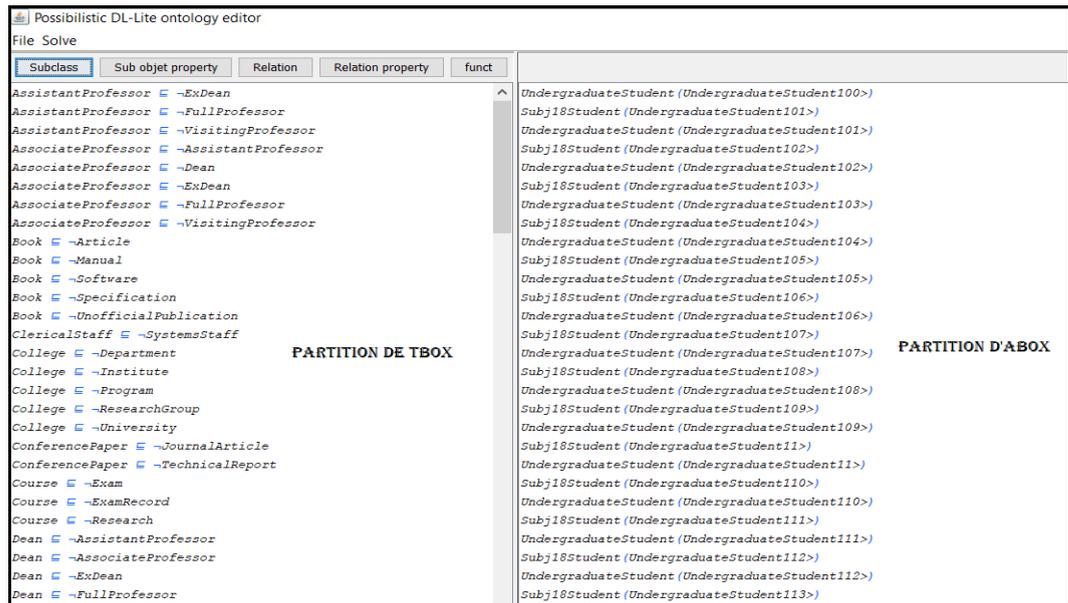


FIGURE 2 – Interface principale

La TBox exportée dans l'éditeur, est représentée en utilisant la syntaxe fonctionnelle de *OWL-QL* comme suit :

```

Declaration(Class(Person))
Declaration(Class(Professor))
Declaration(Class(Article))
Declaration(Class(AssociateProfessor))
Declaration(Class(Publication))
Declaration(ObjectProperty(advisor))
Declaration(ObjectProperty(memberOf))
Declaration(ObjectProperty(member))
Declaration(ObjectProperty(worksFor))
Declaration(DataProperty(age))
DisjointClasses(AssistantProfessor FullProfessor)
SubClassOf(Professor Person)
SubClassOf(Article Publication)
ObjectPropertyDomain(advisor Person)
ObjectPropertyRange(advisor Professor)
InverseObjectProperties(memberOf member)
SubObjectPropertyOf(worksFor memberOf)
DataPropertyDomain(age Person)

```

De même, la ABox est exprimé à l'aide d'assertions et des relations gérées par le moteur SQL-lite<sup>9</sup>, comme suit :

```

ClassAssertion(AssociateProfessor AssociateProfessor4)
ClassAssertion(Person Person4)
ClassAssertion(Professor Professor1)
ClassAssertion(AssociateProfessor AssociateProfessor3)
ClassAssertion(Publication Publication0)
ClassAssertion(AssociateProfessor Dean8)
ClassAssertion(FullProfessor FullProfessor8)
ClassAssertion(Dean Dean8)
ClassAssertion(Dean Publication0)
RoleAssertion(advisor(Person4, Professor1))
RoleAssertion(affiliateOf(University1, Person4))
RoleAssertion(affiliateOf(Person4, Professor1))
RoleAssertion(doctoralDegreeFrom(Person2, University0))

```

## Tâches de raisonnement

L'outil d'éditeur d'ontologie *DL-Lite* met en œuvre deux opérations principales de raisonnement : **Solve** et **Repairs**, qui sont des menus sur l'interface principale de l'éditeur.

### Opération Solve

Cette opération implique trois fonctions; la première concerne la détection des erreurs de modélisation lors de la définition de l'ontologie, à savoir l'incohérence de la TBox (voir la section 2.3.1). Une caractéristique importante de cet outil est que la détection d'incohérence se fait instantanément après avoir rajouté un axiome à TBox. Une fois qu'une incohérence est détectée, l'éditeur indique la source d'incohérence en utilisant *Style italique*. La Figure 5.4 donne un exemple sur la détection de l'incohérence dans TBox.

---

9. <https://www.sqlite.org>

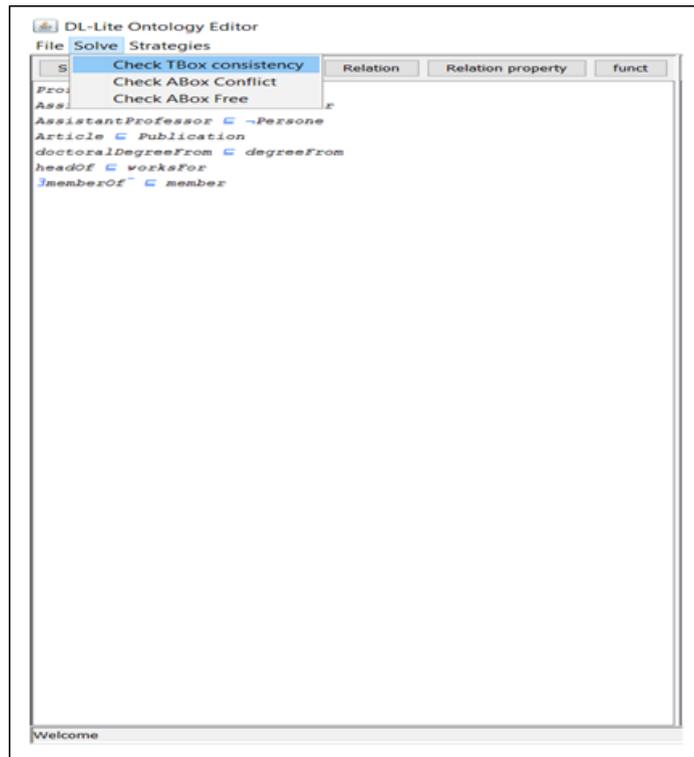


FIGURE 3 – Détection d'incohérence de TBox

Une deuxième fonction fournie par l'opération Solve est le calcul de l'ensemble des conflits (Voir la définition 10). Le résultat de cette tâche est un fichier texte qui contient toutes les paires contradictoires de la forme :

```
(AssociateProfessor(AssociateProfessor8) ; Dean(AssociateProfessor8))
(Publication(Person4) ; Person(Person4))
(Publication(Person6) ; Person(Person6))
(AssociateProfessor(AssociateProfessor3) ; Dean(AssociateProfessor3))
(Publication (Publication0) ; Person(Publication0))
(Publication (Publication2) ; Person(Publication2))
(affiliateOf(Person4, Prefessor1) ; advisor(Person4, Prefessor1))
```

D'autre part, la dernière fonction de l'opération Solve consiste à trouver l'ensemble des éléments libres dans les ABoxes (Voir la définition 11).

## Opération Repairs

Sur cette opération, cinq algorithmes de calcul de réparation ont été implémentés. L'outil calcule d'abord la fermeture négative en utilisant les règles présentées dans la section 2.3.3. Cette fermeture se transforme en requêtes effectuées sur l'ensemble des individus dans les ABoxes afin de calculer le degré d'inconsistance. Une fois que le degré d'inconsistance est calculé, tous les axiomes trouvés dans les ABoxes inférieurs ou égaux à ce degré sont mis en évidence en utilisant le *Style italique*. La Figure 5.4 montre un exemple de détection de l'inconsistance dans les ABoxes.

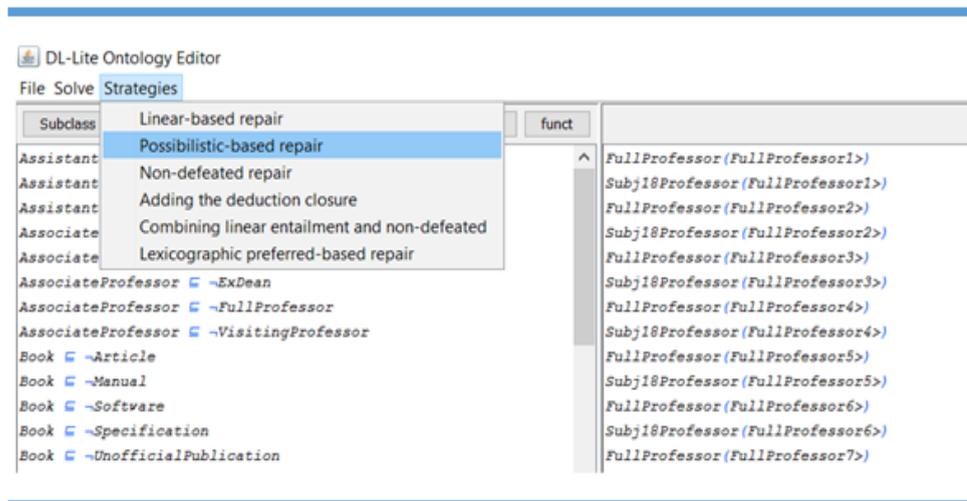


FIGURE 4 – Détection d'inconsistance dans les ABoxes

Pour chaque algorithme, un compteur affiche le temps d'exécution nécessaire pour calculer la réparation et le taux de production (Voir Figure 5.4). Le résultat de ce calcul est enregistré dans un fichier OWL.

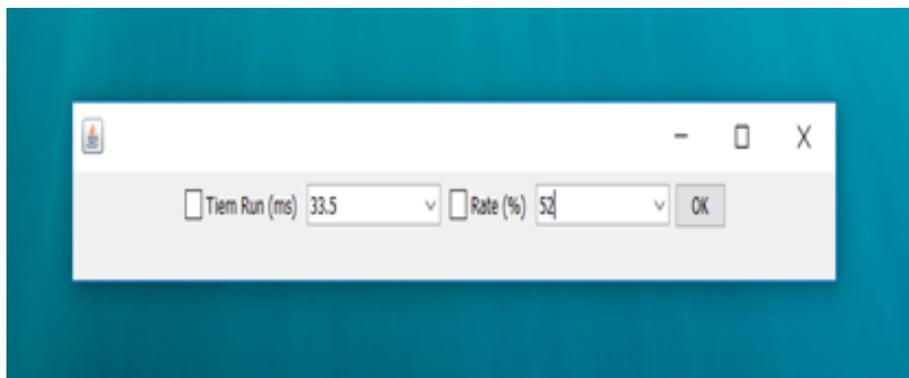


FIGURE 5 – Le temps et le taux de réponse

# Bibliographie

- [1] Baader F., Brandt S., Lutz C., *Pushing the EL envelope*, In Proc. of IJCAI 2005, 364-369, (2005)
- [2] Calvanese D., De Giacomo G., Lembo D., Lenzerini M., Rosati R., *DL-Lite : Tractable description logics for ontologies*, In Proc. of the 20<sup>th</sup> Nat. Conf. on Artificial Intelligence (AAAI 2005), 602 - 607, (2005)
- [3] Calvanese D., De Giacomo G., Lembo D., Lenzerini M., Rosati R., *Tractable reasoning and efficient query answering in description logics : The dl-lite family*, J. Autom. Reasoning, 39(3),385-429, (2007)
- [4] Calvanese D., De Giacomo G., Lembo D., Lenzerini M., and Rosati R., *DL-Lite : Tractable description logics for ontologies*. In Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, AAAI 05, 602-607. AAAI Press / The MIT Press, (2005)
- [5] Alessandro A., Diego C., Roman K., Michael Z., *The DL-Lite family and relations*, J. Artif. Intell. Res. (JAIR), (2009)
- [6] Antonella P., Lembo D., Calvanese D., Giuseppe De G., Maurizio L., and Riccardo R., *Linking data to ontologies*. In Journal of Data Semantics, 133-173, (2008)
- [7] Leopoldo E., Bertossi, Anthony H., and Torsten S., *Introduction to inconsistency tolerance*. In Inconsistency Tolerance, 1-14, (2005)
- [8] Baader F., McGuinness D. L., Nardi D., and Patel-Schneider F., *The description logic handbook : Theory, implementation, and applications*, In Cambridge University Press, (2003)
- [9] Benferhat S., Didier D., Henri P., *Some syntactic approaches to the handling of inconsistent knowledge bases : A comparative study part 1 : The flat case*, Studia Logica, 58(1) :17 - 45, (1997)

- 
- [10] Anthony H., *Handbook of defeasible reasoning and uncertainty management systems.*, Chapter Paraconsistent Logics, 11-36. Kluwer Academic Publishers, Norwell, MA, USA, (1998)
- [11] Hunter A., Konieczny S., *Approaches to Measuring Inconsistent Information.* In : Bertossi L., Hunter A., Schaub T. (eds) Inconsistency Tolerance. Lecture Notes in Computer Science, vol 3300. Springer, Berlin, Heidelberg. (2005)
- [12] Philippe B., and Anthony H., *Elements of Argumentation.* MIT Press, (2008)
- [13] Guilin Q., and Anthony H., *Measuring incoherence in description logic-based ontologies.* In Karl Aberer, Key-Sun Choi, Natasha Fridman Noy, Dean Allemang, Kyung-Il Lee, Lyndon J. B. Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors, The Semantic Web, 6<sup>th</sup> International Semantic Web Conference, 2<sup>nd</sup> Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007., volume 4825 of Lecture Notes in Computer Science, 381-394. Springer, (2007)
- [14] Yue M., and Pascal H., *Distance-based measures of inconsistency and incoherency for description logics.* In Volker Haarslev, David Toman, and Grant E. Weddell, editors, Proceedings of the 23<sup>rd</sup> International Workshop on Description Logics (DL 2010), Waterloo, Ontario, Canada, May 4-7, 2010, volume 573 of CEUR Workshop Proceedings. CEUR-WS.org, (2010)
- [15] Thomas L., Maria V. M., and Gerardo I. S., *Complexity of inconsistency-tolerant query answering in datalog+/-.* In Informal Proceedings of the 26<sup>th</sup> International Workshop on Description Logics, Ulm, Germany, July 23 - 26, 791-803, (2013)
- [16] Lembo D., Lenzerini M., Rosati R., Ruzzi M., Fabio Savo D., *Inconsistency-tolerant semantics for description logics,* In Pascal Hitzler and Thomas Lukasiewicz, editors, RR, volume 6333 of LNCS, 103-117. Springer, (2010)
- [17] Oscar C., Catherine R., Luis Manuel V. B., and Ivan P., *Pattern-based OWL ontology debugging guidelines.* In Eva Blomqvist, Kurt Sandkuhl, François Scharffe, and Vojtech Svatek, editors, Proceedings of the Workshop on Ontology Patterns (WOP 2009) , collocated with the 8th International Semantic Web Conference ( ISWC-2009 ), Washington D.C., USA, 25 October, 2009., volume 516 of CEUR Workshop Proceedings. CEUR-WS.org, (2009)
- [18] Meghyn B., *On the complexity of consistent query answering in the presence of simple ontologies,* In Hoffmann and Selman, (2012)
- [19] Meghyn B., Riccardo R., *Tractable approximations of consistent query answering for robust ontology-based data access,* In Rossi (2013)

- 
- [20] Charalet T., Zacklad M., Kassel G. Bourigault D., *Ingénierie des connaissances évolutions récentes et nouveaux défis*, Eyrolles, Paris, 305-324 (2000)
- [21] Michel B., *Dictionnaire des Concepts philosophiques*, coll. Larousse in extenso, 880, (2012)
- [22] Gruber T.R., *A translation approach to portable ontologies*. Knowledge Acquisition, 5(2),199-220, (1993)
- [23] Neches R., Fikes R., Finin T., Gruber T., Patil R., Senator T., Swartout W., *Enabling technology for knowledge sharing*. AI Magazine, 12(3), 36-56, (1991)
- [24] Staab S., Studer R., Schnurr H.P., Sure Y., *Knowledge processes and ontologies*. Intelligent Systems, IEEE 16, 26 - 34, (2001)
- [25] Sowa J. F., *Knowledge Representation : Logical, Philosophical and Computational Foundations*, Brooks Cole Publising Co, Pacific Grove, California, (2000)
- [26] Mike U., Michael G., *To appear in Knowledge Engineering Review*. Volume 11 Number 2, Mike Uschold, June (1996)
- [27] Zghal S., *Contributions à l'alignement d'ontologies OWL par agrégation de similarités*, Thèse pour l'obtenir le grade de docteur, faculté des science Tunis, Soutenu le 21 décembre (2010)
- [28] Furst F., *L'ingénierie ontologique*. Rapport de recherche N° 02-07. (2002)
- [29] Abdel Kader K., *Conception Coopérative d'Ontologies Pré-Consensuelles :Application au domaine de l'Urbanisme*, Thèse de doctorat, Institut National des Sciences Appliquées de Lyon, Soutenue le 06 Juin (2007)
- [30] Gruber T. R., *Toward Principles for the Design of Ontologies Used for Knowledge Sharing*, presented at International Workshop on Formal Ontology, Padova, Italy, (1993)
- [31] Uschold M., Grüninger M., *ONTOLOGIES : Principles, Methods and applications*, Knowledge engineering review, vol.11, N°2, (1996)
- [32] Guarino, N., *Formal ontology and information systems*, In Proceedings of the International Conference of Formal Ontology in Information Systems, 3-15, Trento, Italy (1998)
- [33] Gomez Pérez A., *Développements récents en matière de conception, de maintenance et d'utilisation des ontologies*, Actes du colloque de Nantes, 9-20, (1999)
- [34] Sowa J.F., *Top-level ontological categories*, International Journal of Human and Computer Studies, 43, 669-685, (1995)
- [35] Mizoguchi R., Bourdeau J., *Using Ontological Engineering to Overcome Common AI-ED Problems*. International Journal of Artificial Intelligence and Education, vol.11 (Special Issue on AIED 2010), 107-121. (2000)

- 
- [36] Mizoguchi R., Kozaki K., Sano T., and Kitamura Y., *Construction and Deployment of a Plant Ontology*, The 12<sup>th</sup> International Conference, EKAW2000, Juan-les-Pins, France, 113-128, (2000)
- [37] Maedche A., *Ontology Learning for the Semantic Web*, Boston, Kluwer Academic Publishers, (2002)
- [38] Michel C., and Marie-Laure M., *Graph-based Knowledge Representation*, Computational Foundations of Conceptual Graphs, Advanced Information and Knowledge Processing, Springer, (2009)
- [39] Sowa J.F., *Semantic networks*, <http://www.jfsowa.com/pubs/semnet.htm>, Encyclopedia of Artificial Intelligence, (1987)
- [40] Franz B., Diego C., Deborah L., Guinness M., Daniele N., Peter F., Patel-Schneider P., *The Description Logic Handbook : Theory, Implementation and Applications*, Cambridge University Press, New York, NY, USA, 2<sup>nd</sup> edition, (2010)
- [41] Nigel S., Tim B., and Wendy H., *The semantic web revisited*, IEEE Intelligent Systems, 21(3) :96-101, (2006)
- [42] Michard A., *XML : Langage et applications Eyrolles ISBN, 2-212-09052-8* (1999)
- [43] François J., *Les langages du Web Sémantique*, INRIA , Mars (2003)
- [44] Horrocks I., *DAML+OIL : A Description Logic for the Semantic Web*, IEEE Data Engineering Bulletin, Num 1, vol.25, 4-9, (2002)
- [45] *W3C. Owl2 web ontology language document overview*, second edition, <http://www.w3.org/2012/pdf/REC-owl2-overview-20121211.pdf>, [Online; W3C Recommendation 11 December 2012]. (2008)
- [46] Horrocks I., Patel-Schneider P. F., and Van Harmelen F., *From SHIQ and RDF to OWL : The making of a web ontology language*, Journal of Web Semantics, vol. 1, n°1, 7-26, (2003)
- [47] Jérôme E., and Pavel S., *Ontology Matching*, Second Edition, Springer, (2013)
- [48] Pavel S., and Jérôme E., *Ontology matching : State of the art and future challenges*, IEEE Trans, Knowl, Data Eng., 25(1) :158-176, (2013)
- [49] Dejing D., Drew V. M., Peishen Q., *Ontology translation on the semantic web*. J. Data Semantics, 2 :35-57, (2005)
- [50] Thomas R., and Gruber. *A translation approach to portable ontology specifications*. Knowledge Acquisition, 5(2) :199-220, June (1993)
- [51] Dejing D., Han Q., and Haishan L., *Semantic translation for rule-based knowledge in data mining*. In Abdelkader Hameurlain, Stephen W. Liddle, Klaus-Dieter Schewe, and Xiaofang Zhou, editors, Database and Expert Systems Applications

- 
- 22<sup>nd</sup> International Conference, DEXA 2011, Toulouse, France, August 29 - September 2, 2011, Proceedings, Part II, volume 6861 of Lecture Notes in Computer Science, pages 74-89. Springer, (2011)
- [52] Thomas A. M., Kevin L., and Richard B., *Knowledge integration for description logics*. In Manuela M. Veloso and Subbarao Kambhampati, editors, Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA, pages 645-650. AAAI Press The MIT Press, (2005)
- [53] Chih-Sheng J. H., Mark A. Musen, and Natalya F. N., *EZPAL : environment for composing constraint axioms by instantiating templates*. Int. J. Hum.-Comput. Stud., 62(5) :578-596, (2005)
- [54] Bernardo C. G., Horrocks I., Yevgeny K., and Ulrike S., *Modular reuse of ontologies : Theory and practice*. J. Artif. Intell. Res. (JAIR), 31 :273-318, (2008)
- [55] Bernardo C. G., Horrocks I., Yevgeny K., and Ulrike S., *Extracting modules from ontologies : A logic-based approach*. In Heiner Stuckenschmidt, Christine Parent, and Stefano Spaccapietra, editors, Modular Ontologies : Concepts, Theories and Techniques for Knowledge Modularization, volume 5445 of Lecture Notes in Computer Science, pages 159-186. Springer, (2009)
- [56] Natalya F. N., Abhita C., William L., and Mark A. Musen. *A framework for ontology evolution in collaborative environments*. In Isabel F. Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Michael Uschold, and Lora Aroyo, editors, The Semantic Web - ISWC 2006, 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006, Proceedings, volume 4273 of Lecture Notes in Computer Science, 544-558. Springer, (2006)
- [57] Peter P., Olga De T., and Sven C., *Understanding ontology evolution : A change detection approach*. J. Web Sem., 5(1) :39-49, (2007)
- [58] Martin O. M., and Marcelo A. F., *A non-monotonic description logics model for merging terminologies*. Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial, 11(35) :77-88, (2007)
- [59] Konstantinos K., George A. V., and Konstantinos S., *Towards automatic merging of domain ontologies : The hcone-merge approach*. J. Web Sem., 4(1) :60-79, (2006)
- [60] Natalya F. N., and Mark A. Musen. *PROMPT : algorithm and tool for automated ontology merging and alignment*. In Henry A. Kautz and Bruce W. Porter, editors, Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, July 30 - August 3, 2000, Austin, Texas, USA., 450-455. AAAI Press / The MIT Press, (2000)

- 
- [61] Thomas L., and Umberto S., *Managing uncertainty and vagueness in description logics for the semantic web*. J. Web Sem., 6(4) :291-308, (2008)
- [62] Thomas L., Maria V. M., Giorgio O., and Gerardo I. Simari., *Heuristic ranking in tightly coupled probabilistic description logics*. In Nando de Freitas and Kevin P. Murphy, editors, Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence, Catalina Island, CA, USA, August 14-18, 2012, 554-563. AUAI Press, (2012)
- [63] Fernando B., Miguel D., and Juan G., *Delorean : A reasoner for fuzzy OWL2*. Expert Systems with Applications, 39(1) :258-272, 2012.
- [64] Philippe A., Philippe C., Francois G., Marie-Christine R., and Laurent S., *Distributed reasoning in a peer-to-peer setting : Application to the semantic web*. J. Artif. Intell. Res. (JAIR), 25 :269-314, (2006)
- [65] Rafael P., and Baris S., *On the complexity of axiom pinpointing in the EL family of description logics*. In Fangzhen Lin, Ulrike Sattler, and Miroslaw Truszczyński, editors, Principles of Knowledge Representation and Reasoning : Proceedings of the Twelfth International Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010. AAAI Press, (2010)
- [66] Michel L., and Rafael P., *Error-tolerant reasoning in the description logic  $\mathcal{EL}$* . In Eduardo Fermé and Joao Leite, editors, Logics in Artificial Intelligence, 14<sup>th</sup> European Conference, JELIA 2014, Funchal, Madeira, Portugal, September 24-26, 2014. Proceedings, volume 8761 of Lecture Notes in Computer Science, 107-121. Springer, (2014).
- [67] Lembo D., Lenzerini M., Rosati R., Ruzzi M., Fabio Savo D., *Inconsistency-tolerant semantics for description logics*, In Pascal Hitzler and Thomas Lukasiewicz, editors, RR, volume 6333 of LNCS, 103-117, Springer, (2010)
- [68] Floriana Di P., Giuseppe De G., Maurizio L., and Riccardo R., *Ontology-based data access with dynamic boxes in DL-Lite*. In Hoffmann and Selmán (2012).
- [69] Leila A., *A unified setting for inference and decision : An argumentation-based approach*, In UAI 05, Proceedings of the 21<sup>st</sup> Conference in Uncertainty in Artificial Intelligence, Edinburgh, Scotland, July 26-29, 2005, 26-33. AUAI Press, (2005)
- [70] Napoli A., *Une introduction aux logiques de descriptions*, Rapport de recherche N°3314. INRIA.72, (1997)
- [71] Michael H., Mark D. R., *Logic in computer science : modelling and reasoning about systems*, 2<sup>nd</sup> edition, Cambridge University Press, (2004)
- [72] Amedeo N., *Une introduction aux logiques de descriptions*. Rapport de recherche, RR-3314, INRIA. pp.72, (1997)

- 
- [73] Baader, F., et Nutt, W., *Appendix 1 : Description logic terminology*. Dans Baader, F., Calvanese D., McGuinness D., Nardi D., et Patel-Schneider P., (éditeurs), *The Description Logic Handbook : Theory, Implementation and Applications*, Cambridge University Press, 495505, (2003)
- [74] Chmidt M., Smolka G., *Attributive concept descriptions with complements*, *Artificial Intelligence*, 48 (1) , 1-26, (1991)
- [75] Kaneiwa K., *Negations in description logic contraries, contradictories, and sub-contraries*, In ICCS05, 66-79, Kassel University Press, (2005)
- [76] Sattler U., *Description Logics for Ontologies*, In Ganter B., de Moor A., Lex W. (eds) *Conceptual Structures for Knowledge Creation and Communication*, ICCS-Concept Struct, Lecture Notes in Computer Science, vol 2746. Springer, Berlin, Heidelberg, (2003)
- [77] Horrocks I., Baader F., Sattler U., *Description Logics as Ontology Languages for the Semantic Web*, In *Festschrift in honor of Jorg Siekmann*, LNAI, Springer-Verlag, (2003)
- [78] Horrocks I., and Peter F., Patel-Schneider P., *Reducing OWL entailment to description logic satisfiability*, *J. Web Sem*, 1(4),345-357, (2004)
- [79] Sirin E., Parsia B., Grau B., Kalyanpur A., and Patel-Schneider P., *A practical owl-dl reasoner*, Submitted for publication to *Journal of Web Semantics*, (2006)
- [80] Haarslev V., and Moller R., *Racer users guide and reference manual*, Version 1.6 , Technical report, University of Hamburg, Computer Science Department, (2001)
- [81] Fournier-Viger P., *Un modèle de représentation des connaissances à trois niveaux de sémantique pour les systèmes tutoriels intelligents*, Mémoire de maîtrise (M.Sc.), Université de Sherbrooke, Sherbrooke, Canada, (2005)
- [82] Padadimitriou C. H., *Computational complexity*, Addison-Wesley Publishing Company, Massachusetts, É.-U. (1994)
- [83] Serge A., Richard H., and Victor V., *Foundations of Databases*, Addison-Wesley, (1995)
- [84] Baral C., *Knowledge Representation, Reasoning and Declarative Problem Solving*, (2003)
- [85] Calvanese D., De Giacomo G., Lembo D., Lenzerini M., Rosati R., *Tractable Reasoning and Efficient Query Answering in Description Logics : The DL-Lite Family*, *Journal of Automated Reasoning*. (2007)
- [86] Francesco M., D., *Complexity of reasoning. In The description logic handbook*, Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter

- 
- F. Patel-Schneider (Eds.). Cambridge University Press, New York, NY, USA 96-136. (2003)
- [87] Kontchakov R., Rodriguez-Muro M., and Zakharyashev M. *Ontology-Based Data Access with Databases : A Short Course*. In : Rudolph S., Gottlob G., Horrocks I., van Harmelen F. (eds) Reasoning Web. Semantic Technologies for Intelligent Data Access. Lecture Notes in Computer Science, vol 8067. Springer, Berlin, Heidelberg (2013)
- [88] Nonfjall H., and Larsen H. L., *Detection of potential inconsistencies in knowledge bases*, International Journal of Intelligent Systems, 7(2),81-96, (1992)
- [89] Giorgos F., Zhisheng H., Jeff Z., Pan Dimitris P., Holger W., *Inconsistencies, negations and changes in ontologies*, In Proc. of AAAI 06, 1295-1300, (2006)
- [90] Garey M.R., David S. J., *Computers and Intractability : A Guide to the Theory of NP-Completeness*, W. H. Freeman, (1979)
- [91] Gerhard B., Preferred S., *An extended logical framework for default reasoning*, In N. S. Sridharan, editor, Proceedings of the 11<sup>th</sup> International Joint Conference on Artificial Intelligence. Detroit, MI, USA, August 1989, 1043-1048. Morgan Kaufmann, (1989)
- [92] Benferhat S., Dubois D., Prade H., *Some syntactic approaches to the handling of inconsistent knowledge bases : A comparative study*, Part 2 : the prioritized case, volume 24, 473-511. Physica-Verlag, Heidelberg, (1998)
- [93] Giuseppe D. G., Maurizio L., Antonella P., Riccardo R., *On the approximation of instance level update and erasure in description logics*, In AAAI, 403-408, (2007)
- [94] Benferhat S., Didier D., Henri P., *Argumentative inference in uncertain and inconsistent knowledge bases*, In David Heckerman and E. H. Mamdani, editors, UAI, 411-419. Morgan Kaufmann, (1993)
- [95] Leila A., Henri P., *Using arguments for making and explaining decisions*, Artificial Intelligence, 173(3-4) :413-436, (2009)
- [96] Benferhat S., Dubois D., Prade H., *Representing default rules in possibilistic logic*. In Knowledge Representation and Reasoning, 673-684. Morgan Kaufmann, (1992)
- [97] Reiter R., *A theory of diagnosis from first principles*, Artificial Intelligence, 32(1) :57-95, (1987)
- [98] Stefan S., Ronald C., *Non-standard reasoning services for the debugging of description logic terminologies*, In Georg Gottlob and Toby Walsh, editors, IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003, 355-362. Morgan Kaufmann, (2003)

- 
- [99] Peter H., Frank van H., Zhisheng H., Heiner S., York S., *A framework for handling inconsistency in changing ontologies*, In Yolanda Gil, Enrico Motta, V. Richard Benjamins, and Mark A. Musen, editors, The Semantic Web - ISWC 2005, 4<sup>th</sup> International Semantic Web Conference, ISWC 2005, Galway, Ireland, November 6-10, 2005, Proceedings, volume 3729 of Lecture Notes in Computer Science, 353-367. Springer, (2005)
- [100] Sven Ove H., *Specified meet contraction*. Erkenntnis, 69(1) :31-54, (2008)
- [101] Thomas L., Maria V. M., Gerardo I. S., *Inconsistency handling in datalog+/- ontologies*, In Luc De Raedt, Christian Bessiere, Didier Dubois, Patrick Doherty, Paolo Frasconi, Fredrik Heintz, and Peter J. F. Lucas, editors, ECAI 2012 - 20th European Conference on Artificial Intelligence, 2012, volume 242 of Frontiers in Artificial Intelligence and Applications, 558-563, IOS Press, (2012)
- [102] Liping Z., Houkuan H., Guilin Q., Yue M., Zhisheng H., Youli Q., *Paraconsistent query answering over DL-Lite ontologies*, Web Intelligence and Agent Systems, 10(1) :19-31, (2012)
- [103] Benferhat S., Bouraoui, Z., Papini O., Würbel W., *A prioritized assertional-based revision for DL-Lite knowledge bases*, In European Conference on Logics in Artificial Intelligence, volume 8761 of LNCS, 442-456. Springer, (2014)
- [104] Nicholas R., and Ruth M., *On inference from inconsistent premisses : Theory and Decision*, 1(2) :179-217, (1970)
- [105] Dubois D., and Prade P., *Possibility theory*, Plenum Press, New-York, (1988)
- [106] Didier D., Lang J., and Henri P., *Possibilistic logic*, Volume 3 of In Handbook on Logic in Artificial Intelligence and Logic Programming, 439-513, Oxford University press, (1994)
- [107] Didier D., and Henri P., *Epistemic entrenchment and possibilistic logic*, Artificial Intelligence, 50(2) :223-239, (1991)
- [108] Benferhat S., and Bouraoui Z., *Min-based possibilistic DL-Lite*, Journal of Logic and Computation, (2015)
- [109] Qi G., Ji Q., Pan Z., and Du J., *Extending description logics with uncertainty reasoning in possibilistic logic*, International Journal of Intelligent Systems, 26(4) :353-381, (2011)
- [110] Benferhat S., Cayrol C., Dubois D., Lang J., and Prade H., *Inconsistency management and prioritized syntax-based entailment* In International Joint Conference on Artificial Intelligence, 640-647. Morgan Kaufmann, (1993)
- [111] Nebel B., *Base revision operations and schemes : Semantics, representation and complexity*, In European Conference on Artificial Intelligence, 341-345, (1994)

- 
- [112] Maria Vanina M., Francesco P., Andrea P., Gerardo I., Simari, and Subrahmanian V. S., *Inconsistency management policies*, In Gerhard Brewka and Jérôme Lang, editors, Principles of Knowledge Representation and Reasoning : Proceedings of the Eleventh International Conference, KR 2008, Sydney, Australia, September 16-19, 2008, 367-377. AAAI Press, (2008)
- [113] Calvanese D., Kharlamov E., Nutt W., and Zheleznyakov D., *Evolution of DL-Lite knowledge bases*, In International Semantic Web Conference (1), 112-128, (2010)
- [114] Slawek S., Jan C., and Jerzy M., *Prioritized repairing and consistent query answering in relational databases*. Ann. Math. Artif. Intell., 64(2-3) :209-246, (2012)
- [115] Jianfeng D., Guilin Q., and Yi-Dong S., *Weight-based consistent query answering over inconsistent SHIQ knowledge bases*, Knowl. Inf. Syst., 34(2) :335-371, (2013)
- [116] Meghyn B., Camille B., and François G., *Querying inconsistent description logic knowledge bases under preferred repair semantics*, In Brodley and Stone, 996-1002, (2014)
- [117] Lutz C., Seylan I., Toman D., and Wolter F., *The combined approach to OBDA : taming role hierarchies using filters*. In The Semantic Web - ISWC, volume 8218 of Lecture Notes in Computer Science, 314-330. Springer, (2013)
- [118] Calvanese D., De Giacomo G., Lembo D., Lenzerini M., and Rosati R., *Data complexity of query answering in description logics*. 260-270. (2006)
- [119] Cristian M., Jan C., and Jerzy M., *Disjunctive databases for representing repairs*. Annals of Mathematics and Artificial Intelligence, 57(2) :103-124, (2009)
- [120] Didier D., Lang J., and Henri P., *Dealing with multi-source information in possibilistic logic*. In ECAI, 38-42, (1992).
- [121] Alessandro A., Roman K., Frank W., and Michael Z., *Temporal description logic for ontology-based data access*, In Rossi (2013)
- [122] Rafael P., and Baris S., *On the complexity of axiom pinpointing in the EL family of description logics*. In Fangzhen Lin, Ulrike Sattler, and Miroslaw Truszczyński, editors, Principles of Knowledge Representation and Reasoning : Proceedings of the Twelfth International Conference, KR 2010, Toronto, Ontario, Canada, May 9-13, 2010. AAAI Press, (2010).
- [123] Bouraoui Z., *Inconsistency and uncertainty handling in lightweight description logics*, Thèse de doctorat, Université d'Artois, Soutenu le 5 juin (2015)
- [124] Ann Kipling B., and Monica P., *Dance notation for beginners*. Dance Books Ltd, 9 Cecil Court, London WC2N 4EZ, first edition. 1984

- 
- [125] Ngo Duc T., *Regional Culture & culture zoning in Vietnam*. Young Publisher, Vietnam. 2004.
- [126] Phung Quang M., *Folk dance Origin of professional dance*. Master thesis, Hanoi Academy of Theatre and Cinema, 2010.
- [127] Nguyen Tam K., Nguyen Thu N., *Classical dance curriculum Vietnam*. Ethnic Culture Publisher, Vietnam. 2014
- [128] Benesh R., and Benesh J., *Reading Dance : The Birth of Choreology*. McGraw-Hill Book Company Ltd, ISBN 0-285-62291-9. 1983
- [129] Benferhat S., Bouraoui Z., Chaudhry H., Shafry Bin Mohd Rahim Fc M., Tabia K., and Telli A., *Characterizing Non-Defeated Repairs in Inconsistent Lightweight Ontologies*. 282-287. 10.1109/SITIS.2016.53.
- [130] Hetty B., Hetty Blades K., *Creative Computing and the Re-Configuration of Dance Ontology*., Electronic Visualisation and the Arts., London, UK, 10 - 12 July 2012. College London 1D Chesham Building Strand, London. WC2R 2LS
- [131] Smeulders W. M., Worring M., Santini S., Gupta A., Jain R., *Content based image retrievals at the end of the early years.*, on Pattern Analysis and Machine Intelligence, Vol.22, pp.1349-1380, IEEE, 2000
- [132] Fernandez M., Gomez-Perez A., et al. *METHONTOLOGY :From ontological art towards ontological engineering*. In Proceedings of the Spring Symposium Series on Ontological Engineering AAAI Press., 1997.
- [133] Staab S., and Studer R., *Web Ontology Language : OWL.*, Handbook on Ontologies., Springer-Verlag, 2004
- [134] Gangemi, A. ; Steve, G. ; Giacomelli, F ; ONIONS : an ontological methodology for taxonomic knowledge integration. Working notes of the workshop Ontological Engineering. ECAI. 29-40. (1996)