THÈSE

présentée à

l'Université Joseph Fourier - Grenoble 1

pour obtenir le grade de

DOCTEUR

spécialité

Informatique

intitulée

Sémantique des réseaux de connaissances :

gestion de l'hétérogénéité fondée sur le principe de médiation

présentée et soutenue publiquement le 17 novembre 2008 par

Antoine Zimmermann

devant le jury composé de :

Jérôme Euzenat	**Directeur de thèse**
François Goasdoué	**Rapporteur**
Amedeo Napoli	**Rapporteur**
Chantal Reynaud	**Rapporteur**
Marie-Christine Rousset	**Examinatrice**

Remerciements

La connaissance est en fin de compte fondée sur la reconnaissance.

Ludwig Wittgenstein

Avant de parler de sémantique formelle, je tiens à remercier certains membres de mon propre réseau de connaissances. En premier lieu, je remercie Jérôme Euzenat, Directeur de Recherche à l'INRIA Grenoble - Rhône-Alpes, pour avoir encadré mon doctorat en laissant beaucoup d'autonomie tout en gardant régulièrement un œil attentif sur l'avancement de mes travaux.

Ensuite, je souhaite remercier les membres du jury pour avoir accepté d'évaluer ma thèse. En particulier : Marie-Christine Rousset, qui m'a fait l'honneur de présider ce jury ; les rapporteurs François Goasdoué, Amedeo Napoli et Chantal Reynaud pour leurs commentaires constructifs et enrichissants et tout particulièrement ceux, nombreux, de François Goasdoué.

Je tiens aussi à remercier les personnes qui ont favorisé ou contribué à l'élaboration de ma thèse. Je les salue de loin : Markus Krötzsch et Pascal Hitzler en Allemagne, Jason Jung en Corée du Sud, Faisal Alkhateeb en Jordanie, Mathieu d'Aquin au Royaume Uni, François Scharffe en Autriche et Frederico Freitas au Brésil. Un grand merci particulièrement à Chan Le Duc qui m'a beaucoup aidé à construire l'une des plus importantes contribution de ma thèse.

Par ailleurs, ces quatre années n'auraient pas été aussi agréables sans les discussions avec mes collègues de l'INRIA : Sébastien, merci d'avoir apporté ta lumière sur les débats insolites de midi, Julien, merci d'avoir alimenté nos conversations de ta finesse, Pierre, Nabil, Jérôme D., Jérôme P., Samuel, Romain, Arun, Seungkeun, Nizar, Dung, Xavier, Mathieu, etc.

Enfin, il va de soi que je remercie ceux qui m'ont soutenu moralement : mes parents que je remercie aussi pour avoir organisé le bon pot de soutenance ; mes frères Sylvain et Vincent ; mes proches amis Camille, Jullian, Nicolas, Aurore, Matthieu, Rodolphe. En dernier lieu, ma plus grande reconnaissance revient à ma meilleure amie et compagne Cécile.

Résumé

L'objectif de cette thèse est de modéliser la sémantique d'un ensemble des connaissances produites indépendamment les unes des autres, mais mises en correspondances. On suppose que ces connaissances forment un réseau où à chaque nœud se trouve une ontologie, reliée aux autres par des correspondances formant des alignements d'ontologies. De telles structures peuvent se trouver sur le Web ou les réseaux pair-à-pair sémantiques. Afin de favoriser l'utilisation d'ontologies indépendantes et pré-existantes, je définis une sémantique formelle exploitant le principe de médiation. Celle-ci définit l'interprétation d'un réseau d'ontologies alignées en affectant à chaque ontologie une interprétation locale selon sa propre logique, et en y ajoutant un domaine d'interprétation supplémentaire (dit global) dans lequel viennent se projeter les domaines locaux par le biais d'une fonction dite d'égalisation. Ce domaine sert de médiateur et c'est dans celui-ci que les alignements viennent poser leurs contraintes pour définir la satisfaction d'un réseau d'ontologies. Quatre applications sont présentées : la sémantique des ontologies modulaires, la sémantique d'un langage d'alignement expressif, la composition d'alignements d'ontologies et le raisonnement distribué. Cette sémantique s'applique aux ontologies modulaires en traitant un module comme médiateur vis-à-vis des modules qu'il importe. Ainsi la sémantique des modules importés (locaux) se distingue de celle du module importateur (global). La distinction entre niveau local et global permet de donner une sémantique à un langage d'alignement expressif, indépendamment de l'expressivité du langage d'alignement. Un opérateur de composition d'alignements fondé sur les algèbres de relation est défini de façon cohérente avec la sémantique du réseau d'ontologie. Enfin, en se restreignant au cas des logiques de description alignées par des correspondances simples, on défini une procédure de vérification de la cohérence d'un réseau d'ontologies alignées. Cette procédure est correcte et complète, et exploite des systèmes de raisonnement locaux existant sans en connaître l'implémentation explicite.

Mots-clés : Web sémantique, représentation de connaissances, alignement d'ontologies, médiation

Abstract

This thesis aims at modelling the semantics of a set of knowledge bases produced independently from each others, but put in correspondence. We assume that this knowledge forms a network where each node is represented as an ontology, linked to the others by correspondences in the form of an ontology alignment. Such structures exist on the Web or semantic peer-to-peer networks. In order to promote the use of independent, pre-existing ontologies, I define a formal semantics which exploit the principle of mediation. It defines the interpretation of a network of aligned ontologies by assigning a local interpretration to each ontology, according to its own logic, and by adding a supplementary domain of interpretation (the global domain) in which the local domains are mapped via a so-called equalising function. This domain can be seen as a mediator in which alignments pose the constraints which serve to define the satisfaction of a network of ontologies. This semantics is applied in four cases : the semantics of modular ontologies ; the semantics of an expressive alignment language ; the composition of ontology alignments ; and distributed reasoning. Such semantics applies to modular ontologies by dealing with a module as if it were a mediator with respect to the modules it imports. Thus, the semantics of imported modules (local) stay distinct from the semantics of importing modules (global). With this distinction between local and global level, the semantics of alignments can differ from the semantics of ontologies, in particular in term of expressivity. An ontology alignment composition operator based on relation algebras is defined as a sound reasoning operation with respect ot the semantics of the network of ontologies. Finally, when restricted to description logics and simple correspondences, we can define a consistency checking procedure for a network of aligned ontologies. It is sound and complete, and exploit existing local reasoner while not imposing any specific implementation.

Keywords : Semantic Web, knowledge representation, ontology alignment, mediation

Table des matières

Introduction

Problématique

Je me propose de modéliser la sémantique de réseaux de connaissances, sous la forme d'une spécification mathématique fondée sur la théorie des modèles. Tandis qu'il existe des outils tant théoriques que techniques pour représenter et raisonner au sein de systèmes localisés à base de connaissances (par exemple une description formalisée du domaine de la bibliographie, sur un site Web de vente de livres), la sémantique d'un réseau composé de multiples sources de connaissances interconnectées (par exemple pour l'exploitation de ressources culturelles et artistiques issues de plusieurs parties du monde) est encore mal définie du fait de la diversité des formes de représentations de connaissances et de systèmes d'information servant à les exploiter.

Plus précisément, un réseau informatique comme le World Wide Web, où des documents non structurés ou faiblement structurés sont accessibles aux personnes (texte, images, sons, vidéos), est en train de se voir complété progressivement par un Web sémantique où les données sont accompagnées d'information décrivant formellement des connaissances, d'une manière utilisable par les machines pour leur traitement automatique. D'autres réseaux, comme les réseaux pair-à-pair, acquièrent graduellement plus de métadonnées sémantiques. Parallèlement à ce foisonnement de sources de connaissances variées, les systèmes d'information sur le Web communiquent de plus en plus entre eux, en échangeant, combinant ou agrégeant des données de toute part.

Il devient donc essentiel de comprendre non pas la sémantique d'un simple système de connaissances, mais de tout un réseau de connaissances. Au delà de la simple connection matérielle, pour que les connaissances forment un réseau au niveau sémantique, il faut établir des connections sémantiques entre différentes connaissances locales. C'est la combinaison d'un ensemble de nœuds de connaissances et de ces connections qui forme un réseau de connaissances.

L'objectif général de ce mémoire est d'étudier la sémantique de ces réseaux.

Solution envisagée

La formalisation de la sémantique d'un réseau de connaissances suppose d'une part que l'on sache représenter chaque nœud de connaissances et en formaliser la sémantique, et d'autre part que l'on puisse représenter les relations entre ces représentations.

Premièrement, on supposera que les nœuds de connaissances sont représentés sous la forme d'ontologies, c'est-à-dire d'une spécification formelle d'un domaine de connaissances (par exemple la bibliographie), dans un format de représentation de connaissance disposant d'une sémantique en théorie des modèles. Ceci implique qu'il existe une notion d'interprétation du vocabulaire dans le format associé à un nœud et une notion de satisfaction reliant les interprétations aux assertions qu'elles satisfont. En outre, on supposera que ces formalismes locaux sont associés à des outils de raisonnement (vérification, déduction, etc.) localisés. Ces ontologies sont a priori conçues indépendamment les unes des autres, leur système d'information associé fournit potentiellement des services à un utilisateur ou un système tiers, mais les nœuds ne sont pas supposés fonctionner en coordination avec les autres nœuds.

Deuxièmement, les relations sémantiques entre nœuds de connaissances seront supposées représentées dans ce qu'on appelle des alignements d'ontologies, que l'on distingue des connaissances locales des ontologies. Ceux-ci peuvent être produits par des outils déconnectés des systèmes d'information liés aux ontologies. L'alignement est donc vu comme un objet à part entière, au même titre que les ontologies.

Ainsi, on suppose que, tout comme un nœud de connaissances est représenté par une ontologie, un réseau de connaissances est représenté par un réseau d'ontologies alignées.

Ces hypothèses, bien que ne couvrant pas nécessairement toute forme de réseaux de connaissances, ne sont pas particulièrement exigeantes et traduisent des contraintes réalistes quant aux technologies d'ores et déjà existantes sur le Web et le Web sémantique.

À cela s'ajoute d'autres exigences particulières à ce mémoire. Le type de réseaux envisagé ici consiste en un ensemble d'ontologies parfaitement indépendantes que l'on souhaite exploiter sans contraindre leur représentation interne, auxquelles s'ajoutent des alignements d'ontologies établis à part sans privilégier l'une ou l'autre des représentations locales.

Les formalismes applicables aux réseaux d'ontologies alignées sont inadaptés à ce cadre particulier. D'un côté, on trouve des formalismes utilisant une sémantique "classique", c'est-à-dire qui traite ontologies et alignements comme des portions d'une même représentation monolithique. Cette approche est peu tolérante à l'hétérogénéité, notamment parce que la logique commune à tout le réseau peut être indécidable. D'un autre côté, on trouve des formalismes tolérants à l'hétérogénéité, mais dont les alignements expriment une relation du point de vue d'un nœud, ce qui tend à forcer les systèmes locaux à tenir compte des connaissances externes. Or, dans le cadre que l'on s'est fixé, on souhaite exploiter des systèmes sémantiques locaux existants, qu'ils soient ou non "conscients" de leur position dans un réseau d'ontologies alignées.

Au contraire, ce qui guide la présente approche est une conception des réseaux d'ontologies où alignements et ontologies sont découplés et traités à part. En cela, notre approche suit les principes de gestion de modèles que l'on trouve en bases de données et favorise la gestion des alignements par un serveur d'alignement. La sémantique générale que nous avons définie respecte ces prérequis et cette vision en donnant au réseau une sémantique qui dépend des diverses sémantiques locales, sans pour autant les perturber. Plus précisément, celle-ci est issue de la combinaison de deux approches :

– le raisonnement contextuel, permettant d'assigner à chaque ontologie un domaine d'interprétation distinct,
– et le principe de médiation, permettant de ne pas imposer une mise en relation directe entre ontologies.

Le premier point permet une plus grande liberté des interprétations locales, donc est conforme au premier critère. Le second point indique que la coordination de la sémantique passe par un intermédiaire et ainsi ne force pas les nœuds à communiquer directement entre eux.

Contributions

Le cœur de la contribution de ce travail est la définition d'une sémantique générique pour les réseaux d'ontologies alignées. L'avantage de cette sémantique réside dans le fait que les alignements d'ontologies sont traités comme des objets à part entière, ayant une sémantique propre, distincte des ontologies. Ceci permet une meilleure réutilisabilité des alignements dans des situations différentes et une séparation du langage d'alignements de celui des ontologies. Cette sémantique a aussi la particularité de bien modéliser le principe de médiation entre ontologies. Ceci est rendu possible en ne reliant pas directement les interprétations des ontologies locales, mais en utilisant un domaine d'interprétation supplémentaire, servant à interpréter les connaissances inter-ontologiques, selon un point de vue plus global et éventuellement plus abstrait.

Cette contribution est prolongée par des applications du formalisme générique aux cas suivants.

- Premièrement, on montre en quoi la sémantique générique s'adapte bien à la définition d'une **sémantique des ontologies modulaires**. En adoptant une approche inspirée de l'ingénierie logicielle, on peut voir un module comme une sorte de médiateur vis-à-vis des modules qu'il importe, puisqu'il sert à intégrer ces imports de façon cohésive. Un tel module peut alors être à son tour importé et donc intégré par un module de plus haut niveau.
- Deuxièmement, grâce à la forte indépendance de la représentation des alignements vis-à-vis de la représentation des ontologies, il est possible de définir un **langage d'alignements expressif**, permettant d'exprimer des correspondances précises, par l'utilisation de constructeurs de formules spécifiques au langage d'alignements. Disposer d'un langage expressif au niveau d'un médiateur est important car celui-ci n'a pas forcément accès aux ontologies locales. Il faut alors disposer des relations sémantiques les plus riches possibles pour favoriser une bonne interaction entre systèmes d'ontologies. En particulier, une forte expressivité est la bienvenue lorsque les ontologies sont inaccessibles. Celle-ci peut améliorer les opérations de traduction de requêtes ou de données, afin de garantir un minimum de perte lors de la traduction. En outre, cette expressivité ne force aucunement les langages locaux à disposer d'une expressivité similaire.
- Troisièmement, l'une des caractéristiques les plus intéressantes de cette sémantique est son comportement vis-à-vis de la **composition d'alignement d'ontologies**, qui forme l'une des contribution de ce mémoire. Cette opération est cruciale dans un vaste réseau comme le Web sémantique car la mise en correspondance directe de deux ontologies est très coûteuse. Par l'utilisation de la théorie des algèbres de relations, on peut construire un opérateur de composition d'alignements à la fois efficace et conforme à la sémantique.
- Quatrièmement, on étudie des procédures de **vérification de la cohérence** d'un réseau d'ontologies alignées en logiques de description. Ceci comprend d'abord la définition d'un formalisme pour décrire des réseaux d'ontologies en logiques de description. Ensuite, une procédure de vérification de la cohérence est proposée, faisant appel à des procédures de déduction locales, mais ne nécessitant pas de connaitre leur implémentation. Aussi, l'expressivité locale n'est pas limitée par l'algorithme. Toutefois, la complexité oblige à critiquer l'approche et des variantes du formalisme sont alors envisagées. On montre comment ces variantes permettent de se rattacher à des logiques distribuées existantes, avec lesquelles il est possible de raisonner.

Organisation de la thèse

Les trois premiers chapitres constituent l'état de l'art :

– Le chapitre 1 détaille les motivations de ce travail en décrivant des scénarios où les réseaux d'ontologies alignées interviennent. Ces scénarios mettent en évidence le besoin d'offrir une sémantique qui accorde une grande indépendance des ontologies et des alignements pour faciliter leur manipulation et leur gestion séparées. En particulier, la sémantique doit garantir le principe de médiation et permettre la composition et le raisonnement sur des alignements.
– Le chapitre 2 introduit des notions générales de représentation de connaissances et d'alignements d'ontologies, qui serviront tout au long de cette thèse. En particulier, ce chapitre décrit la sémantique des ontologies de façon très générale, en théorie des modèles, puis détaille la syntaxe et la sémantique des logiques de descriptions que j'utiliserai principalement pour illustrer les aspects abordés. Il décrit aussi les alignements d'ontologies dans toute leur généralité avant de donner les différentes représentations concrètes existantes.
– Le chapitre 3 entre dans le vif du sujet en présentant les formalismes de représentation de connaissances distribuées proposés dans la littérature, hormis celui développé dans cette thèse. Pour chacun de ces formalismes, j'évoquerai les avantages et les inconvénients, surtout du point de vue de la médiation, et je décrirai les outils logiciels ou algorithmes produits pour les supporter et les mettre en œuvre. Il sera fait aussi état de l'existant en terme de médiation, en partant des techniques liées aux bases de données, jusqu'à la médiation pour les ontologies. On verra que les formalismes existants ne comblent pas bien les besoins en matière de médiation dans des réseaux d'ontologies hétérogènes tels que le Web sémantique.

Le chapitre 4 constitue le cœur de la contribution. Il décrit de façon abstraite et très générale un formalisme modélisant la sémantique des réseaux d'ontologies connectées par médiation. Ce chapitre se dégage de toute représentation concrète des ontologies et des alignements, et s'abstrait même de la (ou des) logique(s) locale(s) utilisée(s), pourvu qu'elles aient une sémantique en théorie des modèles, conformément aux définitions du premier chapitre. Ce chapitre défend aussi le principe même d'une sémantique distinguant niveau local et niveau global.

Alors que le chapitre 4 ne donnait que des arguments rhétoriques pour défendre le formalisme proposé, les quatre chapitres suivants le mettent en application afin de démontrer sa pertinence. Ces quatre chapitres partagent dans les grandes lignes le même schéma : ils décrivent d'abord une application à laquelle j'ai contribué, puis montrent comment le formalisme distribué s'y applique et enfin argumentent sur l'avantage de ce formalisme vis-à-vis de l'application.

– Plus précisément, le chapitre 5 aborde la problématique de la modularité des ontologies, en mettant l'accent sur la réutilisation de modules issus de sources hétérogènes, par exemple, trouvés sur le Web sémantique. Après la présentation d'une infrastructure pour la spécification de modules d'ontologies, utilisant des alignements pour relier les modules importés entre eux ou au module qui les importe, je leur assigne une sémantique conforme à celle du chapitre 4 et compatible avec les caractéristiques désirées des modules hétérogènes. Ce chapitre prend tout son sens si l'on combine la modularité des ontologies à la composition des alignements.
– Le chapitre 6 décrit un langage d'alignements expressif, dont le bien fondé et les impératifs sont soulignés par divers scénarios typiques de la médiation d'ontologies. À ce langage est affectée une sémantique en théorie des modèles basée sur le formalisme générique du chapitre 4 et donnant une interprétation des alignements en fonction des

sémantiques locales. Ce chapitre démontre la possibilité de découpler complètement le langage d'alignements et les langages locaux, ce qui permet d'exprimer des correspondances riches entre deux formalismes pauvres (par exemple, des taxonomies).
- Le chapitre 7 présente le problème de la composition d'alignements, essentielle pour la réutilisabilité des alignements. Je montre qu'une composition utilisant une algèbre de relations permet d'obtenir un opérateur cohérent avec la sémantique. En revanche, cette opérateur ne tient pas compte de la sémantique locale des ontologies, pouvant enrichir l'alignement composé, ni de la possibilité de composer les alignements expressifs du précédent chapitre. Ces deux points renvoient à la nécessité d'une procédure de raisonnement pour ces réseaux d'ontologies alignées, abordée au chapitre 8.
- Le chapitre 8 clôt l'ensemble des contributions en fournissant une procédure de vérification de la cohérence d'un réseau d'ontologies alignées dans le cas du formalisme (introduit dans le chapitre) des logiques de description distribuées intégrées (*Integrated Distributed Description Logics* ou IDDL). Ce chapitre est essentiel puisque, tandis que les trois précédents mettaient en évidence les propriétés intéressantes de la sémantique appliquée à trois cas d'utilisation, celui-ci démontre la possibilité d'automatiser la déduction sur des réseaux d'ontologies. Cette étape constitue alors une prémisse pour l'intégration d'un moteur d'inférence dans chacune des applications qui précèdent.
- La conclusion ouvre les perspectives nombreuses qu'implique l'introduction de ce nouveau formalisme pour le raisonnement distribué. La plus essentielle est la mise en pratique de procédures d'inférence répondant en un temps raisonnable sur des réseaux de taille importante. En effet le raisonnement se répercute sur toutes les applications présentées ici. La composition est aussi un point clé des travaux futurs car, comme on le verra, elle se trouve au cœur de nombreuses utilisations intéressantes.

Première partie

État de l’art

Chapitre 1

Motivation

Résumé

La contribution de ce mémoire est guidée par trois grandes idées autour desquelles s'articule notre vision des réseaux de connaissances. Il s'agit du principe de médiation, la gestion de modèles en bases de données, et la notion de serveur d'alignements. Le présent chapitre décrit, en partant d'exemples concrets, le type de réseaux d'ontologies alignées dont on souhaite définir la sémantique. Avant de présenter les travaux existants sur ces sujet, ce chapitre présente des scénarios où les réseaux d'ontologies alignées jouent un rôle important, et sur l'analyse desquels seront établies les bases sur lesquelles repose la sémantique proposée au chapitre 4. Ces bases sont présentées dans les grandes lignes afin que le lecteur sache ce qui détermine les choix effectués.

Sommaire

1.1 Scénarios de raisonnement distribué

Afin de montrer comment le raisonnement sur des réseaux de connaissances intervient dans les outils informatiques contemporains, on présentera deux types de scénarios. Dans le premier cas, on souhaite exploiter simultanément des connaissances provenant de sources hétérogènes sur le Web sémantique, par le biais d'une seule interface homogène. Dans le second cas de figure, on se trouve dans un réseau pair-à-pair où chaque pair dispose d'une ontologie personnalisée décrivant ses propres données. Pour retrouver les données pertinentes vis-à-vis d'un certain besoin, il faut pouvoir combiner les connaissances des différents pairs.

1.1.1 Une interface homogène pour des sources hétérogènes

Imaginons qu'une personne (appelons la Alice) fait une recherche en histoire de l'art en cherchant des informations précises sur des artistes et œuvres contemporaines sur le Web. Elle dispose pour cela de multiples bases de connaissances en rapport avec la littérature, la musique, le cinéma et les arts plastiques. Si elle veut exploiter toutes ces connaissances, Alice doit trouver chacune de ces sources d'informations, les interroger toutes, puis mettre en commun les résultats qui se recoupent, et ainsi de suite.

On voudrait lui fournir une seule interface utilisant simultanément ces bases de connaissances, et lui retournant des résultats homogènes. Mais puisque les différentes bases de connaissances ont été produites indépendamment les unes des autres, leurs vocabulaires respectifs doivent être mis en correspondances. Néanmoins, chacun des systèmes n'est pas nécessairement apte à coopérer avec les autres donc la mise en correspondance doit se faire par un mécanisme externe.

Présentation du réseau de connaissance

On peut donc décrire le système qu'Alice tente d'exploiter. Il est composé de plusieurs bases de connaissances, chacune munie d'un système de raisonnement automatique. Ces connaissances structurées se conforment à une ontologie décrivant leur domaine de connaissances. Ces ontologies ne sont pas forcément contenues dans les systèmes de raisonnement car il peut suffire de s'y référer par leur identifiant sur le Web, auquel cas il se peut que plusieurs bases de connaissances pointent sur la même ontologie. Si plusieurs de ces ontologies sont rendues accessibles à un moment donné, elles ont pu être mises en correspondance par un système spécialisé, par exemple un serveur d'alignement d'ontologies. L'alignement est alors lui-même rendu disponible sur le Web. Mais alors, l'alignement n'exprime pas *a priori* une connaissance relative à l'un ou l'autre des systèmes qu'Alice souhaite interroger.

On peut donc dire qu'un alignement dans ce contexte n'exprime pas une connaissance du point de vue d'une ontologie ou d'une base de connaissances particulière. Au contraire, l'alignement ne fait qu'affirmer des relations entre deux ontologies sans en privilégier une, puisqu'on ne peut savoir laquelle ou lesquelles seront utilisées par une base de connaissances locales. En l'absence d'information supplémentaire, on peut alors supposer qu'*un alignement n'est pas directionnel*.

D'autre part, toujours dans ce cadre, un alignement ne constitue pas *a priori* un appariement des ontologies locales vers une hypothétique ontologie "globale". Un alignement se contente de définir les relations entre les terminologies locales, et non une terminologie qui lui est propre.

Ainsi, les alignements ne sont que des connaissances qui se *réfèrent* à des ontologies, mais *précisent* les connaissances inter-ontologiques.

Interrogation du réseau

On peut imaginer deux manières d'interroger ce réseau d'ontologies. Dans le premier cas, Alice accède à l'une des bases de connaissances et l'interroge, mais grâce à des alignements de l'ontologie de cette base avec d'autres ontologies, elle peut récupérer des informations déduites en partie de connaissances externes au système qu'elle utilise.

Dans le second cas, Alice accède à un système d'intégration qui sait localiser diverses sources de connaissances, et qui dispose en plus des alignements pour exploiter simultanément ces différentes sources. Ce système d'intégration peut utiliser ou non une ontologie propre,

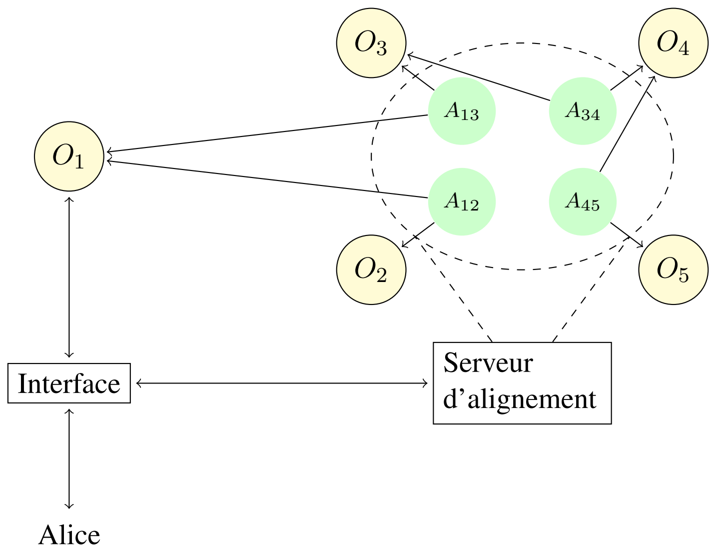


FIGURE 1.1 – Interrogation d'un seul système, exploitant des connaissances externes. Alice interroge le système conformément à une ontologie O_1, mais le système exploite diverses ontologies par le biais du serveur d'alignements qui dispose des correspondances avec d'autres ontologies.

mais les connaissances locales ne sont pas *a priori* appariées avec cette ontologie. Dans les deux cas, il faut pouvoir raisonner avec un réseau d'ontologies alignées, où ontologies et alignements sont conçus indépendamment les uns des autres.

Imaginons qu'Alice cherche des informations sur les surréalistes, et notamment quels artistes ont collaboré pour créer ensemble des œuvres surréalistes. La base de connaissances des arts plastiques indique que Salvador Dali est un peintre surréaliste. Par ailleurs, la base de connaissances cinématographiques indique que Dali a collaboré avec Luis Buñuel pour écrire *Un chien andalou*. Ce résultat de la requête d'Alice ne peut être découvert qu'en combinant les connaissances des deux bases de connaissances. Des cas plus complexes où plus de deux ontologies permettent de déduire un résultat sont envisageables.

Gestion séparée des alignements

Supposons maintenant que dans l'exemple de la figure 1.1, l'ontologie O_3 disparaisse du réseau. Il s'ensuit que O_4 et O_5 sont déconnectées de O_1. Mais grâce à la gestion séparée des alignements d'ontologies par le serveur d'alignements, il suffit de composer A_{13} et A_{34} pour obtenir des correspondances entre O_1 et O_4, et rétablir du même coup le lien avec O_5.

D'autres manières de combiner les alignements sont possibles. Notamment, il peut exister plusieurs alignements des deux mêmes ontologies. D'ailleurs, ce peut être le cas que la composition fournisse deux manières de relier deux ontologies. On peut envisager alors deux opérations. La première est l'union d'alignements, que l'on effectue surtout lorsqu'on admet que deux processus d'alignements incomplets sont complémentaires. L'autre opération est l'intersection, qui suppose que les alignements fournissent trop de correspondances, mais que si deux processus fournissent une même correspondance, alors cela confirme sa validité.

Enfin, on peut imaginer que l'on veuille raisonner sur un ou plusieurs alignements, de façon à fournir des déductions partielles quand bien même les systèmes de raisonnement locaux sont inopérants ou pour éviter de trop les solliciter.

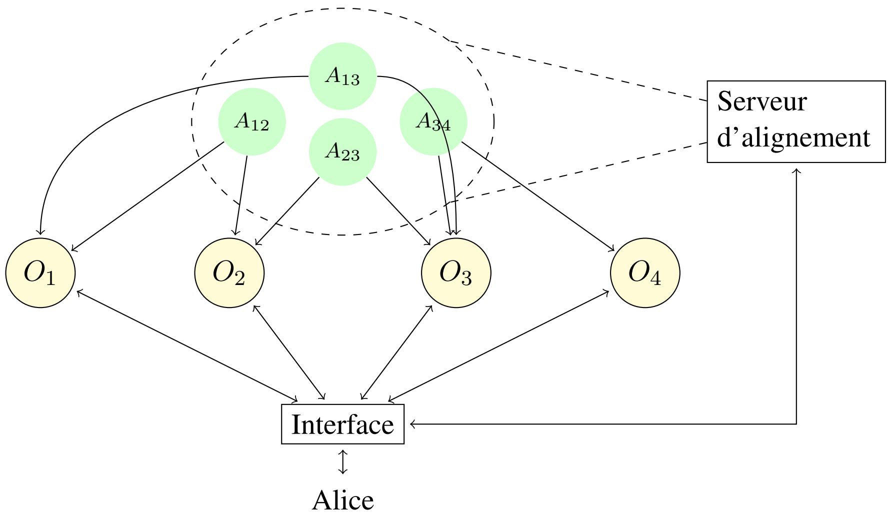


FIGURE 1.2 – Interrogation d'un système intégré, traduisant les requêtes d'Alice vers les systèmes hétérogènes. La traduction peut se faire en demandant au serveur d'alignements les correspondances avec les diverses ontologies.

Pour s'assurer que ces opérations sont valables, elles devront être définies conformément à la sémantique du réseau d'ontologies alignées. C'est pourquoi le chapitre 4 défini une sémantique formelle pour ces réseaux, le chapitre 7 définit une opération de composition conforme à la sémantique et le chapitre 8 donne une procédure de déduction automatique suivant cette sémantique.

1.1.2 Système pair-à-pair à base d'ontologies

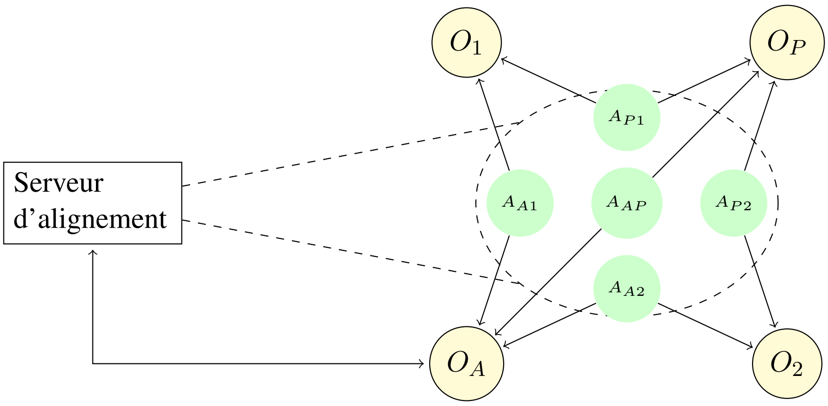
Imaginons maintenant qu'Alice utilise un logiciel pair-à-pair pour trouver les données qu'elle cherche.

Présentation du système

Son propre système indexe les données en les représentant dans des formats sémantiques conformes à une ontologie. L'ontologie qu'elle utilise a été conçue au cours de l'utilisation du logiciel, de façon semi automatique, par les interactions d'Alice avec le système. Elle dispose donc d'une ontologie qui lui est propre, et qui peut différer de toutes les ontologies des autres pairs. Pour pouvoir interroger l'ensemble du réseau pair-à-pair, il faut donc que le système connaisse les correspondances entre les ontologies de chaque pair.

On ne peut supposer qu'Alice se chargera de la mise en correspondance, car cette tâche est à la fois complexe et fastidieuse. En outre, il faudrait que l'ontologie d'Alice soit mise en correspondance avec chacun des pairs avec lesquels elles souhaite échanger de l'information, ce qui renforce encore la difficulté. Afin de ne pas utiliser toutes les ressources d'Alice ou de son système pair-à-pair associé, on peut par exemple faire appel à un service externe d'alignement d'ontologies [1]. Ce service peut identifier des correspondances à la demande, à l'aide d'algorithmes d'alignement d'ontologies. On peut aussi imaginer que le système demande à Alice de valider certaines opérations lors de l'alignement. Mais le service peut aussi, le cas échéant, utiliser un alignement déjà construit, qu'il aura stocké dans sa base de données. En

1. L'utilisation d'un serveur d'alignement commun à tous les pairs ne remet pas en question l'aspect totalement distribué du réseau pair-à-pair, puisque le serveur ne fait pas partie du réseau pair-à-pair en tant que tel.



Requête d'Alice : Trouver x tel que $x \in \mathit{Document}$ et $x.\mathit{sujet} \in \mathit{ArtisteSurrealiste}$?

Ontologie O_A d'Alice : $\mathit{ArtisteSurrealiste} = (\mathit{Artiste}$ et $(\mathit{style} = \mathit{surrealiste}))$

Ontologie O_P de Paul : $d \in \mathit{Biography}$
$d.\mathit{subject} = \mathit{Dali}$
$\mathit{Dali} \in \mathit{Person}$

Ontologie O_1 : $\mathit{Book} \subseteq \mathit{Document}$

Ontologie O_2 : $\mathit{SalvadorDali} \in (\mathit{Peintre}$ et $\mathit{Surrealiste})$

Alignement de O_A et O_P : $\mathit{sujet} = \mathit{subject}$

Alignement de O_A et O_1 : $\mathit{Document} = \mathit{Document}$

Alignement de O_A et O_2 : $\mathit{Artiste} \supseteq \mathit{Peintre}$
$(\mathit{style} = \mathit{surrealiste})) \supseteq \mathit{Surrealiste}$

Alignement de O_P et O_1 : $\mathit{Biography} \subseteq \mathit{Book}$

Alignement de O_P et O_2 : $\mathit{Dali} = \mathit{SalvadorDali}$

FIGURE 1.3 – Un réseau d'ontologies alignées dans un système pair-à-pair sémantique.

particulier, si l'on suppose qu'Alice utilise une ontologie populaire trouvée sur le Web, et qu'il en est de même pour certains pairs auxquels elle s'est connectée, il est probable que l'alignement soit déjà calculé. Il faut aussi remarquer qu'en utilisant un tel service, les alignements resteront disponibles malgré l'évolution de la topologie du réseau pair-à-pair. On pourra donc toujours composer les alignements, même si un pair s'avère manquant.

Il est important de remarquer que si l'alignement est obtenu de la sorte, alors il ne correspond pas nécessairement à des connaissances en conformité avec la conceptualisation personnelle d'Alice, pas plus qu'aux ontologies des systèmes auxquels son logiciel se connecte. Ce que cela implique, c'est qu'un alignement ne se rattache pas à l'une ou l'autre des ontologies alignées, même s'il se réfère aux deux.

Interrogation

Alice cherche des documents textuels ou multimédia au sujet d'artistes surréalistes. Dans le réseau pair-à-pair, Paul dispose d'un document d répondant à cette requête. La figure 1.1.2 montre les connaissances qui doivent être exploitées pour découvrir cette réponse. Pour l'instant, ces connaissances sont décrites dans un pseudo code. On définira formellement les langages de représentation des ontologies et des alignements dans le chapitre 2.

La difficulté du raisonnement à effectuer ici vient du fait que l'on doive exploiter simultanément les connaissances de quatre ontologies et de cinq alignements. Pourtant, on pourrait être tenté de vouloir stocker aussi bien les ontologies au sein du serveur d'alignements, ce qui rendrait le raisonnement centralisé. Mais c'est sans compter sur le fait que chaque pair peut personnaliser son ontologie. Il serait extrêmement ardu de maintenir cet ensemble d'ontologies. Mais bien qu'elles soient personnalisées, ces ontologies sont probablement proches d'ontologies connues réutilisées par différents pairs, et pour lesquelles le serveur peut disposer des alignements.

1.1.3 Résumé des besoins

Les deux scénarios décrits ci-dessus permettent de distinguer des critères pour choisir une sémantique des réseaux d'ontologies alignées qui convienne à notre cadre d'étude.

Premièrement, la sémantique doit offrir une bonne robustesse vis-à-vis de l'hétérogénéité des ontologies. Cela signifie qu'il doit exister un mécanisme tolérant une certaine divergence des représentations locales, et surtout que l'ensemble puisse rester décidable si tous les langages d'ontologies et d'alignements sont eux-mêmes décidables. Ceci se traduira par le choix d'une sémantique qualifiée de contextualisée, c'est-à-dire autorisant des interprétations distinctes des ontologies du réseau.

Deuxièmement, on constate que les alignements se placent toujours en intermédiaire pour connecter les ontologies et que les nœuds du réseau n'ont pas besoin de se mettre en correspondance directement. Cette tâche est reléguée à un système tiers, qui pourra prendre la forme d'un *serveur d'alignements*, que l'on peut qualifier de système médiateur. Ainsi, dans le type de réseaux d'ontologies que l'on souhaite exploiter, l'on souhaite que la sémantique traduise le principe de médiation. L'intérêt de cette approche réside dans le fait qu'elle perturbe au minimum les systèmes locaux.

En outre, comme on a pu le voir dans les scénarios précédents, il serait bon de pouvoir définir des opérations sur les alignements seuls. Il faut donc que la sémantique traite les alignements comme des objets de première classe, c'est-à-dire utilisable pour eux-mêmes, au même titre que les ontologies, et non en tant que référence à d'autres structures. En particulier, l'on souhaite qu'elle autorise la composition d'alignements.

Nous allons donc détailler dans les sections suivantes les notions essentielles qui forment les bases de notre vision d'un réseau de connaissances distribuées.

1.2 La médiation d'ontologie

1.2.1 Principe de la médiation

On cite souvent [76] comme un travail fondateur de l'idée de médiateur, bien que des logiciels ad hoc existaient préalablement. Le besoin de médiation est né de l'explosion de la quantité de données accessibles sur les réseaux informatiques. La multiplication des sources de données a rendu impossible l'assimilation de l'ensemble des informations par un système monolithique. Pour pallier ce problème, Wiederhold propose un modèle d'architectures où un module logiciel est chargé d'accéder à un ensemble de sources de données, tout en offrant aux clients l'illusion d'utiliser un unique système d'information.

Ce module logiciel est appelé le médiateur et son existence est justifiée par deux caractéristiques des réseaux d'informations.

– Le volume de données, empêche de traiter, d'exploiter ou de manipuler l'ensemble des informations simultanément. Il faut donc pouvoir appliquer une sélection sur celles-ci.

– La disparité empêche de regrouper l'ensemble des données pour offrir un système unique de sélection et de manipulation des informations. Il faut donc fournir des moyens de rendre les diverses sources de données interopérables.

La problématique reste la même lorsque les données représentent des connaissances structurées, à la seule différence près que ces connaissances sont munies d'une sémantique formelle. Il devient alors naturel de se poser la question de la sémantique de l'ensemble du réseau, compte tenu de ces deux caractéristiques. Ce problème est rendu difficile par les caractéristiques suivantes des réseaux d'ontologies alignées.

– Il y a plusieurs sources de données, chacune exportant une partie de ses connaissances par le biais d'interfaces.
– Les sources ont été conçues indépendamment.
– Les sources sont autonomes et ont une sémantique indépendante du réseau.

Un médiateur permet de faire interopérer diverses sources d'informations sans en modifier le fonctionnement interne. Un médiateur peut être chargé de localiser des sources de données, de transmettre des requêtes à chaque source, ou d'une source à une autre, de récupérer les réponses aux requêtes pour éventuellement les renvoyer à d'autres sources. Ceci est traduit par la figure 1.4 où l'on voit que le médiateur permet d'accéder à des informations de sources très diverses, en jouant le rôle d'intermédiaire entre l'utilisateur et les différents systèmes d'information.

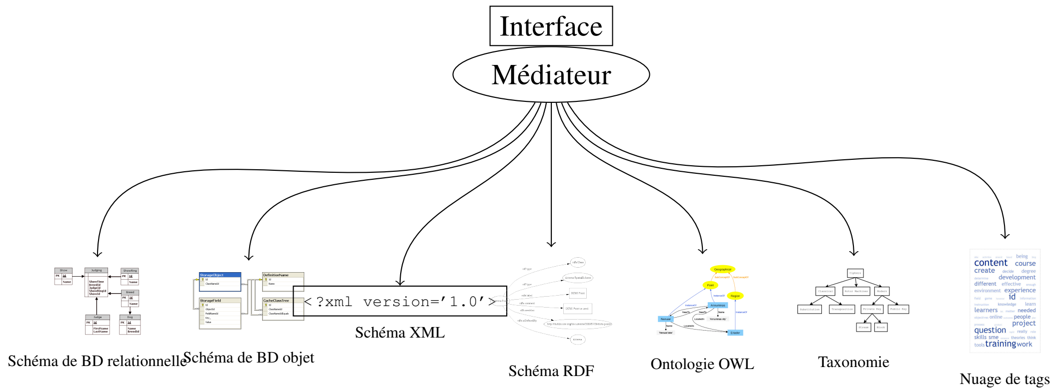


FIGURE 1.4 – Intégration de données.

1.2.2 Systèmes de médiation d'ontologies

La médiation a d'abord été introduite pour les bases de données, mais des systèmes de médiations exploitant une sémantique existent. Cependant, cette sémantique est le plus souvent limitée à celle d'une opération spécifique que le système réalise. Ainsi, on ne peut donner une interprétation générique à un réseau d'ontologies alignées en utilisant les approches de ces systèmes, bien qu'elles permettent de vérifier le bon fonctionnement de certains cas de médiations.

On peut généralement regrouper les systèmes de médiation en systèmes d'intégration, ou en systèmes de transformations. Dans les deux cas, si l'on dispose d'une sémantique formelle des réseaux de connaissances, on peut déterminer la validité des opérations.

Pour des systèmes de transformations ou d'intégration on pourra se référer au chapitre 6 de [34], qui donne un bon aperçu de ce sujet. Bien que ces systèmes ne soient pas tous destinés à produire des alignements d'ontologies, ils manipulent presque tous des structures pouvant être traduits en alignements selon la définition que l'on donnera dans le chapitre 2. En outre, la construction des correspondances se fait indépendamment de la construction des ontologies à aligner. On obtient donc bien, comme on le suppose dans les scénarios, des objets distincts, que l'on aimerait manipuler à part.

Ce besoin de pouvoir gérer les alignements à part rappelle la notion de gestion de modèles.

1.3 Gestion de modèles

La gestion de modèles (*model management*) provient de la communauté des bases de données, qui a reconnu la nécessité de disposer d'outils à haut niveau d'abstraction pour manipuler des schémas de bases de données et leurs correspondances comme des objets à part entière, afin de facilité la gestion, la mise à jour, l'évolution et la combinaison de ces éléments dans un cadre hétérogène (comme le Web par exemple).

Bernstein et ses collègues [17] ont défini ces notions pour les schémas de bases de données, et l'on souhaiterait les voir appliquer aux réseaux d'ontologies. Dans leur approche, les schémas de bases de données autant que les correspondances entre eux sont traités comme des objets de première classe, que l'on peut alors manipuler séparément ou combiner selon diverses opérations.

Dans [17], l'approche reste assez théorique et abstraite, mais plusieurs travaux se sont construits sur cette base. Notamment, [6] décrit plus précisément les opérations selon la sémantique des bases de données. Ces définitions ne permettent pas néanmoins de traiter toute la richesse des ontologies que l'on considère ici, et ne donne pas la possibilité de véritablement raisonner sur le réseau complet. Un des développements importants se trouve dans les travaux sur la composition des appariements de schémas de bases de données (*schema mapping composition*), notamment dans [55] et [16]. Ce dernier article décrit l'implémentation de la procédure de composition et décrit aussi comment des correspondances complexes (de schémas de bases de données) peuvent être composées.

Bien que l'approche proposée ne permette pas complètement de généraliser les opérations pour le cas des ontologies, et qu'elle ne prévoit pas une automatisation du raisonnement sur les connaissances, les principes sous jacents correspondent à ce vers quoi nous souhaiterions tendre en matière de gestion des alignements. Dans cette optique, un serveur d'alignement est un outil primordial.

1.4 Serveur d'alignement

Un serveur d'alignement est une application dont le rôle est de fournir, à la demande, des alignements entre deux ontologies identifiés en entrée du système [31, 36, 32]. Ces alignements peuvent être soit construits à la volée grâce à des algorithmes de mise en correspondance automatiques, soit retrouvés dans une base d'alignement ou bien sur le Web. D'autres fonctionnalités relatives à la manipulation des alignements ou bien à la médiation peuvent être prises en charge par le serveur, ou bien une application autre, ou encore déléguées directement aux systèmes locaux. En reprenant les scénarios de médiation précédents, on peut voir l'utilité d'un tel service, puisqu'il permet de construire un réseau d'ontologies alignées à partir d'ontologies isolées.

Euzenat et ses collègues [32] listent les services que l'on peut associer à un tel système. Il s'agit de la mise en correspondance d'ontologies, du stockage d'alignements, de la recherche d'alignements ou d'informations au sujet d'un alignement, de la mise à jour d'alignements, de la génération de transformation de requêtes ou de données à partir d'un alignement, ou de la génération d'axiomes dans un langage donné traduisant l'alignement et enfin, la recherche d'ontologies alignées avec une autre, avec la possibilité de les ordonner par proximité.

Dans le cas du réseau pair-à-pair hétérogène, lorsque deux pairs se connectent entre eux, ils demandent au serveur un alignement entre leurs ontologies respectives. Cet alignement va permettre de traduire des requêtes éventuelles, ainsi que de retranscrire les réponses conformément aux ontologies de chaque pair. La situation est similaire dans les systèmes multi-agents. La tâche de traduction des données ou de transformation de requêtes peut elle-même être effectuée par le serveur, mais ce n'est pas obligatoire. Ce processus a l'avantage de n'imposer aux nœuds du réseau aucune gestion des correspondances avec les autres nœuds.

De même, un système d'intégration de données peut faire appel à un serveur d'alignement à chaque fois qu'une nouvelle ontologie est ajoutée au réseau. Cet alignement servira à traduire ou bien à fusionner des informations provenant de sources hétérogènes, avec l'illusion d'une information homogène pour l'utilisateur.

1.5 Bilan

Le présent chapitre a montré, par deux scénarios impliquant des connaissances distribuées, les besoins en termes de sémantique pour les réseaux de connaissances. En particulier, on a montré que la sémantique devait garantir le principe de médiation pour ne pas perturber des systèmes d'ontologies hétérogènes existants. On a aussi constaté par ces scénarios que la sémantique doit permettre de gérer séparément les alignements en automatisant des opérations sur les alignements. Ces besoins sont renforcés par la présence de serveurs d'alignement pouvant produire, stocker et raisonner sur les alignements d'ontologies.

Avant de décrire la sémantique des réseaux d'ontologies alignées, on introduit au chapitre suivant des notions préliminaires en représentation de connaissances et en alignement d'ontologies afin de mieux définir notre objet d'étude.

Chapitre 2

Représenter un réseau d'ontologies et d'alignements

Résumé

Ce chapitre donne les éléments théoriques préliminaires permettant de définir un réseau d'ontologies alignées. Il définit les composants d'un réseau d'ontologies alignées, à savoir les ontologies et les alignements, en présentant aussi, pour les ontologies, les notions de sémantiques formelles couramment utilisées dans des systèmes à base de connaissances localisées. Les logiques de description, formalisme important sur le Web sémantique, sont présentées plus particulièrement. Un aperçu des formats de représentation d'alignement est aussi fourni. En donnant des exemples de représentations concrètes de ces deux notions, ce chapitre montre que les réseaux d'ontologies alignées sont une réalité et qu'il est donc justifié de les étudier, notamment du point de vue de la sémantique.

Sommaire

2.1 Représenter des réseaux de connaissances

La problématique posée par ce mémoire suppose l'existence d'ontologies multiples, accompagnées de systèmes les exploitant (moteur d'inférence, service Web, etc.) et implantées dans un réseau où elles peuvent interagir. La mise en correspondance des différentes ontologies se fait par le biais de descriptions explicites de leurs relations : les alignements d'ontologies.

Avant de proposer une approche pour définir la sémantique de tels réseaux d'ontologies, il faut en définir les composants et, en particulier, détailler les notions de sémantique formelle que l'on peut rattacher aux ontologies. C'est sur la base de la (ou des) sémantique(s) locale(s) que doivent être étudiés les réseaux d'ontologies alignées.

Le présent chapitre sert aussi à introduire les notions théoriques adoptées dans l'ensemble du mémoire. Premièrement, on conviendra que des systèmes localisés à base de connaissances ont leur sémantique formelle bien définie et, par conséquent, on présentera les notions générales de représentation des connaissances avec des exemples, en particulier celui du formalisme adopté comme standard pour la description d'ontologie sur le Web, à savoir OWL. Les types d'applications principaux utilisant des ontologies dans des réseaux sont les services Web sémantiques, les moteurs de recherche d'information ou de services, l'un et l'autre pouvant être amenés à fonctionner de pair. Outre le Web sémantique, ces systèmes peuvent fonctionner localement à chaque nœud d'un réseau pair-à-pair.

Deuxièmement, dans l'optique d'une sémantique favorisant la médiation, il faut que les relations entre ontologies soient explicitées pour interpréter de façon cohérente l'ensemble du réseau. Tandis que les ontologies se placent aux nœuds d'un réseau de connaissances, ces relations (les alignements d'ontologies) tiennent lieu de connections entre nœuds, ou, selon le vocabulaire des graphes, d'arcs. Ce chapitre décrit des formats existants pour la représentation des alignements, mais leur sémantique sera décrite dans les chapitres suivants.

Le chapitre s'organise de la façon suivante. Dans une première partie (section 2.2), des notions générales de représentation de connaissances sont abordées afin de fournir les définitions importantes utilisées dans la suite du mémoire. Ces notions sont accompagnées d'exemples donnant un aperçu concret du type d'information existant aux nœuds d'un réseau d'ontologies alignées. Pour donner plus de fluidité aux discours, les notions les moins essentielles sont simplement expliquées en termes courants et seuls les concepts importants sont soulignés par des définitions formelles. Ensuite, en section 2.3, on définit de façon très générale la notion d'alignement d'ontologies et l'on présente des formats pour leur représentation.

2.2 Langages de représentation de connaissances

Les langages de représentation de connaissances [1] se caractérisent par leur syntaxe, c'est-à-dire la manière dont on écrit les phrases ou formules (section 2.2.1), et la sémantique (section 2.2.2), c'est-à-dire la manière dont sont interprétées ces formules. La section 2.2.4 concrétise quelque peu les notions générales présentées auparavant en présentant une famille de langages de représentation de connaissances, à savoir les logiques de description. Cette section a deux intérêts : d'une part, elle précise les notations utilisées tout au long de ce mémoire et, d'autre part, illustre ces notions par des exemples issus des logiques de description, le formalisme à la base de la logique distribuée IDDL pour lequel je donne une procédure de déduction au chapitre 8.

Pour éviter d'accumuler de nombreuses définitions, certains concepts moins essentiels sont exprimés informellement, mais l'annexe A en donne une présentation rigoureuse.

Remarque 2.2.1 *Le lecteur intéressé par des détails sur ces notions pourra consulter les ouvrages sur la représentation des connaissances [68] ou [27], sur la théorie des modèles [45] et sur les logiques de description [8].*

1. On nomme parfois *logique* un langage de représentation de connaissance, accompagné de sa sémantique.

2.2.1 Syntaxe d'un langage de représentation de connaissances

La syntaxe d'un langage de représentation des connaissances indique comment les "phrases" ou assertions doivent être construites à partir d'un ensemble de symboles et de règles de construction. La syntaxe d'un langage L de représentation de connaissances définit :

– un ensemble de symboles pouvant être utilisés pour construire des assertions, ceux-ci pouvant être regroupés en différentes sortes ou types de symboles ;
– un ensemble de règles permettant de construire les assertions à partir des symboles (il s'agit de la grammaire).

Pour désigner un sous-ensemble des symboles du langage, associés à leur type, on parle de signature. La notion de signature permet par exemple de désigner l'ensemble des symboles utilisés à un nœud d'un réseau d'ontologies. On parlera donc de signature d'une ontologie.

Chaque signature permet de définir un ensemble de formules déterminé par le langage. Ces formules servent à énoncer des faits ou des vérités sur le monde. Le plus souvent, les règles de construction des formules d'un langage sont données à l'aide d'une grammaire formelle.

Un ensemble de formules construites à partir d'une même signature constitue une ontologie. Ceci correspond à une description formelle d'un domaine de connaissances. Il s'agit de la représentation syntaxique d'un nœud de connaissances dans un réseau de connaissances interconnectées.

Definition 2.2.1 (Ontologie) *Soit L un langage de représentation de connaissances. Une* ontologie *O de L est un couple $\langle \Sigma(O), A(O) \rangle$ où $\Sigma(O)$ est une signature de L et $A(O)$ est un sous-ensemble des formules que l'on peut construire à partir de $\Sigma(O)$. On appelle les formules de $A(O)$ des* axiomes *de O.*

Pour résumer, on dira que tout langage L de représentation de connaissances définit sa syntaxe par un ensemble de signatures $\mathbf{Sign}_L$, auxquelles sont associées les formules que l'on peut construire avec leurs symboles. Si l'on note $\mathbf{For}_L(\Sigma)$ les formules associées à la signature Σ, alors on peut dire que la syntaxe de L est caractérisée par $\langle \mathbf{Sign}_L, \mathbf{For}_L \rangle$.

2.2.2 Sémantique des langages de représentation de connaissances

La sémantique d'un langage de représentation des connaissances définit la manière dont les symboles doivent être interprétés et comment les formules contraignent ces interprétations. Elle se caractérise par :

– une notion d'interprétation associant aux symboles d'une signature des éléments d'un certain "univers" ou domaine d'interprétation,
– et des contraintes définissant la satisfaction d'une formule.

Definition 2.2.2 (Interprétation) *Une* interprétation *I d'une signature Σ d'un langage L est un couple $\langle \mathbf{I}, \Delta \rangle$ où Δ est une famille d'ensembles non vides (Δ_t) indicée par les types de symboles de L et $\mathbf{I}$ est une famille de fonctions $I_t : \Sigma_t \to \Delta_t$ appelée* fonction d'interprétation. *Δ est appelé le* domaine d'interprétation *de I. Une interprétation d'une ontologie O est une interprétation de la signature de O.*

Exemple 2.2.1 *Étant donnée une signature $\Sigma = \langle \mathcal{C}, \mathcal{R}, \mathcal{U} \rangle$ en logique de description, une interprétation I de Σ associe à chaque concept $C \in \mathcal{C}$ un ensemble $C^I \subseteq \Delta$, à chaque rôle $R \in \mathcal{R}$ une relation $R^I \subseteq \Delta \times \Delta$ et à chaque individu $o \in U$ un élément $o^I \in \Delta$.*

La seule notion d'interprétation ne suffit pas à elle seule pour raisonner sur des connaissances. En effet, elle ne permet que de relier les symboles à des "choses" de l'univers de discours modélisées par le domaine d'interprétation. En revanche, elle ne permet pas de déterminer ce qui est correct vis-à-vis d'une ontologie. La notion cruciale pour cela est celle de satisfaction. Elle sert à relier les formules aux interprétations qui satisfont ces formules, c'est-à-dire qui valident la connaissance exprimée par ces formules.

Definition 2.2.3 (Relation de satisfaction) *La relation de satisfaction $\models_L$ d'un langage L définit une relation entre les interprétations d'une signature Σ et les formules sur cette signature. Formellement, $\models_L$: $\Sigma \mapsto \models_{L,\Sigma} \subseteq \mathbf{Int}_L(\Sigma) \times \mathbf{For}_L(\Sigma)$ en notant $\mathbf{Int}_L(\Sigma)$ l'ensemble des interprétations de Σ. Lorsque $\langle I, f\rangle \in \models_{L,\Sigma}$, on note $I \models_L f$ et on dit que I satisfait f.*

Exemple 2.2.2 *Considérons les concepts C et D en logique de description et la formule $C \sqsubseteq D$. Elle est satisfaite par une interprétation I si et seulement si l'interprétation de C par I est un sous-ensemble de l'interprétation de D par I. Plus formellement, $I \models_L C \sqsubseteq D \Leftrightarrow C^I \subseteq D^I$.*

On étend la notion de satisfaction aux ontologies en disant qu'une interprétation satisfait une ontologie si et seulement si elle satisfait tous les axiomes de l'ontologie. Une interprétation satisfaisant une ontologie est appelée modèle de l'ontologie.

Definition 2.2.4 (Modèle d'une ontologie) *Soit L un langage de représentation de connaissances. Soit O une ontologie de L. Une interprétation I de O satisfait O si et seulement si I satisfait tous les axiomes de O. Dans ce cas, on note $I \models_L O$ et on dit que I est un modèle de O. Par ailleurs, on note $\mathbf{Mod}(O)$ l'ensemble de tous les modèles de O.*

Pour résumer, la sémantique d'un langage L de représentation de connaissances est caractérisée par ses interprétations $\mathbf{Int}_L$ et la relation de satisfaction $\models_L$. De même que pour les ontologies, la sémantique des réseaux de connaissances sera définie par une notion d'interprétation d'un réseau d'ontologies alignées et une relation de satisfaction. C'est ce que je définirai au chapitre 4. Ces notions sont définies dans le but de s'assurer qu'un processus de raisonnement automatique est bien conforme aux connaissances décrites.

2.2.3 Raisonner avec un langage de représentation de connaissances

Nous nous intéressons aux systèmes d'information qui n'ont pas accès aux connaissances réellement décrites par les ontologies. Pour qu'un système puisse raisonner automatiquement, il faut définir des notions indépendantes des connaissances décrites. On définit donc la cohérence d'une ontologie, c'est-à-dire le fait qu'elle ne se contredit pas, et les conséquences logiques d'une ontologie. Chacune de ces notions se traduit par une quantification des interprétations.

Definition 2.2.5 (Cohérence) *Soit L un langage de représentation de connaissances. Une ontologie O de L est cohérente si et seulement s'il existe un modèle de O.*

La notion de cohérence ne détermine pas si une ontologie représente convenablement un domaine de connaissances.

Exemple 2.2.3 *L'ontologie $O_1 = \{$Il n'y a pas de fumée sans feu.$\}$ est cohérente. En revanche, l'ontologie $O_2 = \{$Il n'y a pas de fumée sans feu.,L'azote liquide au contact de l'air produit de la fumée mais pas de feu.,Il existe de l'azote liquide au contact de l'air.$\}$ est incohérente. Plus formellement, on pourrait écrire $O_2 = \{$fumée$\Rightarrow$feu.,azote$\wedge$air$\Rightarrow \neg$feu$\wedge$fumée.,azote$\wedge$air.$\}$.*

Definition 2.2.6 (Conséquence logique) *Soit L un langage de représentation de connaissances. Soit O une ontologie et f une formule de L. On dit que f est une conséquence sémantique de O lorsque tous les modèles de O satisfont f. Dans ce cas, on note $O \models_L f$.*

Dans certains cas, il est possible de réduire le problème de la conséquence sémantique ("f est-elle conséquence de O ?") à celui de la cohérence d'une ontologie (plus exactement, de l'incohérence). En effet, il suffit que le langage de représentation de connaissance puisse exprimer la négation de formules. Si l'on note $\mathrm{non}(f)$ la négation d'une formule f, alors $O \models_L f$ équivalent à dire que $\langle \Sigma(O), A(O) \cup \{\mathrm{non}(f)\}\rangle$ est incohérent.

Déterminer la cohérence d'une ontologie et en déduire les conséquences sémantiques sont des enjeux majeurs de la représentation des connaissances. Disposer d'une sémantique formelle aide justement à construire des procédures automatiques pour répondre à ces questions. Néanmoins, il n'est pas toujours possible de construire de telles procédures dans le cas général. Seuls les langages décidables autorisent la construction de procédures répondant à ces questions quelle que soit l'ontologie.

Definition 2.2.7 (Décidabilité) *Soit L un langage de représentation de connaissances. On dit que L est décidable si et seulement si pour toute ontologie O et toute formule f de $\mathbf{For}_L(\Sigma(O))$, il existe un algorithme permettant de déterminer si f est une conséquence sémantique de O ou non.*

Exemple 2.2.4 *La logique du premier ordre est indécidable.*

Lorsque le langage est décidable, il existe un algorithme déterminant si une formule est conséquence d'une ontologie. Il est donc intéressant de savoir si un algorithme donné détermine bel et bien les conséquences d'une ontologie. On dit qu'un algorithme est correct si tout ce qu'il permet de déduire est effectivement une conséquence sémantique et il est dit complet si toutes les conséquences sémantiques peuvent être déduites.

Definition 2.2.8 (Correction et complétude) *Soit L un langage de représentation de connaissances. Un algorithme A retournant $\mathbf{VRAI}$ ou $\mathbf{FAUX}$ en fonction d'une ontologie O de L et d'une formule f de $\mathbf{for}(\Sigma(O))$ en entrée ($A(O, f) \in \{\mathbf{VRAI}, \mathbf{FAUX}\}$) est dit* correct *si et seulement pour tout O, f, $A(O, f) = \mathbf{VRAI}$ implique $O \models_L f$. Il est dit* complet *si et seulement pour tout O, f, $O \models_L f$ implique $A(O, f) = \mathbf{VRAI}$.*

Habituellement, on cherche à définir des algorithmes à la fois corrects et complets, à condition bien sûr que le langage soit décidable. En revanche, pour des raisons pratiques de complexité du raisonnement, on peut se contenter de la correction ou bien de la complétude si l'opération est suffisamment efficace et l'application moins exigeante.

La complexité du raisonnement est définie soit comme le temps, soit comme la quantité de mémoire nécessaires pour retourner une réponse correcte, en fonction de la taille des données manipulées. Par exemple, on peut définir la complexité comme un nombre d'appels $f(n)$ à des opérations élémentaires, pour une ontologies contenant n termes. Souvent, la fonction n'est pas donnée de façon explicite, mais seul un ordre de grandeur est indiqué. Dans ce cas, la complexité est indiquée comme faisant partie d'une *classe de complexité*. Par exemple, on indique que la complexité est bornée par une fonction polynômiale de la taille des données, traduisant l'appartenance à la classe **P**.

Bien que la décidabilité indique en théorie que l'on peut toujours répondre correctement au problème de la conséquence sémantique, elle n'est pas toujours suffisante. Le temps nécessaire pour obtenir la solution est aussi primordial. Plus précisément, on définit la *complexité* *d'un algorithme* en donnant un ordre de grandeur du nombre d'opérations élémentaires à effectuer pour obtenir la réponse dans le pire cas, en fonction de la taille des données. Quant à la *complexité d'un problème* (ici, pour le problème de conséquence sémantique, on parlera de la complexité de la logique), elle se définit comme la complexité d'un algorithme optimal. Les ordres de grandeurs sont indiqués par des *classes de complexité*, par exemple complexité polynomiale (si la complexité est une fonction polynomiale de la taille de données), complexité exponentielle, etc.

2.2.4 Exemple : logiques de description

Les logiques de description sont une famille de langages de représentations de connaissances. Ces logiques ont été introduites dans les années 80 dans le but de se donner un langage permettant de définir une terminologie d'une façon supposée plus "naturelle" qu'en logique du premier ordre. Elles sont aussi moins expressives que la logique du premier ordre puisque la plupart de ces logiques forment un sous-ensemble décidable de la logique du premier ordre. Maintenant, il existe une grande variété de logiques de description dont l'expressivité et l'efficacité de raisonnement dépend de l'utilisation ou non de constructeurs spécifiques.

Les ontologies en logique de description modélisent les connaissances selon deux niveaux de description : le niveau terminologique (*TBox*) et le niveau factuel (*ABox*). Le niveau terminologique définit des concepts et des rôles représentant respectivement des ensembles d'entités du monde et des relations entre ces entités. Le niveau factuel, quant à lui, énonce des faits sur des individus représentant les entités elles-mêmes.

Niveau terminologique : Les concepts et rôles sont définis soit en les nommant (concepts ou rôles atomiques), soit en les construisant à l'aide d'autres concepts ou rôles et de constructeurs spécifiques. Les connaissances sur ces concepts et rôles sont décrites à l'aide d'axiomes indiquant la subsomption (de concepts ou de rôles) ou la transitivité d'un rôle. La table 8.1 en annexe A.4 donne une liste de constructeurs existant en logiques de description. Elle n'est pas exhaustive.

Une logique de description est un langage utilisant une partie de ces constructeurs. On nomme généralement une logique de description en prenant comme base une logique simple comme $\mathcal{AL}$ et en lui accolant le nom des constructeurs qu'on autorise. Par exemple, $\mathcal{ALCHI}^{\sqcap,\neg}$ autorise les constructeurs d'$\mathcal{AL}$ (voir la figure 2.1) ainsi que la négation de concept, la hiérarchie de rôles, les rôles inverses, la conjonction de rôles et le complément de rôle. Il existe d'autres noms de logiques simples comme $\mathcal{EL}$, $\mathcal{SI}$, DL-Lite, etc. dont certaines imposent des restrictions supplémentaires.

$C \longrightarrow A$	(concept atomique)
$\mid \bot$	(le concept le plus spécifique (ou concept vide))
$\mid \top$	(le concept universel)
$\mid \neg A$	(négation de concept atomique)
$\mid C \sqcap D$	(conjonction de concepts)
$\mid \exists R.\top$	(quantification existentielle limitée)
$\mid \forall R.C$	(quantification universelle complète)

FIGURE 2.1 – Grammaire des expressions de concept en $\mathcal{AL}$

Niveau factuel : Ce niveau n'introduit que des noms d'individus et des propriétés sur ces individus, en exploitant les termes du niveau terminologique. Les axiomes de la ABox sont appelés *assertions*.

Les définitions générales données en section 2.2 peuvent être réécrites de la manière suivante pour les logiques de description.

Definition 2.2.9 (Signature) *Soit LD une logique de description. Une signature de LD est un triplet $\langle \mathcal{C}, \mathcal{R}, \mathcal{U} \rangle$ tel que $\mathcal{C}$ est un ensemble de concepts atomiques, $\mathcal{R}$ un ensemble de rôles atomiques et $\mathcal{U}$ un ensemble d'individus.*

Definition 2.2.10 (Formule) *Soit LD une logique de description. Une formule de LD pour une signature Σ est l'un des axiomes de TBox ou de ABox indiqués dans la table 8.1 et autorisés par le langage, où C et D sont soit des concepts atomiques, soit des concepts complexes obtenus en utilisant les constructeurs du langage ; R et S sont soit des rôles atomiques, soit des rôles construits à l'aide des constructeurs de rôles ; a, a_1 et a_2 sont des individus.*

Definition 2.2.11 (Ontologie) *Soit LD une logique de description. Une ontologie de LD est un couple $\langle \Sigma, F \rangle$ telle que Σ est une signature et $F = \langle T, A \rangle$ contient des axiomes de TBox T et des assertions de ABox A.*

Definition 2.2.12 (Interprétation) *Soit LD une logique de description et $\Sigma = \langle \mathcal{C}, \mathcal{R}, \mathcal{U} \rangle$ une signature de LD. Une interprétation de Σ est un couple $\langle I, \Delta \rangle$ où Δ est un ensemble non vide et I un triplet de fonctions $I_{\mathcal{C}} : \mathcal{C} \rightarrow 2^{\Delta}$, $I_{\mathcal{R}} : \mathcal{R} \rightarrow 2^{\Delta \times \Delta}$ et $I_{\mathcal{U}} : \mathcal{U} \rightarrow \Delta$. On étend la fonction d'interprétation aux concepts complexes en appliquant récursivement les règles de la table 8.1.*

Definition 2.2.13 (Satisfaction) *Soient LD une logique de description, Σ une signature de LD, I une interprétation de Σ et f une formule de $\mathbf{For}(\Sigma)$. I satisfait f si et seulement la contrainte d'interprétation associé à la formule dans la table 8.1 est satisfaite.*

Le format de description d'ontologies privilégié sur le Web sémantique est OWL (Web Ontology Language [50]), recommandé par le World Wide Web Consortium depuis 2004. OWL se décline en trois sous-langages offrant différents degrés d'expressivité : OWL-Lite, OWL-DL et OWL-Full. La sémantique de ce langage repose sur le formalisme des logiques de description, bien que la version la plus expressive (OWL-Full) sorte de ce cadre. OWL-Lite correspond à la logique de description $\mathcal{SHIF}(D)$ et OWL-DL correspond à $\mathcal{SHION}(D)$. Le D entre parenthèses indique que la logique supporte l'utilisation de types de données comme les entiers, les nombres décimaux, les dates, etc.

Les problèmes de raisonnement en logiques de description sont classés dans plusieurs catégories :

- déterminer la cohérence d'une TBox ou bien d'une ABOX ou bien des deux réunis ;
- déterminer la subsomption de deux concepts, c'est-à-dire savoir si $C \sqsubseteq D$ pour C et D deux concepts (potentiellement complexes) de l'ontologie ;
- déterminer si un concept est satisfiable, c'est-à-dire s'il peut être interprété comme non vide ;
- répondre à des requêtes sur la ABox, typiquement en demandant quels sont les individus appartenant à un concept C.

Le plus souvent, ces problèmes peuvent être tous ramenés à celui de la satisfiabilité. Par exemple, si la logique permet d'exprimer la négation de concepts, alors le problème de subsomption pour C et D est équivalent à déterminer la non-satisfiabilité du concept $C \sqcap \not D$.

Tous ces problèmes retrouvent leur équivalent dans les réseaux de connaissances, mais l'on s'intéressera avant tout au problème de la cohérence d'un réseau d'ontologies alignées, au chapitre 8.

2.2.5 Systèmes de raisonnements pour les logiques de description

De nombreux systèmes logiciels ont été implémentés pour raisonner sur diverses logiques de description. La table 2.1 donne une liste de systèmes, accompagnée des logiques de description prises en charge par le raisonneur, des articles connexes, de l'URL à laquelle on peut télécharger ces logiciels, ainsi que le langage d'implémentation.

Système	Logique	Articles	Implémentation
CEL [2]	$\mathcal{EL}^+$	[10]	Lisp
Fact++ [3]	$\mathcal{SHOIQ}(D)$	[72]	C++
fuzzyDL [4]	fuzzy $\mathcal{SHIF}$	[69]	Java/C++
KAON2 [5]	$\mathcal{SHIQ}(D)$	[54]	Java
SPASS/MSPASS [6]	$\mathcal{ALB}$+	[75]	C
Pellet [7]	$\mathcal{SROIQ}(D)$	[67]	Java
QuOnto [8]	DL-Lite + Requêtes	[2]	Java
RacerPro [9]	$\mathcal{SHIQ}(D)$	[59]	Java

TABLE 2.1 – Systèmes de raisonnement sur des logiques de description.

2.2.6 Nœud de connaissances

Les notions importantes de représentation de la connaissance ont été introduites. Il est maintenant possible de modéliser formellement un nœud dans un réseau de connaissances.

À chaque nœud d'un réseau de connaissances est associée une logique L (c'est-à-dire une syntaxe $\mathbf{Sign}_L, \mathbf{For}_L$ et une sémantique $\mathbf{Int}_L, \models_L$ particulières), une ontologie $O = \langle \Sigma(O), A(O) \rangle \in \mathbf{Sign}_L \times \mathbf{For}_L(\Sigma(O))$ dans le langage de cette logique, ainsi qu'un système d'inférence Γ. Ainsi, on peut modéliser un nœud de connaissances par un triplet $\langle L, O, \Gamma \rangle$, ce qui correspond finalement à la définition d'un contexte dans [43].

Definition 2.2.14 (Nœud de connaissances) *Un nœud de connaissances est un triplet $\langle L, O, \Gamma \rangle$ où L est une logique dont l'ensemble des signatures est $\mathbf{Sign}_L$, les formules sont déterminées par $\mathbf{For}_L$, les interprétations sont définies par $\mathbf{Int}_L$ et la relation de satisfaction est donnée par $\models_L$; O est une ontologie dans le langage L, c'est-à-dire $\langle \Sigma(O), A(O) \rangle \in \mathbf{Sign}_L \times \mathbf{For}_L(\Sigma(O))$ et Γ est un système d'inférence pour le langage L.*

Ces nœuds constituent les sommets du graphe formé par le réseau. Il convient maintenant de définir ce qui tient lieu d'arcs et connecte les ontologies entre elles, à savoir les alignements d'ontologies.

2. `http://lat.inf.tu-dresden.de/systems/cel/`
3. `http://owl.man.ac.uk/factplusplus/`
4. fuzzy $\mathcal{SHIF}$ est la logique de description $\mathcal{SHIF}$ étendue à l'aide d'opération sur la logique floue. `http://gaia.isti.cnr.it/~straccia/software/fuzzyDL/fuzzyDL.html`
5. `http://kaon2.semanticweb.org/`
6. SPASS est un raisonneur pour la logique du premier ordre ainsi que de nombreuses logiques. Il permet de raisonner sur des logiques de description très expressives. `http://www.cs.man.ac.uk/~schmidt/mspass/`
7. `http://pellet.owldl.com/`
8. `http://www.dis.uniroma1.it/~quonto/`
9. `http://www.racer-systems.com/`

2.3 Alignement d'ontologies

Dans sa plus grande généralité, un alignement d'ontologie est l'expression des relations existantes entre plusieurs ontologies. Ainsi, un alignement formalise les connections entre nœuds de connaissances. Il permet donc de faire transiter et coordonner les connaissances de l'ensemble. Cette section ne décrit que la partie syntaxique de l'alignement et se décompose comme suit : la section 2.3.1 définit les notions générales sur les alignements d'ontologies. La section 2.3.2 passe en revue divers formats de représentation des alignements. La sémantique des alignements fait partie intégrante de l'apport de cette thèse, et sera donc discutée plus longuement par la suite, en particulier dans les chapitres 3 et 4.

2.3.1 Langage d'alignements d'ontologies

La représentation abstraite d'un alignement est assez similaire à celle d'une ontologie. La différence essentielle entre ces deux objets est que l'ontologie définit un vocabulaire, tandis que l'alignement ne fait que relier des éléments de vocabulaires existants. Mis à part cette différence, on retrouve des similitudes entre ces deux notions. L'alignement fournit aussi un ensemble de formules, que l'on appelle des correspondances. Ces formules décrivent une relation sémantique entre des entités d'ontologies différentes.

Les entités mises en relation ne sont pas forcément de simples termes ou symboles de la signature d'une ontologie. Non seulement ils peuvent être des éléments construits du langage de représentation local (par exemple $\exists R.C$ en logique de description), mais ils peuvent aussi être construit selon une syntaxe propre au langage de représentation des alignements.

Ici, on se contentera de représenter des relations entre paires d'ontologies, mais les définitions pourraient être généralisées. Ainsi, une correspondance identifie une entité e dans une première ontologie, une entité e' dans une seconde ontologie et une relation r se tenant entre e et e'.

Un langage d'alignements d'ontologies doit donc définir les entités pouvant être identifiées dans une ontologie, ainsi que les relations exprimables.

Definition 2.3.1 (Langage d'entités) *Un* langage d'entités *définit une fonction* Ent *associant à toute signature d'ontologies* Σ *un ensemble* $\mathrm{Ent}(\Sigma)$ *contenant au moins* Σ*. On étend la définition de* Ent *aux ontologies, c'est-à-dire en notant* $\mathrm{Ent}(O)$ *pour* $\mathrm{Ent}(\Sigma(O))$*. Les éléments de* $\mathrm{Ent}(O)$ *sont appelées les* entités alignables *de* O*.*

Exemple 2.3.1 *Un exemple de langage d'entités est celui permettant d'identifier des conjonctions ou disjonctions de termes d'une ontologie, c'est-à-dire pour tout* Σ, $\mathrm{Ent}(\Sigma) = \Sigma \cup \{A \wedge B \mid A, B \in \Sigma\} \cup \{A \vee B \mid A, B \in \Sigma\}$.

Dans le cas des logiques de description, les entités alignées sont généralement des concepts, des rôles ou des individus. Le chapitre 6 décrira un langage d'alignements permettant d'identifier des entités complexes. Le langage d'entités fait partie intégrante du langage d'alignements. En plus de cela, un langage d'alignements définit un ensemble de relations.

Definition 2.3.2 (Langage d'alignements) *Un* langage d'alignements *est un couple* $\langle \mathrm{Ent}, \mathcal{R} \rangle$ *où* Ent *est un langage d'entités et* $\mathcal{R}$ *un ensemble de symboles de relations.*

Ces langages permettent de décrire des correspondances.

Definition 2.3.3 (Correspondance) *Soit O_1 et O_2 deux ontologies. Soit $L_A = \langle \mathrm{Ent}, \mathcal{R} \rangle$ un langage d'alignements. Une* correspondance *est un triplet $\langle e_1, e_2, r \rangle$ où*

- *$e_1 \in \mathrm{Ent}(O_1)$ et $e_2 \in \mathrm{Ent}(O_2)$ sont des entités alignables des ontologies O_1 et O_2 respectivement ;*
- *$r \in \mathcal{R}$ dénote une relation existant entre e_1 et e_2.*

Exemple 2.3.2 *Par exemple $\langle Homme \sqcup Femme, Personne, = \rangle$ est une correspondance possible représentant le fait que l'union de l'ensemble des hommes et des femmes (termes d'une ontologie O_1) est équivalent à l'ensemble des personnes (terme d'une autre ontologie O_2).*

Un alignement d'ontologies est alors un ensemble de correspondances. Il existe d'autres définitions des alignements, en particulier dans [34] où la notion de correspondance comprend en plus une valeur qui exprime le degré de confiance que l'on associe à la relation entre e_1 et e_2. Le traitement de ces valeurs au niveau sémantique est problématique mais celles-ci peuvent être manipulées à part, indépendamment de tout processus déductif.

La manière dont les entités des ontologies sont représentées peut dépendre à la fois du langage d'ontologies et du langage d'alignements. On verra au chapitre 6 qu'il est préférable que la construction de ces entités ne dépende que du langage d'alignements, afin de mieux assurer l'indépendance du médiateur et des nœuds de connaissances.

Ces définitions donnent donc une description abstraite de l'alignement d'ontologies. La section suivante décrit des formats de représentation concrets pour les alignements d'ontologies.

2.3.2 Les formats de représentation d'alignements

Cette section décrit divers formats proposés pour représenter des alignements dans le but de les stocker, les comparer, les combiner ou les utiliser dans des opérations de médiation. Je n'entrerai pas dans les détails techniques de ces formats et pour une comparaison détaillée je renverrai à l'ouvrage de référence dans le domaine [34].

Dans l'ensemble, ces formats se répartissent entre d'un côté des langages expressifs mais souvent contraignants pour la représentation locale des ontologies et de l'autre des formats polyvalents mais limités en expressivité. En fait, on verra au chapitre 6 que l'on peut concilier ces deux qualités en adoptant la sémantique de médiation développée au chapitre 4.

OWL et langages d'ontologies

Comme indiqué en section 2.2.4, le langage OWL est un langage d'ontologies fondé sur les logiques de description. Mais on peut aussi le considérer comme un langage pour exprimer des correspondances entre ontologies. En fait, les primitives `equivalentClass` et `equivalentProperty` ont été introduites pour relier des éléments d'ontologies décrivant le même domaine de connaissances. D'ailleurs, cette utilisation a même été affirmée par le groupe de travail sur les bonnes pratiques (*best practice working group* [74]). Cependant, ces primitives ne sont que des raccourcis pour d'autres primitives (à savoir `subClassOf`, `subPropertyOf`) qui permettent aussi de mettre en relation des entités. Par exemple l'ontologie :

```
<owl:Class rdf:about="http://www.example.com/onto1#Personne">
 <owl:equivalentClass rdf:resource="http://www.example.com/onto2#Humain"/>
</owl:Class>

<owl:Class rdf:about="http://www.example.com/onto1#Chercheur">
```

```
 <owl:subClassOf rdf:resource="http://www.example.com/onto2#Adulte"/>
</owl:Class>

<owl:Property rdf:about="http://www.example.com/onto1#pereDe">
 <owl:subPropertyOf rdf:resource="http://www.example.com/onto2#parentDe"/>
</owl:Property>
```

peut être vue comme un alignement exprimant l'équivalence des classes `Humain` et `Personne`, la subsomption entre les classes `Chercheur` et `Adulte`, et la relation de sous propriété entre `pereDe` et `parentDe`. Plus généralement, dès qu'une ontologie fait intervenir des termes provenant de différentes ontologies, on peut la voir comme l'expression d'un alignement. Par exemple, l'ontologie

```
<owl:Class rdf:ID="Femme">
 <owl:equivalentClass>
  <owl:Class>
   <owl:subClassOf rdf:resource="http://www.example.org/ontologie2#Personne"/>
   <owl:subClassOf>
    <owl:Restriction>
     <owl:onProperty rdf:resource="http://www.example.org/ontologie2#sexe"/>
     <owl:hasValue rdf:resource="http://www.example.org/ontologie2#F"/>
    </owl:Restriction>
   </owl:subClassOf>
  </owl:Class>
 </owl:equivalentClass>
</owl:Class>
```

exprime le fait que la classe `Femme` dans l'ontologie `http://www.example.org/ontologie1` est équivalente à une restriction de la classe `Personne` dans l'ontologie `http://www.example.org/ontologie2` pour laquelle le genre est `F`.

Utiliser un langage d'ontologies comme OWL pour l'expression des alignements est possible. D'ailleurs, ce langage est assez riche pour exprimer des correspondances subtiles. En outre, il dispose d'une sémantique formelle. Néanmoins, cette approche a deux défauts :

1. un langage d'ontologies peut être utilisé pour définir de nouveaux termes ou des connaissances sur un domaine, indépendamment des relations qui rapprochent deux ontologies. Il devient alors difficile de séparer ce qui est de l'ordre des connaissances locales et ce qui exprime les liens sémantiques entre ontologies. Il est toutefois possible d'imaginer qu'un médiateur dispose lui-même de connaissances spécifiques, en plus des relations entre les ontologies qu'il doit faire interopérer. Mais en l'absence de séparation des deux représentations, l'échange et le stockage d'alignements seuls, en vue d'une réutilisation, est compromis.
2. à supposer que cette méthode soit utilisée pour aligner des ontologies en OWL, il implique vraisemblablement que la sémantique de OWL sera utilisée pour interpréter l'ensemble du réseau d'ontologies. Or, on verra dans le chapitre 3 que ceci implique des contraintes trop fortes sur des réseaux d'ontologies indépendantes et hétérogènes.

Il convient donc de s'intéresser aux formats ayant pour objectif spécifique de représentation des relations sémantiques entre ontologies.

MAFRA

MAFRA [66, 49] signifie *MApping FRAmework*. Il s'agit d'un système complet pour extraire des alignements d'ontologies et les exécuter en tant que transformation de données d'une ontologie à une autre. Ce système a d'abord été conçu pour fonctionner avec le langage DAML+OIL [22].

Dans MAFRA, le format d'alignement est spécifié par une ontologie appelée *Semantic Bridging Ontology* (SBO). Une instantiation de cette ontologie constitue un document d'alignement d'ontologies (il s'agit d'une instance de la classe `Mapping`).

Les correspondances dans MAFRA peuvent être très précises et se combinent à des services, eux aussi spécifiés selon l'ontologie SBO, précisant les transformations à effectuer sur les données pour rendre interopérables.

Des relations peuvent être explicitées entre concepts, relation ou attribut, que l'on peut davantage préciser par l'utilisation de conditions. Ce système est avant tout destiné à traduire des instances d'une ontologie vers une autre. Ainsi, lorsqu'une instance d'une entité identifiée dans une correspondance est repérée et qu'elle vérifie les conditions, elle est traduite en une instance de l'entité identifiée dans la seconde ontologie, tout en lui appliquant des conversions par le biais des services.

Ce format est donc très expressif, mais il manque de polyvalence car il est fortement lié à l'architecture de MAFRA. Les correspondances sont décrites autant en termes de relations sémantiques qu'en termes d'opérations à effectuer pour la transformation. Ces deux aspects ne sont pas bien délimités. En outre, l'accent est mis sur l'opération de transformation, sans chercher à partager les alignements pour d'autres tâches, indépendamment de l'infrastructure MAFRA. Il peut être alors difficile de les manipuler dans d'autres cas d'utilisation (voir section 1.1).

En outre, la sémantique formelle des ponts sémantiques n'est pas clairement établie. En effet, le résultat attendu d'une transformation ne peut être déterminé que grâce à l'algorithme propre à MAFRA. Seuls des spécifications informelles ou implicites, ainsi que l'implémentation elle-même, permettent de deviner la signification des concepts introduit par l'ontologie de MAFRA.

C-OWL (*Contextualized OWL***)**

C-OWL est une extension du langage OWL utilisée pour exprimer des relations entre concepts ou rôles d'ontologies décrites en OWL, mais hétérogènes du point de vue de leur conception. Les constructeurs de C-OWL sont appelés "passerelles" (*bridge rules* en anglais) et elles permettent d'exprimer une famille de relations sémantiques entre concepts, rôles et individus interprétés dans des domaines hétérogènes. Étant données deux ontologies O_1 et O_2, une passerelle de O_1 vers O_2 exprime une relation entre un concept/rôle/individu de O_1 perçu du point de vue de O_2 et un concept/rôle/individu de O_2. Les passerelles sont directionnelles puisque la relation est exprimée du point de vue de l'ontologie cible. Donc une passerelle de O_1 vers O_2 n'est pas équivalente à une passerelle inverse de O_2 vers O_1.

C-OWL est une application du formalisme des logiques de description distribuées (DDL) que je décrirai plus en détail dans le chapitre 3. Je ne détaille donc pas ici la syntaxe du langage.

L'avantage de C-OWL par rapport aux autres formats est qu'il dispose d'une sémantique formelle conforme au paradigme du raisonnement contextuel de [43]. En particulier, une correspondance indique une relation directionnelle, établie du point de vue d'une ontologie, et non une assertion globale partagée par les ontologies mises en commun. Ainsi une correspondance C-OWL peut établir que, selon O_2, le concept A_1 de l'ontologie O_1 est perçu comme plus spécifique que le concept B_2 de O_2, mais cela n'indique pas que O_1 perçoit B_2 comme un sous-concept de A_1. Le chapitre 3 reviendra plus en détail sur la sémantique de ces passerelles.

C-OWL permet d'exprimer des alignements relativement simples (on ne peut pas utiliser de classes construites dans une passerelle, seulement des classes nommées). La partie la plus expressive se trouve dans les relations utilisées dans les correspondances. Ce langage est conçu pour mettre en relation des ontologies en OWL uniquement.

SWRL

SWRL [46] est le langage de règles pour le Web sémantique (SWRL signifie *Semantic Web Rule Language*). Le langage SWRL étend le langage OWL en ajoutant la notion de règles de déduction explicites, interprétées comme des clauses de Horn. Une règle se compose d'un antécédent ou "corps" et d'un conséquent ou "tête". Une règle du type `tête` ← `corps` signifie que si l'on a pu démontrer la partie `corps`, alors la partie `tête` est vraie. Ces règles peuvent être vues comme des correspondances entre ontologies lorsque les entités mises en jeu dans l'antécédent et le conséquent appartiennent à des ontologies différentes.

Une règle peut représenter une correspondance complexe puisque les constructeurs de OWL ainsi que des variables peuvent être utilisées pour les décrire. Ainsi, on peut indiquer qu'une `Personne` dans une ontologie O_1 avec l'attribut `sexe` égal à `F` est une `Femme` dans une ontologie O_2. Les variables permettent notamment de représenter des chemins de relations, comme par exemple le fait qu'un `oncle` est équivalent à la relation obtenue par composition de `parent` et `frere`. SWRL fournit aussi un ensemble de prédicats prédéfinis sur les types de données XML, ainsi que des opérateurs sur des ensembles.

De même que les axiomes des langages d'ontologies comme OWL peuvent exprimer des correspondances, les règles de SWRL peuvent être utilisées pour spécifier des alignements. Ces règles ont l'avantage de se distinguer des axiomes et assertions des ontologies, donc peuvent permettre une certaine indépendance des connaissances locales et des connaissances inter-ontologies.

Néanmoins, SWRL n'est en fin de compte qu'un langage d'ontologies plus expressif que OWL. Il souffre alors d'inconvénients similaires. Sa sémantique formelle ne lui permet pas de vraiment séparer les interprétations liées aux ontologies et celle liée à l'alignement. Les règles ne fonctionnent pas comme des transformations mais juste comme des règles logiques, donc l'interprétation d'un réseau d'ontologies complet en OWL et SWRL revient à fusionner l'ensemble des connaissances du réseau en une seule ontologie. Le chapitre 3 reviendra sur cet inconvénient vis-à-vis de la médiation dans les réseaux d'ontologies alignées.

Le format de l'API d'alignement

Le format associé à l'API d'alignement [30] se veut un format versatile pour le stockage, la manipulation et l'utilisation d'alignements au sein de diverses applications, de façon indépendante du langage utilisé pour décrire les ontologies. Ce format est intégré à une interface de programmation d'application (*Application Programming Interface* ou API) développée afin de fournir des opérations courantes sur les alignements et des primitives pour leur production et leur utilisation. Il est en outre assez souple pour être étendu selon les besoins. En revanche, il n'a pas vocation à disposer d'une sémantique formelle. Au contraire, l'API permet de produire, à partir d'un alignement dans ce format, toute sorte de formats structurés pouvant, quant à eux, disposer de leur propre sémantique. Notamment, un alignement peut être converti en axiomes OWL ou bien en règles SWRL.

Pour faire simple, un alignement dans ce format est constitué d'informations de haut niveau sur la nature de l'alignement — son arité (bijectif, partiel, injectif, etc.), son niveau (un indicateur de l'expressivité des correspondances), la date de création, l'algorithme de génération, les ontologies alignées — et d'un ensemble de correspondances (*Cell*) comprenant cinq composants :

L'entité 1 (`entity1`) est la première entité alignée. Elle est identifiée par un URI et correspond à une entité langage d'entités, comme par exemple une classe en OWL, une règle en SWRL, ou même un enregistrement dans une base de données.

L'entité 2 (`entity2`) est la seconde entité alignée, soumise aux mêmes contraintes que l'entité1.

La `relation` (par défaut "=") est la relation entre les entités. Elle n'est pas restreinte à la seule équivalence et peut être sophistiquée, comme par exemple la subsomption, l'incompatibilité [44], ou même quelque relation floue.

La force (`strength`) (par défaut $\top$) dénote le degré de confiance que l'on a envers cette correspondance. Puisque de nombreuses méthodes de mise en correspondance calculent la "force" de la relation entre les entités, cette force peut être indiquée par une mesure normalisée. La mesure doit appartenir à un ensemble ordonné M incluant un élément maximum $\top$ et un élément minimum $\bot$. Actuellement, l'API contraint ces valeurs à être des nombres décimaux entre 0 et 1. Néanmoins, ceci pourrait être généralisé ci nécessaire à n'importe quel treillis.

id (`id`) est un identifiant pour la correspondance.

Le format d'alignement a été spécifiquement conçu pour décrire des correspondances indépendamment du langage d'ontologies, mais aussi indépendamment de la tâche pour laquelle il sera utilisé au final. En le combinant aux fonctionnalités de l'API d'alignement, il permet entre autre de générer d'autres formats plus spécifiques. En revanche, contrairement à certains langages introduits jusqu'à présent, il n'est pas pourvu d'une sémantique formelle.

Toutefois, une bonne caractéristique de ce format est son ouverture qui permet d'introduire de nouvelles relations et si nécessaire de nouveaux type d'expressions tout en restant compatible avec des langages plus pauvres. Justement, on verra au chapitre 6 qu'un langage très expressif peut être implanté dans ce format.

SKOS

SKOS [53, 52] signifie *Simple Knowledge Organisation System* ou système simple d'organisation de connaissances. Le vocabulaire de SKOS est donné sous la forme d'un schéma RDF visant à exprimer les relations entre ontologies "légères" (parfois appelées *folksonomies* par contraction des termes *folk* et taxonomie) ou thésaurus.

L'objectif de SKOS est d'offrir une couche supplémentaire pour faire le lien entre différents formalismes, soit pour établir des correspondances entre entités, soit pour indiquer des relations de plus haut niveau entre les ontologies elles-mêmes ou entre les langages d'ontologies. Il s'agit plutôt d'annotations ou de méta-données pour les ontologies plutôt qu'un véritable format de spécification.

Il est néanmoins possible d'identifier des concepts et de les relier entre eux, donc d'établir, d'une certaine manière, des correspondances entre ontologies. Pour cela, SKOS définit des "relations sémantiques" qui expriment des relations entre concepts SKOS. Les relations entre concepts peuvent indiquer une inclusion (plus générale (*broader*) ou plus spécifique (*narrower*)) de concepts ou bien une relation informelle (*related*, c'est-à-dire lié).

SKOS a l'avantage d'être un vocabulaire léger pour définir des relations entre entités ou ensemble d'entités, à différents niveau de granularité. En effet, il est possible de décrire les concepts mais aussi les ontologies elles-mêmes et de les mettre en relation assez librement. SKOS se distingue donc du langage d'ontologies, car il permet de manipuler tout type de descriptions organisées identifiables par des URI comme s'il s'agissait d'un ensemble de classes utilisables. Les relations reliant ces entités sont très générales et, par conséquent, ne disposent pas d'une sémantique formelle précise. Il est donc moins intéressant à utiliser au sein d'un système de médiation devant garantir la correction des opérations. Mais la souplesse de SKOS permet de définir des relations sémantiques entre le vocabulaire de SKOS lui-même et d'autres vocabulaires.

Conclusion

La diversité des formats de représentation d'alignements montre que le besoin d'un format à part a été identifié. Cependant, il est bien trop fastidieux de concevoir des alignements manuellement et, de ce fait, on ne peut soutenir l'idée que les réseaux d'ontologies sont répandus sans montrer l'existence d'outils pour leur construction automatique ou semi-automatique.

Considérant la complexité de certaines ontologies, les alignements peuvent difficilement être conçus manuellement. C'est pourquoi, des systèmes automatiques ou semi automatiques de découverte ou de construction d'alignements ont été proposés. Une présentation détaillée et comparative des systèmes se trouve au chapitre 6 de [34].

2.4 Réseau d'ontologies alignées

Toutes les composantes d'un réseau de connaissances peuvent donc être représentées. Les nœuds correspondent à des ontologies, dans une logique munie d'une sémantique formelle. Les connaissances reliant les nœuds sont représentés par des alignements d'ontologies. Par conséquent, la représentation d'un réseau de connaissances est un *réseau d'ontologies alignées*, dont voici la définition.

Definition 2.4.1 (Réseau d'ontologies alignées) *Un* réseau d'ontologies alignées *est un couple* $\langle \mathbf{N}, (A_{ij})_{i,j\in\mathcal{N}} \rangle$ *où* $\mathcal{N}$ *désigne un ensemble fini de nœuds de connaissances (définition 2.2.14) et pour tout* $n, n' \in \mathbf{N}$*,* $A_{n,n'}$ *est un alignement entre les ontologies de* n *et de* n'*.*

On peut remarquer que cette définition indique qu'il y a exactement un alignement par couple n, n'. Si en pratique certains couples d'ontologies ne sont pas alignées, on peut toujours formaliser cela en un alignement vide. Si en revanche il existe plusieurs alignements entre deux même ontologies, on peut considérer l'alignement résultant de leur union, puisqu'un alignement est un simple ensemble de correspondances. Ce mémoire a pour objectif de fournir une sémantique formelle à ce type de réseaux.

2.5 Bilan

Ce chapitre a introduit le vocabulaire et les notions usuels de représentation de connaissances. On a vu qu'un langage de représentation de connaissances L est caractérisé par ses signatures $\mathbf{Sign}_L$ et ses formules $\mathbf{For}_L$ pour la syntaxe, des interprétations $\mathbf{Int}_L$ et une relation de satisfaction $\models_L$ pour la sémantique. Les logiques de description ont été présentées comme exemple typique de langage de représentation de connaissances sur le Web. La syntaxe des alignements a aussi été présentée, en montrant que divers formats existent pour les représenter.

On cherche ici à définir une sémantique pour les réseaux d'ontologies alignées et donc, en fin de compte, définir la notion d'interprétation et de satisfaction d'une combinaison d'ontologies et d'alignements. Ce chapitre a complètement déterminé la syntaxe de ces réseaux et de tels réseaux d'ontologies alignées existent sur le Web. Pour s'en convaincre, on peut prendre pour exemple les diverses ontologies développées indépendamment par la FAO [10] et alignées par des algorithmes dans le cadre de la compétition annuelle de mise en correspondances d'ontologies [11].

10. L'organisation des Nations Unies pour l'alimentation et l'agriculture (*Food and Agriculture Organization* `http://www.fao.org/`).
11. *Ontology Alignment Evaluation Initiative* (OAEI `http://oaei.ontologymatching.org/`).

Le chapitre suivant décrit des sémantiques de réseaux d'ontologies existantes et montre que celles-ci ne conviennent pas aux critères que l'on s'est fixé au chapitre 1. Par conséquent, le chapitre 4 définira notre propre approche de la sémantique des réseaux d'ontologies, favorisant la gestion séparée des alignements et donc la médiation. Puis, après trois chapitres montrant des applications importantes de la sémantique, on fournira au chapitre 8 une procédure correcte et complète pour la vérification de cohérence d'un réseau d'ontologies en logiques de description.

Chapitre 3

Sémantiques distribuées

Résumé

Ce chapitre présente un état de l'art des formalismes pour le raisonnement distribué. On trouve d'un côté des systèmes ne définissant pas une sémantique particulière pour des réseaux d'ontologies. Dans ce cas, on considère que l'ensemble du réseau forme une unique théorie dont chaque nœud représente une partie de la connaissance. Les alignements complètent eux aussi cette même théorie en joignant les différents blocs de connaissances. La médiation est facilitée dans ce cas puisque tous les nœuds s'interprètent dans un même contexte, seul le vocabulaire et la couverture du domaine de connaissance varient d'une ontologie à l'autre. Cette approche est malheureusement inadaptée à la médiation entre ontologies de provenances et de contextes hétérogènes. D'un autre côté, on trouve des formalismes dont la sémantique est spécifiquement conçue pour tolérer des contextes différents, en affectant à chaque ontologie une interprétation qui lui est propre. Ces formalismes sont les logiques du premier ordre distribuées (*Distributed First Order Logics* ou DFOL), les logiques de description distribuées (*Distributed Description Logics* DDL), les $\mathcal{E}$-connections et les logiques de description par paquetages (*Package-based Description Logics* ou P-DL). Ils diffèrent par la manière dont les différentes interprétations sont reliées entre elles. Ces formalismes sont assez bien adaptés aux raisonnement multi-contextes mais ne sont pas conformes au principe de médiation dans le sens où ils obligent chaque nœud à gérer localement la mise en correspondance avec les autres ontologies.

Sommaire

3.1 Introduction

Le raisonnement distribué peut être étudié selon deux grands axes, l'un au niveau de la procédure de déduction, l'autre au niveau de la sémantique. Une procédure distribuée consiste à utiliser séparément différents systèmes de raisonnement localisés, mis en commun par un protocole d'échange de messages. Ces messages peuvent transiter via un médiateur ou non. Définir une procédure distribuée pour le raisonnement requiert que la sémantique soit connue. Quant à la sémantique distribuée, elle consiste à affecter une interprétation distincte à chaque nœud d'un réseau d'ontologies, avec des règles de compatibilité entre interprétation.

Plus précisément, on peut regrouper l'ensemble des définitions d'interprétation distribuée en une définition générique homogène qu'il suffira d'instancier pour chaque formalisme. Pour cela, on rappelle que chaque ontologie est décrite dans un langage dont on connait la sémantique. Par conséquent, on peut affecter à chaque ontologie une interprétation selon sa propre sémantique.

Cet ensemble d'interprétations locales forme la base d'une interprétation d'un réseau d'ontologies. Pour vérifier qu'une interprétation distribuée satisfait le réseau entier, il faut premièrement que chaque interprétation locale soit un modèle de son ontologie associée. Deuxièmement, les combinaisons d'interprétations doivent satisfaire des contraintes supplémentaires posées par les alignements.

Ainsi, les alignements restreignent les combinaisons de modèles autorisées pour valider le réseau. La figure 3.1 illustre le principe en représentant schématiquement les ensembles d'interprétations et de modèles des ontologies, et montre la compatibilité par des arcs reliant les modèles compatibles. En effet, puisque l'alignement de cette figure restreint les couples de modèles validant le réseau, cela revient à dire que l'alignement définit une relation ensembliste entre les ensembles de modèles relatifs aux ontologies.

Ainsi, d'une manière très générale, on peut définir l'interprétation d'un alignement d'ontologies comme une relation entre les ensembles de modèles des ontologies alignées, soit un sous-ensemble de $\mathrm{Mod}_{L_1}(O_1) \times \mathrm{Mod}_{L_2}(O_2)$.

Definition 3.1.1 (Modèle d'ontologies alignées) *Soient O_1 une ontologie dans une logique L_1 dont les modèles sont $\mathrm{Mod}_{L_1}(O_1)$ et O_2 une ontologie de L_2 dont les modèles sont $\mathrm{Mod}_{L_2}(O_2)$ et soit A un alignement de O_1 et O_2. Les modèles des ontologies alignées $\langle O_1, O_2, A\rangle$ forment un ensemble $\mathrm{Mod}(A, O_1, O_2) \subseteq \mathrm{Mod}_{L_1}(O_1) \times \mathrm{Mod}_{L_2}(O_2)$.*

D'une façon plus abstraite, on peut représenter cela par un diagramme, comme dans la figure 3.2. On pourra faire le rapprochement avec la formalisation des alignements en théorie des catégories dans [79].

Cette notion se généralise aux réseaux d'ontologies alignées en utilisant une relation n-aire au lieu d'une relation binaire.

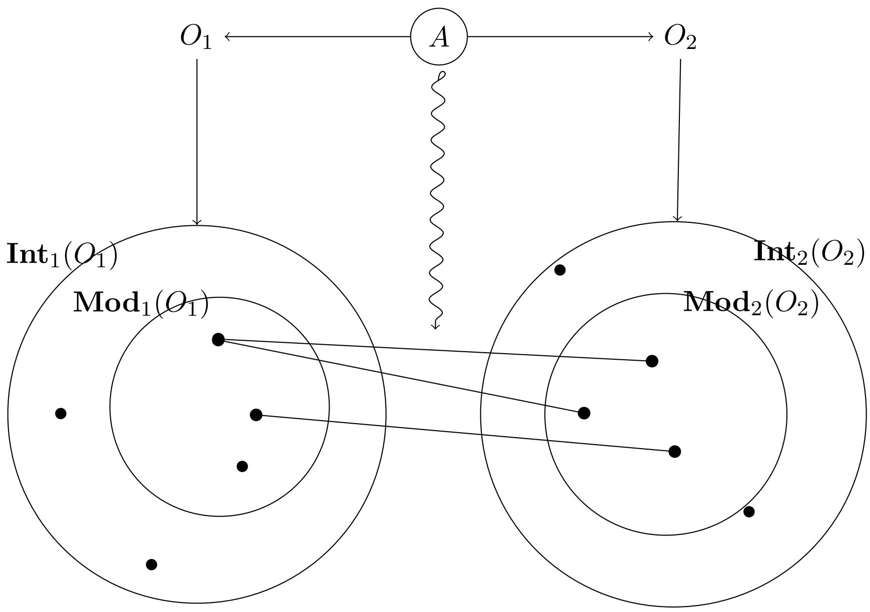


FIGURE 3.1 – O_1 (respectivement O_2) est une ontologie dont l'ensemble des interprétations est $\mathbf{Int}_1(O_1)$ (respectivement $\mathbf{Int}_2(O_2)$) et l'ensemble des modèles est $\mathbf{Mod}_1(O_1)$ (respectivement $\mathbf{Mod}_2(O_2)$). L'alignement A détermine les règles de compatibilité des modèles. Les couples de modèles compatibles sont représentés par des arcs.

Definition 3.1.2 (Modèle d'un réseau d'ontologies alignées) *Soit $S = \langle \mathbf{O}, \mathbf{A} \rangle$ un réseau d'ontologies alignées où chaque ontologie $O_i \in \mathbf{O}$ est associée à une logique L_i. Les modèles du réseau S forment un ensemble* $\mathrm{Mod}(S) \subseteq \mathrm{Mod}_{L_1}(O_1) \times \cdots \times \mathrm{Mod}_{L_n}(O_n)$.

Définir une sémantique pour les réseaux d'ontologies alignées revient finalement à définir précisément les modèles des ontologies alignées en fonction des alignements et des sémantiques locales. Cette définition générale couvre les différentes sémantiques distribuées proposées dans la littérature et développées ci-dessous.

3.2 Raisonnement distribué en logique "classique"

Le raisonnement distribué à l'aide de logiques classiques n'impose pas la définition d'un nouveau formalisme ni de sa sémantique. Avec cette approche, on considère que les ontologies, de même que les alignements, décrivent chacun une partie d'une unique théorie globale. On peut alors interpréter l'ensemble du réseau comme on interprète une simple ontologie. Seul subsiste le problème, certes complexe, du raisonnement distribué, c'est-à-dire par utilisation conjointe de plusieurs raisonneurs locaux, communiquant par messages. Je vais d'abord présenter le principe du raisonnement distribué en logique classique (section 3.2.1), puis les systèmes ou algorithmes proposés dans la littérature (section 3.2.2). Cette approche est critiquée dans la section 3.2.3.

3.2.1 Principe

Le raisonnement distribué en logique classique consiste à supposer que chaque nœud de connaissances décrits une partie d'un même monde et que, par conséquent, selon l'hypothèse du monde ouvert, chaque ontologie vient compléter la description obtenue par mise en commun de l'ensemble. Bien entendu, malgré cette vision, il reste indispensable d'exprimer les

$$\begin{tikzcd}
O_1 \arrow[d] & A \arrow[l] \arrow[r] \arrow[d, rightsquigarrow] & O_2 \arrow[d] \\
\mathbf{Mod}_1(O_1) & \mathrm{Mod}(A) \arrow[l] \arrow[r] \arrow[d, hook] & \mathbf{Mod}_2(O_2) \\
& \mathbf{Mod}_1(O_1) \times \mathbf{Mod}_2(O_2) \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

FIGURE 3.2 – L'ensemble des modèles de l'alignement est une relation entre les modèles des deux ontologies, c'est-à-dire que $\mathrm{Mod}(A) \subseteq \mathbf{Mod}_1(O_1) \times \mathbf{Mod}_2(O_2)$.

relations existantes entre ontologies, puisque le vocabulaire n'est pas nécessairement le même. Néanmoins, selon cette perspective, un alignement d'ontologies ne fait que compléter à son tour la description de l'ensemble.

De cette manière, la sémantique du réseau et les déductions réalisables sont tout simplement équivalentes à une unique ontologie dans laquelle seraient mises en commun toutes les connaissances, à la fois des ontologies et des alignements. Une telle approche n'empêche pas le raisonnement distribué, ni la médiation. En effet, en admettant que le raisonnement soit monotone, c'est-à-dire qu'aucune nouvelle connaissance ne peut invalider ce que l'on pouvait prouver sans elle, alors des systèmes de raisonnement différents et répartis peuvent apporter chacun leur part de déduction, qu'ils pourront alors échanger éventuellement avec un système associé au médiateur. Une telle solution ne pose, en fin de compte, que des problèmes algorithmiques et technique de parallélisation et répartition du système d'inférence.

En fait, le langage d'ontologies pour le Web sémantique a été pensé dès l'origine dans une optique de répartition des ontologies, en reliant des ontologies distantes par le biais de la primitive d'importation `import`. En outre, il est possible, au sein d'une ontologie, de se référer à n'importe quel concept ou propriété définis dans une autre ontologie, en utilisant son identifiant unique, l'URI. Aussi, comme on l'a vu dans le chapitre 2, le langage OWL lui-même peut être utilisé pour définir des correspondances entre ontologies. La combinaison de ces différentes fonctionnalités permet de construire des réseaux d'ontologies alignées entièrement à l'aide du langage OWL.

La sémantique d'un tel réseau, selon [58] correspond à la sémantique d'une unique ontologie obtenue par l'union de tous les axiomes de la clôture transitive des ontologies importées. Par conséquent, tout ce qui est affirmé en n'importe quel nœud du réseau d'ontologies est aussi vrai en tout autre nœud.

La figure 3.3 montre que les modèles du réseau d'ontologies selon cette approche doivent être à la fois des modèles de O_1 et de O_2. D'autre part, pour que le réseau d'ontologies soit satisfait, il faut aussi que ces modèles satisfassent les alignements. Puisque cette approche considère aussi les alignements comme des axiomes d'une même théorie, ils restreignent encore les modèles valides du réseau d'ontologies. C'est pourquoi un modèle de $\langle O_1, O_2, A\rangle$ doit se trouver à l'intersection des modèles de O_1, de O_2 et de A.

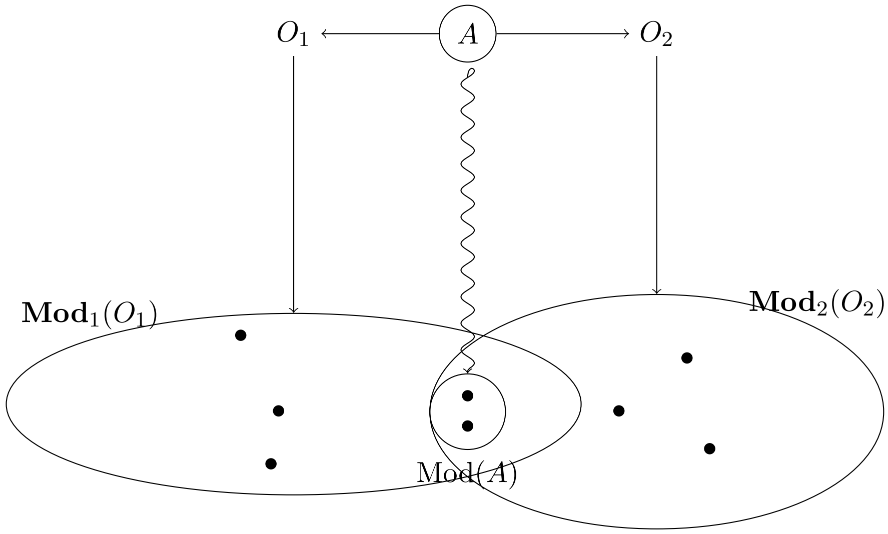


FIGURE 3.3 – Sémantique des ontologies alignées en logique classique. Les modèles du réseau d’ontologies se trouve dans l’intersection $\mathrm{Mod}_1(O_1) \cap \mathrm{Mod}_2(O_2) \cap \mathrm{Mod}(A)$.

3.2.2 Applications

Des systèmes pair-à-pair ou de médiation à base de connaissances admettent explicitement ou implicitement une sémantique en logique classique. Notamment on trouve les systèmes Edutella [56], SomeWhere [3, 4] ou son extension SomeRDFS [5].

Edutella

Edutella est une infrastructure pour les réseaux pair-à-pair, permettant de rechercher des méta données du Web sémantique, mais ne permettant pas de récupérer les données elles-mêmes sur son réseau. Ce système utilise des requêtes dans un format exprimé en RDF, transmises à travers le réseau et retranscrites par chaque pair en leur format interne de gestion des données. Mais ce système ne dispose pas d’alignements d’ontologies car les requêtes doivent être conformes au vocabulaire de l’ontologie cible. En quelque sorte, on peut dire qu’un pair Edutella ne s’intéresse pas aux connaissances issues d’une autre conceptualisation que la sienne et donc l’aspect distribué ne concerne ici que la procédure et non la sémantique.

SomeWhere - SomeRDFS

Des systèmes de raisonnement distribué en logique classique ont été proposés. Par exemple, SomeWhere [3] est un système pair-à-pair, exploitant des connaissances sous forme clausale afin de retrouver des fichiers échangés. Les correspondances établies entre pairs sont des clauses faisant apparaitre des termes de deux ontologies différentes. La sémantique de l’ensemble du réseau est équivalente à celle de l’union de toutes les théories locales et des correspondances.

Le raisonnement se fait par échange de messages entre pairs. L’objectif du système de raisonnement est d’obtenir de nouvelles conséquences sémantiques par des inférences sur l’ensemble du réseau. La méthode est dite “anytime” (“n’importe quand”) car elle permet d’obtenir des inférences partielles mais correctes en l’arrêtant à tout moment. Plus précisément, le système fournit des instances répondant à une requête en construisant l’ensemble de réponses au fur et mesure de son exécution. À tout moment (*at any time*), on peut arrêter le programme et consulter l'ensemble de réponses partiellement construits avec l'assurance qu'il s'agit de réponses correctes à la requête. Lorsqu'il est exécuté suffisamment longtemps, l'algorithme est complet, mais la propriété "anytime" permet le plus souvent d'obtenir une réponse satisfaisante avant même la fin de l'exécution.

En outre, SomeWhere possède des fonctionnalités de gestion des incohérences. Le parti pris de SomeWhere vis-à-vis de l'incohérence est que les ontologies locales ne sont pas elles-mêmes en cause. Si une incohérence existe dans l'ensemble du réseau d'ontologies, elle est la conséquence d'une mauvaise mise en correspondance. Ainsi, un nouveau pair entrant dans le réseau ne peut pas être rejeté à cause de sa propre représentation des connaissances. En revanche, lorsqu'une correspondance est ajoutée, la cohérence globale est vérifiée, ce qui permet de le notifier à l'utilisateur. De cette manière, le système SomeWhere tolère, dans une certaine mesure, une hétérogénéité des ontologies locales.

Deux extensions de SomeWhere permettent de raisonner, selon le même principe, sur des connaissances plus expressives. SomeRDFS gère des connaissances décrites en un fragment de RDFS appelé CoreRDFS [5] tandis que SomeOWL gère une portion de OWL appelée OWL-PL [4].

Extension conservative

La notion d'extension conservative peut être utilisée pour favoriser le raisonnement distribué. Celle-ci peut se définir ainsi. Étant données deux théories (ou ontologies) T_1 et T_2, on dit que $T_1 \cup T_2$ est une extension conservative de T_1 si, pour toute formule α dans le vocabulaire de T_1 on a $T_1 \cup T_2 \models \alpha \Rightarrow T_1 \models \alpha$. Informellement, cela signifie que T_2 n'apporte aucune nouvelle connaissance sur les termes de T_1. Autrement dit, la nouvelle théorie ne permet pas de prouver plus de théorèmes que l'ancienne théorie. Ainsi, pour toute déduction dans $T_1 \cup T_2$ relative au vocabulaire de T_1, il suffit d'utiliser un raisonneur rattaché à T_1 seul.

L'extension conservative est une notion de la logique mathématique qui a été récemment mise en avant dans le cadre de la modularité des ontologies. Or, une ontologie modulaire peut être perçue comme un réseau d'otologies. Dans [23], un module d'ontologie (c'est-à-dire un nœud du réseau) peut indiquer explicitement qu'il utilise des termes d'un autre module, ce qui correspond en quelque sorte à notre notion d'alignement. La sémantique de l'ontologie modulaire obtenue est une logique de description classique, mais pour traiter le problème de disparité et de répartition des modules, une propriété est exigée pour que l'ontologie soit qualifiée de modulaire. Cette propriété indique que toute union de module $M \cup M'$ doit être une extension conservative de M' lorsque M se réfère à M'.

Dans [25], les auteurs indiquent que l'utilisation de service de raisonnement comme la vérification de la conservativité pourrait être envisagée comme alternative à l'utilisation d'une sémantique non standard (comme DDL, $\mathcal{E}$-connection, etc. que nous aborderons par la suite).

Cependant, pour le type de réseaux d'ontologies envisagés dans ce mémoire, il est clair que cette notion restrictive de modularité n'est pas particulièrement désirable. D'ailleurs, [1] indique qu'une extension *non* conservative peut être justement utile pour compléter ou préciser les connaissances d'une ontologie en exploitant les connaissances des autres ontologies d'un réseau.

3.2.3 Critiques

Dans un réseau d'ontologies où la connaissance est décrite par de multiples pairs selon un unique contexte d'utilisation (comme c'est le cas de l'application développée pour SomeWhere), la médiation est favorisée par l'utilisation d'un unique formalisme dont l'interprétation est commune à tous les nœuds du réseau. En outre, on peut qualifier cette approche d'intuitive, puisque tout ce qui est affirmé par un pair est vrai pour tout le réseau. On définit donc facilement les conséquences sémantiques d'un tel réseau.

En revanche, si l'on s'intéresse à la sémantique des réseaux fortement hétérogènes présentés au chapitre 1, cette approche souffre de plusieurs inconvénients.

Premièrement, si les logiques locales sont différentes, établir une sémantique en logique classique ne peut se faire que sur l'union des logiques utilisées. Or, il est notoire que l'union de logiques décidables peut être indécidable. Deuxièmement, elle suppose que toute connaissance dans le réseau décrit une partie d'une seule et unique modélisation du monde. Or, ceci est discutable car on peut modéliser un domaine de connaissance de différentes manières, selon le contexte ou le point de vue.

Le premier point a été étudié dans le cadre des logiques modales [47] puis des logiques de description [9] en utilisant la notion de *fusion de logiques* au lieu de l'*union de logiques*. La fusion des logiques est une logique moins expressive que l'union qui consiste à ne pas mélanger les constructeurs des deux langages au sein d'une même formule. Ceci permet parfois de pouvoir réutiliser les procédures de raisonnement des logiques données pour raisonner sur la fusion. Nous verrons au chapitre 8 que la sémantique proposée respecte dans une certaine mesure ce principe.

Le second point peut être en partie résolu par l'utilisation du raisonnement contextuel introduit dans [43]. Le principe du raisonnement contextuel est d'ajouter aux règles d'inférence locales des règles supplémentaire utilisant à la fois une formule locale et une formule d'un autre langage d'ontologies. En définissant adéquatement ces règles, on peut contrôler les transferts de connaissances d'une ontologie à l'autre. Ce principe peut être traduit en théorie des modèles par le fait qu'on affecte à chaque ontologie une interprétation distincte, auxquelles s'ajoutent des relations particulières au formalisme considéré.

La section suivante présente des logiques qualifiées de distribuées car elles favorisent la séparation des raisonnement en interprétant chaque ontologie différemment, conformément au principe de raisonnement contextuel.

3.3 Les logiques distribuées

Toutes les sémantiques décrites dans cette section ont en commun l'attribution d'interprétations distinctes à chaque ontologie du réseau, conformément au paradigme du raisonnement contextuel décrit dans [43]. Ce point est aussi commun à l'approche que je propose dans ce mémoire, détaillée dans le chapitre 4. En revanche, elles diffèrent par la manière dont les interprétations sont reliées entre elles, en tenant compte des relations sémantiques exprimées dans les alignements.

Cette section se décompose ainsi. En section 3.3.1, on abordera rapidement le cas d'une complète séparation des interprétations. La section 3.3.2 décrit la logique du premier ordre distribuée, mise en pratique directe des principes développés dans [43]. La section 3.3.3 présente les logiques de description distribuées, restriction de DFOL aux logiques de description. La section 3.4 présente le formalisme des $\mathcal{E}$-connections, dont on peut montrer qu'il couvre en grande partie DDL. Enfin, on présente en section 3.5 les logiques de description par paquetages, mieux adaptées à la modularité des ontologies.

3.3.1 Interprétation séparée

On a indiqué précédemment que le fait d'interpréter l'ensemble des connaissances d'un réseau d'ontologies comme une unique théorie fusionnée peut mener à des incohérences non désirables. Or, si les connaissances d'un nœud peuvent influencer le raisonnement sur l'ensemble du réseau, on ne peut garantir que le réseau sera cohérent quand bien même tous les nœuds seraient cohérents. Une solution naïve mais efficace consiste à ne pas faire interagir les connaissances des différents nœuds.

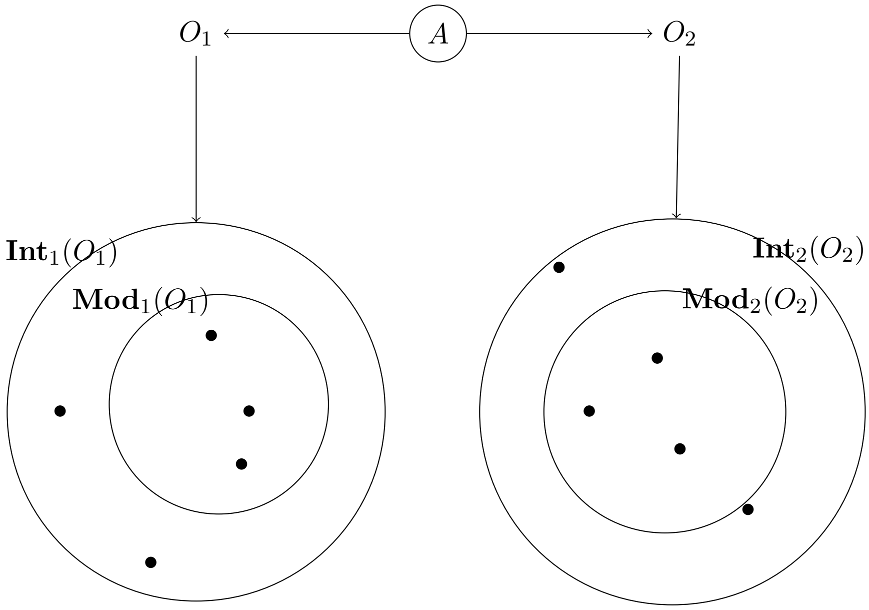


FIGURE 3.4 – Interprétations séparées de O_1 et O_2. Les modèles dans $\mathrm{Mod}_1(O_1)$ et $\mathrm{Mod}_2(O_2)$ ne sont pas reliés.

Critique

Il est clair que cette approche est fortement limitée puisqu'elle ne permet pas, en fin de compte, de combiner des connaissances multiples pour établir de nouvelles conclusions. On peut néanmoins imaginer un système récoltant des résultats d'inférences de systèmes distincts et les exploitant indépendamment de leur sémantique.

Finalement, avec cette approche, le médiateur est dépourvu de sémantique et les ontologies sont purement et simplement isolées. Il n'y a donc aucun problème de compatibilité des points de vue puisqu'ils ne sont pas mis en commun, mais les divers systèmes locaux ne peuvent bénéficier d'une connaissance externe. Il est donc raisonnable d'explorer des possibilités se situant entre cette voie et la sémantique monolithique de la section précédente.

3.3.2 Logique du premier ordre distribuée

La logique du premier ordre distribuée (*Distributed First Order Logics* ou DFOL [38]) offre un cadre très général pour le principe de raisonnement contextuel avancé par [43]. Elle permet de formaliser la sémantique de réseaux d'ontologies juxtaposant plusieurs logiques différentes en les reliant par des formules particulières appelées *passerelles* (*bridge rules* en anglais).

Tandis que chaque ontologie est interprétée séparément, les passerelles fixent des contraintes sur plusieurs interprétations à la fois.

Syntaxe

Dans ce formalisme, les ontologies sont décrites dans des logiques pouvant être exprimées en logique du premier ordre. Quant aux relations sémantiques entre ontologies, elles expriment une relation entre ontologies selon un contexte particulier, c'est-à-dire du point de vue d'une ontologie particulière. En effet, elles peuvent contenir des variables pouvant faire référence à des éléments d'un autre domaine d'interprétation.

En plus des symboles de logique du premier ordre habituels (fonctions, prédicats, constantes, variables et constructeurs booléens) il est possible d'introduire des variables spéciales dites *variables fléchées* dans une ontologie O_i du réseau. Elles s'écrivent $x^{\rightarrow j}$ ou $x^{j\rightarrow}$ (pour $j \neq i$) et servent à faire référence à des variables de l'ontologie O_j au sein de l'ontologie O_i.

À l'aide de ces variables et des autres symboles, DFOL autorise la définition de *contraintes d'interprétation* qui expriment des relations entre formules de différentes ontologies.

Definition 3.3.1 (Contrainte d'interprétation) *Une contrainte d'interprétation de $i_1, \ldots, i_n$ vers i où $i_k \neq i$ pour $1 \leq k \leq n$ est une expression de la forme*

$$i_1 : \phi_1, \ldots, i_n : \phi_n \longrightarrow i : \phi$$

On peut considérer qu'une contrainte d'interprétation désigne une correspondance entre différentes ontologies.

Sémantique

Dans un réseau d'ontologies en DFOL, chaque ontologie est interprétée selon la sémantique de sa propre logique, devant être un fragment de la logique du premier ordre. Puisque les domaines d'interprétation de différentes ontologies peuvent être hétérogènes, une interprétation distribuée décrit aussi comment deux domaines différents sont interconnectés. Cela se traduit dans l'interprétation du réseau d'ontologies par des relations ensemblistes entre domaines d'interprétation, comme on peut le voir dans la définition suivante.

Definition 3.3.2 (Relation de domaine) *Soient Δ_i et Δ_j deux domaines d'interprétation. Une* relation de domaine *r_{ij} de i vers j est un sous-ensemble de $\Delta_i \times \Delta_j$.*

Une relation de domaine r_{ij} décrit comment l'ontologie O_j "perçoit" les éléments du domaine de l'ontologie O_i. Par exemple, O_i peut décrire un certain livre en tant qu'œuvre littéraire d'un auteur, tandis que O_j décrit des livres comme objets physiques individuels. La relation r_{ij} pourrait alors associer l'œuvre littéraire à l'ensemble d'objets physiques qui la contiennent. En effet, selon O_j, l'œuvre peut être considérée comme une représentation de l'ensemble de ces objets. En revanche, il est possible que la relation r_{ji} ne soit pas l'inverse de r_{ij}, par exemple si O_j distingue spécifiquement l'œuvre intellectuelle des objets qui la publient, c'est-à-dire qu'elle distingue les instances d'œuvres littéraires en tant que création intellectuelle, et les instances de livres physiques. Ces relations évacuent en partie les problèmes d'hétérogénéité dans la conception des ontologies et traduisent bien la notion de contexte dans la sémantique.

Pour définir la satisfaction globale du réseau d'ontologies, il faut d'une part donner un sens aux variables fléchées, puis définir la sémantique des contraintes d'interprétation. Pour cela, on commence par étendre la notion d'affectation.

Definition 3.3.3 (Affection) *Soit $\mathcal{M} = \langle(\mathcal{M}_i), (r_{ij})\rangle$ une interprétation distribuée en DFOL. Une affection a est une famille (a_i) de fonctions partielles de l'ensemble des variables et variables fléchées vers les domaines $\Delta^{\mathcal{M}_i}$ telles que*

1. $a_i(x) \in \Delta^{\mathcal{M}_i}$,
2. $a_i(x^{j\rightarrow}) \in r_{ji}(a_j(x))$,
3. $a_j(x) \in r_{ij}(a_i(x^{\rightarrow j}))$.

Une affectation a est *admissible* pour une formule $i : \phi$ si a_i assigne toutes les variables fléchées apparaissant dans ϕ. En outre, a est admissible pour un ensemble de formules Γ si elle est admissible pour toutes les formules de Γ. Une affectation a est strictement admissible pour un ensemble de formules Γ si elle est admissible pour Γ et n'assigne que les variables fléchées qui apparaissent dans Γ.

En utilisant la notion d'affectation admissible ci-dessus, la relation de satisfaction d'une formule locale peut être définie en logique du premier ordre distribuée.

Definition 3.3.4 (Satisfaction) *Soient $\mathcal{M} = \langle(\mathcal{M}_i), (r_{ij})\rangle$ une interprétation distribuée, $m \in \mathcal{M}_i$ un modèle local et a un affectation. Une i-formule ϕ est satisfaite par m, vis-à-vis de a, symboliquement $m \models_D \phi[a]$ si*

1. *a est admissible pour $i : \phi$ et*
2. *$m \models \phi[a_i]$, où $\phi[a_i]$ correspond à la formule ϕ dans laquelle on a remplacé les variables par leur affectation.*

$\mathcal{M} \models \Gamma[a]$ si pour tout $i : \phi \in \Gamma$ et $m \in \mathcal{M}_i, m \models_D \phi[ai]$.

Maintenant, il faut définir quand une interprétation satisfait les contraintes d'interprétation.

Definition 3.3.5 (Satisfiabilité des contraintes d'interprétation) *Une modèle $\mathcal{M}$ satisfait une contrainte d'interprétation $i_1 : \phi_1, \ldots, i_n : \phi_n \longrightarrow i : \phi$ si pour tout affectation a strictement admissible pour $\{i_1 : \phi_1, \ldots, i_n : \phi_n\}$, si $\mathcal{M} \models i_k : \phi_k[a]$ pour $1 \leq k \leq n$, alors a peut être étendu à un affectation a' admissible pour $i : \phi$ et tel que $\mathcal{M} \models i : \phi[a']$.*

On peut remarquer que, selon qu'une variable fléchée $x^{\rightarrow}$ apparait à gauche ou droite d'une contrainte, $x^{\rightarrow}$ est quantifiée universellement ou existentiellement. Aussi, grâce aux contraintes d'interprétation sur l'égalité, on peut formaliser des contraintes sur les relations de domaine, par exemple en les forçant à être injectives, surjectives, partielles, etc.

Algorithme de déduction

Le formalisme de la logique du premier ordre distribuée n'est pas décidable, mais il est possible de définir l'ensemble des règles d'inférences nécessaires à un raisonnement correct et complet [38]. Il n'y a pas d'implémentation supportant ce formalisme.

Critique

L'avantage de ce formalisme est son fort pouvoir d'expressivité et sa généricité. En effet, il permet de définir des relations entre multiples ontologies grâce aux contraintes d'interprétation. Le formalisme DFOL s'affranchit du problème d'hétérogénéité en distinguant les interprétations des diverses théories locales. Les contraintes d'interprétation déterminent les contraintes de compatibilité des connaissances inter-ontologies et sont elles-mêmes contextuelles puisqu'elles s'ont dirigées vers une ontologie particulière, donc exprimées selon un point de vue.

Néanmoins, le fait que les contraintes d'interprétation soient exprimées du point de vue d'une ontologie particulière posent certains problèmes. Premièrement, il n'est pas possible de composer les relations entre ontologies, comme on le voit dans la figure 3.6. Le problème de la composition est en fait lié au fait que les passerelles sont directionnelles. Elles indiquent, en principe, comment une ontologie (l'ontologie cible, identifiée à droite des passerelles) perçoit les connaissances d'autres ontologies. Par conséquent, une passerelle est en quelque sorte rattachée à une ontologie comme s'il s'agissait d'un axiome propre à l'ontologie. D'ailleurs, la partie droite de la passerelle doit être construite dans le langage de l'ontologie cible. Par conséquent, on perd l'indépendance du langage d'alignements vis-à-vis du langage d'ontologies. Si, dans certains cas, on peut supposer que les alignements expriment une connaissance d'un point de vue local, on ne peut l'affirmer en général dans le cadre qu'on souhaite modéliser, où les alignements énoncent des connaissances inter-ontologiques, sans plus de précision.

Toutefois, on peut constater que les auteurs évoquent la notion de médiation vis-à-vis de ce formalisme, notamment dans [37] et [38]. En effet, il est indiqué qu'en restreignant les relations à un appariement entre ontologies locales et ontologie d'un médiateur, on peut représenter la médiation à l'aide de DFOL, comme on peut le voir dans la figure 3.5

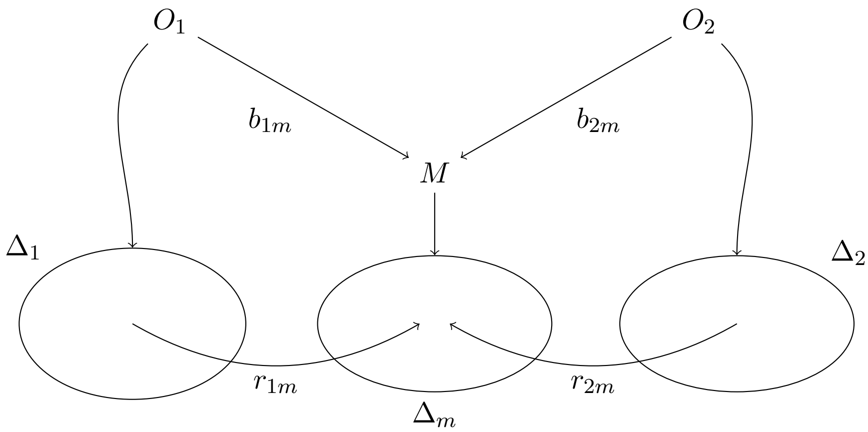


FIGURE 3.5 – Médiation à l'aide de DFOL. Les contraintes d'interprétations (ou les passerelles) se dirigent toutes vers le médiateur M. Ainsi, le médiateur permet de mettre en relation les connaissances des deux ontologies sans obliger ces systèmes locaux à communiquer directement entre eux.

On verra par la suite qu'il s'agit en fait, à peu de chose prêt, du principe qui sera retenu dans la solution proposée ici pour la sémantique d'un réseau d'ontologies alignées.

3.3.3 Logiques de description distribuées

Tout en gardant le principe du raisonnement contextuel, il est intéressant de réduire les logiques locales à des logiques de description, puisqu'il s'agit du formalisme de prédilection du Web sémantique. C'est ce qu'offrent les logiques de description distribuées.

En utilisant le même principe que DFOL, les logiques de description distribuées (DDL) permettent de relier et de raisonner avec de multiples ontologies sur le Web sémantique. Les passerelles en logiques de description distribuées se restreignent aux relations entre concepts, rôles et individus d'ontologies différentes (voir la syntaxe ci-après). Leur sémantique permet avant tout de déduire des relations de subsomption qu'une ontologie seule ne permet pas d'obtenir (voir la sémantique ci-après). Afin de représenter des réseaux d'ontologies avec DDL, une syntaxe concrète fondée sur OWL a été définie et un système de raisonnement distribué offre un procédure de raisonnement pour un fragment de ce langage (voir l'implémentation en section 3.3.4).

Syntaxe

Un réseau d'ontologies en DDL est composé de diverses bases de connaissances en logique de description, dont la syntaxe a été présentée au chapitre 2. Par ailleurs, les ontologies sont reliées entre elles par le biais des passerelles. Celles-ci sont représentées, dans la syntaxe abstraite comme suit.

Definition 3.3.6 (Passerelles) *Soient O_i et O_j deux ontologies. Une* passerelle *de O_i vers O_j ($i \neq j$), est une expression de l'une des formes suivantes :*
- $i\!:\!X \xrightarrow{\sqsubseteq} j\!:\!Y$ *est une règle "intra" (into-bridge rule) ;*
- $i\!:\!X \xrightarrow{\sqsupseteq} j\!:\!Y$ *est une règle "extra" (onto-bridge rule) ;*
- $i\!:\!a \mapsto j\!:\!b$ *est une correspondance d'individus (individual correspondence).*

où $i\!:\!X$ et $j\!:\!Y$ sont soit des concepts soit des rôles de O_i et O_j respectivement et $i\!:\!a$ est un individu de O_i et $j\!:\!b$ est un individu de O_j.

À l'origine, les passerelles étaient conçues pour associer des concepts entre eux, ou bien des rôles entre eux, mais les derniers travaux sur DDL [40, 39, 41] permettent dorénavant d'associer un rôle et un concept et inversement. Dans ce cas, on parle de passerelles hétérogènes. Pour les distinguer des règles dites homogènes, on les notera $i : R \overset{\sqsubseteq}{\twoheadrightarrow} j : C$ ou $i : C \overset{\sqsubseteq}{\twoheadrightarrow} j : R$. Celles-ci indiquent en fait que les éléments d'un concept correspondent à une réification d'une relation. Pour faire simple, on peut dire que si $i\!:\!C \overset{\sqsubseteq}{\twoheadrightarrow} j\!:\!R$, cela signifie que tous les éléments $d_i \in C^{I_i}$ sont associés à des couples $(d_j, d'_j) \in R^{I_j}$. Ces notions seront formalisées dans la section suivante.

Sémantique

Dans un réseau d'ontologies en DDL, on affecte à chaque ontologie une interprétation en logique de description. Il peut exister différentes logiques de description à chaque nœud du réseau. Pour relier les connaissances de deux ontologies, DDL utilise les relations de domaines.

Definition 3.3.7 (Relation de domaines (homogène)) *Soient Δ_i et Δ_j deux domaines d'interprétation. Une relation de domaine homogène r_{ij} de i vers j est un sous-ensemble de $\Delta_i \times \Delta_j$. Pour tout $d \in \Delta^{I_i}$, on utilise $r_{ij}(d)$ pour dénoter l'ensemble $\{d' \in \Delta^{I_j} | \langle d, d' \rangle \in r_{ij}\}$, pour tout $D \subseteq \Delta^{I_i}$, on utilise $r_{ij}(D)$ pour dénoter l'ensemble $\bigcup_{d \in D} r_{ij}(d)$ et pour tout $R \subseteq \Delta^{I_i} \times \Delta^{I_j}$ $r_{ij}(R)$ dénote $\bigcup_{\langle d,e \rangle \in R} r_{ij}(d) \times r_{ij}(e)$.*

Une relation de domaines r_{ij} représente une manière possible d'apparier les éléments de Δ_i avec des éléments de Δ_j, selon le point de vue de j. Les relations de domaines servent à interpréter les passerelles homogènes.

Definition 3.3.8 *Une relation de domaine r_{ij} satisfait une passerelle homogène vis-à-vis de deux interprétations locales I_i et I_j (noté $\langle I_i, r_{ij}, I_j \rangle \models rp$) si et seulement si :*
- $\langle I_i, r_{ij}, I_j \rangle \models i\!:\!X \xrightarrow{\sqsubseteq} j\!:\!Y$ *si et seulement si* $r_{ij}(X^{I_i}) \subseteq Y^{I_j}$,

– $\langle I_i, r_{ij}, I_j\rangle \models i:X \xrightarrow{\sqsupseteq} j:Y$ *si et seulement si* $r_{ij}(X^{I_i}) \supseteq Y^{I_j}$,
– $\langle I_i, r_{ij}, I_j\rangle \models i:a \mapsto j:b$ *alors* $\langle a^{I_i}, b^{I_j}\rangle \in r_{ij}$.

Les relations de domaine ne suffisent pas à traiter correctement des relations hétérogènes. Comme indiqué dans la section précédente, une relation hétérogène indique une association entre un élément d'un concept et la réification d'une relation, donc elle associe un objet du domaine à une paire d'objets. Par exemple, on peut vouloir relier un mariage m au couple de personnes mariées $(\text{Marie}, \text{Pierre})$. C'est selon ce principe qu'a été définie d'abord la sémantique des règles hétérogènes dans [39]. Mais ceci a été révisé par la suite afin d'affecter à un objet d'un concept, non pas un couple mais un triplet $(\text{Marie}, \text{mariéÀ}, \text{Pierre})$. Ceci correspond mieux à l'idée de réification. Ainsi, pour une interprétation I_i, on notera Σ_i l'ensemble des triplets $\langle x, R, y\rangle$ tels que $x, y \in \Delta_i$ et R est un rôle de O_i.

Definition 3.3.9 (Relations de domaine hétérogènes) *Soient I_i et I_j deux interprétations. Une relation de domaine concept-rôle cr_{ij} de i vers j est un sous-ensemble de $\Delta_i \times \Sigma_j$. Une relation de domaine rôle-concept rc_{ij} de i vers j est un sous-ensemble de $\Sigma_i \times \Delta_i$.*

La relation rc_{ij} représente une manière possible de réifier des relations entre objets, tandis que la relation cr_{ij} représente le processus inverse.

Definition 3.3.10 *Une relation de domaine concept-rôle cr_{ij} satisfait une passerelle hétérogène vis-à-vis de deux interprétations locales I_i et I_j lorsque :*

– $\langle I_i, cr_{ij}, I_j\rangle \models i : C \overset{\sqsubseteq}{\twoheadrightarrow} j : R$ *si et seulement si pour tout* $x \in C^{I_i}$ *et toute paire* $(x, \langle x_1, X, x_2\rangle) \in cr_{ij}$, $X^{I_j} \subseteq R^{I_j}$,
– $\langle I_i, cr_{ij}, I_j\rangle \models i : C \overset{\sqsupseteq}{\twoheadrightarrow} j : R$ *si et seulement si pour tout* $(x_1, x_2) \in R^{I_j}$, *il existe une paire* $(x, \langle x_1, X, x_2\rangle) \in cr_{ij}$ *telle que* $X^{I_j} \supseteq R^{I_j}$ *et* $x \in C^{I_i}$.

Une relation de domaine rôle-concept rc_{ij} satisfait une passerelle hétérogène vis-à-vis de deux interprétations locales I_i et I_j lorsque :

– $\langle I_i, rc_{ij}, I_j\rangle \models i : R \overset{\sqsubseteq}{\twoheadrightarrow} j : C$ *si et seulement si pour tout* $(x_1, x_2) \in R^{I_i}$ *et pour toute paire* $(\langle x_1, X, x_2\rangle, x) \in rc_{ij}$ *telle que* $X^{I_i} \supseteq R^{I_i}$, $x \in C^{I_j}$,
– $\langle I_i, rc_{ij}, I_j\rangle \models i : R \overset{\sqsupseteq}{\twoheadrightarrow} j : C$ *si et seulement si pour tout* $x \in C^{I_j}$, *il existe une paire* $(\langle x_1, X, x_2\rangle, x) \in rc_{ij}$ *telle que* $X^{I_i} \subseteq R^{I_i}$.

Ainsi, lorsqu'une règle rôle-concept est satisfaite, les couples d'éléments de la relation R^{I_i} sont associés à des éléments de C^{I_j}. Une interprétation distribuée se construit à partir de ces différentes relations ainsi que des interprétation locales affectées à chaque ontologie.

Definition 3.3.11 (Interprétation distribuée) *Soit $S = \langle \mathbf{O}, \mathbf{B}\rangle$ un réseau d'ontologies et de passerelles. Une* interprétation distribuée $\mathcal{I} = \langle \mathbf{I}, \mathbf{r}, \mathbf{cr}, \mathbf{rc}\rangle$ *de S affecte à chaque ontologie O_i une interprétation $I_i = \langle \Delta^{I_i}, \cdot^{I_i}\rangle$ de $\Sigma(O_i)$ et à chaque paire $i \neq j$, une relation de domaine $r_{ij} \subseteq \Delta^{I_i} \times \Delta^{I_j}$, une relation concept-rôle $cr_{ij} \subseteq \Delta^{I_i} \times \Sigma^{I_j}$ et une relation rôle-concept $rc_{ij} \subseteq \Sigma^{I_i} \times \Delta^{I_j}$.*

Une interprétation distribuée satisfait un réseau d'ontologies S quand (1) ses interprétations locales satisfont (selon la sémantique locale) leur ontologie respective et (2) les contraintes imposées par les passerelles sont satisfaites.

Definition 3.3.12 (Modèle d'un réseau d'ontologies) *Un* modèle *d'un système $\langle \mathbf{O}, \mathbf{B}\rangle$ est une interprétation distribuée $\mathcal{I} = \langle \mathbf{I}, \mathbf{r}, \mathbf{cr}, \mathbf{rc}\rangle$ telle que pour tout $i \in K$, $I_i \models O_i$ et pour tout*

$i, j \in K$, $\langle I_i, r_{ij}, I_j\rangle$ satisfait toutes les passerelles homogènes de B_{ij}, $\langle I_i, cr_{ij}, I_j\rangle$ satisfait toutes les passerelles concept-rôle de B_{ij} et $\langle I_i, rc_{ij}, I_j\rangle$ satisfait toutes les passerelles rôle-concept de B_{ij}. On note cela $\mathcal{I} \models_d S$.

Un axiome (respectivement une passerelle) α (respectivement β) est une conséquence sémantique de S si tous les modèles de S satisfont α (respectivement β).

D'après ces définitions, on peut observer qu'il n'est pas nécessaire que $r_{ij} = r_{ji}^{-1}$ et donc, si $r_{ij}(C^{I_i}) \subseteq D^{I_j}$, il ne peut pas être déduit que $r_{ji}(D^{I_j}) \supseteq C^{I_i}$. En effet, les passerelles sont liées à un contexte donné.

3.3.4 Implémentation

Une implémentation partielle de DDL existe. La logique de description locale considérée est $\mathcal{SHOIN}b$ et le langage concret pour l'implémentation est appelé C-OWL, en référence à OWL qu'il étend pour prendre en compte le raisonnement contextuel (d'où la lettre "C"). Le système de raisonnement distribué pour C-OWL s'appelle Drago [64] et fonctionne à la manière d'un système pair-à-pair. Chaque pair intègre un raisonneur local de logique de description utilisant un algorithme de tableaux, auquel s'ajoute des règles d'expansion relatives aux passerelles. Ces règles d'expansion particulières font appel aux raisonneurs des autres pairs mis en correspondances. Il revient donc à chaque pair la tâche de gérer les correspondances dans le raisonnement distribué.

À l'origine, Drago n'implémentait pas les correspondances d'individus, mais une extension permet dorénavant le raisonnement avec les correspondances totales d'individus [65]. Les correspondances hétérogènes ne sont pas encore prises en charge dans l'implémentation.

Une des utilisations intéressantes d'un raisonneur DDL est présentée dans [51]. Le raisonnement est effectué dans le but de vérifier si les passerelles DDL sont cohérentes et donc d'améliorer les résultats d'outils d'alignement automatique. Pour cela, il n'est pas utile d'avoir un système de raisonnement distribué et donc cette vérification n'utilise pas une architecture constituée de multiples pairs.

3.3.5 Critique

Le formalisme DDL possède une grande robustesse face au problème d'hétérogénéité. Les passerelles rendent compte des relations existant entre ontologies et sont elles-mêmes contextuelles. Aussi, DDL bénéficie d'une implémentation sous forme de système pair-à-pair. Chaque pair dispose d'une ontologie et de toutes les passerelles qui le relient aux autres pairs. En outre, tous les pairs implémentent un algorithme de tableaux distribués, mais la gestion de la partie locale du raisonnement est assez indépendante du raisonnement distribué.

Néanmoins, il partage des inconvénients de DFOL vis-à-vis des types de réseaux d'ontologies et d'alignements que l'on souhaite modéliser, comme par exemple l'impossibilité de composer les passerelles, ainsi que leur orientation, toujours exprimée du point de vue d'un des nœud et donc contraire au principe de médiation. La composition permet d'obtenir de bons alignements à moindre coût, lorsque les alignements composés sont eux-mêmes de qualité. La figure 3.6 montre comment se comporte DDL vis-à-vis de la composition.

Enfin, DDL impose l'utilisation des logiques de description au sein des systèmes locaux. Or, on aimerait pouvoir exploiter des connaissances exprimées dans d'autres formalismes, plus ou moins expressifs.

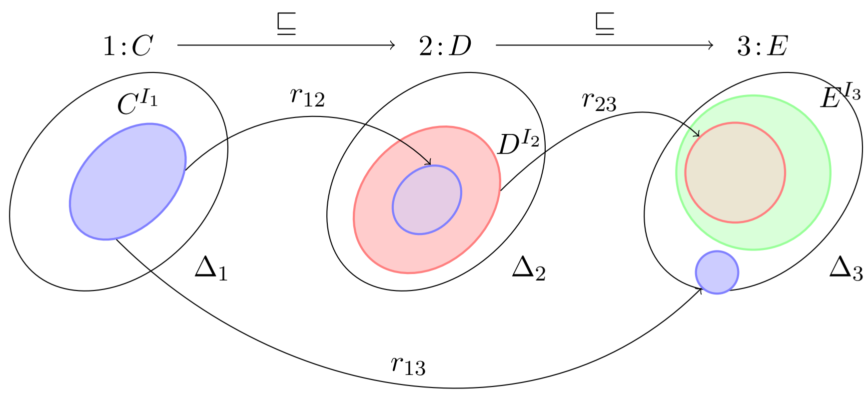


FIGURE 3.6 – Le problème de la composition des passerelles en DDL.

3.4 $\mathcal{E}$-connection

L'idée derrière les $\mathcal{E}$-connections était à l'origine de partitionner des bases de connaissances en des sous-parties exprimables dans des formalismes décidables, mais dont l'union n'est pas forcément décidable. Chaque partie étant connectée par des liens assez peu contraints pour garantir une bonne indépendance des raisonnements locaux. Les $\mathcal{E}$-connections ont rapidement été exploitées pour la modularisation des ontologies. Les relations entre ontologies ne sont pas exprimés d'une façon compatible avec la définition 2.3.3. Les relations entre ontologies sont représentées par des termes particuliers appelés liens qui font partie intégrante du vocabulaire des ontologies.

3.4.1 Syntaxe

Les $\mathcal{E}$-connections utilisent la notion de système de description abstrait (*Abstract Description System* ou ADS) pour représenter les diverses logiques locales. Ce concept couvre un grand nombre de logiques de description et de logiques modales, mais pour simplifier la notion, je me cantonnerai aux logiques de description. Ainsi, la syntaxe locale utilise les constructeurs et le vocabulaire de logiques de description, auxquels sont ajoutés les éléments suivants.

Pour chaque paire d'ontologies O_i et O_j d'un réseau d'ontologies, la syntaxe des $\mathcal{E}$-connections introduit un ensemble E_{ij} de termes appelés *liens*, qui représentent des sortes de rôles reliant des instances de l'ontologie i à des instances de l'ontologie j. Ces liens sont utilisés dans des axiomes de manière similaire aux rôles. Dans l'ontologie O_j, on peut utiliser un lien noté $\langle E\rangle^i$ pour indiquer une relation provenant de l'ontologie O_i et les constructeurs suivants peuvent être appliqués.

– si $\langle E\rangle^i$ est un lien alors $\langle\neg E\rangle^i$ est aussi un lien,
– si $\langle E_1\rangle^i$ et $\langle E_2\rangle^i$ sont des liens alors $\langle E_1 \sqcap E_2\rangle^i$ et $\langle E_1 \sqcup E_2\rangle^i$ sont aussi des liens.

Les liens interviennent dans la définition de concept d'une ontologie, par exemple $\forall\langle E\rangle^i C$, $\exists\langle E\rangle^i C$, $\leq n\langle E\rangle^i C$ ou $\geq \langle E\rangle^i C$. Ces constructions dénotent des concepts de l'ontologie O_j.

3.4.2 Sémantique

De même que pour les autres logiques distribuées, une interprétation locale I_i est affectée à chaque ontologie O_i. La sémantique des concepts, rôles et individus est la même qu'en logique de description pour tous les constructeurs habituels. En revanche, l'interprétation des liens $\langle E \rangle$ diffère légèrement de l'interprétation de rôles habituels. En fait, la seule différence est que les liens maintiennent une séparation des domaines d'interprétation, c'est-à-dire qu'au lieu d'être interprétés comme une relation sur un domaine d'interprétation, ils sont interprétés comme une relation sur un couple de domaines d'interprétation. Ainsi, pour interpréter un lien $\langle Z \rangle^j$ dans une ontologie O_i, on ajoute les règles suivantes.

- $(\langle E \rangle^j)^{I_i} \subseteq \Delta^{I_i} \times \Delta^{I_j}$,
- $(\langle \neg E \rangle^j)^{I_i} = (\Delta^{I_i} \times \Delta^{I_j}) \setminus (\langle E \rangle^j)^{I_i}$,
- $(\langle E_1 \sqcup E_2 \rangle^j)^{I_i} = (\langle E_1 \rangle^j)^{I_i} \cup (\langle E_2 \rangle^j)^{I_j}$,
- $(\langle E_1 \sqcap E_2 \rangle^j)^{I_i} = (\langle E_1 \rangle^j)^{I_i} \cap (\langle E_2 \rangle^j)^{I_j}$,
- $(\forall \langle E \rangle^i C)^{I_i} = \{x \in \Delta^{I_i} \mid \forall y \in \Delta^{I_j}, (x, y) \in (\langle E \rangle^j)^{I_i} \Rightarrow y \in C^{I_j}\}$,
- $(\exists \langle E \rangle^i C)^{I_i} = \{x \in \Delta^{I_i} \mid \exists y \in C^{I_j}, (x, y) \in (\langle E \rangle^j)^{I_i}\}$,
- $(\leq n \langle E \rangle^i C)^{I_i} = \{x \in \Delta^{I_i} \mid \sharp\{y \in C^{I_j} \mid (x, y) \in (\langle E \rangle^j)^{I_i}\} \leq n\}$,
- $(\geq n \langle E \rangle^i C)^{I_i} = \{x \in \Delta^{I_i} \mid \sharp\{y \in C^{I_j} \mid (x, y) \in (\langle E \rangle^j)^{I_i}\} \geq n\}$.

3.4.3 Implémentation

Pellet intègre un raisonneur sur les $\mathcal{E}$-connections [26]. Ce raisonneur autorise l'utilisation d'ontologie en $\mathcal{SHIQ}$, $\mathcal{SHOQ}$ ou $\mathcal{SHIO}$ mais ne couvre pas les liens complexes.

3.4.4 Critique

La principale critique que l'on peut émettre vis-à-vis de la médiation est que ce formalisme définit les relations inter-ontologies en mêlant liens, concepts locaux et concepts externes au sein des définitions de concepts complexes. Il est donc difficile, dans le cas général, de représenter les relations sémantiques sous forme de correspondances distinctes des axiomes locaux. Par ailleurs, les procédures de raisonnement obligent chaque nœud à traiter ses liens et par conséquent ne peuvent être reléguées à un système externe de médiation.

3.5 Package-based Description Logics

Package-based Description Logics ou P-DL [13, 14] ne fait pas appel, à proprement parler, à la notion d'alignement. P-DL est avant tout destiné à concevoir des ontologies modulaires de façon plus adéquate que le système d'importation défini par OWL. Les différentes ontologies du réseau d'ontologies sont reliées entre elles par la notion d'importation de concepts, de rôles ou d'individus. On peut ensuite utiliser les termes importés comme s'il s'agissait de concepts locaux dans des axiomes.

3.5.1 Syntaxe

La syntaxe est basée sur les logiques de description, auxquelles est ajoutée la notion d'import. Une ontologie locale en P-DL est appelé *package* et la signature d'un package P est divisée en termes locaux $\mathrm{Loc}(P)$ et en termes externes $\mathrm{Ext}(P)$. Une des particularités de la syntaxe est qu'elle désigne explicitement la négation de façon locale (autrement appelée *négation contextualisée*) en ajoutant en indice l'identifiant du package auquel elle se rapporte (par exemple, $\neg_i$). Ceci permet de s'affranchir d'un problème récurrent dans les logiques distribuées, à savoir que la négation est relative à un domaine d'interprétation et donc ne peut être "transférée" d'une base de connaissance à une autre. De même, le concept universel $\top$ est contextualisé de la même manière (par exemple $\top_i$).

Pour indiquer qu'un package P_i importe un terme t provenant d'un package P_j, on ajoute $P_j \xrightarrow{t} P_i$ dans le package P_i. Dans ce cas, le terme t doit appartenir à $\mathrm{Loc}(P) \cap \mathrm{Ext}(P)$.

3.5.2 Sémantique

[12] donne une sémantique de P-DL exploitant la notion de relation de domaine présente dans DDL et DFOL, en leur attribuant des contraintes supplémentaires. Ainsi, l'interprétation d'un réseau d'ontologies en P-DL est similaire à celle des logiques de description distribuées, excepté que les termes $\top_k$ et les constructeurs $\neg_k$ doivent pouvoir être interprétés localement.

Ajouté à cela, les relations de domaine en P-DL doivent être à la fois injectives et transitives. La figure 3.7 représente un exemple d'interprétation distribuée en P-DL.

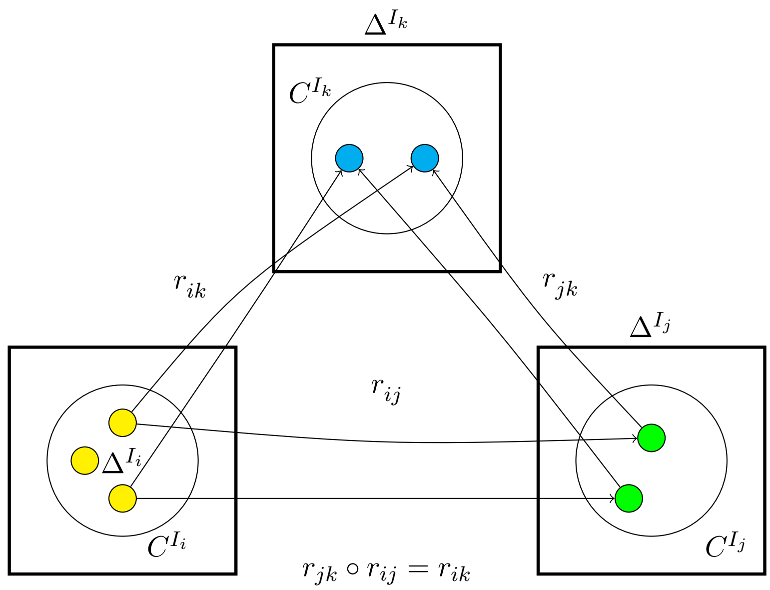


FIGURE 3.7 – Aperçu de la sémantique de P-DL. Les relations de domaine sont injectives et transitives [2].

3.5.3 Critique

P-DL est avant tout un langage pour la modularisation des ontologies sur le Web sémantique. Il n'est pas conçu pour rétablir l'interopérabilité de systèmes hétérogènes, puisque les termes ne peuvent être qu'importés. Il n'y a donc pas d'alignement à proprement parler. L'approche est intéressante surtout pour la modularisation d'ontologies conçues selon un même point de vue, mais nécessitant une séparation des différentes parties. Elles n'est donc pas convenable pour traiter une approche par médiation pour des ontologies hétérogènes.

3.6 Ontology Integration Framework (OIS)

Le formalisme développé dans [20] est plus proche de la problématique de médiation qui nous intéresse. Cette approche généralise les travaux existant sur l'intégration de schéma

1. Figure adapté de [12], avec l'aimable autorisation de Jie Bao.

fondée sur les vues (par exemple dans [73]) et subsume d'autres approches pour interconnecter des modèles de logiques de description par des règles.

OIS suppose l'existence d'un modèle global g auquel les modèles locaux s sont appariés. Au niveau sémantique, les domaines locaux sont supposés inclus dans un domaine global. Par ailleurs, dans OIS les constantes sont supposées désigner les mêmes objets dans tous les domaines.

Selon ces suppositions, les appariements décrivent des correspondances entre modèles locaux et modèle global. Les interprétations de ces correspondances sont définies comme suit.

Definition 3.6.1 ([20]) *Les correspondances entre ontologies sources et ontologie globale sont de l'une des trois formes suivantes :*

1. *$\mathcal{I}$ satisfait $\langle \phi, \psi, sound \rangle$ vis-à-vis de l'interprétation locale $\mathcal{D}$, si tous les tuples satisfaisant ψ dans $\mathcal{D}$ satisfont aussi ϕ dans $\mathcal{I}$*
2. *$\mathcal{I}$ satisfait $\langle \phi, \psi, complete \rangle$ vis-à-vis de l'interprétation locale $\mathcal{D}$, si aucun tuple autre que ceux satisfaisant ψ dans $\mathcal{D}$ ne satisfont ϕ dans $\mathcal{I}$,*
3. *$\mathcal{I}$ satisfait $\langle \phi, \psi, exact \rangle$ vis-à-vis de l'interprétation locale $\mathcal{D}$, si l'ensemble des tuples qui satisfont ψ dans D est exactement l'ensemble des tuples satisfaisant ϕ dans $\mathcal{I}$.*

De ces conditions, on peut remarquer que $\langle \phi, \psi, exact \rangle$ est équivalent à la conjonction de $\langle \phi, \psi, sound \rangle$ et de $\langle \phi, \psi, complete \rangle$.

3.6.1 Critiques

Le problème de cette approche vient du fait que des ontologies locales ne sont pas mises en correspondance entre elles, mais reliées à une ontologie globale. Par conséquent, cette ontologie globale doit contenir une description suffisamment vaste et suffisamment précise pour que toutes les ontologies puissent s'apparier avec l'ontologie globale. Dans un cadre d'application ouvert comme le Web sémantique, ou un réseau pair-à-pair, il n'est pas envisageable de disposer d'une unique représentation couvrant toutes les connaissances potentiellement exprimables dans le réseau d'ontologies.

3.7 Logique de description pour l'intégration d'information

Une approche légèrement différente de l'intégration de différents modèles en logique de description est décrite dans [21]. Cette approche suppose que les domaines des modèles M_i et M_j se superposent plutôt qu'ils soient inclus complètement dans un domaine global. Pour le reste, les mêmes suppositions sont faites que pour OIS décrit dans la section précédente.

Une interprétation $\mathcal{I}$ associe à chaque M_i un domaine Δ_i. Ces différents modèles sont connectés par des assertions inter-schéma. La satisfiabilité des assertions inter-schéma est définie comme suit [2]

Definition 3.7.1 (Satisfiabilité des assertions inter-schéma) *Soit $\mathcal{I}$ est une interprétation pour*

2. Pour simplifier la définition, on introduit la notation $\top_{nij}^{\mathcal{I}} = \top_{ni}^{\mathcal{I}} \cap \top_{nj}^{\mathcal{I}}$ pour tout $n > 1$. À noter que $\top_{nij}^{\mathcal{I}} = \Delta_i^n \cap \Delta_j^n$.

M_i et M_j. On dit que $\mathcal{I}$ satisfait l'assertion inter-schéma si et seulement si :

(3.1) $$\phi \sqsubseteq_{ext} \psi, \textit{ if } \phi^{\mathcal{I}} \subseteq \psi^{\mathcal{I}}$$

(3.2) $$\phi \equiv_{ext} \psi, \textit{ if } \phi^{\mathcal{I}} = \psi^{\mathcal{I}}$$

(3.3) $$\phi \not\sqsubseteq_{ext} \psi, \textit{ if } \phi^{\mathcal{I}} \not\subseteq \psi^{\mathcal{I}}$$

(3.4) $$\phi \not\equiv_{ext} \psi, \textit{ if } \phi^{\mathcal{I}} \neq \psi^{\mathcal{I}}$$

(3.5) $$\phi \sqsubseteq_{int} \psi, \textit{ if } \phi^{\mathcal{I}} \cap \top^{\mathcal{I}}_{nij} \subseteq \psi^{\mathcal{I}} \cap \top^{\mathcal{I}}_{nij}$$

(3.6) $$\phi \equiv_{ext} \psi, \textit{ if } \phi^{\mathcal{I}} = \psi^{\mathcal{I}} \cap \top^{\mathcal{I}}_{nij}$$

(3.7) $$\phi \not\sqsubseteq_{ext} \psi, \textit{ if } \phi^{\mathcal{I}} \not\subseteq \psi^{\mathcal{I}} \cap \top^{\mathcal{I}}_{nij}$$

(3.8) $$\phi \not\equiv_{ext} \psi, \textit{ if } \phi^{\mathcal{I}} \cap \top^{\mathcal{I}}_{nij} \neq \psi^{\mathcal{I}} \cap \top^{\mathcal{I}}_{nij}$$

Comme auparavant, on peut définir $\equiv_{ext}$ et $\equiv_{int}$ comme conjonction de $\sqsubseteq_{ext}$ et $\sqsubseteq_{int}$. Tandis que l'interprétation extentionnelle correspond à la sémantique des appariements OIS et l'interprétation intentionnelle correspond à la sémantique des passerelles en C-OWL. Ainsi, en utilisant la distinction faite par cette approche on obtient une explication des différentes conceptualisations sous jacentes aux sémantiques de C-OWL et de OIS qui utilisent une interprétation extensionnelle et intentionnelle, respectivement.

3.7.1 Critiques

Le premier problème de cette approche est que l'ensemble des sources de connaissances utilisées doit être apparié avec une ontologie globale qui devra être suffisamment riche et détaillée pour intégrer toute la diversité des connaissances que le réseau est amené à utiliser. Néanmoins, cela risque fortement de ne pas suffire pour prendre en compte de nouveaux arrivants dans le réseau, à moins de s'autoriser à redéfinir l'ontologie globale régulièrement. Aussi, puisque les ontologies ne sont pas directement mises en correspondance, mais plutôt reliées à l'ontologie globale, on ne peut pas facilement exprimer une opération de composition d'alignements. Enfin, le fait que les domaines se superposent oblige les ontologies à être relativement homogènes, du moins dans la partie commune du domaine de connaissance. Ce point est critiquable au même titre qu'il l'est pour P-DL.

En revanche, les logiques de description pour l'intégration d'information proposent un formalisme adapté à un système d'intégration où des requêtes sont soumises de façon homogène pour interroger des sources hétérogènes.

3.8 Bilan

Dans ce chapitre, j'ai exposé les différents formalismes pouvant être utilisés pour raisonner sur un réseau d'ontologies et d'alignements. Tous ces formalismes ont leur légitimité dans un certain cadre d'application, mais ne garantissent pas complètement les propriétés dont on souhaiterait disposer pour les types de réseaux d'ontologies alignées que l'on considère. On a vu d'abord qu'un tel système ne devrait pas, de préférence, être interprété comme une seule et même base de connaissance, du fait de l'hétérogénéité des langages et des représentations locales, éventuellement incompatibles. Pour cette raison, d'autres formes de raisonnement, se fondant sur la notion de contexte, ont été proposés. Ces formalismes, tout en résolvant, dans une certaine mesure, la question de l'hétérogénéité et tout en offrant une bonne indépendance des systèmes localisés au nœud du réseau, définissent tous la sémantique des alignements comme relative à une ontologie particulière. Ceci ne favorise pas un traitement des alignements séparé de celui des ontologies. Or le principe de médiation doit servir à déporter la coordination vers des processus externes, sans imposer aux différentes parties de tenir compte de connaissances extérieures. Le chapitre suivant présente un formalisme qui préserve la notion de contexte local, mais favorise le principe de médiation en distinguant mieux la sémantique locale (intra-ontologique) et celle des alignements (inter-ontologique).

Deuxième partie

Contribution

Chapitre 4

Sémantique des réseaux d'ontologies alignées

Résumé

Ce chapitre présente une sémantique générique pour les réseaux d'ontologies alignées utilisant le paradigme des domaines d'interprétation multiples. Le principe essentiel de cette sémantique est qu'elle distingue les différentes interprétations locales propres à chaque ontologie et l'interprétation globale du réseau d'ontologies, mettant en jeu les alignements, donc les connaissances inter-ontologies, qui correspondent en fait au point de vue du médiateur.

Sommaire

4.1 Introduction

Dans le chapitre 2, nous avons vu comment sont représentés les réseaux de connaissances. Par ailleurs, j'ai aussi défini de manière très générale les notions permettant de décrire une sémantique formelle pour un langage de représentation des connaissances, en particulier la notion d'interprétation et celle de modèle. Ici, il s'agit de définir l'interprétation puis la satisfaction d'un réseau d'ontologies et d'alignements. Celles-ci se définissent en fonction de l'interprétation des ontologies qui composent le réseau.

Ce chapitre ne décrit pas à proprement parler une logique distribuée, mais une sémantique générique paramétrée par les logiques locales et le langage d'alignements. Pour obtenir la logique concrète, il conviendra d'instancier ces langages, ce que je ferai dans les exemples. Ainsi, on ne tiendra compte que des définitions abstraites et générales des ontologies et des alignements pour définir cette sémantique.

On se donne donc un ensemble d'ontologies (O_i), chacune composée d'un ensemble de termes $\Sigma(O_i)$ et d'axiomes $A(O_i)$. Ces ontologies sont mises en correspondances deux à deux par des alignements A_{ij}. La combinaison de ces deux ensembles forme le réseau d'ontologies alignées S.

Une sémantique pour des réseaux d'ontologies alignées définit une notion d'interprétation et de satisfaction de façon analogue aux logiques locales, c'est-à-dire qu'il nous faut définir les interprétations d'un réseau d'ontologies alignées et la relation de satisfaction. Or, le réseau S contient des ontologies ayant d'ores et déjà leur propre sémantique. Le présent chapitre montre comment ces sémantiques locales viennent interagir avec une sémantique des alignements, en utilisant le principe des fonctions d'égalisation initialement proposé dans [78].

Comme on l'a vu en introduction du chapitre 2, les alignements imposent des restrictions sur les interprétations locales compatibles dans le réseau. Pour montrer comment ces restrictions agissent sur les interprétations locales, on commence par définir une sémantique des alignements indépendante des ontologies (section 4.2). Cette manière de procéder permet de définir plus facilement des opérations sur les alignements. Les interprétations locales et l'interprétation des alignements sont ensuite reliées entre elles par un outil technique appelé fonction d'égalisation, qui fait partie intégrante d'une interprétation de réseau d'ontologies alignées (section 4.3).

4.2 Sémantiques des alignements

Comme on a pu le voir au chapitre 2, un langage d'alignements est un couple $\langle \mathrm{Ent}, \mathcal{R} \rangle$, où Ent est un langage d'entités et $\mathcal{R}$ un ensemble de symboles de relations. Un alignement est composé de correspondances $\langle e_i, e_j, r \rangle \in \mathrm{Ent} \times \mathrm{Ent} \times \mathcal{R}$. Commençons par donner une interprétation aux entités, puis nous verrons comment les relations sont interprétées.

4.2.1 Interprétation des alignements

La sémantique d'un langage d'alignements, tout comme celle d'une logique pour la représentation de connaissances, définit la notion d'interprétation pour les entités identifiées dans les correspondances. Comme il a été dit dans le chapitre 2, un alignement introduit de nouvelles connaissances au sujet d'ontologies, mais ne définit pas de nouveaux termes. Par conséquent, l'interprétation des entités d'un alignement peut être ramenée à l'interprétation des termes des ontologies. Ensuite, la sémantique du langage d'alignements étend l'interprétation à toutes les entités, par application de règles d'interprétation.

Definition 4.2.1 (Interprétation d'un ensemble d'entités) *L'interprétation d'un ensemble d'entités $E \subseteq \mathrm{Ent}$ est un couple $\langle I, D \rangle$ tel que pour tout $e \in E$, $I(e) \in D$. On appelle D le* domaine d'interprétation global.

Exemple 4.2.1 *Considérons le langage d'entités suivant : les termes des ontologies sont des entités et pour tout ensemble fini de termes $\{t_i\}_{1 \leq i \leq n}$, $\{t_1, \ldots, t_n\}$ est une entité alignable. On définit alors la sémantique suivante. Dans ce langage, on définit l'interprétation d'un terme d'ontologie comme un élément d'un ensemble non vide E. Une interprétation des termes est* *alors étendue à une interprétation des entités alignables en utilisant la règle* $I(\{t_1, \ldots, t_n\}) = \{I(t_1), \ldots, I(t_n)\}$.

Un alignement fait intervenir des entités provenant de deux langages d'entités potentiellement différents. On définit alors une interprétation d'un alignement comme la combinaison de deux interprétations ayant le même domaine global. La figure 4.1 illustre cette notion.

Definition 4.2.2 (Interprétation d'un alignement) *Soit A un alignement d'ontologies. Une interprétation de A est un triplet $\langle I, I', D\rangle$ tel que $\langle I, D\rangle$ interprète les entités à gauche des correspondances et $\langle I', D\rangle$ interprète celles à droite selon le même domaine d'interprétation.*

De même, le langage définit l'interprétation des symboles de relation.

Definition 4.2.3 (Interprétation des relations) *Pour tout symbole de relation $r \in \mathcal{R}$, l'interprétation de r vis-à-vis du domaine global D est une relation $r^D \subseteq D \times D$. On appelle r^D l'interprétation de la relation r vis-à-vis du domaine D.*

La relation r^D ne dépend que du domaine global et du langage d'alignements. Il est donc indépendant de l'alignement interprété.

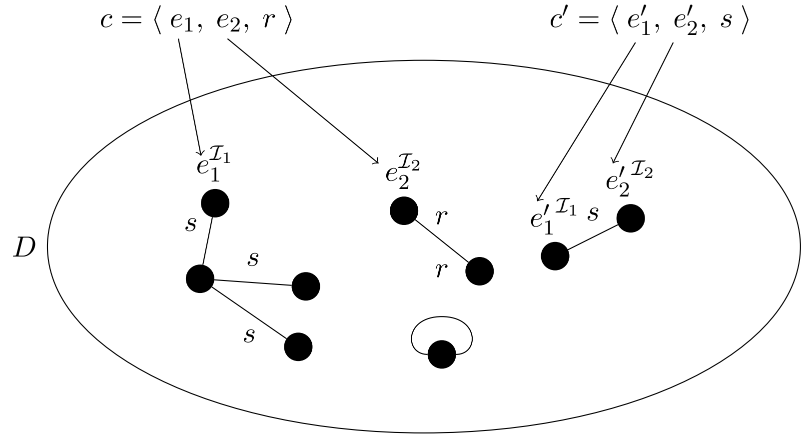


FIGURE 4.1 – Exemple d'interprétation d'alignements. La correspondance c n'est pas satisfaite ($\mathcal{I}_1, \mathcal{I}_2 \not\models c$) tandis que la correspondance c' est satisfaite ($\mathcal{I}_1, \mathcal{I}_2 \models c'$).

Exemple 4.2.2 *Prenons par exemple le langage d'alignements permettant de mettre en correspondance des concepts d'ontologies en logique de description selon les relations de subsomption $\sqsubseteq$ ou d'exclusion $\perp$. On interprète alors les entités (ici, les concepts) comme des sous-ensemble d'un univers U. De ce fait, une interprétation d'entités mises en correspondance est un couple $\langle I, \mathcal{P}(U)\rangle$. Les symboles de relation $r \in \{\sqsubseteq, \perp\}$ sont interprétés comme suit : pour chaque ensemble U, $\sqsubseteq^{\mathcal{P}(U)} = \{(x, y) \in \mathcal{P}(U)^2 \mid x \subseteq y\}$ et $\perp^{\mathcal{P}(U)} = \{(x, y) \in \mathcal{P}(U)^2 \mid x \cap y = \emptyset\}$.*

Notons qu'une telle interprétation contraint le domaine global à contenir des éléments qui sont eux-mêmes des ensembles. Dans une telle situation, il est alors pratique de définir un domaine global à la manière des logiques de description, c'est-à-dire en indiquant seulement un domaine des objets Δ, et de définir le domaine global comme $\Delta \cup 2^{\Delta}$.

À l'instar des sémantiques formelles en théorie des modèles, il convient maintenant de définir la relation de satisfaction $\models$ permettant de relier les interprétations aux formules qu'elles satisfont.

4.2.2 Modèle d'un alignement

La notion de modèle d'alignement ressemble fortement à celle de modèle d'une ontologie. Dans un langage d'alignements, les formules ou axiomes sont représentées par des correspondances. On définit la relation $\models$ entre les interprétations définies précédemment et les correspondances. Le principe est simple : une interprétation $\langle I, I', D\rangle$ satisfait une correspondance $\langle e, e', r\rangle$ lorsque le couple $\langle I(e), I'(e')\rangle$ appartient à la relation r^D.

Definition 4.2.4 (Satisfaction d'une correspondance) *Soit $\langle I, I', D\rangle$ une interprétation d'un alignement A. Soit $c = \langle e, e', r\rangle$ une correspondance sur les entités de A. On dit que $\langle I, I', D\rangle$ satisfait c et on note $I, I' \models c$ lorsque $\langle I(e), I(e')\rangle \in r^D$.*

Definition 4.2.5 (Modèle d'un alignement) *Un modèle d'un alignement A est une interprétation I de A telle que pour tout $c \in A$, $I \models c$. On le note $I \models A$.*

Ces notions d'interprétations permettent de raisonner sur les alignements de la même façon que l'on raisonne sur des ontologies. Il devient alors possible de vérifier la cohérence et l'adéquation des procédés de manipulation des alignements sans avoir accès aux ontologies qu'ils mettent en correspondance. En particulier, l'opération de composition sera détaillée dans le chapitre 7.

Néanmoins, l'objectif de cette sémantique est de faire interagir des connaissances multiples mises en correspondance. Il faut donc relier cette sémantique des alignements aux sémantiques des ontologies. C'est ce que décrit la section suivante.

4.3 Interprétation d'un réseau d'ontologies et d'alignements

4.3.1 Logique distribuée concrète

Comme énoncé au précédemment, ce chapitre décrit une sémantique générique, ou plus exactement une classe de sémantique distribuée. Dans cette section, on définit ce qui caractérise une sémantique distribuée particulière. Par la suite, on utilisera la notation Ld pour désigner une logique distribuée concrète, c'est-à-dire une logique distribuée utilisant une sémantique distribuée particulière.

Ce qui caractérise la syntaxe d'une logique distribuée concrète est la (ou les) syntaxe(s) de la (des) logique(s) locale(s), ainsi que la syntaxe du langage d'alignements. Pour caractériser la sémantique, il faut une notion supplémentaire, à savoir celle de fonction d'égalisation.

Afin de tirer profit des connaissances décrites dans plusieurs ontologies et de les relier à l'interprétation d'un alignement, on définit la notion de fonction d'égalisation, qui sert à relier les interprétations locales (qui, rappelons-le, peuvent contenir des domaines complètement différents) en les projetant dans un domaine dit global, dans lequel elles pourront être comparées au regard des correspondances.

Definition 4.3.1 (Fonction d'égalisation) *Soit Ω un ensemble d'ontologies. Soit $\mathbf{I} = (I_o, D_o)_{o\in\Omega}$ une famille d'interprétations des ontologies de Ω. Une fonction d'égalisation pour $\mathbf{I}$ est une famille de fonctions $\boldsymbol{\varepsilon} = (\varepsilon_o : D_o \to \Delta)_{o\in\Omega}$ ayant même codomaine Δ.*

Grâce à ces fonctions, des sémantiques locales distinctes peuvent être comparées, sans toutefois nécessiter une transformation des ontologies vers un langage ou un contexte commun.

La fonction d'égalisation est ici définie de la façon la plus générale possible. Étant donnée la définition d'interprétation utilisée (voir la définition 2.2.2, au chapitre 2), on peut associer à l'interprétation d'un concept de logique de description aussi bien un ensemble de couples qu'un élément atomique. Or, il est parfois utile d'imposer qu'un ensemble soit associé à un ensemble et une relation à une autre relation. On reviendra sur ce point plus loin dans ce chapitre, ainsi que dans le chapitre 8.

Par conséquent, une logique distribuée concrète définit aussi un sous-ensemble des fonctions d'égalisation. En résumé, on peut décrire formellement les caractéristiques d'une logique distribuée concrète.

Definition 4.3.2 (Logique distribuée concrète) *Une logique distribuée Ld est définie par :*

1. *un langage d'alignements $A(Ld)$, c'est-à-dire les constructeurs sur les entités alignables et les relations exprimables entre elles ainsi que leur interprétation ;*
2. *un ensemble de logiques locales autorisées $LL(Ld)$;*
3. *un ensemble $E(Ld)$ de fonctions d'égalisation autorisées.*

Dans la suite du chapitre, on supposera que l'on se place dans une logique distribuée concrète quelconque.

4.3.2 Interprétation et modèles d'ontologies alignées

Avant de s'intéresser à l'interprétation d'un réseau d'ontologies alignées, il convient de présenter d'abord le cas d'une simple paire d'ontologies alignées. Dans cette section, on supposera que Ld désigne une logique distribuée concrète.

Definition 4.3.3 (Paire d'ontologies alignées) *Lorsque A est un alignement pour les ontologies o et o', on appelle le triplet $\langle o, o', A\rangle$ une* paire d'ontologies alignées.

Les réseaux d'ontologies réduits à une paire d'ontologies alignées correspond à un cas de figure très fréquent, tant pour l'intégration d'information que pour traduire des requêtes posées en terme d'une ontologie en requêtes posées en terme d'une autre ontologie ou encore pour transformer des données de la première ontologies en données conformes à la seconde.

Definition 4.3.4 (Interprétation d'une paire d'ontologies alignées) *On définit une interprétation d'une paire d'ontologies alignées $\langle o, o', A\rangle$ (dans Ld) comme un couple $\langle \mathbf{I}, \boldsymbol{\varepsilon}\rangle$ dans lequel $\mathbf{I}$ désigne une paire d'interprétations $(\langle I_o, D_o\rangle, \langle I_{o'}, D_{o'}\rangle)$ pour $\{o, o'\}$ et $\boldsymbol{\varepsilon} \in E(Ld)$ est une fonction d'égalisation pour $\mathbf{I}$ (et autorisée dans Ld), tel que $\varepsilon_o \circ I_o$ (respectivement $\varepsilon_{o'} \circ I_{o'}$) est une interprétation des entités à gauche de A (respectivement à droite de A). Dans ce cas, le codomaine de ε est appelé* domaine d'interprétation global.

Exemple 4.3.1 *On imagine une base de connaissances d'un libraire, mise en correspondance avec une ontologie bibliophile contenant des instances de la classe* `Roman`*. Chez le libraire, on trouve notamment sept instances $t_1, \ldots, t_7$ identifiant les livres intitulés Du côté de chez Swann, À l'ombre des jeunes filles en fleurs, Le côté de Guermantes, Sodome et Gomorrhe, La prisonnière, Albertine disparue et Le temps retrouvé. Dans l'ontologie bibliophile, on trouve une instance i identifiant l'œuvre nommée À la recherche du temps perdu. En supposant que le langage d'entités permet de définir des ensembles énumérés, on considère l'alignement contenant la correspondance $\langle\{t_1, t_2, t_3, t_4, t_5, t_6, t_7\}, i, =\}$. On imagine en outre que le langage d'alignements définit l'interprétation d'une énumération comme l'ensemble des interprétations de chaque élément énuméré. Ces instances sont interprétées localement comme des* *individus d'un domaine d'interprétation quelconque. La fonction d'égalisation est alors définie comme suit : pour $1 \leq k \leq 7$, $\varepsilon_1(t_k^{I_1}) = x_k$ et $\varepsilon_2(i^{I_2}) = \{x_1, \ldots, x_7\}$ avec $\Delta = \{x_1, \ldots, x_7, \{x_1, \ldots, x_7\}\}$.*

L'exemple précédent montre notamment que la fonction d'égalisation peut agir comme une fonction d'abstraction.

On peut facilement faire le lien entre la définition précédente et celle d'une interprétation générale d'un alignement. En effet, si $\langle \mathbf{I}, \boldsymbol{\varepsilon} \rangle$ est une interprétation d'une paire d'ontologies alignées $\langle o, o', A \rangle$ avec Δ comme codomaine de $\boldsymbol{\varepsilon}$, alors $\langle \varepsilon_o \circ I_o, \varepsilon_{o'} \circ I_{o'}, \Delta \rangle$ est une interprétation de l'alignement A. [1]

La définition d'un modèle d'une paire d'ontologies alignées est alors très naturelle. Pour qu'une interprétation d'une paire d'ontologies alignées soit un modèle, il faut que les interprétations locales satisfassent les ontologies correspondantes et que l'alignement soit satisfait.

Definition 4.3.5 (Modèle d'ontologies alignées) *Soit $\langle o, o', A \rangle$ une paire d'ontologies alignées. Une interprétation $\mathcal{I} = \langle \mathbf{I}, \boldsymbol{\varepsilon} \rangle$ est un modèle de $\langle o, o', A \rangle$ si et seulement si $I_o \models o$, $I_{o'} \models o'$ et $\boldsymbol{\varepsilon} \circ I, \boldsymbol{\varepsilon}' \circ I', \Delta \models A$. Dans ce cas, on note simplement $I_o, I_{o'} \models_{\boldsymbol{\varepsilon}} A$.*

Une paire d'ontologies alignées forme un petit réseau d'ontologies et d'alignements. On peut généraliser facilement les définitions précédentes au cas d'un réseau de taille finie quelconque.

4.3.3 Modèle d'un réseau d'ontologies alignées

On étend d'abord la notion d'interprétation à un réseau de taille quelconque.

Definition 4.3.6 (Interprétation distribuée) *Soit $S = \langle \mathbf{O}, \mathbf{A} \rangle$ un réseau d'ontologies et d'alignements. Une interprétation de S est un couple $\langle \mathbf{I}, \boldsymbol{\varepsilon} \rangle$ tel que $\mathbf{I} = (I_o, D_o)_{o \in \mathbf{O}}$ assigne à chaque ontologie une interprétation selon la sémantique du langage associé et $\boldsymbol{\varepsilon} \in E(Ld)$ est une fonction d'égalisation pour $\mathbf{I}$, autorisée pour la logique Ld.*

Pour satisfaire toutes les connaissances exprimées dans un réseau d'ontologies alignées, une interprétation distribuée doit satisfaire localement les axiomes des ontologies mises en jeu et doit aussi satisfaire chacun des alignements.

Definition 4.3.7 (Modèle d'un réseau d'ontologies alignées) *Soit $S = \langle \mathbf{O}, \mathbf{A} \rangle$ un réseau d'ontologies et d'alignements. Une interprétation $\mathcal{I} = \langle \mathbf{I}, \boldsymbol{\varepsilon} \rangle$ de S est un modèle pour S si et seulement si :*

1. *pour tout $o \in \mathbf{O}$, $I_o \models o$;*
2. *pour toute paire d'ontologies alignées $\langle o, o', A \rangle \in \mathbf{O} \times \mathbf{O} \times \mathbf{A}$, $I_o, I_{o'} \models_{\boldsymbol{\varepsilon}} A$.*

Dans ce cas, on note encore $\mathcal{I} \models S$.

Definition 4.3.8 (Réseau d'ontologies alignées cohérent) *Un réseau d'ontologies alignées S est cohérent si et seulement s'il existe un modèle de S.*

1. En réalité, $\varepsilon_o \circ I_o$ n'est pas à proprement parler une interprétation du langage d'entités associé à o. Pour obtenir une véritable interprétation, il faut encore que le langage d'alignements définisse des règles pour interpréter d'éventuels constructeurs. Ce problème particulier est abordé au chapitre 6.

Puisque l'on dispose maintenant de la notion de modèle pour un réseau d'ontologies et d'alignements, il est possible de définir à son tour la notion de conséquence sémantique. Les conséquences sémantiques sont les formules satisfaites par tous les modèles d'un réseau d'ontologies. Or, dans un réseau d'ontologies et d'alignements, il existe deux types de formules : les axiomes des ontologies et les correspondances des alignements.

Definition 4.3.9 (Conséquence sémantique) *Soit $S = \langle \mathbf{O}, \mathbf{A} \rangle$ un réseau d'ontologies et d'alignements. Soit α une formule du langage de l'ontologie $o \in \mathbf{O}$ (respectivement c une correspondance du langage d'alignements entre o et o'). On dit que α (respectivement c) est une conséquence sémantique de S si et seulement si, pour tout modèle $\langle \mathbf{I}, \boldsymbol{\varepsilon} \rangle$ de S, $I_o \models \alpha$ (respectivement $I_o, I_{o'} \models_{\boldsymbol{\varepsilon}} c$).*

On peut définir une notion affaiblie de la cohérence d'un réseau d'ontologies, en n'imposant que la satisfaction des alignements. Dans ce cas on dit que les alignements sont cohérents. Si une interprétation distribuée ne satisfait que les alignements, on parle de *quasi modèle* du réseau d'ontologies.

Definition 4.3.10 (Quasi modèle d'un réseau d'ontologies alignées) *Soit $S = \langle \mathbf{O}, \mathbf{A} \rangle$ un réseau d'ontologies et d'alignements. Une interprétation $\mathcal{I} = \langle \mathbf{I}, \boldsymbol{\varepsilon} \rangle$ de S est un quasi modèle pour S si et seulement si pour toute paire d'ontologies alignées $\langle o, o', A \rangle \in \mathbf{O} \times \mathbf{O} \times \mathbf{A}$, $I_o, I_{o'} \models_{\boldsymbol{\varepsilon}} A$. Dans ce cas, on note encore $\mathcal{I} \models_{\approx} S$.*

Tous les modèles d'un réseau d'ontologies alignées sont des quasi modèles. On peut aussi définir la notion de quasi conséquence, correspondante aux conséquences restreintes aux quasi modèles.

Definition 4.3.11 (Quasi conséquence sémantique) *Soit $S = \langle \mathbf{O}, \mathbf{A} \rangle$ un réseau d'ontologies et d'alignements. Soit c une correspondance du langage d'alignements entre o et o'. On dit que c est une quasi conséquence sémantique de S si et seulement si, pour tout quasi modèle $\langle \mathbf{I}, \boldsymbol{\varepsilon} \rangle$ de S, on a $I_o, I_{o'} \models_{\boldsymbol{\varepsilon}} c$. Dans ce cas, on note $S\mathcal{I} \models_{\approx} S$.*

Toute quasi conséquence est alors une conséquence sémantique du réseau d'ontologies alignées. Ainsi, déterminer des quasi conséquences revient en fait à négliger les fonctions d'égalisation. Dans ce cas, il suffit de ne considérer que la sémantique du langage d'alignements et le raisonnement est alors équivalent à un raisonnement local sur le langage d'alignements.

4.3.4 Restriction sur les fonctions d'égalisation

De manière générale, une fonction d'égalisation peut associer n'importe quel type d'élément dans un domaine d'interprétation à un élément du domaine global. Il est donc possible de mettre en correspondance des individus avec des classes ou des rôles, par exemple. Or cette liberté dans les associations empêche un grand nombre de connaissances locales d'être transférées au niveau global et réciproquement. Or, on souhaiterait que les connaissances locales servent à définir de nouvelles déductions, soit au niveau de l'alignement, soit au niveau des autres ontologies. Le théorème suivant indique que seule l'équivalence se propage du niveau local au niveau global.

Théorème 4.3.1 *Soit $S = \langle \mathbf{O}, \mathbf{A} \rangle$ un réseau d'ontologies et d'alignements. Si les alignements sont cohérents et que S est incohérent alors il existe une ontologie O_i telle que pour tout modèle $M_i \in \mathbf{Mod}(O_i)$, il existe e et e' deux entités de O_i apparaissant dans les alignements tels que $M_i(e) = M_i(e')$.*

Preuve : On suppose que la propriété n'est pas vraie. Donc, quelle que soit l'ontologie O_i, on peut trouver un modèle local M_i pour lequel toutes les entités de l'ontologie apparaissant dans les alignements sont distinctes. Par conséquent, pour la $i^{ème}$ fonction d'égalisation ε_i on peut choisir n'importe quelle image pour toutes les entités alignées. On peut donc en particulier choisir des images qui valident toutes les correspondances, puisque les alignements sont cohérents. □

Si cette propriété permet de s'affranchir de la majorité des problèmes d'incohérence globale, elle est justement indésirable si l'on veut profiter des connaissances de l'ensemble des ontologies du réseau.

Par conséquent, il est convenable de définir des contraintes sur les fonctions d'égalisation lorsqu'un formalisme distribué est concrètement défini. On pourra remarquer que la nécessité de restreindre les relations de domaine a été aussi identifiée dans la logique du premier ordre [38] et dans les $\mathcal{E}$-connections [48]. Il peut par exemple être souhaitable d'aligner des éléments de même type entre eux. Par exemple, en logique de description, on pourrait souhaiter apparier des concepts entre eux, ou bien des rôles, ou bien des individus.

Le chapitre 8 reviendra sur la question des restrictions sur les fonctions d'égalisation.

4.4 Réseau d'ontologies alignées hybride

Dans le chapitre 3, différentes sémantiques pour les réseaux d'ontologies ont été présentées. On a pu remarquer que chacune dispose d'avantages et de caractéristiques adaptées à des cas d'utilisation précis. Par conséquent, il semble raisonnable que ces formalismes soient utilisés dans ces cas de figures spécifiques. En fait, la sémantique présentée ici ne se place pas en compétition face à ces logiques établies, mais au contraire en complément. D'ailleurs, la figure 4.2 illustre comment différents formalismes peuvent être amenés à fournir une part d'information dans un système de médiation à plusieurs niveaux de distributivité.

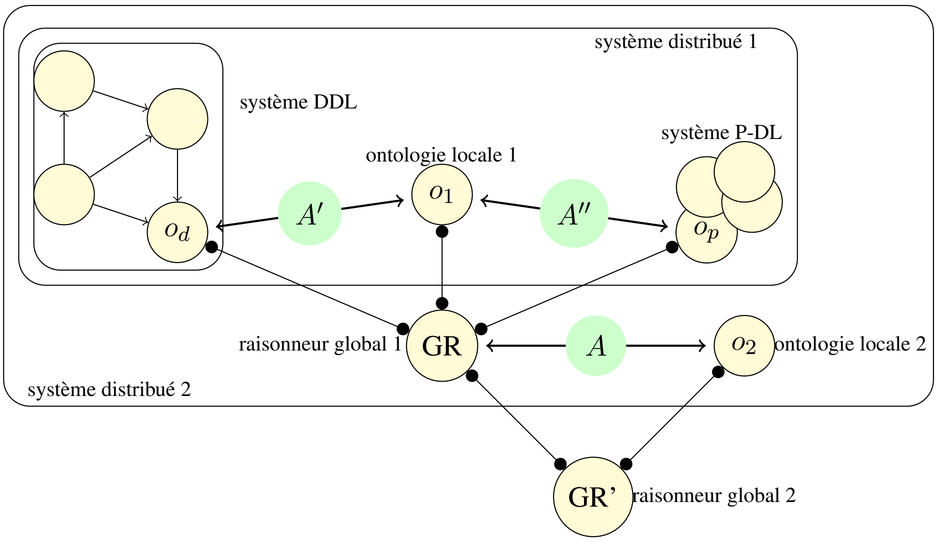


FIGURE 4.2 – Un exemple de réseau d'ontologies alignées hybride.

Cette mise en commun des formalismes est possible grâce à la définition générique de la sémantique du réseau d'ontologies, *en fonction* des sémantiques locales. Or, au niveau local, rien d'autre qu'une notion d'interprétation et de satisfaction n'est exigée pour participer à la définition d'interprétation et de modèle d'un réseau d'ontologies alignées. D'ailleurs, la sémantique donnée dans ce chapitre définit elle-même une notion d'interprétation et de satisfaction pouvant intervenir comme sémantique locale pour un autre réseau d'ontologies alignées de plus haut niveau. Ceci permet ainsi d'entrevoir les possibilités de modularité offertes par cette sémantique. Le chapitre suivant traite justement de ce sujet.

4.5 Raisonnement local vis-à-vis d'un réseau d'ontologies alignées

Un réseau d'ontologies alignées dans son ensemble peut être étudié et exploité de la même manière qu'une seule ontologie, en déterminant par exemple sa cohérence, ses conséquences sémantiques ou en posant des requêtes sur l'ensemble du réseau d'ontologies. Mais les logiques distribuées posent aussi de nouveaux problèmes relatifs aux conséquences du réseau rejaillissant sur les connaissances locales.

Dans cette section, on se focalise sur une ontologie particulière d'un réseau d'ontologies alignées et l'on étudie les implications du réseau sur l'ontologie. On peut aborder la question de deux manières. D'un côté, on peut se demander comment le réseau d'ontologies alignées influence le raisonnement local. D'un autre côté, on peut à l'inverse se demander comment une ontologie particulière influence le réseau d'ontologies alignées.

4.5.1 Modèles locaux vis-à-vis d'un réseau d'ontologies alignées

La définition de modèle d'un réseau d'ontologies et d'alignements montre clairement que les modèles locaux compatibles avec les alignements forment une restriction des modèles des ontologies prises individuellement. En d'autres termes, une ontologie plongée dans un réseau d'ontologies et d'alignements peut enrichir ses propres déductions grâce aux connaissances extérieures alignées avec elle.

Dans les définitions de la section précédente, on prenait le point de vue d'une entité supérieure ayant accès à l'ensemble du réseau. Mais il est aussi envisageable de prendre le point de vue d'une seule ontologie souhaitant exploiter les connaissances externes, qui ont pu être mises en correspondances par un serveur d'alignement par exemple. Ce cas de figure correspond à un scénario de type pair-à-pair ou bien celui d'un système multi-agents.

Avant de poursuivre, on peut se demander si, dans cette situation où l'on est intéressé par le raisonnement dans un contexte donné, il ne serait pas plus judicieux d'utiliser des correspondances contextuelles comme c'est le cas en DDL, DFOL ou $\mathcal{E}$-connections. En fait, il est légitime de se demander si l'utilisation d'un formalisme qui a recourt à un domaine d'interprétation global est appropriée alors que l'on souhaite raisonner localement. À cela on peut opposer les arguments suivants.

- Les ontologies peuvent avoir été mises en correspondance par un système tiers, auquel cas les alignements ne prennent pas le point de vue de l'une ou l'autre des ontologies ;
- un système de médiation peut se charger de transmettre des messages d'un nœud à l'autre du réseau, on s'assurant que la cohérence globale est préservée ;
- enfin, le domaine d'interprétation global n'est qu'un outil technique abstrait qui ne se traduit pas nécessairement par l'existence d'un système existant au niveau global. Ce domaine ne sert en fait qu'à décrire une hypothétique vision omnisciente du système, permettant de valider une procédure de déduction. Cette procédure peut parfaitement être localisée et contextualisée.

4.5.2 Conséquence sémantique locale

On s'intéresse à un réseau $S = \langle \mathbf{O}, \mathbf{A} \rangle$ dans lequel on choisit une ontologie particulière $\omega \in \mathbf{O}$. Dans le cas idéal, on souhaiterait définir l'ensemble des modèles de ω qui entrent en jeu dans un modèle de réseau d'ontologies et d'alignements. Ceci peut être défini comme suit.

Definition 4.5.1 *Les modèles de ω vis-à-vis de S, sont les modèles locaux m_ω de ω tels qu'il existe une famille de modèles $(m_j)_{j\neq\omega}$ des ontologies de S et une fonction d'égalisation ε pour $\mathbf{m} = (m_j)_{j\in\mathbf{O}}$ tels que $\langle \mathbf{m}, \varepsilon\rangle \models S$. On note l'ensemble de ces modèles $\mathrm{Mod}_S(\omega)$.*

Cette notion de modèle permet alors de définir les conséquences sémantiques locales, vis-à-vis d'un réseau d'ontologies alignées.

Definition 4.5.2 (Conséquence sémantique locale) *Un formule locale α du vocabulaire de ω est une conséquence sémantique locale si et seulement si pour tout $m \in \mathrm{Mod}_S(\omega)$, $m \models \alpha$.*

On peut remarquer qu'une formule α est une conséquence sémantique locale si et seulement si elle est une conséquence du réseau d'ontologies alignées, selon la définition 4.3.9. L'inconvénient de cette définition est qu'elle exige de connaitre la totalité du réseau et d'exploiter les connaissances dans leur ensemble. Or, dans un système pair-à-pair par exemple, cela exige que tous les pairs soient accessibles depuis n'importe quel autre pair et de préférence en un temps raisonnable.

Il est alors intéressant de définir une notion restreinte des modèles locaux n'utilisant que le voisinage de l'ontologie.

Definition 4.5.3 *On définit un modèle local vis-à-vis du voisinage de ω comme un modèle de ω tel qu'il existe une famille de modèles $(m_j)_{j\in\mathbf{O}\setminus\{\omega\}}$ des ontologies de S et une fonction d'égalisation ε pour $\mathbf{m} = (m_j)_{j\in\mathbf{O}}$ tels que pour tout $j \neq \omega$, $m, m_j \models_\varepsilon A_{\omega j}$. On note l'ensemble de ces modèles $\mathrm{Mod}^+_S(\omega)$.*

Cette définition n'est pas équivalente à la précédente puisqu'elle n'exige pas que les alignements entre deux autres ontologies que ω soient validés. Néanmoins, l'ensemble des alignements peut être en partie exploité en utilisant l'opération de composition d'alignement dont je reparlerai au chapitre 7. Toutefois, si l'on souhaite utiliser plus précisément chaque alignement, on peut définir une notion comparable mais utilisant la notion de point fixe.

Definition 4.5.4 *On note $\mathrm{Mod}^\star_S(\omega)$ l'ensemble des modèles de ω tels qu'il existe une famille de modèles $(m_j)_{j\in\mathbf{O}\setminus\{\omega\}}$ des ontologies de S et une fonction d'égalisation ε pour $\mathbf{m} = (m_j)_{j\in\mathbf{O}}$ tels que pour tout $j \neq \omega$, $m, m_j \models_\varepsilon A_{\omega j}$ et $m_j \in \mathrm{Mod}^\star_S(O_j)$.*

Cette définition n'est pas non plus équivalente à celle de modèle local vis-à-vis d'un réseau. Pour s'en convaincre, considérons un réseau composé de trois ontologies O_1, O_2 et O_3, contenant respectivement les termes C, D et E. On introduit alors les correspondances $\langle C, D, \equiv\rangle$, $\langle D, E, \equiv\rangle$ et $\langle E, C, \perp\rangle$. On suppose que pour tout domaine global Δ, $\equiv^\Delta$ est l'identité sur Δ et que $\perp^\Delta$ est l'exclusion. Le réseau est donc incohérent, mais pourtant, tous les modèles locaux satisfont la définition précédente. Ceci vient du fait que la définition ne tient compte que des alignements avec le voisinage. En revanche, en composant les alignements (ce qui est possible avec ce formalisme, mais pas avec DDL, DFOL ou $\mathcal{E}$-connection) on aurait pu déduire que $\langle C, E, \perp\rangle$. Mais la composition n'offre pas un mécanisme de déduction complet comme on le verra au chapitre 7.

Avec ces définitions, il n'est plus utile de connaitre la totalité du réseau pour effectuer des déductions. Néanmoins, il est encore nécessaire de prendre en compte simultanément un nombre d'alignements égal au nombre d'ontologies alignées avec ω. Dans un large réseau, ceci peut-être encore trop exigeant. On affaiblit encore la notion de modèle local vis-à-vis du réseau d'ontologies avec la définition suivante.

Definition 4.5.5 *Soit* $\mathrm{Mod}^{\leftrightarrows}_S(\omega)$ *l'ensemble des modèles* m_ω *de* ω *tel qu'il existe une fonction* $\varepsilon_\omega : \Delta^{m_\omega} \to \Delta$ *et pour tout* $j \neq \omega$*, il existe un modèle* m_j *de* j *et une fonction* $\varepsilon_j : \Delta^{m_j} \to \Delta$ *tels que* $m_\omega, m_j \models_{\boldsymbol{\varepsilon}} A_{\omega j}$*.*

De même que précédemment, on peut modifier la définition pour obtenir une version avec point fixe $\mathrm{Mod}^{\leftrightarrows\star}_S(\omega)$.

Si l'on compare les différents ensembles de modèles donnés ici, on peut constater qu'ils contiennent tous l'ensemble $\mathrm{Mod}_S(\omega)$ (voir le theorème 4.5.1). Par conséquent, chacune de ces définitions permet de décrire une forme de raisonnement correcte, bien qu'incomplète.

Propriété 4.5.1 *Les différents types de modèles locaux vis-à-vis d'un réseau d'ontologies alignées sont reliés par les inégalités suivantes.*

$$\begin{tikzcd}
 & \mathrm{Mod}^+_S(\omega) \arrow[dr, phantom, "\subsetneq" sloped] & \\
\mathrm{Mod}_S(\omega) \subseteq \mathrm{Mod}^\star_S(\omega) \arrow[ur, phantom, "\subsetneq" sloped] \arrow[dr, phantom, "\subsetneq" sloped] & & \mathrm{Mod}^{\leftrightarrows}_S(\omega) \subseteq \mathbf{Mod}(\omega) \\
 & \mathrm{Mod}^{\leftrightarrows\star}_S(\omega) \arrow[ur, phantom, "\subsetneq" sloped] &
\end{tikzcd}$$

Cette comparaison de différentes notions de modèles montre que l'on peut accommoder de différentes manières la notion de modèle afin de trouver le compromis souhaité entre efficacité du raisonnement et complétude des déductions. En effet, sur le Web sémantique par exemple, le passage à l'échelle est un problème crucial et les procédures de raisonnement incomplètes offrent une piste intéressante. Une autre approche consiste à déterminer des propriétés permettant de garantir que le raisonnement peut se faire de façon simplifiée.

4.5.3 Extension conservative et localité

Soit S un réseau d'ontologies alignées admettant un modèle $\mathcal{I} = \langle \mathbf{I}, \boldsymbol{\varepsilon} \rangle$. On définit alors le réseau d'ontologies S' obtenu à partir de S en y ajoutant une ontologie ω et des alignements $A_{\omega i}$ pour toute ontologie O_i dans S.

Definition 4.5.6 (Extension d'un modèle de réseau d'ontologies) *On dit que* $\mathcal{I}$ *peut être étendu à un modèle* $\mathcal{I}'$ *de* S' *s'il existe un modèle* I_ω *de* ω *et une fonction d'égalisation* $\boldsymbol{\varepsilon}'$ *pour* $\mathbf{I} \cup \{I_\omega\}$ *telle que* $\langle \mathbf{I} \cup \{I_\omega\}, \boldsymbol{\varepsilon}' \rangle$ *est un modèle de* S'*.*

Definition 4.5.7 (Extension conservative) *On dit que* $\langle \omega, (A_{\omega i})_{i \in S'} \rangle$ *forme une extension conservative de* S *si tous les modèles de* S *peuvent être étendus à des modèles de* S'*.*

On retrouve la propriété définie pour les logiques de description dans [42]. On peut remarquer que si les alignements associés à la nouvelle ontologie sont vides et que l'ontologie est cohérente, alors on obtient nécessairement une extension conservative. Par conséquent, si l'on tient à garantir cette propriété, il suffit de contraindre la construction de nouveaux alignements pour qu'ils ne transmettent pas de connaissances indésirables d'une ontologie à l'autre.

Le fait d'avoir une extension conservative est surtout intéressant pour l'efficacité du raisonnement, puisqu'alors toutes les conséquences sémantiques locales d'une ontologie de S vis-à-vis de S' sont aussi les conséquences locales vis-à-vis de de S. Donc, à supposer que l'on puisse calculer cette propriété au préalable, elle permet d'obtenir des réseaux d'ontologies alignées dans lesquels le raisonnement est fortement parallélisable.

Par ailleurs, se savoir en présence d'une extension non conservative assure que les connaissances extérieures à une ontologie ont bien servi à compléter les connaissances locales. Une version affaiblie de la conservativité fait intervenir les conséquences sémantiques plutôt que les modèles.

Definition 4.5.8 (Extension conservative affaiblie) *On dit que $\langle \omega, (A_{\omega i})_{i \in S} \rangle$ forme une extension conservative affaiblie de S si pour toute ontologie O_i de S et toute formule α dans le vocabulaire de O_i, $S \models_d \alpha \Leftrightarrow S' \models_d \alpha$.*

Cette notion d'extension conservative est à rapprocher de celle d'extension conservative déductive, tandis que la précédente peut être qualifiée d'extension conservative de modèles ("*model-theoretic conservative extension*" [24]). Toutefois, ces notions ne coincident pas exactement puisque le concept d'extension conservative déductive ou de modèles est plus général et s'applique à toute logique, tandis qu'ici, cette notion est restreinte au cas où l'on peut exhiber différentes ontologies au sein d'un réseau.

4.6 Bilan

Ce chapitre a présenté de manière très générale la sémantique d'un réseau d'ontologies alignées, en faisant abstraction des logiques utilisées localement, ainsi que du langage concret d'alignement. La possibilité de combiner différents formalismes distribués a aussi été présentée. Enfin, de nombreuses extensions du raisonnement local vis-à-vis d'un réseau d'ontologies alignées ont été proposées.

Par ailleurs, les trois chapitres suivants mettent en application cette sémantique générique dans le cadre des ontologies modulaires (chapitre 5), puis dans la définition d'une sémantique d'un langage d'alignements très expressif (chapitre 6) et enfin pour la composition d'alignements d'ontologies (chapitre 7).

Tous ces problèmes de raisonnement ne peuvent pas être abordés dans le cadre de ce mémoire, mais le chapitre 8 proposera une procédure de vérification de la cohérence d'un réseau d'ontologies alignées, dans le cas concret des ontologies en logiques de description.

Chapitre 5

Application à la modularité

Résumé

Une des applications importantes de la médiation entre ontologies est la conception d'ontologies modulaires par réutilisation des connaissances issues de sources variées. L'objectif de la *modularité* est d'offrir des méthodes et outils pour construire des ontologies complexes à partir de modules simples, en les assemblant selon des règles bien définies. Je distingue ce problème de celui de la *modularisation* qui consiste à mettre en évidence et à extraire de tels modules simples à l'intérieur d'une vaste ontologie monolithique. Dans ce chapitre, je décris une infrastructure pour la conception d'ontologies modulaires ayant la propriété d'encapsuler les ontologies, d'une manière qui rappelle l'ingénierie logicielle. Je montre que la sémantique que je propose permet bien de garantir cette encapsulation et permet de réutiliser des modules hétérogènes au sein d'une même ontologie.

Sommaire

5.1 Objectif

Les études concernant les modules d'ontologies se divisent en deux approches principales. D'un côté on trouve la *modularisation*, qui consiste à former des modules de taille réduite à partir d'une large ontologie monolithique. On parle alors souvent d'une approche par partionnement. Ceci ne sera pas abordé dans ce chapitre. D'un autre côté, on trouve la modularité, qui consiste à concevoir d'emblée des ontologies à partir de modules plus simples et plus réduits. Cette approche est aussi appelé composition de modules. Nous préférons utiliser le terme de composition dans son sens mathématique, pour la composition de fonctions, de morphismes ou de relations (et par extension, d'alignements d'ontologies).

Cet aspect des ontologies est lié au raisonnement distribué car on peut s'attendre à ce que, sur le Web sémantique, les ontologies réparties soient réutilisées dans divers contextes. Cependant, dans l'état actuel du Web sémantique, la réutilisation est difficile à accomplir avec les outils disponibles. En particulier, le langage d'ontologies recommandé pour le Web sémantique, OWL, offre une primitive `owl:import` qui ne parvient pas à fournir assez de contrôle sur ce qui est importé.

Du point de vue de l'ingénierie des ontologies, c'est un défaut de la définition des ontologies sur le Web sémantique. D'un point de vue sémantique, cela ne favorise pas les possibilités de répartition des mécanismes de déduction et donc leur passage à l'échelle mais l'on peut faire du raisonnement distribué sans faire appel à `owl :import`, comme dans SomeOWL [4] par exemple.

Les modules d'ingénierie logicielle reposent sur la séparation entre l'interface et l'implémentation d'un module. L'interface décrit les entités du module qui peuvent être accédées de l'extérieur du module, tandis que l'implémentation n'est normalement pas disponible depuis l'extérieur, hormis à travers l'interface. Ceci aide à contrôler ce qui est fourni par un module. Le format proposé ici est conforme à ce désir et la sémantique proposée traduit ce principe.

Pour cela, il faut être capable d'exprimer ce qui est visible ou caché, grâce à la définition d'interfaces. De surcroît, pour réutiliser des modules conçus indépendamment, l'infrastructure de module doit fournir des moyens de les rattacher l'un à l'autre de façon cohérente. Ceci est accompli en spécifiant — ou en se référant à — des alignements d'ontologie dans les modules. De cette manière, le système de modules peut profiter des technologies d'alignement d'ontologies, ainsi que de l'opération de composition d'alignements.

Ces modules peuvent remplacer les ontologies où qu'elles soient utilisées. C'est pourquoi il convient de définir leur sémantique formelle, c'est-à-dire ce qui peut être déduit d'une ontologie modulaire.

Ce chapitre se divise en trois sections. Dans un premier temps, je décris un exemple détaillé d'ontologie modulaire utilisant des modules de sources différentes, reliés par des alignements (section 5.2). Cet exemple sert à la fois de motivation pour ce chapitre, mais aussi d'exemple d'application pour les chapitres à venir. Ensuite, je présente le langage de spécification de modules permettant leur réutilisation, l'encapsulation et leur mise en correspondance par les alignements. Ce format de description de modules a été proposé dans [35]. Enfin, je fournis la sémantique formelle de ce langage de module, en restant assez général pour couvrir un grand nombre de formalismes d'ontologies.

5.2 Exemple détaillé

Imaginons une application intégrée permettant d'accéder à des informations culturelles, en particuliers sur des œuvres littéraires, musicales ou cinématographiques, et offrant la possibilité d'acheter certaines de ces œuvres en fournissant un portail commun à de nombreux libraires, disquaires ou vidéothèques, entre autres services. Il est nécessaire que l'agent logiciel soit capable d'utiliser des informations fournies par d'autres sources. Pour cela, il profitera d'autres modules. Par exemple, un client peut vouloir acheter en même temps un livre et un disque provenant de deux sites marchands différents. Le système doit alors être capable de trouver des informations décrites par ces deux parties, bien qu'elles définissent leurs informations selon des ontologies différentes (discographie et littérature). Elles-même utilisent potentiellement d'autres modules, comme par exemple une ontologie technique sur la musique, pour le disquaire. Il est aussi possible que plusieurs ontologies d'un même domaine soient utilisées, par exemple selon que la littérature est décrite du point de vue du libraire, ou bien de celui des chercheurs en littérature. La figure 5.3 à la page 81 décrit la situation.

Exemple 5.2.1 *L'exemple repose sur des ontologies existantes, que l'on peut trouver, pour la plupart, sur le Web.*

- foaf *est l'ontologie Friend Of A Friend au sujet des personnes et des relations sociales* [1]*,*
- time *est une ontologies développée par le W3C pour les concepts et relations temporels* [2]*,*
- event *est une ontologie des évènements, particulièrement de types sociaux (réunions, spectacles, etc.)* [3]*,*
- mo *est une ontologie de la musique (The Music Ontology), principalement centrée sur ce qui a trait à l'industrie du disque* [4]*,*
- music1 *est une ontologie de la musique centrée sur l'opéra, les orchestres, la musique classique et symphonique, et possédant une classification des instruments de musique* [5]*,*
- Bio *est une ontologie pour les informations biographiques* [6]*,*
- movies *est une ontologie du cinéma* [7]*,*
- Biblio *est une ontologie des références bibliographiques (The Bibliographic Ontology* [8]*),*
- order *est une ontologie des transactions commerciales, notamment pour les commandes en ligne* [9].

Par ailleurs, des bases de données ou base de connaissances "ouvertes" sont utilisées.

- musicbrainz *est une base de connaissances sur la musique, utilisant l'ontologie* mo [10]*,*
- last.fm *est une base de données musicale qui se met à jour en fonction des métadonnées présents dans les MP3 écoutés par les utilisateurs* [11]*,*
- discogs *est une base de données musicales, mise à jour par des contributions d'utilisateurs et des professionnels* [12]*,*
- imdb *est une base de données sur le cinéma (The Internet Movie Data Base* [13]*),*
- dblp *est une base de connaissances sur les publications scientifiques en informatique* [14]*,*
- dbpedia *est une base de connaissances généralistes traduisant les informations structurées de Wikipédia en RDF* [15]*,*
- Amazon *correspond à la base de données du site de vente en ligne* [16]*,*
- iTunes *est la base de données du site de vente de musique numérique* [17].

Les autres modules correspondent à des ontologies fictives introduites pour l'exemple.

1. http://www.foaf-project.org/
2. http://www.w3.org/2006/time#
3. http://purl.org/NET/c4dm/event.owl#
4. http://purl.org/ontology/mo/
5. http://www.cours.polymtl.ca/inf6410/Documents/music1.owl
6. http://purl.org/vocab/bio/0.1/
7. http://www.csd.abdn.ac.uk/~ggrimnes/dev/imdb/IMDB
8. http://bibliontology.org
9. http://www.dayf.de/2004/owl/order.owl
10. http://musicbrainz.org/
11. http://www.last.fm/
12. http://www.discogs.com/
13. http://www.imdb.com/
14. http://www.informatik.uni-trier.de/~ley/db/
15. http://dbpedia.org/
16. http://www.amazon.com/
17. http://www.apple.com/itunes/

5.2.1 Définir les importations de modules

Un module comme PortailCulturel est chargé de fournir des informations culturelles ou techniques sur des œuvres artistiques, audiovisuelles ou littéraires, et propose, lorsque c'est possible, d'acheter les produits associés. Ce module fournit une interface homogène à d'autres ontologies hétérogènes. En effet, cette ontologie s'appuie sur diverses ontologies plus spécifiques : Cinema, Litterature, Musique. Des ontologies plus génériques sont réutilisées par plusieurs modules : Foaf permet de décrire les artistes musicaux ainsi que les techniciens du cinéma. Le module time indique les informations temporelles, qu'elles soient liées à des évènements comme des spectacles ou bien à des périodes ou instants particuliers de la vie des artistes (naissances, décès, etc.). Dans la figure 5.3, les flèches simples indiquent la relation d'importation entre modules. Ainsi, le langage de modules doit permettre d'exprimer par exemple que :

– *le Module* PortailCulturel *utilise les Modules* Cinema, Musique *et* Litterature,
– *le Module* Bio *utilise les Modules* Foaf, time *et* dbpedia.

En fait, Musique n'importe que des informations spécifiques des autres modules : les propriétés $genre$ et $composer$ de l'ontologie mo mais pas les propriétés $produced\_sound$ et $sample\_rate$; $String\_Instrument$ et $Woodwind\_Instrument$ dans le module music1, mais pas $Symphonic\_Poem$. Pour cela, le langage doit exprimer que :

– *le Module* Musique *importe la Classe* $MusicalWork$ *et* $Record$ *et les Propriétés* $composer$, $genre$, has_track, $performed$, $release\_type$ *du Module* mo.
– *le Module* Cinema *importe la Classe* $Movie$ *et les Propriétés* $participates$, $character$, has_music *et* $Runtime$ *du Module* movies.

5.2.2 Déclarer le contenu du module

Le but du module PortailCulturel est de décrire des informations liées aux œuvres artistiques et leurs auteurs, tout en fournissant des critiques ou chroniques, ainsi que des références à des produits commerciaux. En particulier, ce module définit les classes $OeuvredArt$ et $Artiste$.

– *le Module* PortailCulturel *définit la Classe* $OeuvredArt$ *avec les Propriétés* $auteur$ *qui doit être un* $Artiste$, $dateDeCreation$ *qui doit être une Date et* $description$ *qui doit être une Chaîne de caractères.*
– *le Module* PortailCulturel *définit la Classe* $Artist$ *comme une sorte de* $Personne$.

En outre, le module PortailCulturel peut utiliser les termes importés dans la définition de son ontologie. Par exemple, la classe $MusicArtist$ du module Musique (elle même importée du module mo) est une sous classe de $Artiste$, idem pour la classe $Realisateur$ dans le module Cinema. Aussi, l'on peut préciser qu'un $Realisateur$ est nécessairement lié à un $Film$ par la propriété $realise$.

– *Un* $MusicArtist$ *est un* $Artiste$.
– *Un* $Realisateur$ *est un* $Artiste$ *qui* $realise$ *au moins un* $Film$.

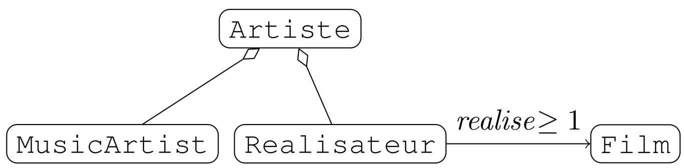


FIGURE 5.1 – Représentation graphique du contenu du module.

5.2.3 Aligner des modules différents

L'ontologie Musique utilise des concepts et relations de deux ontologies de la musique, mo et music1. Or des alignements entre les ontologies de domaines similaires existent sur le Web, qui permettent d'établir les correspondances entre celles-ci. D'autres correspondances peuvent être aussi ajoutées au sein du module qui les importe.

– *la Classe $Musical\_Work$ dans le Module* music1 *correspond à la classe $MusicalWork$ dans le Module* om.
– *la Classe $Musical\_Representation$ dans le Module* music1 *correspond à la classe $MusicalExpression$ dans le Module* om.
– *la Propriété $published$ du Module* music1 *correspond à la Propriété $published\_as$ du Module* om *restreinte aux seuls $Libretto$, $Score$ et $Lyrics$.*
– *la Propriété $represents$ du Module* music1 *est plus générale que la Propriété $performs$ du Module* mo.
– *la Classe $Musician$ dans le Module* music1 *est plus génerele que la portée (range) de la Propriété $supporting\_musician$ dans le Module* mo.
– *la Propriété $plays$ dans le Module* Boatmusic1 *est plus spécifique que la composition des Propriétés $performs$ et $instrument$ dans le Module* mo.

Le module mo est lui-même issu de l'intégration de diverses ontologies (event, time, etc.) par l'utilisation de la primitive `owl:import`. Mais il est de la responsabilité du module mo de mettre ces ontologies en correspondance et cela ne concerne pas le module Musique. Il est notable que de tels alignements pourraient exister dans un dépôt d'alignements et l'infrastructure de module doit permettre de s'y référer. On peut remarquer que la correspondance impliquant la propriété $published$ n'est pas exprimable en OWL pour l'instant, puisque les restrictions de portée ne s'appliquent qu'aux classes, pas aux propriétés.

5.2.4 Résumé des besoins

En résumé, dans cet exemple, on souhaiterait utiliser certains modules externes de telle sorte que :

– Ce qui est exporté par chaque module est précisément défini : mo exporte $Record$;
– Le module importateur peut restreindre les entités importées : Musique peut importer $Concert$ mais pas $Recital$ depuis music1 par exemple ;
– Les entités importées peuvent être mises en correspondance avec des entités importées d'autres modules : un $Instrument$ est mis en correspondance avec un $Musical\_Instrument$.

Mais ce faisant, on souhaiterait connaitre les conséquences sémantiques appliquées aux modules. En particulier on voudrait que :

– ce qui est vrai au sujet de $Livre$ dans Litterature soit aussi vrai pour un $Roman$ correspondant dans PortailCulturel ;
– Cinema ne peut rien connaitre des conséquences sémantiques qui n'impliquent pas des entités importés de Bio ou movies.

Ces intuitions seront rendues plus précises par la suite.

5.2.5 Avantages

Cette organisation permet une bonne indépendance des différents modules participant au réseau de modules. Le seul contrat entre le module importé et le module qui l'importe est de définir les termes exportés.

En particulier, les modules n'offrent aucune garantie sur l'"implémentation" de ces termes, c'est-à-dire sur les axiomes qui gouvernent les concepts définis dans le module. Par conséquent, l'implémentation d'un module peut s'améliorer sans altérer l'interface d'export sans empêcher l'application de fonctionner. C'est l'*encapsulation*.

Au contraire, si l'interface d'export est modifiée de telle sorte que l'application ne puisse plus fonctionner, ceci peut être vérifié sans observer les axiomes. Cela assure la propriété de *séparation du développement* pour ces modules. Par exemple, le module Achats peut intégrer des ontologies de nouvelles plateformes de vente de disques en ligne.

Cela permet une plus facile *réutilisation* de modules : si un meilleur module fournit l'information concernant mo (par exemple, un module mo.v2.0, il peut, tant qu'il fournit la même interface d'export, remplacer l'ancien sans briser l'application complète.

Ces propriétés requièrent une sémantique particulière pour ces modules qui diffère de la sémantique classique qui résulterait de la clôture transitive des importations en OWL (à savoir l'union de tous les axiomes de la clôture transitive). En particulier, les conséquences d'un module peuvent seulement être des formules faites de concepts et d'attributs dans les interfaces des modules et, sitôt qu'un module n'exporte pas les entités qu'il importe, elles ne sont plus visibles depuis les ontologies qui importent le dernier module.

Pour remplir ces objectifs, la section suivante fournit une syntaxe à laquelle pourra être greffée une sémantique adhérant aux besoins formulés ici.

5.3 Syntaxe

Cette section introduit les divers composants entrant dans la définition d'un module. Ceci fournit une syntaxe abstraite au langage de spécification de modules. On se servira essentiellement des logiques de description (comme OWL) en tant que langage d'ontologies. Cependant, la sémantique peut s'adapter à d'autres langages d'ontologies comme on le verra en section 5.4.

5.3.1 Composants des modules

Un module contient une définition d'ontologie qui peut utiliser des entités (c'est-à-dire des classes et propriétés) de modules importés. Donc la syntaxe des modules proposées ici impose que les modules importés soient clairement spécifiés et seules les entités de ces modules sont utilisées, en plus de celles définies localement. Pour corréler des modules importés potentiellement hétérogènes, ils sont reliés par des alignements d'ontologies, pouvant être définis localement dans le module ou référencés à l'aide d'un URI. Enfin, pour assurer le principe d'encapsulation, la définition du module établit explicitement quelles entités sont exportées, c'est-à-dire lesquelles peuvent être utilisées par d'autres modules qui l'importent. Accessoirement, il devrait y avoir une description textuelle indiquant ce que représentent les concepts exportés et qui devraient expliquer ce que le module garantit être vrai tout au long de l'évolution du développement du module (à la manière d'une documentation d'API indiquant le fonctionnement des méthodes sans en fournir le code source). Cette description spécifie donc une partie du comportement du module sans divulguer les détails d'implémentation. En particulier, seuls les termes donnés dans l'interface d'exportation ont besoin d'être décrits aux utilisateurs extérieurs. La table 5.1 résume tout cela.

Bien entendu, le contenu du module et les alignements ne peuvent se référer qu'aux entités des ontologies faisant partie de l'interface d'importation. Quant à celle-ci, elle ne peut que se référer aux entités faisant partie de l'interface d'exportation des modules importés.

Les alignements et les ontologies peuvent être référencés par leur URI, ce qui rend possible la réutilisation d'alignements et d'ontologies déjà publiés indépendamment de la définition d'un module.

Le graphe formé par la relation d'importation, c'est-à-dire la relation *utilise* dans la table 5.1, doit être acyclique. On reviendra sur ce point dans la description de la sémantique.

Composant	Description	Représentation syntaxique
id	Identifiant de module	URI
INTERFACE D'IMPORTATION		
utilise	Liste de modules	URI
entités importées	Liste d'entités appartenant à l'interface d'export des modules importés	URI ou le mot clé `ALL` (toutes les entités de l'interface d'exportation sont importées)
DÉFINITIONS		
alignements	Liste d'alignements	URI ou correspondances localement définies, ou encore, composition de plusieurs alignements par référence à leurs URI
contenu	Définitions d'entités locales et d'axiomes	soit une ontologie externe identifiée par un URL, soit des axiomes qui peuvent utilisés les classes et propriétés importées
INTERFACE D'EXPORTATION		
entités exportées	Liste d'entités pouvant être définies soit dans le contenu de l'ontologie, soit dans les entités importées	données par des URI ou les mots clé `ALL` ou `ALL*`
commentaires	Description textuelle de ce pour quoi le module est fait	chaîne de caractères
MÉTA DONNÉES		
langage d'ontologies	Un identifiant du langage d'ontologies (par exemple OWL, RDFS, SWRL, etc.)	chaîne de caractères
langage d'alignements	Un identifiant du langage d'alignements (peut utiliser l'attribut "level" du langage de l'API d'alignement)	chaîne de caractères
autres méta-données	auteur du module, date de création, version et ainsi de suite	chaîne de caractères

TABLE 5.1 – Composants des modules, avec leur signification et leurs valeurs possibles.

Il existe une syntaxe concrète en RDF/XML découlant de cette même structure. Celle-ci n'est pas présentée ici mais une description plus détaillée se trouve dans [35].

5.3.2 Syntaxe abstraite

Pour pouvoir par la suite définir la sémantique d'un réseau de modules, on donne d'abord une syntaxe abstraite plus facile à manipuler dans les définitions d'interprétation et modèle.

Definition 5.3.1 (Module d'ontologie) *Un* module d'ontologie $\mathfrak{M} = \langle \mathrm{id}, \mathbf{M}, \mathbf{\Pi}, \mathbf{A}, O, E \rangle$ *est un tuple tel que :*

- id *est un URI identifiant le module ;*
- $\mathbf{M} = (M_i)_{i \in J}$ *est l'ensemble des modules importés définis sur un ensemble d'indices* J *;*
- $\mathbf{\Pi} = (\Pi_i)_{i \in J}$ *est l'interface d'importation pour* $\mathbf{M}$ *(c'est-à-dire autant d'ensembles d'entités qu'il y a de modules importés) ;*
- $\mathbf{A} = (A_{ij})_{i,j \in J}$ *est un ensemble d'alignements reliant les ontologies de* $\mathbf{M}$ *et*
- O *est l'ontologie contenue dans le module, définissant des termes et des axiomes locaux*

pouvant utiliser des termes importés ;
– *E est l'interface d'exportation de $\mathfrak{M}$.*

Un *module de base* est un module qui n'importe pas d'autres modules. Il correspond à une ontologie isolée que l'on "encapsule" dans un module en rendant certains de ses termes accessibles via l'interface d'exportation. De façon abstraite, un module de base s'écrit $\langle \mathrm{id}, \emptyset, \emptyset, \emptyset, O, E\rangle$.

En ce qui concerne les alignements utilisés pour relier les modules importés entre eux, ils suivent la syntaxe abstraite générique introduite au chapitre 2. Leur représentation concrète peut être différente d'une application à l'autre, mais si l'on utilise le format d'alignement proposé par [30], le système de modules bénéficie des outils associés à l'API d'alignement. Notamment, ce format intègre le langage expressif décrit dans le chapitre 6. L'ontologie contenue dans le module peut être écrite dans n'importe quel langage possédant une sémantique formelle du type de celles décrites dans le chapitre 2.

5.4 La sémantique des modules

Dans cette section, nous allons voir comment le formalisme du chapitre 4 peut être appliqué à ce langage de module pour donner une sémantique aux ontologies modulaires. Tout comme au chapitre précédent, la sémantique est présentée de façon très générale et pour une logique concrète il conviendra de fixer les logiques locales que l'on s'autorise à utiliser, le langage d'alignements pour lier les modules et une restriction sur les fonctions d'égalisations utilisables.

5.4.1 Sémantique locale

La sémantique locale est celle d'un nœud de connaissance dans un réseau de connaissance. On rappelle qu'on associe à un nœud une ontologie O écrite dans un langage L de représentation de connaissances. Un tel langage détermine la syntaxe (les termes et les formules) dans laquelle est écrite O et la sémantique avec la notion d'interprétation $\mathbf{Int}_L$ et de satisfaction $\models_L$.

Ayant supposé cela, il est alors possible de définir l'interprétation d'un module.

5.4.2 Interpréter des modules

Le principe permettant d'appliquer la sémantique du chapitre 4 est le suivant. Si un module $\mathfrak{M}$ importe les modules $(M_i)_{i\in J}$ et utilise les alignements $(A_{ij})_{i,j\in J}$ (où J est un ensemble fini) alors $\langle (M_i), (A_{ij})\rangle$ se comporte comme un réseau d'ontologies alignées et le module $\mathfrak{M}$ joue le rôle du médiateur intégrant les connaissances du réseau. Selon cette perspective, on peut schématiser la sémantique des modules, comme l'illustre la figure 5.2.

L'intuition donnée par cette figure ne suffit pas à donner une définition formelle à l'interprétation. En effet, pour donner une interprétation d'un réseau d'ontologies alignées, il faut disposer de la sémantique des nœuds. Or, ici, on cherche justement à définir la sémantique des modules. On pourrait par exemple utiliser, pour les modules importés, l'interprétation de l'ontologie contenue dedans. Mais alors, on ne profiterait pas de la connaissance importée par ces modules. Cette approche ne permettrait pas de faire transiter la connaissance à plus haut niveau.

Pour utiliser toute la chaine d'importation, on utilisera donc une définition *récursive* des interprétations. Cette définition suppose qu'il n'y a pas de cycles dans la chaine d'importation

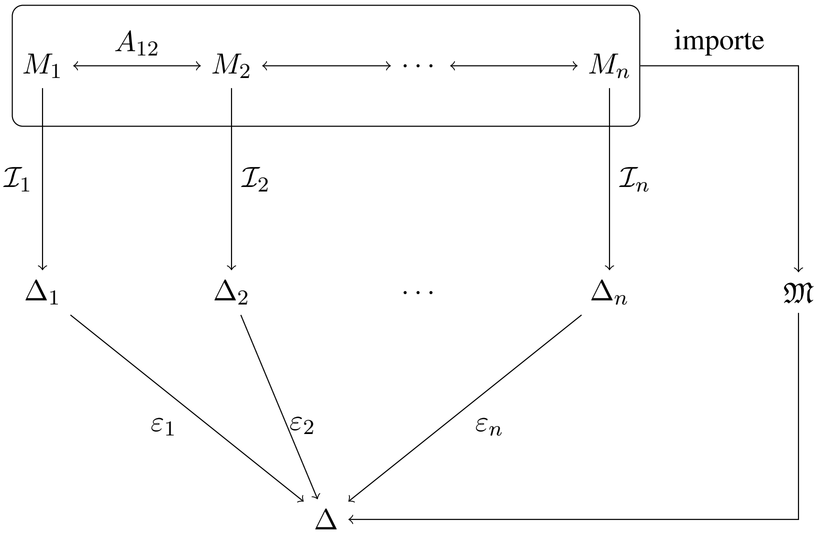


FIGURE 5.2 – Les modules importés sont interprétés localement, indépendamment les uns des autres. Le module qui les importe a son propre domaine d'interprétation, qui correspond alors au domaine global du réseau d'ontologies formé par son interface d'importation.

et donc la succession d'importation mène toujours à un module de base sans import. Étant donnée l'approche par intégration illustrée dans la figure 5.2, cette supposition est naturelle. La détection des cycles peut être facilement vérifiée syntaxiquement. Cette contrainte est d'ailleurs toujours imposée en ingénierie des logiciels, c'est pourquoi il ne s'agit pas d'une forte limitation. En effet, lorsqu'un nouveau module est conçu, il est normal qu'il importe des modules existants.

Definition 5.4.1 (Interprétation d'un module de base) *Soit $\mathfrak{M} = \langle \mathrm{id}, \emptyset, \emptyset, \emptyset, O, E\rangle$ un module de base. Une interprétation de $\mathfrak{M}$ est une interprétation locale $\mathcal{I}$ de l'ontologie O dans le contenu de $\mathfrak{M}$, avec un domaine d'interprétation Δ.*

Ceci constitue le cas de base de la définition récursive. Le cas général est alors celui-ci.

Definition 5.4.2 (Interprétation de module) *Soit $\mathfrak{M} = \langle \mathrm{id}, \mathbf{M}, \mathbf{\Pi}, \mathbf{A}, O, E\rangle$ un module. Une interprétation de $\mathfrak{M}$ est un triplet $\mathfrak{I} = \langle \mathcal{I}, (\mathcal{I}_m)_{m\in\mathbf{M}}, \varepsilon\rangle$ tel que :*
- *$\mathcal{I}$ est une interprétation locale de l'ontologie O contenue dans $\mathfrak{M}$, sur le domaine d'interprétation Δ. Δ est aussi appelé le* domaine d'interprétation *du module $\mathfrak{M}$;*
- *pour tout module importé $m \in \mathbf{M}$, $\mathcal{I}_m$ est une* interprétation de module *de m sur un domaine d'interprétation Δ_m ;*
- *ε est une fonction d'égalisation pour les interprétations $(\mathcal{I}_m)_{m\in M}$, ayant pour domaine global Δ (et compatible avec la logique distribuée concrète).*

Il faut noter que le domaine Δ est à la fois domaine d'interprétation de $\mathcal{I}$ et domaine global pour ε. Il faut donc que la sémantique locale du module et la sémantique du langage d'alignements soient compatibles. On peut s'assurer de cela en utilisant les fonctionnalités de l'API d'alignement. En effet, l'API offre des primitives pour produire des axiomes d'un langage d'ontologies à partir d'un alignement. Il suffit alors de produire les axiomes dans le langage du contenu du module.

Pour qu'une interprétation satisfasse un module, elle doit vérifier quatre conditions.

1. l'interprétation locale doit être un modèle du contenu du module, c'est-à-dire que tous les axiomes locaux doivent être satisfaits ;
2. les modules importés doivent être satisfaits par leurs interprétations respectives ;
3. les alignements entre les modules importés doivent être satisfaits par les paires d'interprétations respectives ;
4. enfin, l'interprétation locale et les interprétations de modules importés doivent s'accorder sur l'interprétation des interfaces de $\mathbf{\Pi}$.

Le quatrième point est un peu ambigu. La notion d'"accord" entre les interprétations doit être précisée. On peut remarquer que les termes importés ont une interprétation locale dans le domaine d'interprétation du module considéré. Or, dans ce domaine, les termes importés ont aussi une interprétation par le biais de la fonction d'égalisation composée avec l'interprétation locale du module importé.

L'accord entre l'interprétation locale et les interprétations de modules importés signifie simplement que les termes importés sont interprétés localement identiquement à leur interprétation par voie de fonction d'égalisation. Formellement, la notion de modèle d'un module se définit alors ainsi :

Definition 5.4.3 (Modèle d'un module) *Soit* $\mathfrak{M} = \langle \mathrm{id}, \mathbf{M}, \mathbf{\Pi}, \mathbf{A}, O, E\rangle$ *un module et* $\mathfrak{I} = \langle \mathcal{I}, (\mathcal{I}_m)_{m\in\mathbf{M}}, \boldsymbol{\varepsilon}\rangle$ *une interprétation de* $\mathfrak{M}$. $\mathfrak{I}$ *est un* modèle *de* $\mathfrak{M}$ *(noté* $\mathfrak{I} \models \mathfrak{M}$*) si et seulement si :*
- $\mathcal{I}$ *est un modèle de* O *(c'est-à-dire que le contenu local de* $\mathfrak{M}$ *est satisfait) ;*
- *pour chaque module importé* $m \in \mathbf{M}$, $\mathcal{I}_m \models m$ *(c'est-à-dire que chaque module importé est localement satisfait).*
- *pour chaque paire de modules* $m, m' \in \mathbf{M}$, $\mathcal{I}_m, \mathcal{I}_{m'} \models_{\boldsymbol{\varepsilon}} A_{m,m'}$ *(c'est-à-dire tous les alignements sont satisfaits) ;*
- *pour chaque élément* e_m *de l'interface d'importation* I_m, $\boldsymbol{\varepsilon}_m(e_m^{\mathcal{I}_m}) = e_m^{\mathcal{I}}$ *(c'est-à-dire les modules importés et qui importent s'accordent sur l'interprétation de l'interface).*

L'ensemble des modèles d'un module $\mathfrak{M}$ *est écrit* $\mathrm{Mod}(\mathfrak{M})$.

Encore une fois, le langage de modules définit une logique en donnant une notion d'interprétation et de satisfaction, de la même façon qu'un langage quelconque de représentation de connaissances, comme on l'a vu au chapitre 2. Ainsi, on peut à nouveau utiliser une ontologie modulaire dans ce format au sein d'un réseau d'ontologies alignées et, par exemple, le mettre en relation avec des ontologies non modulaires ou décrites selon une autre logique. Aussi, de la notion de modèle découle immédiatement la définition de conséquences sémantiques d'un module.

Definition 5.4.4 (Conséquences d'un module) *Soit* $\mathfrak{M} = \langle \mathrm{id}, \mathbf{M}, \mathbf{\Pi}, \mathbf{A}, O, E\rangle$ *un module. Soit* δ *une formule construite sur la signature du contenu de* $\mathfrak{M}$ *(qui inclut les interfaces d'importation de* $\mathbf{\Pi}$*).* δ *est une conséquence de* $\mathfrak{M}$*, noté* $\mathfrak{M} \models \delta$ *si et seulement si pour tout modèle* $\langle \mathcal{I}, (\mathcal{I}_m)_{m\in\mathbf{M}}, \boldsymbol{\varepsilon}\rangle$ *de* $\mathfrak{M}$, $\mathcal{I} \models \delta$.

Évidemment, si une formule est une conséquence de l'ontologie contenue dans le module, alors c'est une conséquence du module lui-même. [18] En outre, il est souhaitable de retirer des connaissances des modules importés. Cependant, si quelque chose est vrai au sujet d'un concept C dans un module, ce n'est pas forcément vrai dans un autre module qui importe C.

On peut d'ailleurs comparer cette sémantique avec une sémantique classique qui traite toutes les ontologies et les alignements au même niveau. Le module résultant est alors une

18. À noter que seules les conséquences liées aux termes exportés sont utiles à un module externe qui les importe.

simple ontologie. Dans ce cas, les modèles de cette ontologie peuvent être convertis en modèles de l'ontologie modulaire dans la sémantique distribuée. En effet, si l'on dispose d'un modèle $\mathcal{I}$ de l'ontologie en logique classique, alors c'est un modèle pour toute sous-ontologie, en particulier pour tous les modules qui la composent. En ajoutant à cela la fonction d'égalisation égale à l'identité du domaine d'interprétation de $\mathcal{I}$, on peut en déduire que $\langle \mathcal{I}, (\mathcal{I})_{m \in M}), \mathrm{id} \rangle$ est un modèle distribué du module. Toutefois, il faut que la logique distribuée concrète utilisée accepte l'identité comme fonction d'égalisation.

En revanche, le contraire n'est pas toujours vrai : un modèle distribué ne permet pas forcément de construire un modèle en logique classique. Un exemple sera donné pour la logique distribuée concrète IDDL au chapitre 8.

Deux raisons expliquent ce phénomène. D'une part, les domaines d'interprétation des modules importés ne sont pas les mêmes que celui du module qui les importe. Par conséquent, certaines formules, dont l'interprétation dépend du domaine, ne peuvent être interprétées de la même manière dans le module importé et dans le module qui l'importe. Notamment, la négation ne se transmet pas par l'importation. Deuxièmement, les fonctions d'égalisation peuvent transformer la structure des interprétations, en associant des interprétations différentes aux termes.

5.4.3 Raisonner avec un réseau de modules hétérogènes

Le raisonnement sur des ontologies modulaires renvoie au besoin de procédures de déduction pour des réseaux d'ontologies alignées, conformément à la sémantique définie au chapitre 4. Le chapitre 8 définit une telle procédure, dans un cadre plus restreint. Mais la modularité diffère un peu du raisonnement sur un simple réseau d'ontologies alignées, car en plus des ontologies et des alignements, il faut pouvoir utiliser les liens d'importation dont la sémantique n'est pas identique aux liens exprimés par les alignements d'ontologies. Ceci n'est pas pris en compte dans le chapitre 8 et nécessiterait une étude complémentaire.

En outre, le raisonnement sur les ontologies modulaires pose des problèmes particuliers, car il implique que toute la chaîne d'importation est accessible au moment du raisonnement. Pour éviter cela, il est intéressant de compiler les connaissances des modules importés pour les intégrer directement dans le module qui les importe (voir par exemple [19, 70]). De cette manière, il n'est plus utile que d'accéder au réseau composé des modules compilés (qui deviennent alors des modules de base) et des alignements (eux aussi potentiellement compilés). Néanmoins, tous les langages ne permettent pas de représenter en une seule ontologie les connaissances d'un ensemble de modules interconnectés.

Enfin, il convient de revenir à la discussion entamée au chapitre 4, dans la section 4.3.4, au sujet des fonctions d'égalisation. Il y était mentionné qu'une sémantique concrète pour des réseaux d'ontologies alignées peut poser des contraintes sur les fonctions d'égalisation. Dans le cas des ontologies modulaires, c'est essentiel, puisque l'on souhaite profiter au maximum des connaissances déjà décrites dans les modules importés. Néanmoins, pour préserver la tolérance vis-à-vis de l'hétérogénéité des modules, on souhaite justement que certaines connaissances ne soient que partiellement transférées d'un module importé à un autre qui l'importe. Le formalisme IDDL décrit dans le chapitre 8 offre un bon compromis entre robustesse à l'hétérogénéité et transfert de connaissances.

5.5 Bilan

La modularité est un aspect important des ontologies sur le Web sémantique, dont le soutien par le standard OWL fait particulièrement défaut. Un grand nombre de travaux se concentre sur le partitionnement et d'autres sur la définition de propriétés devant être assurées par un module (on a mentionné au chapitre 3 le cas de l'extension conservative). Les logiques distribués (DFOL, DDL, $\mathcal{E}$-connections, P-DL) sont souvent qualifiés de langages d'ontologies modulaires [25]. Mais ces travaux ne considèrent pas l'encapsulation ou la réutilisabilité ou forcent une certaine implémentation par un langage spécifique. Enfin, aucun ne distingue clairement la relation d'importation des relations sémantiques présentes dans les alignements.

Dans ce chapitre, on s'est intéressé à la modularité dans le cas de modules issus de sources variées, mis en correspondances, éventuellement, par des outils tiers. Dans cette optique, j'ai présenté un système de modules réutilisant des ontologies et des alignements existants, en les identifiants sur le Web. J'y ai associé une sémantique formelle qui exploite les sémantiques locales des ontologies, sans les perturber. Grâce à cela, les conséquences sémantiques d'un module sont clairement définies. Ainsi, cette sémantique de modules bénéficie d'une bonne tolérance à l'hétérogénéité et d'une bonne réutilisabilité des alignements.

En utilisant des alignements comme objets distincts des ontologies au sein du module, il est notamment possible d'utiliser des représentations différentes pour les correspondances et les axiomes et, par conséquent, avoir une expressivité différente pour le langage d'alignements et pour le langage d'ontologies. Le chapitre suivant décrit justement un langage très expressif pour les correspondances et fournit sa sémantique en fonction des sémantiques des ontologies alignées, conformément au formalisme du chapitre 4.

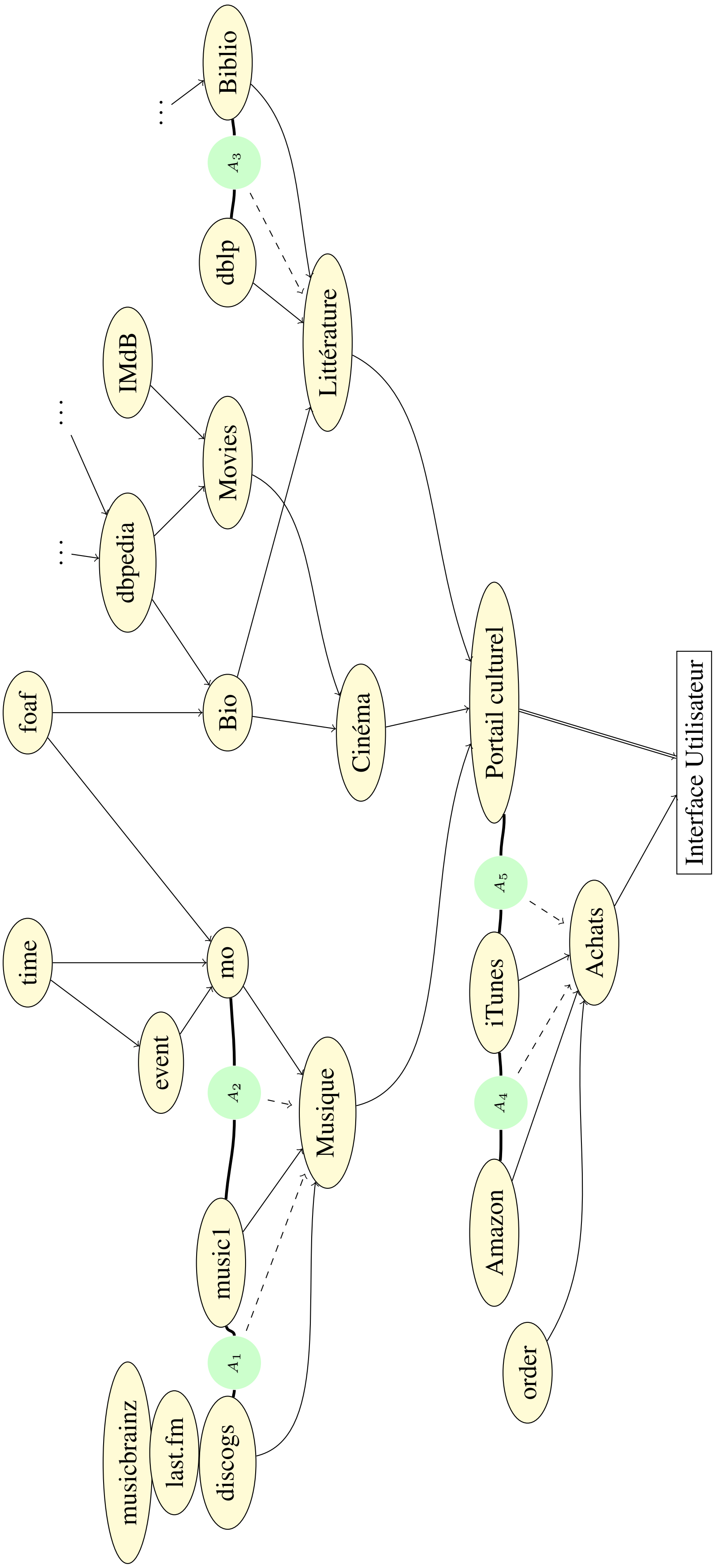


FIGURE 5.3 – Un exemple d'ontologies modulaires.

Chapitre 6

Langage d'alignements expressif

Résumé

La sémantique générale présentée au chapitre 4 permet de découpler complètement le langage de représentation local, c'est-à-dire le langage d'ontologies, et le langage de représentation des alignements. Par conséquent, l'expressivité du langage d'alignements peut être arbitrairement élevée, quelle que soit la logique utilisée localement. Ce chapitre décrit comment un langage d'alignements d'ontologies à la fois expressif et indépendant des langages d'ontologie peut être construit. Ce langage fournit des constructeurs similaires à ceux des logiques de description, mais que les logiques locales ne prennent pas nécessairement en charge. Une syntaxe abstraite est proposée, à laquelle on associe une sémantique formelle en théorie des modèles venant instancier la sémantique générale présentée dans le chapitre 4.

Sommaire

6.1 Introduction

Comme on l'a vu précédemment, la sémantique des réseaux d'ontologies alignées repose sur la sémantique des ontologies, au niveau local, ainsi que sur la sémantique d'un langage d'alignements, au niveau global. Les outils de mise en correspondance automatique génèrent habituellement des alignements dans un langage simple, le plus souvent caractérisé simplement par un ensemble de relations binaires associant des entités nommées dans les ontologies.

Toutefois, un langage d'alignements peut aussi définir des constructeurs spécifiques inexistant dans les langages d'ontologies. Dans ce chapitre, je décris un langage d'alignements ayant à la fois la faculté de définir des correspondances complexes, tout en restant indépendant des langages d'ontologies. Ce langage a été conçu à partir du travail de François Scharffe [62], auquel j'ai apporté une sémantique formelle. Ce chapitre montre comment la sémantique générale du chapitre 4 s'applique aussi à un langage d'alignements expressif.

En plaçant l'expressivité du côté du langage d'alignements, il devient possible de déclarer des relations subtiles entre des ontologies très pauvres en termes d'expressivité. Par exemple, il peut être nécessaire de combiner différents concepts dans une simple taxonomie pour les mettre en correspondance avec des concepts d'une autre ontologie. Par ailleurs, il peut arriver que l'implémentation des ontologies ne soit pas accessible mais que l'alignement puisse être échangé. Un langage expressif et indépendant des langages d'ontologies est alors primordial. En outre, pour la transformation de requêtes ou d'instances, seules des équivalences peuvent garantir une traduction correcte et complète de l'information originale. Or, s'il n'existe pas de terme exactement équivalent, il est fort utile de pouvoir en combiner plusieurs pour retrouver l'équivalence.

Ce chapitre présente volontairement un grand nombre de constructeurs qui rendent le langage complexe. Pour des raisons pratiques, par exemple si un raisonnement correct et complet au sujet des alignements est nécessaire, il se peut que seul un sous-langage soit utilisé. Mais, si la nécessité de représenter des choses complexes prime, on peut utiliser ce langage au prix d'une déduction incomplète.

Ce chapitre se focalisera principalement sur les aspects sémantiques du langage et ne discutera pas, notamment, de l'implémentation. Ainsi, j'articule ce chapitre de la façon suivante. En premier lieu, dans la section 6.2, la nécessité d'un langage très expressif est motivée par une série d'exemples représentatifs des problèmes de conception et d'utilisation d'alignements. Puis, dans la section 6.3, un exemple de langage d'alignements très expressif est donné sous la forme d'une syntaxe abstraite et d'une sémantique en théorie des modèles. La présentation de ce langage, qui n'est qu'un exemple possible de langage d'alignements, suit en grande partie les conventions de notations utilisées en logique de description. La section 6.4 discute des caractéristiques de la sémantique et des difficultés d'un raisonnement avec un tel langage.

6.2 Utilité d'un langage d'alignements expressif

Ce langage a été défini initialement pour des raisons pragmatiques. Initialement, il s'agissait de pouvoir décrire dans une syntaxe d'échange les relations entre représentations internes de services Web. À cela, j'ai ajouté une sémantique formelle. Un langage d'alignements expressif n'a pas pour fonction d'être utilisé dans une procédure de raisonnement générique sur un réseau d'ontologies quelconque. L'objectif est de pouvoir spécifier des relations précises lorsqu'elles sont nécessaires, afin d'effectuer des traitements automatiques, notamment sur les données ou requêtes transférées d'un système à un autre. La sémantique formelle évite aussi les ambiguïtés lorsque les alignements sont échangés.

6.2.1 Exemples de correspondances

Cette section présente un bestiaire des types de correspondances que l'on souhaiterait typiquement représenter dans un alignement. Tous ces exemples s'appuient sur deux ontologies du cinéma, toutes deux ontologies disponibles sur le Web. Les deux ontologies du cinéma sont décrites en RDFS, un langage assez peu expressif. La première comprend 95 classes et 385 propriétés [1]. La seconde ontologie contient 2 classes et 53 propriétés [2]. Pour simplifier

1. `http://139.91.183.30:9090/RDF/VRP/Examples/imdb.rdf`
2. `http://www.csd.abdn.ac.uk/~ggrimnes/dev/imdb/IMDB`

les explications, on nommera la première ontologie $\texttt{Cinema}_1$ et la seconde $\texttt{Cinema}_2$ et l'on ajoutera les indices correspondants aux termes de ces ontologies.

Les deux ontologies de la musique sont décrites en OWL. La première définit 60 classes, 133 propriétés d'objets (*ObjectProperty* en OWL) et 20 propriétés de données (*DatatypeProperty* en OWL)[3]. On la nommera $\texttt{Musique}_1$. La seconde définit 109 classes, 23 propriétés d'objets et 11 propriétés de données. Celle-ci sera nommé $\texttt{Musique}_2$.

En essayant d'aligner les deux ontologies, je montrerai que de simples correspondances par appariement de termes atomiques ne sont pas suffisantes pour représenter un alignement complet des deux ontologies.

Subsomption et équivalence

Il existe une évidente équivalence entre les deux concepts principaux $\texttt{Movie}_1$ et $\texttt{Movie}_2$ existant dans les deux ontologies. Cette correspondance aurait pu facilement être découverte par un algorithme de mise en correspondance automatique, par simple observation des libellés. En revanche, les multiples propriétés de chaque concept requièrent un examen plus approfondi et parfois des relations plus complexes que de simples équivalences de termes.

La seconde classe de $\texttt{Cinema}_2$ est $\texttt{Role}_2$, que l'on peut faire correspondre exactement avec $\texttt{Role}_1$ dans $\texttt{Cinema}_1$. Aussi, bien que $\texttt{Cinema}_2$ ne la définisse pas, elle utilise la classe $\texttt{Person}$ de l'ontologie FOAF. À celle-ci on peut associer dans $\texttt{Cinema}_1$ la classe $\texttt{MoviePerson}_1$, mais les deux classes ne sont pas équivalentes : $\texttt{MoviePerson}_1$ représente l'ensemble des personnes qui participent à la création d'un film, qu'ils soient acteurs, réalisateurs, maquilleurs ou chorégraphes. Il s'agira donc d'une relation de subsomption.

Diverses équivalences de propriétés peuvent être établies, comme par exemple $\texttt{Birthdate}_1$ (respectivement $\texttt{Trivia}_1$, $\texttt{directs}_1$, $\texttt{Genre}_1$, $\texttt{role}_1$) dans $\texttt{Cinema}_1$ avec $\texttt{birth\_date}_2$ (respectivement $\texttt{trivia}_2$, $\texttt{directed}_2$, $\texttt{acted}_2$, $\texttt{genres}_2$) dans $\texttt{Cinema}_2$.

Dans un premier temps, formalisons ces correspondances en logique du premier ordre, avant de rendre explicite la syntaxe du langage d'alignements. Dans ces expressions logiques, les variables sont quantifiées sur un ensemble d'individus dont l'interprétation est supposée commune aux deux ontologies. Par exemple, $\forall x, \texttt{Movie}_1(x) \Leftrightarrow \texttt{Movie}_2(x)$ signifie que tout x classé dans $\texttt{Movie}_1$ dans la première ontologie, doit être classé dans $\texttt{Movie}_2$ dans la seconde ontologie et vice versa.

Les correspondances évoquées jusqu'à présent peuvent être transcrites facilement comme suit :

$$
\begin{aligned}
&\forall x, [\texttt{Movie}_1(x) \Leftrightarrow \texttt{Movie}_2(x)]\\
&\forall x, [\texttt{Role}_1(x) \Leftrightarrow \texttt{Role}_2(x)]\\
&\forall x, [\texttt{MoviePerson}_1(x) \Leftarrow \texttt{Person}(x)]\\
&\forall x, [\texttt{Birthday}_1(x) \Leftrightarrow \texttt{birth\_day}_2(x)]\\
&\forall x, [\texttt{Trivia}_1(x) \Leftrightarrow \texttt{trivia}_2(x)]\\
&\forall x, [\texttt{directs}_1(x) \Leftrightarrow \texttt{directed}_2(x)]\\
&\forall x, [\texttt{role}_1(x) \Leftrightarrow \texttt{acted}_2(x)]\\
&\forall x, [\texttt{Genre}_1(x) \Leftrightarrow \texttt{genres}_2(x)]
\end{aligned}
$$

3. http://musicontology.com/ et [60]

Relation inverse

Lorsque l'on décrit un domaine de connaissances, il arrive fréquemment que l'on se focalise sur des concepts privilégiés, parce qu'ils ont plus d'importance dans l'application considérée. Par exemple, dans le domaine du cinéma, on peut s'intéresser avant tout au personnel de l'industrie du film, ou bien plutôt aux caractéristiques des films. Dans le premier cas, on s'intéressera au rôle des personnes impliquées dans la conception d'un film. Par exemple "Guy Debord a écrit et réalisé *La société du spectacle*". Dans le second cas, on s'intéressera aux attributs des films. Par exemple, *Window Water Baby Moving* est un film de Stan Brakhage. Ces deux approches mènent à des relations différentes : `realise` et `est_realise_par`. Il est très fréquent que la relation inverse n'existe pas dans l'ontologie. Si l'on souhaite mettre ces deux relations en correspondance, on doit pouvoir exprimer la relation inverse au niveau de l'alignement.

Ainsi, dans l'ontologie $\texttt{Cinema}_1$, on trouve les relations $\texttt{character}_1$, $\texttt{manager\_produces}_1$, $\texttt{participates}_1$, $\texttt{makesup}_1$, $\texttt{artdepartments}_1$, etc. équivalentes à l'inverse des relations $\texttt{film}_2$, $\texttt{make\_up}_2$, $\texttt{teammember}_2$, $\texttt{art\_department}_2$, $\texttt{production\_manager}_2$, etc. respectivement dans $\texttt{Cinema}_2$. On peut remarquer que $\texttt{Cinema}_1$ est plus centré sur le personnel (et définit d'ailleurs la classe `MoviePerson`) alors que $\texttt{Cinema}_2$ se focalise sur les films (d'ailleurs, cette ontologie ne définit pas de terme propre pour les personnes).

La représentation en logique du premier ordre est la suivante :

$$\begin{gathered}
\forall x, y[\texttt{participates}_1(x,y) \Leftrightarrow \texttt{teammember}_2(y,x)] \\
\forall x, y[\texttt{artdepartments}_1(x,y) \Leftrightarrow \texttt{art_department}_2(y,x)] \\
\forall x, y[\texttt{makesup}_1(x,y) \Leftarrow \texttt{make_up}(y,x)] \\
\forall x, y[\texttt{manager_produces}_1(x,y) \Leftrightarrow \texttt{production_manager}_2(y,x)] \\
\forall x, y[\texttt{character}_1(x,y) \Leftrightarrow \texttt{film}_2(y,x)]
\end{gathered}$$

Restriction de valeurs

Dans l'exemple suivant, nous allons voir qu'il est parfois nécessaire de restreindre l'ensemble d'entités entrant dans le cadre d'une correspondance particulière. Dans l'ontologie $\texttt{Cinema}_1$ la classe $\texttt{Drama}_1$ représente les films dramatiques. Mais dans l'ontologie $\texttt{Cinema}_2$, il n'existe pas d'équivalent. On pourrait se contenter d'indiquer que $\texttt{Drama}_1$ est une sous-classe de $\texttt{Movie}_2$, mais comme indiqué en préambule, on recherche de préférence des équivalences. Or, grâce à la propriété `genres` de l'ontologie $\texttt{Cinema}_2$, on peut identifier les films dramatiques. Pour définir une correspondance entre la classe des films dramatiques et ceux dont le genre est "drama", il faut pouvoir restreindre la classe $\texttt{Movie}_2$ aux seules instances qui ont leur `genres` égale à "drama", ainsi que le montre la figure 6.1.

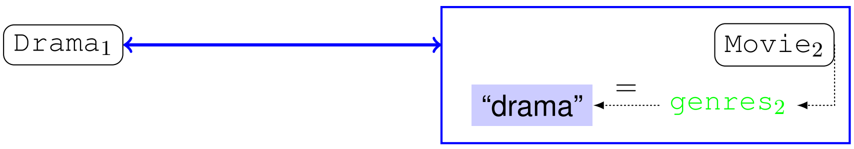


FIGURE 6.1 – Correspondance entre les films dramatiques et les `Drama`.

En calcul des prédicats, ceci peut être transcrit par :

$$\forall x, [\texttt{Drama}_1(x) \Leftrightarrow \texttt{Movie}_2(x) \wedge \texttt{genres}_2(x, < \#Drama >)]$$

où $< \#Drama >$ dénote un individu particulier de l'ontologie $\texttt{Cinema}_2$.

Ici, on considère que $< \#Drama >$ est un individu car l'ontologie $\mathtt{Cinema}_2$ ne précise pas de quel type doivent être les éléments liés à la relation $\mathtt{genres}_2$, tandis que dans $\mathtt{Cinema}_1$, la propriété $\mathtt{Genre}_1$ relie des films à des chaînes de caractères uniquement.

Conversion de types

Un des attributs des films est leur durée. Dans $\mathtt{Cinema}_1$, celle-ci est représentée par la propriété $\mathtt{Runtime}_1$ dont la valeur est un entier, sans précision sur sa véritable signification. Il existe par ailleurs, dans XML Schema, un type de données appelé "duration" permettant de représenter une durée dans le temps. Même si, conceptuellement, les propriétés sont équivalentes, lors de la transformation de données par exemple, il faut que les types de données soient convertis.

Pour cela, la correspondance indique quelle relation est entretenue entre les valeurs d'instances, c'est-à-dire quelle transformation est requise pour correctement associer un entier à son équivalent en durée et réciproquement.

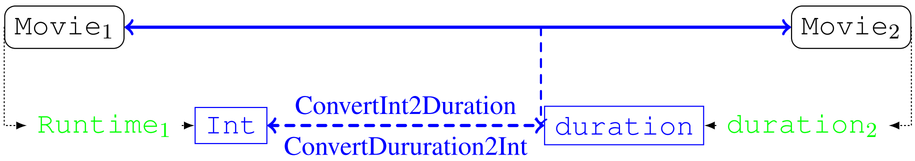


FIGURE 6.2 – Correspondance entre durées.

Cette correspondance peut être représentée ainsi en logique des prédicats :

$$\forall x, [(\mathtt{Movie}_1(x) \wedge \exists y : Int; \mathtt{Runtime}_1(x,y) \Rightarrow \mathtt{duration}_2(x, \mathrm{ConvertInt2Duration}(y))$$
$$\wedge(\mathtt{Movie}_2 \wedge \exists y : Duration; \mathtt{duration}_2(x,y) \Rightarrow \mathtt{Runtime}_1(x, ConvertDururation2Int(y))]$$

Contraintes sur des propriétés

Une définition simpliste du cinéma d'auteur indique qu'un film d'auteur est un film écrit par le réalisateur. Pour identifier une telle catégorie de film, il faut pouvoir poser une contrainte sur deux propriétés, à savoir $\mathtt{director}_2$ et $\mathtt{writer}_2$, en les comparant. En effet, dans $\mathtt{Cinema}_2$, $\mathtt{director}_2$ (respectivement $\mathtt{writer}_2$) relie un film à son (ou ses) réalisateur(s) (respectivement à son (ou ses) auteurs).

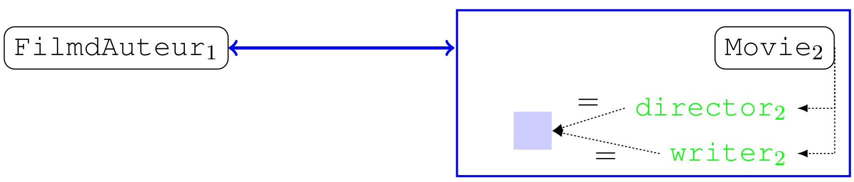


FIGURE 6.3 – Correspondance avec une contrainte sur des relations.

La transcription en logique du premier ordre est alors :

$$\forall x, [\mathtt{FilmdAuteur}_1(x) \Leftrightarrow (\mathtt{Movie}_2(x) \wedge \exists y; \mathtt{director}_2(x,y)) \wedge \mathtt{writer}_2(x,y))]$$

Remarque 6.2.1 *Ici, on compare les deux propriétés avec l'égalité (de même que dans la section 6.2.1), mais on pourrait aussi utiliser une inégalité. Par exemple, pour identifier les* *longs métrages dans une ontologie qui n'a pas ce concept, on peut comparer la propriété "durée" à une certaine valeur. Cette valeur peut être une constante ou bien une valeur définie ailleurs dans l'ontologie. On verra par la suite comment traiter ces exemples.*

Composition de relations

On peut supposer qu'une ontologie du travail, plus générale que $\mathtt{Cinema}_1$, est mise en correspondance avec $\mathtt{Cinema}_1$. On imagine que cette ontologie contient la propriété `worksWith` reliant des personnes qui ont travaillé ensemble. On peut alors supposer que toute paire de personnes ayant participé à un même film ont de ce fait travaillé ensemble. Par exemple, David réalise *Eraserhead* et Alan est ingénieur du son sur le film *Eraserhead*. On en déduit que David et Alan travaille ensemble. Ceci peut être représenté en utilisant une composition de relation.

En composant la relation $\mathtt{participates}_1$ avec son inverse, on trouve toutes les paires de personnes ayant participé au même film. Cependant, il subsiste un léger problème. La relation $\mathtt{participates}_1 \circ \mathtt{participates}_1^{-1}$ est une relation réflexive. Or, dans le langage courant, on ne considère pas qu'une personne travaille avec elle-même. Par conséquent, pour être conforme à la terminologie commune, il faut retirer l'identité de la relation. Ainsi, on obtient la formule du premier ordre suivante :

$$\forall x, y[\exists y; \mathtt{participate}_1(x, z) \wedge \mathtt{participate}_1(y, z) \wedge x \neq y \Leftrightarrow (\mathtt{worksWith}(x, y))]$$

Les correspondances présentées dans cette section sont issues de cas réels qui ne sont en rien extraordinaire. On peut s'attendre à les voir apparaitre fréquemment dans le cadre de la médiation entre ontologies. Même si ces correspondances peuvent facilement être exprimées en langue naturelle, elles requièrent l'utilisation d'un formalisme expressif pour pouvoir les représenter complètement.

6.2.2 Synthèse

Si le découplage du langage d'alignements d'une part et des langages d'ontologies d'autre part est essentiel à la médiation et l'intégration d'ontologies hétérogènes, il impose des besoins supplémentaires en termes d'expressivité : tous les exemples ci-avant font appel à des expressions complexes que peu de langages supportent, comme on peut s'en convaincre en observant les formats d'alignements décrits en section 2.3.2. En outre, la complexité des correspondances ne dépend en rien de la complexité du langage d'ontologies.

Le langage décrit ici reprend en grande partie celui proposé initialement dans le rapport D2.2.10 de Knowledge Web [33] (dont la syntaxe est due essentiellement à François Scharffe) mais en modifie certains aspects. L'objectif n'est pas de fournir une spécification complète et fonctionnelle, mais plutôt de montrer comment intégrer dans la sémantique distribuée un langage d'alignements expressif, riche de nombreux constructeurs.

On présente d'abord la syntaxe abstraite en montrant qu'elle exprime bien les correspondances données en exemple, puis la sémantique formelle illustrée par des exemples.

6.3 Syntaxe abstraite et sémantique

Cette section décrit un langage expressif de spécification d'alignements d'ontologies. Puisqu'il s'agit d'une version modifiée du langage décrit dans [33], je mettrai en évidence les modifications effectuées.

Dans ce qui suit, la syntaxe abstraite est d'abord présentée, permettant d'exprimer les correspondances indiquées en exemples dans la section précédente. Ensuite, on fournira la sémantique formelle, qui fait le lien entre la sémantique du langage et celle des ontologies.

6.3.1 Syntaxe abstraite

Le langage défini ici a une très forte expressivité afin de décrire les correspondances les plus précises. Il serait toutefois naturel de réduire son étendue à certains constructeurs, afin d'en réduire la complexité pour des besoins d'efficacité algorithmique. Je reviendrai sur ce point dans la section 6.4. La syntaxe reprend en grande partie les conventions utilisées dans les logiques de description [8], bien que parfois elle s'écarte de ce formalisme pour plus d'ouverture ou des raisons esthétiques (par exemple, en évitant les doubles exposants).

Les alignements forment un ensemble de correspondances (X) satisfaisant la grammaire suivante. Les correspondances (X) elles-mêmes correspondent à des relations entre entités (E). Ici, ces relations sont restreintes à l'équivalence ($\equiv$), la subsomption ($\sqsubseteq$ ou $\sqsupseteq$) ainsi que l'appartenance ($\in$ ou $\ni$) d'individu (u) à une classe (C) :

$$X \coloneqq E \equiv E \mid E \sqsubseteq E \mid E \sqsupseteq E \tag{6.1}$$

$$\mid u \in C \mid C \ni u \tag{6.2}$$

Le fait d'avoir un ensemble de relations sémantiques connues ($\equiv$, $\sqsubseteq$, $\sqsupseteq$, $\in$, $\ni$) permet de préciser la sémantique de ces relations. Cependant, si l'on considère des correspondances de façon plus générale, telles qu'on peut les définir avec le format d'alignement de l'API d'alignement [30], il serait possible d'étendre l'ensemble des relations, de sorte que la syntaxe générale serait plutôt :

$$X :\coloneqq E\ rel\ E \tag{6.3}$$

Cela permet par exemple d'établir des relations comme $parentDe$ entre individus.

Les entités que l'on trouve dans les correspondances sont des classes (C), relations (R), propriétés (P) et instances ou individus (u) :

$$E :\coloneqq C \mid R \mid P \mid u \tag{6.4}$$

Cette séparation en quatre types d'entités ontologiques est justifiée parce que les définitions d'expressions varient légèrement selon ces types et leur sémantique diffère quelque peu. Les *expressions de classe* représentent des classes ou des ensembles de classes liés entre eux via des opérateurs. Les *expressions de propriété* représentent des relations dont le codomaine (ou la portée — *range*) est un type de données. Les *relations* se situent entre deux classes. Par conséquent, la différence entre Relation et Propriété correspond à la différence entre *ObjectProperty* et *DatatypeProperty* en OWL. Enfin, les instances de classes sont mises en correspondance via des *expressions d'instance*. Pour rendre la distinction entre Relations et Propriété plus explicite, on utilisera parfois les variantes “relation de classe” et “propriété de données”. La structure des expressions varie en fonction du type d'expressions. Par exemple, les expressions de Propriété peuvent avoir des restrictions de valeurs, alors que les restrictions d'expressions de Classe sont plutôt rattachées à leurs instances ou bien aux valeurs des attributs de cette classe. Le constructeur principal permet de donner directement l'URI d'une entité pour construire une expression utilisant des constructeurs. Les constructeurs indiquent comment grouper les ensembles d'entités donnés dans l'expression et sont interprétés en termes de théorie des modèles dans la section 6.3.2. Les constructeurs d'expressions peuvent être composés de sous-expressions. Les conditions sur les expressions de classe, propriété et relation restreignent l'étendue de l'ensemble d'entités construites dans l'expression.

Les alignements mettent en relations des entités provenant de langages d'ontologies tels que OWL, F-logics ou autre. On considèrera que ces entités sont typées. Voici les différents types qui seront considérés :

- Classes : c ;
- Relations : r ;
- Propriétés : p ;
- Instances : i.

Les valeurs et types de données doivent aussi faire partie des entités à considérer :

- Data type : d ;
- Data value : v ;

Ils sont cependant considérés comme extérieurs au langage d'alignements.

À partir de ces entités, le langage d'alignements permet de créer des expressions plus complexes à l'aide de constructeurs. Ces constructeurs sont des expressions classiques d'algèbre booléenne (*et*, *ou*, *non*) de même que des contraintes sur les expressions de classe ou de valeur. Ces contraintes étant exprimées à l'aide d'opérateurs externes, elles ne sont pas définies comme en logique de description, donc il ne faut pas confondre $\exists K$ dans cette syntaxe abstraite et $\exists R.C$ en logique de description, même si ces deux expressions dénotent une classe. On verra par la suite quelle forme peuvent prendre les contraintes K.

$$C ::= c \tag{6.5}$$
$$\mid C \sqcup C \mid C \sqcap C \mid \neg C \tag{6.6}$$
$$\mid \exists K \tag{6.7}$$

On peut d'ores et déjà exprimer les correspondances de subsomption ou équivalence entre classes, illustrées par les exemples de la section 6.2.1.

Les expressions de *relation* sont aussi créées à partir des relations primitives, leur combinaison booléenne et des contraintes sur leur domaine ou portée, ainsi que par des transformations via une fonction, ou encore par leur inverse, leur clôture symétrique, transitive ou réflexive. Aux constructeurs du langage de [33] est ajoutée la composition de relations $R \circ R$ et la relation self dénote la relation identité.

$$R ::= r \tag{6.8}$$
$$\mid \mathrm{self} \tag{6.9}$$
$$\mid R \sqcup R \mid R \sqcap R \mid \neg R \mid R \circ R \tag{6.10}$$
$$\mid \mathrm{dom}(C) \mid \mathrm{range}(C) \tag{6.11}$$
$$\mid \mathrm{inv}(R) \tag{6.12}$$
$$\mid \mathrm{sym}(R) \mid \mathrm{trans}(R) \mid \mathrm{refl}(R) \tag{6.13}$$

Il apparait essentiel d'ajouter la composition car elle permet d'exprimer le type de correspondances de la section 6.2.1, qui ne sont pas exprimables avec le langage de [33]. Par exemple, la correspondance $\mathit{parent} \circ \mathit{frere} \equiv \mathit{oncle}$ n'a pas d'équivalent dans le langage de [33], qui ne permet que d'affirmer que la classe des oncles est équivalente à la classe des personnes qui sont frères d'un parent.

Ce manque est en contradiction avec l'objectif d'expressivité maximale de ce langage, c'est pourquoi j'ajoute cet opérateur. Par ailleurs, le constructeur self est introduit pour indiquer l'identité des individus (c'est-à-dire la relation contenant tous les couples (x, x)). Ainsi, la correspondance de la section 6.2.1 s'écrit $\mathtt{participates}_1 \circ \mathrm{inv}(\mathtt{participates}) \sqcap \neg\mathrm{self} \sqsubseteq$

`worksWith`. L'intersection avec la négation de l'identité permet de se débarasser de la réflexivité de la relation.

Les expressions de *propriété* sont semblables mais moins complexes puisque certains constructeurs des relations ne peuvent s'appliquer aux propriétés (symétrie, transitivité, inverse et réflexivité) :

$$P ::= p \tag{6.14}$$

$$\mid P \sqcup P \mid P \sqcap P \mid \neg P \tag{6.15}$$

$$\mid R.P \tag{6.16}$$

$$\mid \mathrm{dom}(C) \mid \mathrm{range}(d) \tag{6.17}$$

Le constructeur $R.P$ correspond à la composition d'une relation R et d'une propriété P. La composition ne peut se faire que dans ce sens car sinon les valeurs des types de données et les instances de classes seraient mêlées. Or, c'est en séparant les instances et les valeurs de données que les opérations sur les types de données peuvent être convenablement intégrées à la sémantique de ce langage. Ce constructeur est un autre ajout par rapport au langage de [33].

[33] introduit la notion de chemin qui, d'une certaine manière, vient partiellement compenser l'absence de la composition. Elle correspond à une séquence de relations (éventuellement vide) et (éventuellement) finissant par une propriété. Dans le présent chapitre, l'ajout de la composition de relations et de self rend obsolète cette notion. On peut donc remplacer toute occurence des chemins par une relation ou une propriété.

Il faut aussi noter qu'ici, la propriété P dans la composition de relation et propriété, ainsi que la relation R dans la composition de relations ne sont pas restreintes, contrairement à [33] qui indique que celles-ci doivent être atomiques dans l'expression d'un chemin. Cette restriction n'est pas justifiée dans le rapport D2.2.10, mais elle permet en fait de ne pas avoir d'interdépendance entre les constructeurs de relations et les constructeurs de classes. Cette caractéristique rend la présentation du langage plus simple, au détriment d'une limitation en terme d'expressivité.

Du fait de cette interdépendance des expressions de classe et de relation, il faudra modifier significativement la présentation de la sémantique, mais je m'attacherai à ne donner que les points clés car les détails techniques sont donnés dans un tableau en annexe B et des indications supplémentaires se trouvent dans [33].

Les valeurs que l'on trouve dans les contraintes de classe peuvent être des valeurs explicites de type de données (un entier, une date, etc.), des instances particulières, des valeurs associées à une propriété ou relation, ou l'évaluation d'une opération *transf* appliquée à un ensemble de valeurs :

$$V ::= v \mid i \mid R \mid P \mid \mathrm{transf}(V*) \tag{6.18}$$

Les opérations que l'on peut appliquer aux valeurs sont utiles pour exprimer des contraintes sur des domaines concrets. C'est ce qui doit être fait pour convertir les entiers en date, dans la correspondance de la section 6.2.1 dans laquelle les entiers sont convertis en date. Ces fonctions (qui renvoient des valeurs de type de données) et les comparateurs de valeurs (qui renvoient un booléen) sont des opérations externes dont la sémantique est supposée connue et ne sera pas décrite dans ce chapitre. On peut tout de même indiquer que les opérateurs et comparateurs de XQuery [18] conviennent à ce cadre d'utilisation. Pour information, l'annexe C décrit ces opérateurs et comparateurs.

On peut alors écrire des expressions comme :

$$\mathit{length}(\mathit{collection}) \text{ less-than } \mathit{multiply}(\mathit{integer}, \mathit{integer})$$

Une fois ces valeurs disponibles, des prédicats peuvent être évalués pour comparer ces valeurs. Ceci forme la base des contraintes pouvant s'appliquer à des propriétés ou relations pour comparer leurs valeurs, leurs types de données ou leurs cardinalités :

(6.19)	$K ::= R\ cp\ V \mid P\ cp\ V$	(Restriction de valeur)
(6.20)	$\mid P\ cp\ d$	(Restriction de type)
(6.21)	$\mid \lvert R\rvert\ cp\ n \mid \lvert P\rvert\ cp\ n$	(Restriction de cardinalité)

On peut utiliser ces différentes restrictions pour former des expressions de la forme $String$ contains $String$, $age \in [12\ 16]$ ou $|child| \geq 3$. Toutes ces restrictions permettent d'exprimer plus étroitement les relations entre concepts ou propriétés de différentes ontologies.

Exemple 6.3.1 *Dans l'ontologie* $\texttt{Cinema}_1$*,* $\texttt{Birthcountry}_1$ *est une relation entre les personnes et leur lieu de naissance. On peut par exemple définir la classe des réalisateurs ayant mis en scène un film écrit par une personne de même nationalité :*

$$\texttt{MoviePerson}_1 \sqcap \exists \left((((\texttt{directs}_1 \circ \texttt{writes}_1^-) \sqcap \neg\text{self}) \circ \texttt{Birthcountry}_1) = \texttt{Birthcountry}_1 \right)$$

Ici, on compare deux pays de naissance. D'un côté, le pays de naissance d'un individu x qui est aussi une $\texttt{MoviePerson}_1$*. De l'autre, un individu y qui a écrit quelque chose qui a été réalisé par l'individu x.*

6.3.2 Principes pour la sémantique

La difficulté de définir l'interprétation des expressions d'entité provient du fait que les expressions sont, à la base, construites à partir d'entités d'ontologies, qui ont leur propre interprétation dans le langage d'ontologies. Par ailleurs, la sémantique des langages des deux ontologies alignées peut différer grandement. Aussi, l'interprétation des constructeurs est spécifique au langage d'alignements.

C'est pourquoi la sémantique du langage d'alignements doit se conformer au modèle général donné au chapitre 4. Plus précisément, il faut dans un premier temps donner une sémantique aux alignements indépendamment des ontologies alignées. Cela permet d'interpréter les expressions complexes du langage sans faire référence aux interprétations locales des ontologies. Ensuite, cette sémantique est reliée à la sémantique des ontologies en faisant correspondre l'interprétation des termes des ontologies au niveau local et global, grâce à la fonction d'égalisation. La figure 6.4 illustre la situation.

Dans la section suivante, seule une partie de la sémantique est présentée, afin de concrétiser le principe évoqué, mais sans être exhaustif. Pour plus de détail, on pourra consulter le tableau récapitulatif de la sémantique du langage, présenté en annexe B.

6.3.3 Interpréter des expressions

Les sous-sections suivantes définissent les règles régissant l'interprétation des constructeurs et des termes intervenant dans les expressions. La première sous-section n'appartient pas, à proprement parler, à la sémantique du langage défini ici. Les types de données, les opérateurs, les comparateurs et les transformations de données sont gérés par l'application et l'on supposera que tout cela est déjà défini et peut être utilisé dans les définitions suivantes. La sémantique est donnée en se conformant à la syntaxe abstraite, mais l'on ne donnera qu'une partie des définitions puisque l'objectif est d'expliquer le principe plutôt que de définir tout un langage.

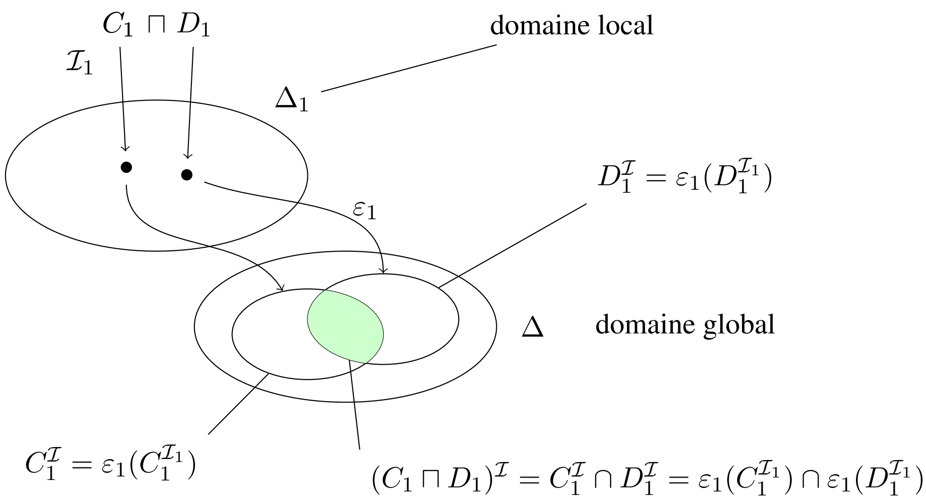


FIGURE 6.4 – Interprétation des expressions complexes.

Ici, on se contentera de définir l'interprétation des contraintes $\exists K$, qui font la richesse de ce langage. Pour cela, on introduira les interprétations des types de données et leurs opérations, nécessaire à l'interprétation des valeurs dans les contraintes.

Types de données, opérateurs, comparateurs et transformations

Les types de données sont interprétés de la même façon qu'en RDF et OWL. Différentes applications peuvent apporter leurs propres types de données et leur gestion est indépendante de la sémantique. Ces types de données sont identifiés dans la syntaxe par :

– Type de données : d ;
– Valeur de donnée : v ;

Un type de données décrit l'ensemble des valeurs appartenant à ce type, l'ensemble des chaînes de caractères représentant ces valeurs et la manière de convertir ces chaînes en valeurs.

Definition 6.3.1 (Type de données) *Un* type de données d *est caractérisé par :*
– *son espace lexical* $L(d)$ *(un ensemble de chaînes de caractères) ;*
– *son espace de valeurs* $V(d)$ *;*
– *une fonction* $L2V(d) : L(d) \to V(d)$ *de l'espace lexical vers l'espace de valeurs.*

Exemple 6.3.2 *Considérons le type de données* `integer` *représentant les entiers dans les types de données XML Schema. L'espace lexical est constitué de toutes les séquences finies de caractères chiffrées (*`0`*,* `1`*, …,* `9`*), éventuellement précédés d'un caractère de signe* `+` *ou* $-$*. L'espace de valeur correspond aux nombres entiers relatifs (on peut remarquer qu'il est infini dans ce cas). La fonction* $L2V(\texttt{integer})$ *interprète les séquences de chiffres comme leur écriture en base 10.*

Definition 6.3.2 (Datatype map) *Un* datatype map D *est une fonction partielle associant à certains URI un type de données.*

Le *datatype map* permet de retrouver un type de données à partir d'un URI qui l'identifie.

Ce langage d'alignements utilise des opérateurs et des comparateurs qui sont liés à des types de données spécifiques. Il faut donc que l'implémentation de ce langage supporte ces types de données.

Interpréter des opérateurs : Les opérateurs sont utilisés pour définir des valeurs qui ne sont pas explicitement fournies par un littéral ou un URI, mais qui peuvent être calculées avec des valeurs existantes. Pour connaître la fonction à utiliser, on fournit dans l'expression un URI associée à l'opération que la syntaxe abstraite nomme transf.

Definition 6.3.3 (Opérateur de données) *Un* opérateur de données *est un tuple* $\langle n, (t_i)_{1 \leq i \leq n}, t_r, f\rangle$ *tel que :*

- n *est l'*arité *de l'opérateur ;*
- $\forall 1 \leq i \leq n$*,* t_i *est le* type de données *de la i*^{ème} *opérande ;*
- t_r *est le type de données du résultat de l'opération ;*
- $f : t_1 \times \cdots \times t_n \to t_r$ *est une fonction.*

Exemple 6.3.3 *L'opérateur de concaténation est le tuple :*

$$\langle 2, (\mathtt{string}, \mathtt{string}), \mathtt{string}, \mathrm{concat}\rangle$$

Definition 6.3.4 (Operator map) *Un* operator map *est une fonction partielle associant à certains URI un opérateur de données.*

Interpréter des comparateurs : les comparateurs servent à comparer deux valeurs ou éléments. En fait, un comparateur est un cas particulier d'opérateur, à savoir un opérateur binaire dont le type de résultat est booléen.

Definition 6.3.5 (Comparateur) *Un* comparateur *est un opérateur de données*

$$cp = \langle 2, (t, t'), \mathrm{Bool}, f\rangle$$

Un *comparator map* est aussi un cas particulier d'*operator map*. La table 8.4 en Annexe donne une bibliothèque de comparateurs usuels.

Interpréter des transformations de données : Les transformations de données sont des fonctions plus complexes utilisées pour convertir des données en d'autres données, afin de rendre possible la médiation entre deux systèmes conceptuellement similaires mais divergeant dans leur représentation concrète. Une transformation de données apparait dans la sémantique comme un opérateur, dont les opérandes sont composées de la donnée en entrée et des paramètres de la fonction et le résultat de l'opération est la sortie de la fonction.

L'implémentation d'une transformation de données peut exister hors de l'application, par exemple en tant que service Web. Par conséquent, dans la sémantique, elle est interprétée comme une fonction.

Dans la suite, on supposera que les types de données, les opérateurs et les comparateurs sont fournis par l'application et qu'un *datatype map* D et un *comparator map* $g_\cdot : \mathrm{cp} \mapsto g_{\mathrm{cp}}$ existent lorsque l'on définit la sémantique.

De même que pour les types de données, différentes applications peuvent implémenter différents opérateurs de sorte que la liste des opérateurs interprétables peut être étendue. En guise d'exemple, on peut se référer à l'annexe C pour trouver une liste d'opérateurs de données dans la table 8.3.

Interpréter des littéraux

Un littéral v peut être *simple*, auquel cas son type de données n'est pas spécifié et il doit être interprété comme lui-même, ou bien il peut être typé, ce qui s'écrit $"v"\hat{}\hat{}d$, avec d l'URI du type de données. Les valeurs d'un littéral v sont interprétées selon un *datatype map* D.

Definition 6.3.6 (Interprétation des littéraux) *Soit D un datatype map et soit $\mathcal{L}$ un ensemble de littéraux. Une interprétation de $\mathcal{L}$ est une paire $\langle \mathbb{D}, \cdot^{\mathcal{I}_L} \rangle$ avec $\mathbb{D}$ le domaine d'interprétation des littéraux et $\cdot^{\mathcal{I}_L} : \mathcal{L} \rightarrow \mathbb{D}$ tel que :*

- *$("v")^{\mathcal{I}_L} = "v"$, c'est-à-dire que les littéraux simples sont interprétés comme eux-mêmes, et*
- *$("v"\hat{}\hat{}d)^{\mathcal{I}_L} = L2V(D(d))(v)$ si $v \in L(D(d))$.*

Lorsqu'un littéral est mal formé, c'est-à-dire qu'il n'appartient pas à l'espace lexical du type auquel il est associé (comme, par exemple, 2.7^^`xsd:boolean`) alors aucune interprétation ne peut exister.

Interpréter des URI

On peut interpréter librement les URI car ils peuvent apparaître indifféremment dans une expression d'instance, de classe, de relation, de propriété ou en tant que valeur individuelle. De manière générale, un ensemble $\mathcal{U}$ d'URI est interprété simplement comme une fonction des éléments de $\mathcal{U}$ vers un domaine d'interprétation. Ce domaine contient un ensemble d'objets, les sous-ensembles de cet ensemble d'objets, les relations sur cet ensemble d'objets et les relations entre cet ensemble d'objets et un domaine d'interprétation de littéraux.

Un ensemble d'URI contient quatre sous-ensembles $\mathcal{O}, \mathcal{C}, \mathcal{R}, \mathcal{P}$, pas nécessairement disjoints, identifiant les instances, les classes, les relations et les propriétés, respectivement.

Definition 6.3.7 (Interprétation d'URI) *Une* Interprétation d'URI *d'un ensemble $\mathcal{U}$ d'URI est une paire $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$ telle que $\cdot^{\mathcal{I}} : \mathcal{U} \rightarrow \Delta \cup 2^{\Delta} \cup 2^{\Delta \times \Delta} \cup 2^{\Delta \times \mathbb{D}}$ tel que $\mathbb{D}$ est un domaine d'interprétation de littéraux et :*

- *si $u \in \mathcal{O}$ alors $u^{\mathcal{I}} \in \Delta$,*
- *si $u \in \mathcal{C}$ alors $u^{\mathcal{I}} \subseteq \Delta$,*
- *si $u \in \mathcal{R}$ alors $u^{\mathcal{I}} \subseteq \Delta \times \Delta$,*
- *si $u \in \mathcal{P}$ alors $u^{\mathcal{I}} \subseteq \Delta \times \mathbb{D}$.*

Le domaine Δ (domaine d'interprétation global) peut contenir des éléments qui sont eux-mêmes des ensembles d'éléments, des ensembles de paires ou encore des ensembles d'ensembles. Ainsi, il est possible que l'interprétation d'une URI soit à la fois dans Δ et dans 2^{Δ}. Par exemple, si $\Delta = \{x, y, (x, y), \{(x, y)\}\}$ alors $\{E\} \in \Delta \cap 2^{\Delta} \cap 2^{\Delta \times \Delta}$. Ceci permet d'exprimer des équivalences entre différents types d'entités, comme un concept avec un rôle, ou un individu et un concept, etc. En contrepartie, en l'absence de restriction sur le langage, cela peut rendre la déduction indécidable.

Interprétation des expressions complexes

À partir d'une interprétation d'URI et de littéraux, on peut définir par récurrence l'interprétation d'une expression. Certaines règles d'interprétation seront détaillées par la suite, mais on peut déjà déclarer qu'une interprétation d'expressions étend une interprétation d'URI de la façon suivante.

Definition 6.3.8 (Interprétation d'expressions) *Une interprétation d'URI $\langle \mathcal{I}, \Delta \rangle$ est étendue en une* interprétation d'expressions *en appliquant les règles d'interprétation récursives du langage, de sorte que :*
- *si C est une expression de classe, alors $C^{\mathcal{I}} \subseteq \Delta$,*
- *si R est une expression de relation, alors $R^{\mathcal{I}} \subseteq \Delta \times \Delta$ et*
- *si P est une expression de propriété, alors $P^{\mathcal{I}} \subseteq \Delta \times \mathbb{D}$.*

La plupart des constructeurs du langage correspondent à des constructeurs de logiques de description qui ne seront pas détaillés ici (voir pour cela le chapitre 2). En revanche, la particularité de ce langage se situe dans les expressions de contrainte dont l'interprétation n'est pas immédiate.

Interpréter des valeurs

L'interprétation des valeurs est plus difficile parce que l'on peut utiliser des littéraux, des individus ou bien des propriétés ou relations pour les définir. Or, les propriétés ou relations sont interprétées comme des relations ensemblistes, donc des ensembles de paires, alors que les littéraux ou individus sont interprétés comme des éléments. Ces relations permettent de comparer la valeur pointée provenant d'une même instance. Ainsi, afin de donner une interprétation homogène de ces valeurs, elles sont toutes interprétées comme des ensembles de paires.

$$V ::= v \mid i \mid R \mid P \mid \mathrm{transf}(V*)$$

Dans la syntaxe abstraite, v dénote une valeur simple (c'est-à-dire un littéral), i dénote un URI, R est une relation, P une propriété et transf dénote un opérateur, identifié par son URI (voir la section 6.3.3). On suppose que le nombre d'opérandes entre parenthèses est égale à l'arité de l'opérateur associé et la sémantique n'indique rien si tel n'est pas le cas. La vérification de la bonne utilisation d'un opérateur est à la charge de l'application. Une interprétation de valeur est alors définie à partir d'une interprétation de valeurs plus simples et d'une interprétation des expressions qui la compose.

Definition 6.3.9 (Interprétation de valeurs) *Soient une interprétation de littéraux $\mathcal{I}_L$ et une interprétation d'expressions $\mathcal{I}$ sur le domaine Δ. On suppose que les interprétations de V_1, ..., V_n sont définies. On définit alors l'interprétation de la valeur V comme une fonction $\mathcal{I}_V$ à valeurs dans l'ensemble $2^{\Delta \times (\Delta \cup \mathbb{D})}$ telle que :*
- $v^{\mathcal{I}_V} = \Delta \times \{v^{\mathcal{I}_L}\}$ *;*
- $i^{\mathcal{I}_V} = \Delta \times \{i^{\mathcal{I}}\}$ *;*
- $R^{\mathcal{I}_V} = R^{\mathcal{I}}$ *;*
- $P^{\mathcal{I}_V} = P^{\mathcal{I}}$ *;*
- $\mathrm{transf}(V_1, \ldots, V_n)^{\mathcal{I}_V} = \{\langle x, h_{\mathrm{transf}}(y_1, \ldots, y_n)\rangle / \langle x, y_1\rangle \in V_1^{\mathcal{I}_V} \wedge \cdots \wedge \langle x, y_n\rangle \in V_n^{\mathcal{I}_V}\}$.

En interprétant constamment ces valeurs comme des couples d'éléments, il n'est pas utile de séparer les cas dans les définitions qui suivent.

Interpréter des restrictions

Les restrictions sont de trois sortes : restrictions de valeurs, restrictions de type et restrictions de cardinalité. Quel que soit le type, elles correspondent à une définition de classe et leur syntaxe abstraite est similaire. Cependant, chaque type de restrictions est interprété différemment des autres.

$$
\begin{aligned}
K ::= & \; R \text{ cp } V \mid P \text{ cp } V && \text{(Restriction de valeur)} \\
& \mid P \text{ cp } d && \text{(Restriction de type)} \\
& \mid R \text{ cp } n \mid P \text{ cp } n && \text{(Restriction de cardinalité)}
\end{aligned}
$$

Le comparateur est dénoté par cp et il est identifié par un URI. R est une expression de relation, P est une expression de propriété et V est une expression de valeur.

Interpréter les restrictions de valeurs : Une restriction de valeur fournit la classe d'individus pour lesquels la comparaison cp est avérée entre une valeur à l'extrémité d'une relation ou propriété et une valeur dénotée par V. Pour simplifier la définition, on utilisera la notation Q pour dénoter soit une relation, soit une propriété.

Definition 6.3.10 (Interprétation de restrictions de valeur) *Étant donnée une interprétation de valeur $\mathcal{I}_V$ construite à partir de $\mathcal{I}$ et $\mathcal{I}_L$, on étend $\mathcal{I}$ à l'interprétation de restrictions de valeur en appliquant la règle suivante :*

$$(Q \text{ cp } V)^{\mathcal{I}} = \{x \in \Delta \mid \exists y, y' \in \Delta, \langle x, y\rangle \in Q^{\mathcal{I}} \wedge \langle x, y'\rangle \in V^{\mathcal{I}_V} \wedge g_{\text{cp}}(y, y')\}$$

où g_{cp} est l'opérateur de données associé à cp *et Δ est le domaine d'interprétation de $\mathcal{I}$.*

Un exemple typique de la restriction de valeur a lieu dans la correspondance de la figure 6.1, reliant la classe des films dramatiques ($\texttt{Drama}_1$ dans $\texttt{Cinema}_1$) aux films ayant pour genre la chaîne de caractère "drama" ($\texttt{Movie}_2 \sqcap \exists(\texttt{genres}_2 =$"$drame$") dans $\texttt{Cinema}_2$).

Interpréter des restrictions de type : Dans ce cas, d dénote un ou plusieurs types de données. La restriction de type impose que les valeurs pointées par la relation ou propriété Q appartiennent au type de données identifié par d.

Definition 6.3.11 (Interprétation des restrictions de type) *On étend une interprétation d'expressions $\mathcal{I}$ à l'interprétation de restrictions de type en appliquant la règle suivante :*

$$(P \text{ cp } d)^{\mathcal{I}} = \{x \in \Delta \mid \forall y \in \Delta, \langle x, y\rangle \in P^{\mathcal{I}} \Rightarrow g_{\text{cp}}(y, V(D(d)))\}$$

où g_{cp} est l'opérateur de données associé à cp *et Δ est le domaine d'interprétation de $\mathcal{I}$.*

Interpréter des restrictions de cardinalité : Dans ce cas, n dénote un entier (ou un ensemble d'entiers), sensé représenter une cardinalité. Le comparateur compare ce nombre avec la cardinalité de l'attribut dénoté par Q. Par exemple, il existe une correspondance entre les réalisateurs ($\texttt{Director}_1$ dans $\texttt{Cinema}_1$) et les personnes ayant réalisé au moins un film ($\texttt{Person}_2 \sqcap \exists(\texttt{directed}_2 \geq 1)$ dans $\texttt{Cinema}_2$).

Pour donner une définition plus concise, on définit d'abord la cardinalité d'une relation ou propriété associée à l'instance x.

Definition 6.3.12 (Cardinalité d'une relation) *Soit Q une relation ou propriété et $\mathcal{I}$ une interprétation d'expressions de domaine Δ et soit $x \in \Delta$. La* cardinalité *de Q depuis x, modulo $\mathcal{I}$ est le nombre cardinal* $\text{Card}_{\mathcal{I}}(x, Q)$ *égal à la taille de l'ensemble* $\{y \in \Delta / \langle x, y\rangle \in Q^{\mathcal{I}}\}$.

Definition 6.3.13 (Interprétation des restrictions de cardinalité) *On étend une interprétation d'expressions $\mathcal{I}$ à l'interprétation de restrictions de cardinalité en appliquant la règle suivante :*

$$(Q \text{ cp } n)^{\mathcal{I}} = \{x \in \Delta \mid g_{\text{cp}}(\text{Card}_{\mathcal{I}}(x, Q), n)\}$$

où g_{cp} est l'opérateur de données associé à cp et Δ est le domaine d'interprétation de $\mathcal{I}$.

Les comparateurs pour les restrictions de cardinalité sont maxCardinality ($\leq$), minCardinality ($\geq$) et cardinality ($=$).

Interpréter des correspondances

Grâce aux règles d'interprétation, on peut étendre n'importe quelle interprétation d'URI à une une interprétation d'expressions complexes. Il est alors possible de comparer les deux entités apparaissant dans une correspondance. Cette comparaison détermine la validité de l'interprétation vis-à-vis de la correspondance.

Conformément à la définition 4.2.3, il faut d'abord définir la sémantique des relations intervenant dans les correspondances. Ces relations sont $\equiv$, $\sqsubseteq$, $\sqsupseteq$, $\in$ et $\ni$, mais pourraient être étendues à d'autres relations. Dans la définition d'interprétation d'expressions qui précède, le domaine Δ est en fait le domaine des objets duquel on peut déduire le domaine général (selon la définition 2.2.2) contenant Δ, 2^{Δ}, $2^{\Delta\times\Delta}$ et $2^{\Delta\times\mathbb{D}}$. Pour plus de lisibilité, on notera $D(\Delta)$ l'ensemble $\Delta \cup 2^{\Delta} \cup 2^{\Delta\times\Delta} \cup 2^{\Delta\times\mathbb{D}}$. Or, selon la définition 4.2.3, une interprétation d'une relation r associe à chaque domaine D une relation $r^D \subseteq D \times D$. Ici, on associera donc à chaque r et chaque Δ (selon la définition 6.3.8) une relation $r^{\Delta} \subseteq D(\Delta) \times D(\Delta)$.

Definition 6.3.14 *Soit Δ un ensemble non vide. On définit les interprétations de relation sur Δ de la façon suivante :*
- $\sqsubseteq^{\Delta} =_{def} \{(x, y) \in D(\Delta) \times D(\Delta) \mid x \subseteq y\}$,
- $\sqsupseteq^{\Delta} =_{def} (\sqsubseteq^{\Delta})^{-1}$,
- $\equiv^{\Delta} =_{def} \sqsubseteq^{\Delta} \cap \sqsupseteq^{\Delta}$,
- $\in^{\Delta} =_{def} \{(x, y) \in D(\Delta) \times D(\Delta) \mid x \in y\}$,
- $\ni^{\Delta} =_{def} (\in^{\Delta})^{-1}$.

À partir des définitions maintenant introduites, il est possible de définir la satisfaction de correspondance et les modèles des alignements en suivant les définitions 4.2.4 et 4.2.5.

Interpréter des ontologies alignées

Jusqu'à présent, les définitions introduites n'ont pas fait intervenir la sémantique locale des ontologies. Pourtant, un alignement doit servir, entre autre, à établir de nouvelles connaissances locales en fonction des connaissances extérieures issues d'ontologies alignées avec l'ontologie locale. De même, un médiateur, détenant un alignement, doit être capable d'évaluer si les correspondances sont en accord avec les connaissances respectives décrites dans les deux ontologies.

Dans ce langage, les entités apparaissant sous forme d'URI dénotent des instances, des classes ou des propriétés ayant une interprétation à la fois dans la sémantique du langage d'ontologies et celle du langage d'alignements. Afin de relier l'interprétation des alignements aux interprétations locales, on fait appel à la notion de fonction d'égalisation car les interprétations locales et l'interprétation (globale) des expressions ne sont pas forcément compatibles.

Definition 6.3.15 (interprétation d'ontologies alignées) *Soient O_1 et O_2 deux ontologies alignées avec un alignement A. Une interprétation de O_1, O_2, A est un triplet $\langle \mathbf{I}, \varepsilon, \Delta \rangle$ tel que :*

- *$\mathbf{I}$ est une paire $\langle I_1, I_2 \rangle$ d'interprétations locales de O_1 et O_2 respectivement,*
- *Δ est un ensemble non vide,*
- *$\boldsymbol{\varepsilon} = \langle \varepsilon_1, \varepsilon_2 \rangle$ est une fonction d'égalisation pour $\mathbf{I}$ dans le domaine global $\Delta \cup 2^{\Delta} \cup 2^{\Delta \times \Delta} \cup 2^{\Delta \times \mathbb{D}}$,*
- *$\varepsilon_1 \circ I_1$ est une interprétation des entités à gauche de A et $\varepsilon_2 \circ I_2$ est une interprétation des entités à droite de A.*

Cette définition peut facilement être étendue à l'interprétation de tout un réseau d'ontologies alignées, en utilisant la définition 4.3.6. Les définitions de modèles d'ontologies alignées, puis de conséquences sémantiques suivent naturellement. Ainsi, il est possible de définir formellement ce qui peut être déduit des alignements expressifs de ce langage.

6.3.4 Synthèse

Cette section a fourni une sémantique pour un langage d'alignements expressif. La syntaxe abstraite et la sémantique sont résumées dans la table de l'annexe B.

La spécificité de cette sémantique est sa capacité à interpréter des réseaux d'ontologies reliant des formalismes hétérogènes.

Cette sémantique couvre un ensemble assez riche de constructeurs et d'opérateurs qui lui donne une forte expressivité. Les exemples montrent les possibilités offertes par le langage. Cependant, cette expressivité est atteinte au prix d'une difficile automatisation du raisonnement.

6.4 Raisonner avec un langage d'alignements expressif

La richesse du langage présenté ici rend le raisonnement difficile, tant en terme de complexité algorithmique que de démonstration formelle. Deux caractéristiques rendent très certainement la déduction indécidable : d'une part, le langage permet d'exprimer la logique $\mathcal{ALC}(\cap, \neg, \circ)$, correspondant à $\mathcal{ALC}$ étendue avec l'intersection, la négation et la composition de rôle. Cette logique de description est indécidable, comme cela est démontré dans [63]. En effet, pour exprimer le concept $\exists R.C$ il suffit d'utiliser la contrainte $\exists(|R \sqcap \text{range}(C)| \geq 1)$. D'autre part, il est possible d'exprimer l'équivalence d'une classe et d'un individu. Or c'est ce qui distingue le langage OWL Full indécidable du langage OWL DL décidable [28].

Néanmoins, l'objectif d'un tel langage est avant tout de pouvoir représenter les correspondances les plus précises possibles, afin de les manipuler à part, de les échanger ou de les combiner. Bien que la capacité à raisonner ne doive pas être séparée de la capacité à exprimer des connaissances, certaines situations exigent de savoir représenter des connaissances complexes [29]. Il faut alors s'en accommoder par des mécanismes de raisonnement restreints, par exemple incomplet. Or, dans certains cas, un raisonnement incomplet est suffisant.

En particulier, les alignements peuvent servir à produire une fonction de transformation de requêtes ou d'instances. Plus précisément, étant donnée une requête q_1 posée en termes d'une ontologie O_1, on souhaite obtenir une requête q_2 en termes de O_2, mais sémantiquement équivalente à la première. Or, s'il est peu probable que l'on trouve un terme de O_2 équivalent à chaque terme de q_1, il est en revanche fort possible qu'une combinaison de termes de O_2 et de constructeurs puisse être équivalente aux termes de q_1. Ainsi, il sera plus facile de construire la requête q_2.

Aussi est-il possible de restreindre l'expressivité du langage dès que le raisonnement devient nécessaire. Dans tous les cas, on peut distinguer deux formes de raisonnements, à savoir le raisonnement sur les alignements seuls et le raisonnement sur les ontologies alignées.

6.4.1 Raisonnement sur les alignements seuls

Il est parfois intéressant de ne raisonner qu'à partir des alignements. En effet, il peut arriver que les ontologies alignées ne soient pas accessibles, soit du fait d'une coupure du réseau, soit du fait d'une politique de protection de données. Il peut aussi s'avérer utile de ne pas accéder aux données des ontologies pour améliorer les performances d'un processus. Cette possibilité est facilitée par la sémantique à deux niveaux proposée. Dans ce cas, il n'est pas utile de tenir compte de la sémantique locale des ontologies et les fonctions d'interprétation peuvent être négligées. On se retrouve alors dans le cas d'un raisonnement classique. Les opérations pouvant être effectuées dans ce cadre sont :

- la déduction de nouvelles correspondances (équivaut à la déduction de conséquences sémantiques),
- la transformation de requêtes ou d'instances,
- la composition d'alignements,
- la vérification de cohérence des alignements,
- la vérification de cohérence de transformations.

La composition d'alignements est en fait un sous-problème de déduction puisqu'elle consiste à déduire des correspondances entre deux ontologies non encore alignées, à partir d'une succession d'alignements entre ces ontologies et une ou plusieurs ontologies intermédiaires. Le chapitre 7 reviendra sur cette opération. La vérification de transformations se distingue de la transformation elle-même. Elle suppose l'existence d'une procédure de transformation, potentiellement établie indépendamment de l'alignement, dont on souhaite vérifier la validité vis-à-vis d'un alignement.

Une autre possibilité consiste à manipuler un alignement d'une ontologie avec elle-même. Ce type d'opération est utile lorsque le langage d'ontologies ne permet pas d'exprimer certaines propriétés que l'on souhaiterait expliciter. Par exemple, dans l'ontologie $\texttt{Cinema}_2$ utilisée dans les exemples, on souhaiterait expliciter le fait que $\texttt{director}_2$ est l'inverse de $\texttt{directed}_2$. Or, ceci ne peut être exprimé en RDFS. En outre, il est possible de traiter les classes comme des instances, donc de raisonner au sujet de l'ontologie et non plus seulement au sujet des concepts internes de l'ontologie. On retrouve alors certaines fonctionnalités du langage SKOS présenté brièvement au chapitre 2.

En outre, le raisonnement peut être enrichi des connaissances locales des ontologies.

6.4.2 Raisonnement sur les ontologies alignées

Dans le cas du raisonnement combinant les ontologies et les alignements, il faut tenir compte de la fonction d'égalisation et de la sémantique des ontologies. Or, on verra au chapitre 8 que la complexité du raisonnement distribué dépend fortement de la complexité des alignements. Il n'est donc pas vraisemblable que l'on souhaite raisonner avec l'ensemble du langage d'alignements, à moins peut-être de réduire les capacités de déductions. Ainsi, pour ce problème particulier, il sera judicieux de n'employer qu'un sous-langage assez restreint.

Toutefois, il faut rappeler que le but de ce langage expressif est avant tout de pouvoir exprimer les correspondances les plus riches en s'abstrayant du langage d'ontologies, notamment lorsque celui-ci a une faible expressivité. À l'extrême, on pourrait envisager d'aligner de simples ensembles de concepts et relations sans aucune assertion associée. Pour profiter de connaissances locales expressives, il est préférable de réduire drastiquement l'expressivité du langage d'alignements. Le chapitre 8 reviendra justement sur les procédures de déduction dans un tel cas.

6.5 Bilan

La syntaxe et la sémantique présentées dans ce chapitre servent à définir un langage complet de représentation des connaissances. Il est possible d'utiliser ce langage pour modéliser des correspondances complexes entre ontologies. Comme on l'a vu sur des exemples simples entre deux ontologies du domaine du cinéma, de telles correspondances complexes peuvent apparaître lorsque les langages d'ontologies sont eux-mêmes peu expressifs.

Bien qu'à ce jour la sémantique ne soit pas exploitée explicitement, un langage expressif de ce type est d'ores et déjà intégré au format d'alignement de l'API d'alignement (voir [33]). Il offre une réponse au besoin de certains services de médiation particulièrement important lors de la composition des services Web sémantiques, entre autres applications.

En revanche, en tentant d'être aussi expressif que possible, la complexité des vérifications et déductions n'a pas été un critère. De ce fait, les applications voulant garantir la décidabilité de leurs procédures de décision disposent d'un langage qu'elles doivent ajuster en fonction du compromis entre expressivité et efficacité voulu.

En outre, l'utilisation d'une sémantique séparant l'interprétation des ontologies et celle des alignements assure un bon découplage des différents langages mis en jeu. Ainsi, elle permet de concevoir des procédures de vérification pour la médiation et permet un échange et une réutilisation des alignements entre différentes applications. Parmi les opérations de manipulation d'alignements figure la composition. Celle-ci est de première importance dans un environnement distribué de grande ampleur et fortement évolutif, et fait l'objet du chapitre suivant.

Chapitre 7

Composition d'alignements d'ontologies

Résumé

Dans une infrastructure aussi vaste que le Web sémantique, il n'est pas raisonnable de supposer que toutes les ontologies traitant d'un domaine de connaissances sont alignées deux à deux. De surcroît, les techniques d'alignements de qualité sont très coûteuses en temps ou en ressource machine. Les alignements de qualité sont donc précieux et gagneraient à être réutilisés. L'opération de composition le permet en produisant un nouvel alignement entre deux ontologies (appelons-les O_1 et O_3) à partir de deux alignements successifs entre O_1 et O_2 d'une part, et O_2 et O_3 d'autre part. Ce chapitre définit la notion de composition sémantique, correspondant à l'alignement contenant toutes les correspondances entre O_1 et O_3 pouvant être déduites du réseau composé des trois ontologies et des deux alignements. On démontre qu'un opérateur de composition d'alignements fondé sur les algèbres de relation donne des résultats cohérents avec la sémantique générique introduite au chapitre 4. On montre aussi que cette opération intervient dans divers cas d'applications et non seulement pour construire un nouvel alignement inexistant.

Sommaire

7.1 Le problème de la composition

Le problème de la composition d'alignements est fondamental dès lors que la quantité d'ontologies disponibles dans le réseau dépasse le nombre d'alignements pouvant être calculés entre des paires d'ontologies. Typiquement, le Web sémantique correspond à un tel réseau et l'on peut imaginer que de plus en plus d'ontologies de diverses natures vont se multiplier grâce aux outils simplifiant leur conception.

Disposer d'un alignement convenable, quelle que soit la paire d'ontologies considérée, n'est que rarement possible lorsque des contraintes de temps ou de ressources sont imposées. Ainsi, les alignements existant et en particulier ceux de bonne qualité, devraient être réutilisés autant que possible afin de faire interopérer les applications plus rapidement.

La principale opération de réutilisation est la composition. Elle consiste, étant données trois ontologies O_1, O_2 et O_3 et deux alignements A_{12} et A_{23} entre O_1 et O_2 puis O_2 et O_3 respectivement, à construire un nouvel alignement A_{13} entre O_1 et O_3. Intuitivement, il semble naturel que l'on puisse déterminer des relations entre O_1 et O_3 lorsque l'on sait les relier à une même troisième ontologie. Pourtant, l'une des caractéristiques importantes du formalisme décrit au chapitre 4 qui le distingue de DDL, DFOL ou $\mathcal{E}$-connection, est sa capacité à définir un opérateur de composition intuitif et conforme à la sémantique.

Ce chapitre a pour but de proposer des opérateurs de composition d'alignements efficaces. Le chapitre commence par décrire les applications de la composition (section 7.2). Ensuite, on introduit une définition formelle d'un opérateur de composition. On constate que, selon cette définition, il existe de nombreux opérateurs de composition, mais l'on verra que l'on peut aussi définir un opérateur dit "idéal" permettant d'obtenir l'alignement le plus précis possible compte tenu des alignements existants (section 7.3). Puis, on définit un opérateur de composition générique, pouvant s'adapter à de nombreux types de relations grâce au formalisme des algèbres de relations (section 7.4). D'autres opérateurs plus complexes sont envisagés par la suite (section 7.5). Enfin, des problèmes connexes sont discutés (section 7.6)

7.2 Les applications de la composition

Dans tous les cas de figures, la composition d'alignements est utilisée dans l'espoir d'obtenir des correspondances dont on ne disposait pas auparavant. Ceci étant dit, il existe des cas d'applications où la composition est d'une utilité particulière.

7.2.1 Aligner deux ontologies non encore alignées

Aligner deux ontologies non encore alignées est sans doute le cas de figure le plus évident. On suppose qu'on ne dispose d'aucun alignement entre O_1 et O_3, mais que l'on dispose d'alignements entre O_1 et une troisième ontologie O_2, elle-même alignée avec O_3. Dans certains cas, on peut se passer d'un alignement direct entre O_1 et O_3 en utilisant le système lié à O_2 comme intermédiaire, par exemple pour traduire une requête ou des instances provenant de O_2 d'abord dans le vocabulaire de O_2 par le truchement du premier alignement, puis dans le vocabulaire de O_3 par le biais du second alignement. Mais il peut arriver que le système de O_2 ne soit pas accessible, ou bien qu'il n'accepte pas de jouer le rôle de médiateur, ou encore que cette traduction intermédiaire conduise à un surcoût inacceptable.

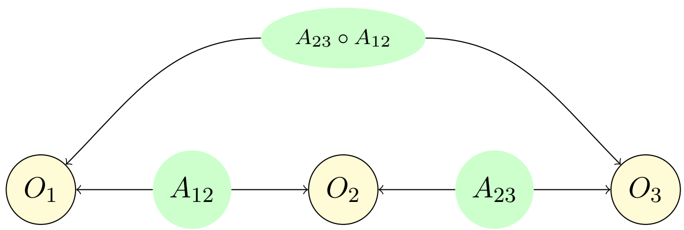


FIGURE 7.1 – Composition d'alignements. Au départ, on ne dispose d'aucun alignement.

7.2.2 Enrichir des alignements existants

Lorsqu'un alignement existe déjà entre deux ontologies, calculer la composition permet d'enrichir cet alignement, soit par l'union des alignements pour obtenir un alignement plus complet, soit par intersection pour obtenir un alignement plus précis ou plus sûr. Il peut aussi arriver que plusieurs compositions possibles permettent d'obtenir un alignement entre deux mêmes ontologies, comme le montre la figure 7.2.

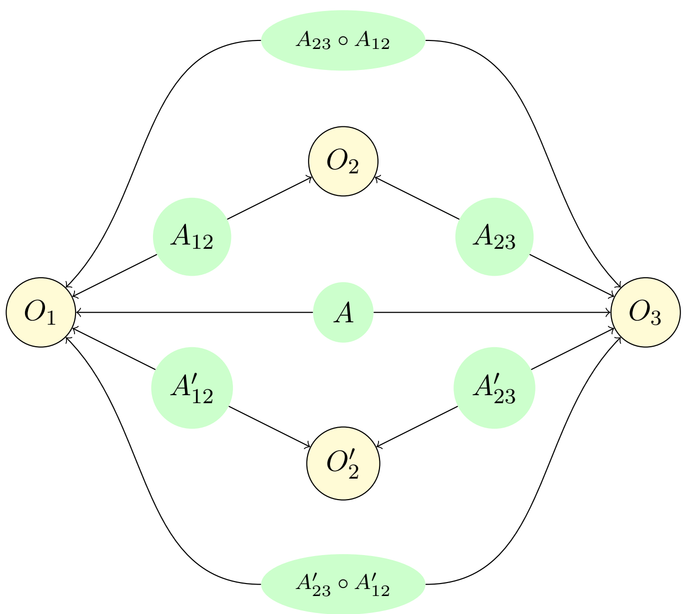


FIGURE 7.2 – Enrichissement d'alignements. Deux manières de composer les alignements mènent à un alignement de O_1 et O_3. En les combinant, on obtient un alignement A plus complet (union) ou plus précis (intersection).

Une autre possibilité pour enrichir un alignement consiste à utiliser un auto-alignement, c'est-à-dire un alignement d'une ontologie avec elle-même. Si l'ontologie est accessible, il est relativement facile de produire un auto-alignement à partir des axiomes de l'ontologie.

7.2.3 Mise à jour d'alignements

Lorsqu'une ontologie évolue, passant d'une version à l'autre, seuls peu de concepts ou expressions sont modifiés. Il est donc assez aisé de définir un alignement très précis entre deux versions successives. En le composant avec un alignement de l'ancienne version d'ontologie avec une seconde ontologie, on obtient un alignement mis à jour pour la nouvelle version.

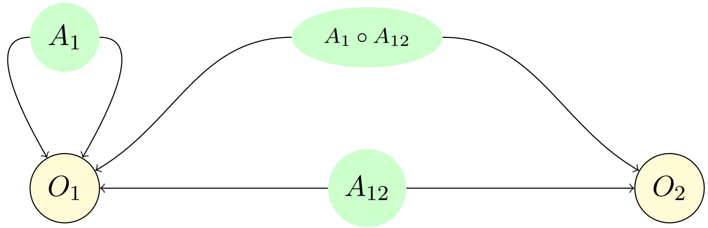


FIGURE 7.3 – Enrichissement d'alignements par un auto-alignement.

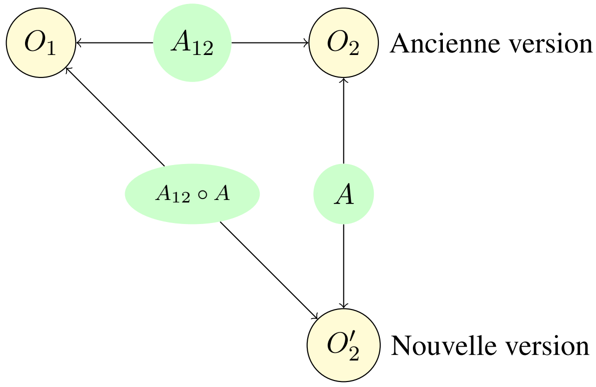


FIGURE 7.4 – Mise à jour d'alignement. Lorsqu'une ontologie est mise à jour, il suffit de garder la trace des changements effectués pour obtenir un alignement entre deux versions successives. On obtient ensuite des alignements à jour avec d'autres ontologie, en les composant avec l'alignement entre versions.

7.3 Opérateur de composition d'alignements

En première approximation, on peut dire qu'un opérateur de composition sémantique est une fonction associant à une paire d'alignements successifs $\langle A_{12}, A_{23}\rangle$ un nouvel alignement $A_{23} \circ A_{12}$ tel que $A_{23} \circ A_{12}$ peut être déduit de $\langle A_{12}, A_{23}\rangle$. Cependant, cette définition approximative est restrictive car elle ne prend pas en compte les ontologies alignées, dont la connaissance, on le verra, peut influencer l'opération de composition. On propose donc la définition suivante.

Definition 7.3.1 (Opérateur de composition) *On appelle* opérateur de composition *une fonction* comp *associant à tout réseau d'ontologies S composé de trois ontologies O_1, O_2 et O_3, et de deux alignements A_{12}, A_{23} entre O_1 et O_2, d'une part et O_2, O_3 d'autre part, un alignement $A_{13} = \mathrm{comp}(S)$ de O_1 et O_3 tel que $S \models A_{13}$ (c'est-à-dire pour toute correspondance c entre O_1 et O_3, $A_{13} \models c \Rightarrow S \models c$).*

On peut facilement remarquer qu'un tel opérateur n'est pas unique, même pour un langage d'alignements particulier et des ontologies particulières. En effet, l'opérateur qui renvoie toujours l'alignement vide est un opérateur de composition d'alignements. Par ailleurs, toute procédure de déduction distribuée correcte permet d'obtenir un opérateur de composition. Bien entendu, on comprend que l'objectif d'un opérateur de composition est d'obtenir l'alignement le plus précis possible. On définit alors la notion d'opérateur idéal.

Definition 7.3.2 (Opérateur de composition idéal) *Soit $S = \langle (O_1, O_2, O_3), (A_{12}, A_{23})\rangle$ un réseau d'ontologies alignées. On appelle* composition idéale *de S un alignement A_{13} tel que* *pour toute correspondance c entre O_1 et O_3, $S \models c \Leftrightarrow A_{13} \models c$. Un opérateur de composition* comp *est dit idéal si pour tout réseau S de trois ontologies et deux alignements,* $\mathrm{comp}(S)$ *est idéale.*

Pour construire un opérateur de composition idéal, il ne suffit pas de disposer d'une procédure de déduction correcte et complète pour les réseaux d'ontologies. En effet, si l'on souhaite produire un alignement en pratique, il faut trouver un ensemble fini de correspondances équivalent à la totalité des correspondances déductibles. Selon les langages d'ontologies et d'alignement utilisés, ceci n'est pas forcément possible. Par exemple, certaines logiques permettent de quantifier universellement un entier ($\forall n \in \mathbb{N}, f(n) = 1$). En imaginant un langage d'alignements dont les entités sont des constantes numériques (par exemple $f(1)$, $f(5)$, $f(100)$), il devient nécessaire de pouvoir énumérer tous les entiers dans l'alignement. Ce problème vient du fait que le langage d'alignements peut être différent du langage d'ontologies.

7.4 Composition à l'aide d'algèbre de relations

Une première approche consiste simplement à utiliser les propriétés des relations intervenant dans les correspondances. La plupart des langages d'alignement permettent de définir des relations comme l'équivalence, la subsomption, l'exclusion ou encore la superposition. À ces relations sont généralement associées des interprétations ensemblistes comme l'inclusion, l'égalité, l'exclusion, etc. dont les propriétés sont bien connues. Or, ces relations ensemblistes peuvent être composées en exploitant la transitivité ou bien en combinant plusieurs de ces relations. Par exemple, si $A \subseteq B$ et $B \subseteq C$, alors $A \subseteq C$. De cette propriété simple, on peut construire un opérateur de composition tel que si $\langle e_1, e_2, \sqsubseteq\rangle \in A_{12}$ et $\langle e_2, e_3, \sqsubseteq\rangle$ alors $\langle e_1, e_3, \sqsubseteq\rangle \in A_{13}$.

7.4.1 Approche préliminaire

De façon plus générale, on peut définir une opération de composition en ne donnant que la composition des symboles de relation. Ainsi, pour un langage d'alignements dont l'ensemble des relations est dénoté par R, on définit une loi $\bullet : R \times R \to R$, qui permet de définir l'opérateur suivant.

Definition 7.4.1 (Composition syntaxique) *On suppose qu'il existe $\bullet : R \times R \to R$. On définit alors l'opérateur $\circ$ qui associe à tout alignement A_{12} de O_1 et O_2, et A_{23} de O_2 et O_3 l'alignement $A_{23} \circ A_{12} = \{\langle e_1, e_3, r\rangle \mid \exists e_2, r_1, r_2, \langle e_1, e_2, r_1\rangle \in A_{12} \wedge \langle e_2, e_3, r_2\rangle \in A_{23} \wedge r = r_1 \bullet r_2\}$.*

Cependant, cet opérateur n'est pas nécessairement un opérateur de composition. Pour cela, il faut qu'il vérifie la propriété suivante [78].

Propriété 7.4.1 *Si pour tout domaine global Δ, toutes relations $r_1, r_2 \in R$ et pour tout $x, y, z \in \Delta$, $(x, y) \in r_1^\Delta \wedge (y, z) \in r_2^\Delta \Rightarrow (x, z) \in (r_1 \bullet r_2)^\Delta$, alors la composition syntaxique fournit un opérateur de composition d'alignements.*

Preuve : Soit $c = \langle e_1, e_3, r\rangle$ une correspondance telle que $A_{23} \circ A_{12} \models c$. Par définition de la composition syntaxique, il existe e_2, r_1, r_2 tel que $\langle e_1, e_2, r_1\rangle \in A_{12}$, $\langle e_2, e_3, r_2\rangle \in A_{23}$ et $r = r_1 \bullet r_2$. Soit $\mathcal{I} = \langle \mathbf{I}, \boldsymbol{\varepsilon}\rangle$ un modèle du système $S = \langle (O_1, O_2, O_3), (A_{12}, A_{13})\rangle$. Puisque $\mathcal{I}$ satisfait les alignements, $\langle e_1^{I_1}, e_2^{I_2}\rangle \in r_1^\Delta$ et $\langle e_2^{I_2}, e_3^{I_3}\rangle \in r_2^\Delta$, d'après la propriété 7.4.1, on en déduit que $\langle e_1^{I_1}, e_3^{I_3}\rangle \in r^\Delta$, donc que le système satisfait c. $\square$

Notons que la loi de composition interne $\bullet$ peut être définie par une table de composition comme celle donnée dans l'exemple 7.4.1.

Exemple 7.4.1 *Dans la table de composition suivante,* $=$ *est l'égalité d'ensemble,* $\subset$ *est l'inclusion stricte,* $\supset$ *est l'inclusion inverse stricte,* $\perp$ *est l'exclusion et* $\between$ *est la superposition partielle.*

$\mathbf{R_1}$ \ $\mathbf{R_2}$	$=$	$\subset$	$\supset$	$\perp$	$\between$
$=$	$\{=\}$	$\{\subset\}$	$\{\supset\}$	$\{\perp\}$	$\{\between\}$
$\subset$	$\{\subset\}$	$\{\subset\}$	$\{=,\subset,\supset,\perp,\between\}$	$\{\perp\}$	$\{\subset,\perp,\between\}$
$\supset$	$\{\supset\}$	$\{=,\subset,\supset,\between\}$	$\{\supset\}$	$\{\supset,\perp,\between\}$	$\{\supset,\between\}$
$\perp$	$\{\perp\}$	$\{\subset,\perp,\between\}$	$\{\perp\}$	$\{=,\subset,\supset,\perp,\between\}$	$\{\subset,\perp,\between\}$
$\between$	$\{\between\}$	$\{\subset,\between\}$	$\{\supset,\perp,\between\}$	$\{\supset,\perp,\between\}$	$\{=,\subset,\supset,\perp,\between\}$

TABLE 7.1 – Table de composition de l'exemple 7.4.1.

Grâce à cette table, on dispose d'une manière générique et compacte de définir un opérateur de composition. On pourra remarquer au passage que cette méthode ne permet pas, en revanche, de définir un opérateur de composition pour les logiques de description distribuées, ni pour DFOL, ni pour $\mathcal{E}$-connections.

Preuve : Soient trois ontologies O_1, O_2 et O_3 ne contenant que les concepts C_1, C_2 et C_3 respectivement, et ne contenant pas d'axiomes. Supposons que seules les passerelles $1:C_1 \xrightarrow{\sqsubseteq} 2:C_2$ et $2:C_2 \xrightarrow{\sqsubseteq} 3:C_3$ existent dans le réseau d'ontologies S mettant en jeu O_1, O_2 et O_3. La composition syntaxique de ces passerelles est la passerelle $c = 1:C_1 \xrightarrow{\sqsubseteq} 3:C_3$. Montrons que $S \not\models c$, ce qui implique que la composition syntaxique ne constitue pas un opérateur de composition.

Un modèle du système S peut être défini comme suit, en affectant à O_1 (respectivement O_2, O_3) l'interprétation $\langle \Delta_1, I_1 \rangle$ (respectivement $\langle \Delta_2, I_2 \rangle$, $\langle \Delta_2, I_2 \rangle$) :

$$\begin{aligned}
\Delta_1 &= \{a\} & I_1(C_1) &= \{a\} \\
\Delta_2 &= \{b\} & I_2(C_2) &= \{b\} \\
\Delta_3 &= \{c,d\} & I_3(C_3) &= \{c,d\} \\
r_{12} &= \{\langle a,b \rangle\} & r_{21} &= \{\langle b,a \rangle\} \\
r_{23} &= \{\langle b,c \rangle\} & r_{32} &= \{\langle c,b \rangle\} \\
r_{13} &= \{\langle a,d \rangle\} & r_{31} &= \{\langle d,a \rangle\}
\end{aligned}$$

Il est facile de vérifier que ceci est un modèle de S et il est aussi facile de vérifier que ce modèle ne satisfait pas c. □

Puisque DDL est une restriction de DFOL, ce contre-exemple convient aussi pour DFOL, en remplaçant les passerelles $i:C_i \xrightarrow{\sqsubseteq} j:C_j$ par $i:C_i(x^{\to j}) \to j:C_j(x)$.

En ce qui concerne $\mathcal{E}$-connection, les relations entre ontologies sont exprimées par les liens (*links*) qui, tout comme les rôles en logiques de description, sont définis de façon propre à chaque ontologie. Il existe une infinité de liens possibles et leurs propriétés peut varier d'un réseau d'ontologie à un autre. Ainsi, parler de leur composition syntaxique n'a pas vraiment de sens.

Dans la section suivante, nous allons voir que l'on peut définir un opérateur de composition plus intéressant en utilisant les algèbres de relations.

7.4.2 Algèbres de relations

Ce qui précède fait penser à la composition de relations dans la théorie des algèbres de relations développées par [71]. Ce formalisme offre une infrastructure adaptée à la manipulation de relations qui va même au delà de la seule composition. Plus précisément, dans le cadre des correspondances, on s'intéresse plus particulièrement aux algèbres de relations binaires. Formellement, on définit une algèbre de relations binaires comme suit :

Definition 7.4.2 (Algèbre de relations binaires) *Une algèbre de relations binaires est un tuple* $\langle R, \cap, \cup, \cdot^{-1}, 1, 0, I, \circ \rangle$ *tel que* $\langle R, \cap, \cup, \cdot^{-1}, 1, 0 \rangle$ *est une algèbre booléenne et* $\circ$ *est une loi de composition interne sur* R *associative admettant un élément neutre* I *(donc* $\langle R, \circ, I \rangle$ *est un monoïde).*

On rappelle qu'une algèbre booléenne est une structure $\langle R, \cap, \cup,^{-1}, 1, 0 \rangle$ telle que R est un ensemble, $0, 1 \in R$, $\cap$ et $\cup$ sont des lois de composition internes sur R associatives, commutatives et distributives l'une par rapport à l'autre et pour tout $a, b \in R$:

$$\begin{aligned} a \cup (a \cap b) &= a \\ a \cap (a \cup b) &= a \\ a \cup a^{-1} &= 1 \\ a \cap a^{-1} &= 0 \end{aligned}$$

Cette structure abstraite peut être instanciée très simplement pour répondre aux besoins de la composition d'alignements. Il s'agit d'associer à R l'ensemble des symboles de relations du langage d'alignements, tout en vérifiant les contraintes suivantes.

- R est défini comme l'ensemble des parties d'un ensemble de relations élémentaires $\bar{R}$ mutuellement exclusives et exhaustives. Du point de vue de la sémantique, ceci signifie que pour tout $r_1, r_2 \in \bar{R}$ et tout domaine Δ, $r_1 \neq r_2 \Leftrightarrow r_1^\Delta \cap r_2^\Delta = \emptyset$ (mutuellement exclusives) et que pour tout $\langle x, y \rangle \in \Delta \times \Delta$, il existe $r \in \bar{R}$ tel que $\langle x, y \rangle \in r^\Delta$ (exhaustif).
- r^{-1} est la relation inverse de r, ce qui implique, d'un point de vue sémantique, que $\langle e, e' \rangle \in r^\Delta$ si et seulement si $\langle e', e \rangle \in (r^{-1})^\Delta$.
- $\cap$ et $\cup$ correspondent à l'intersection et à l'union ensemblistes. D'un point de vue sémantique, cela implique que si $\langle e, e' \rangle \in r_1^\Delta \cap r_2^\Delta$ (respectivement $\langle e, e' \rangle \in r_1^\Delta \cup r_2^\Delta$), alors $\langle e, e' \rangle \in (r_1 \cap r_2)^\Delta$ (respectivement $\langle e, e' \rangle \in (r_1 \cup r_2)^\Delta$).
- La loi de composition $\circ$ doit être conforme à la sémantique, c'est-à-dire que si $\langle e, e' \rangle \in r_1^\Delta$ et $\langle e', e'' \rangle \in r_2^\Delta$, alors $\langle e, e'' \rangle \in (r_1 \circ r_2)^\Delta$.
- 0 est l'ensemble vide et 1 est l'ensemble $\bar{R}$.
- L'élément neutre I est l'identité, c'est-à-dire $I^\Delta = \{\langle x, x \rangle \mid x \in \Delta\}$.

Pour obtenir une loi de composition, il suffit de donner la composition pour les relations élémentaires. La composition sur R entier se construit par union des résultats sur chaque composante. En outre, la relation 0 ne correspond à aucune relation possible entre deux entités. Ainsi, si elle apparait dans un alignement, cela signifie qu'il y a une incohérence. Quant à la relation 1, elle dénote une relation indéterminée et n'apporte aucune information supplémentaire.

Le principal intérêt de décrire la composition sous cette forme est que l'on peut exploiter alors les nombreux résultats théoriques obtenus sur les algèbres de relations. Entre autre, [15] définit des conditions générales pour qu'une table de composition fournisse un mécanisme de déduction complet et traitable en pratique. Ces conditions satisfont la contrainte de la propriété 7.4.1 donc suffisent à fournir un opérateur de composition d'alignements. Dans notre cadre, le raisonnement par cette approche ne peut généralement pas être complet puisqu'il néglige entièrement les connaissances décrites dans les ontologies. En revanche, il est très intéressant pour déduire les quasi conséquences (définition 4.3.11) du système.

Outre l'algèbre d'ensembles de l'exemple 7.4.1, on peut citer l'algèbre temporelle de Allen [7] ou encore l'algèbre spatiale RCC8 [61] obéissant aux conditions ci-dessus. Ainsi, en alignant des ontologies au sujet d'intervalles temporels ou bien d'étendues spatiales, on peut par exemple utiliser un langage d'alignements utilisant les relations de Allen plutôt que la subsomption, l'exclusion, etc. La composition sera alors effectuée de la même manière que dans un graphe d'intervalles temporels.

7.5 Améliorer la composition

Jusqu'à présent, les opérateurs envisagés sont restés relativement simples, puisqu'ils ne font intervenir la sémantique que par le biais des relations de correspondances. Dans cette section, nous allons introduire des opérateurs exploitant soit une sémantique plus riche des alignements, soit la sémantique des ontologies alignées.

7.5.1 Composition de relations complexes

Dans les opérateurs de la section précédente, la forme des expressions d'entités alignées n'était pas exploitée. Or, si l'on utilise le langage décrit dans le chapitre 6, on souhaiterait utiliser la sémantique des constructeurs pour définir un opérateur de composition plus riche. À titre d'exemple, on peut observer que si $\langle A, B, \sqsubseteq \rangle$ et $\langle C, D, \sqsubseteq \rangle$ appartiennent à un premier alignement A_{12} et que $\langle B \sqcap D, E, \equiv \rangle$ appartient à un deuxième alignement A_{23}, alors on devrait pouvoir compter $\langle A \sqcap C, E, \sqsubseteq \rangle$ au nombre des correspondances de l'alignement composé. Cependant, dans cette optique, il n'est plus possible d'utiliser simplement les algèbres de composition.

Il convient alors de définir de nouveaux algorithmes pour traiter ces cas. Traiter ce cadre en toute généralité est très difficile et demanderait une étude au cas par cas des langages d'alignements. Ainsi, pour tout de même offrir une part de solution à ce problème, on se restreindra au cas où la seule relation utilisée dans les correspondances est l'équivalence.

Afin de définir une procédure, il faut d'abord introduire la définition suivante.

Definition 7.5.1 *Soit E une expression d'entité complexe dans un langage d'alignements. Soit X un ensemble d'expressions, simples ou complexes, dans le langage d'alignements. On dit que E* est construit à partir *de X si et seulement s'il est possible d'obtenir E en utilisant les constructeurs du langage d'alignements et une et une seule fois chacune des expressions de X.*

Si E peut être construit à partir de X, alors on dit qu'il existe un plan de construction E_X *permettant d'obtenir E à partir de X par application des règles de la grammaire du langage.*

Si Y est une famille d'expressions indicée par X dont les types correspondent deux à deux (c'est-à-dire que pour tout $e \in X$, y_e est du même type que e, par exemple classe, relation, propriété, instance, etc.), alors on peut définir une expression $E_X(Y)$ construite à partir de Y par application du plan de construction E_X.

D'une certaine manière, $E_X(Y)$ correspond à une substitution de variables dans l'expression E. On peut remarquer que $E_X(X) = E$.

Exemple 7.5.1 *Prenons l'expression $E = C \sqcap \exists(R \circ S = \neg dom(D))$ et l'ensemble $X =$*

$\{C, R{\circ}S, D\}$. *E peut être construit à partir de X. Par ailleurs, prenons* $Y = \{A{\sqcup}\neg B, P, \exists(|F| \leq 1)\}$. *Alors* $E_X(Y) = A \sqcup \neg B \sqcap \exists(P = \neg dom(\exists(|F| \leq 1)))$

La proposition suivante montre une manière de construire des correspondances issues de la composition de deux alignements complexes.

Proposition 7.5.1 *Supposons que E est une expression d'entité pour l'ontologie O_2, alignée avec O_1 par A_{12} d'une part, et O_3 par A_{23} d'autre part. Alors, s'il existe deux ensembles d'expressions X et Y sur le vocabulaire de O_2, tels que :*
- *pour tout $e \in X$, il existe une expression d'entité $f(e)$ de l'ontologie O_1 telle que $A_{12} \models \langle f(e), e, \equiv\rangle$,*
- *pour tout $e' \in Y$, il existe une expression d'entité $g(e)$ de l'ontologie O_3 telle que $A_{23} \models \langle e, g(e), \equiv\rangle$ et*
- *E peut être construit à partir de X et à partir de Y,*

alors la correspondance $\langle E_X(\{f(e) \mid e \in X\}), E_Y(\{g(e) \mid e \in Y\}), \equiv\rangle$ est une conséquence des alignements A_{12} et A_{23}.

Preuve : Pour prouver cette proposition, on commence par établir le lemme suivant.

Lemme 7.5.1 *Soient deux familles d'expressions $Z = (z_e)_{e \in X}$ et $Z' = (z'_e)_{e \in X}$ indicées par X. Soit $\mathcal{I}$ (respectivement $\mathcal{I}'$) une interprétation des termes de Z (respectivement de Z'). Si, pour tout $e \in X$, $z_e^{\mathcal{I}} = {z'_e}^{\mathcal{I}'}$, alors $E_X(Z)^{\mathcal{I}} = E_X(Z')^{\mathcal{I}'}$.*

Ce lemme est immédiat si l'on observe la manière dont sont construites les interprétations dans le langage d'alignements expressif. En effet, une interprétation d'expression est entièrement déterminée par l'interprétation de ses sous-expressions. Prenons maintenant un modèle $\mathcal{I}$ du réseau d'ontologies alignées comprenant les deux alignements A_{12} et A_{23}. Pour tout $e \in X$, $A_{12} \models \langle f(e), e, \equiv\rangle$ implique que $f(e)^{\mathcal{I}} = e^{\mathcal{I}}$. Par conséquent, d'après le lemme, $E_X(\{f(e) \mid e \in X\})^{\mathcal{I}} = E_X(X)^{\mathcal{I}} = E^{\mathcal{I}}$. De la même manière, $E_Y(\{g(e) \mid e \in Y\})^{\mathcal{I}} = E_Y(Y)^{\mathcal{I}} = E^{\mathcal{I}}$. Donc $E_X(\{f(e) \mid e \in X\})^{\mathcal{I}} = E_X(\{g(e) \mid e \in X\})^{\mathcal{I}}$. □

Dans cette preuve, le fait que $\mathcal{I}$ est construite à partir des interprétations locales et d'une fonction d'égalisation n'a aucune importance. Cela signifie que l'on peut utiliser cette construction en raisonnant exclusivement avec les alignements donc la composition peut se faire hors ligne. Cette proposition permet aussi de construire itérativement des correspondances composées complexes.

Exemple 7.5.2 *On peut reprendre l'exemple précédent. Supposons que l'alignement A_{12} contient les correspondances $\langle A \sqcup \neg B, C, \equiv\rangle$, $\langle P, R \circ S, \equiv\rangle$, $\langle \exists(|F| \leq 1), D, \equiv, \rangle$ et que A_{23} contient $\langle C, \exists(K = H), \equiv\rangle$, $\langle \exists(R \circ S = \neg\mathrm{dom}(D)), F, \equiv\rangle$. Alors on peut en déduire la correspondance entre O_1 et O_3 : $\langle A \sqcup \neg B \sqcap \exists(P = \neg dom(\exists|F| \leq 1)), \exists(K = H) \sqcap F, \equiv\rangle$.*

Corollaire 7.5.1 *Soit E une expression d'entité pour l'ontologie O_2, alignée avec O_1 par A_{12}. Alors, s'il existe un ensemble d'expressions X sur le vocabulaire de O_2, tels que :*
- *pour tout $e \in X$, il existe une expression d'entité $f(e)$ de l'ontologie O_1 telle que $A_{12} \models \langle f(e), e, \equiv\rangle$,*
- *E peut être construit à partir de X,*

alors la correspondance $\langle E_X(\{f(e) \mid e \in X\}), E, \equiv\rangle$ est une conséquence de l'alignement A_{12}.

Une fois que l'on a obtenu des expressions complexes par ce moyen, on peut utiliser la composition avec des algèbres de relation, en traitant les expressions comme des entités comme les autres. Il serait intéressant de disposer d'une proposition similaire permettant de calculer des relations autres que l'équivalence. Mais la difficulté réside alors dans le fait que chaque constructeur peut influencer différemment les relations. Tandis que l'équivalence est préservée par application des mêmes constructeurs à droite et à gauche d'une relation, ce n'est pas le cas pour la subsomption par exemple.

Une approche similaire et plus détaillée de procédure de composition de correspondances complexes est proposé par [16] dans un cadre plus restreint, pour les appariements de schémas de bases de données relationnelles. Une approche similaire serait à envisager pour le cas plus général de notre langage d'alignements.

7.5.2 Exploitation des ontologies

La deuxième extension que l'on peut faire consiste à utiliser les connaissances internes aux ontologies pour améliorer la composition. Pour tenir compte de la sémantique des ontologies dans la composition d'alignements, il est indispensable de disposer d'une procédure de raisonnement distribué, ne serait-ce que partielle. Ceci fait l'objet du chapitre suivant, mais l'on peut d'ores et déjà indiquer comment les connaissances locales interagissent avec la composition.

Supposons que l'on ait les correspondances $\langle e_1, e_2, r\rangle$ et $\langle e'_2, e_3, r'\rangle$ où $e_2 \neq e'_2$ et e_2, e'_2 sont des entités atomiques. Dans ce cas, il n'est pas possible de déduire quoi que ce soit en utilisant les opérateurs de composition présentés jusqu'à présent. En revanche, en supposant que l'on puisse déduire, dans l'ontologie intermédiaire O_2, que $e_2 \equiv e'_2$, alors il devient possible d'appliquer la première méthode avec laquelle on obtient $\langle e_1, e_3, r \bullet r'\rangle$.

7.6 Bilan

Dans ce chapitre, j'ai présenté le problème de la composition d'alignements d'ontologies, en le motivant d'abord par plusieurs applications. En fournissant une définition formelle d'un opérateur de composition, j'ai présenté diverses opérations bien fondées pour la composition.

La composition est une opération qui a vocation à être intégrée à un serveur d'alignement. En effet, lorsque le serveur ne dispose pas d'un alignement en stock pour deux ontologies et que les contraintes de temps ou de ressources sont telles que l'on ne peut pas se permettre de lancer une procédure complète de mise en correspondance, il est intéressant de combiner des alignements déjà présents pour fournir un résultat en temps raisonnable. Néanmoins, il subsiste un problème non abordé. Il peut y avoir plusieurs manières de composer les alignements pour obtenir le résultat souhaité. Selon les contraintes, il n'est pas forcément possible de calculer la composition pour toutes les séquences d'alignements donnant le résultat escompté. Ainsi se pose la question du choix de la meilleure séquence d'alignements à utiliser.

Utiliser la plus courte séquence n'est pas toujours le choix le plus judicieux, car les alignements peuvent être de piètre qualité. Un début de réponse peut être apporté en étudiant le réseau formé par les ontologies alignées. On peut supposer que la composition sera d'autant meilleure que l'on passe par des ontologies sémantiquement proches des ontologies à aligner.

L'autre point délicat évoqué concerne l'utilisation de la sémantique des ontologies pour améliorer la composition. Celle-ci renvoie aux besoins de raisonner sur un réseau d'ontologies. Ce point est abordé dans le chapitre qui suit.

Chapitre 8

Application aux logiques de description

Résumé

Ce chapitre décrit les logiques de description distribuées intégrées (*Integrated Distributed Description Logics* ou IDDL) en instanciant le formalisme générique du chapitre 4 pour le cas où les ontologies sont représentées en logiques de description et le langage d'alignements est réduit à des relations de subsomption, d'exclusion et appartenance. Ensuite, le raisonnement sur ce formalisme est étudié en commençant d'abord par établir un premier résultat qui montre la décidabilité de IDDL en fonction de la décidabilité des logiques locales. À partir de ce résultat, on décrit une procédure de déduction dont l'intérêt est l'exploitation des procédures de déduction locales existantes, sans présupposer de l'expressivité des langages utilisés, et selon un mécanisme de communication minimal. Néanmoins, la complexité de la procédure dans le cas général est telle que nous envisageons diverses approches pour améliorer le résultat. Dans un premier temps, on évoque quelques méthodes pour une optimisation directe de la procédure, mais à ce stade de l'étude, ces optimisations ne suffisent pas encore à rendre le mécanisme exploitable en pratique, dans le cas général. On étudie alors une restriction, pour laquelle on peut encore utiliser le même principe d'algorithme avec une complexité acceptable. Enfin, on étudie des variantes de IDDL permettant de faire le lien avec d'autres formalismes distribués.

Sommaire

8.1 Motivations

Les logiques de description se sont imposées comme le formalisme de rigueur sur le Web sémantique. On trouve maintenant de nombreuses ontologies écrites dans le standard du W3C, OWL. Certaines de ces ontologies sont utilisées dans des systèmes accessibles aux utilisateurs, contenant toute la machinerie nécessaire au raisonnement. Afin d'exploiter conjointement ces ontologies et ces systèmes existants, des alignements sont produits pour définir les correspondances. Nombre des alignements créés automatiquement décrivent des équivalences de termes, des subsomptions et des exclusions.

Le présent chapitre s'attache précisément à ce type de réseaux d'ontologies alignées et aborde le problème du raisonnement. De ce fait, il est le point de convergence des applications de la sémantique du chapitre 4, puisque la sémantique ne vaut que si elle permet des vérifications ou des déductions formelles.

8.1.1 Exemple

Dans l'exemple suivant, considérons un réseau de trois ontologies construites indépendemment O_1, O_2 et O_3 connectées via les alignements A_{12}, A_{13} et A_{23} calculés par des outils de mise en correspondances tiers. Bien que ces ontologies soient très petites, on peut imaginer qu'elles représentent la part pertinente d'une plus grande ontologie. La première ontologie décrit des informations sur des personnes et leur localisation (par exemple, des clients et leur adresse). La deuxième est une ontologie des personnes et leurs relations, notamment familiales. La troisième traite des régions géographiques de la Terre (voir la figure 8.1.1).

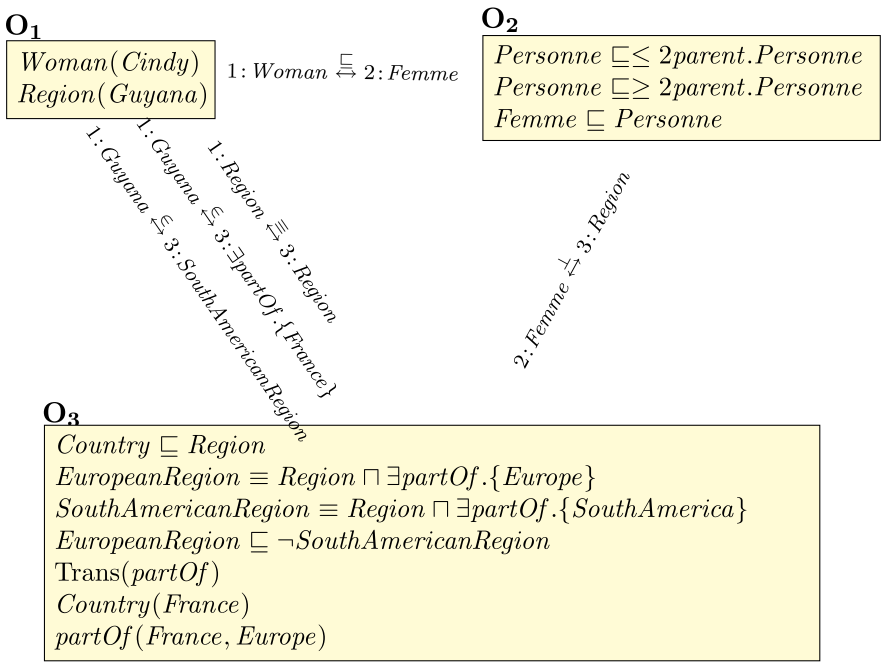


FIGURE 8.1 – Un exemple de réseau d'ontologies alignées.

Les formules apparaissant entre deux ontologies dénotent des correspondances entre termes de différentes ontologies. Leur signification formelle sera expliquée en détails dans la prochaine section. Ici, nous ne donnons qu'une description intuitive de leur sens. La correspondance entre O_1 et O_2 affirme simplement qu'une femme ($Woman$) dans O_1 est une sorte de $Femme$ dans O_2. La correspondance entre O_2 et O_3 assure que les concepts $Femme$ et $Region$ sont disjoints. Enfin, l'alignement entre O_1 et O_3 dénotent que $Region$ dans O_1 définit la même chose que $Region$ dans O_3 ; que $Guyana$ est une partie de la $France$ et que $Guyana$ est une région de l'Amérique du Sud.

On pourrait interpréter simplement les correspondances comme des axiomes ajoutant de l'information au sujet des ontologies fusionnées. Bien que ce soit une possibilité très naturelle, cela mène à des incohérences inattendues ou indésirables. En effet, puisque la Guyane ($Guyana$) est une partie de la France et la relation $partOf$ est transitive, la Guyane est donc aussi une partie de l'Europe. En outre, la Guyane est une région, et ainsi elle doit être une région Européenne. Par ailleurs, une correspondance indique que c'est aussi une région sud-américaine. Ceci est en contradiction avec l'axiome de O_3 assurant que les régions européennes et les régions sud-américaines sont disjointes.

Le problème vient du fait que la relation $partOf$ est utilisée différemment dans des contextes distincts : dans O_3, elle indique une relation topologique, tandis que dans l'alignement entre O_1 et O_3, elle indique une relation géopolitique. La sémantique de IDDL autorise l'utilisation d'un terme dans des contextes différents sans briser la cohérence globale.

8.2 Integrated Distributed Description Logics (IDDL)

Les logiques de description distribuées intégrées (Integrated Distributed Description Logics ou IDDL) ont été présentées tout d'abord dans [77]. IDDL est issu de la combinaison des logiques de description et de la sémantique générale décrite au chapitre 4. Cette logique forme ce que j'ai nommé au chapitre 4 une logique distribuée concrète, qui instancie la sémantique générique.

8.2.1 Réseau d'ontologies alignées en IDDL

Dans IDDL, la notion de réseau d'ontologies alignées est quelque peu particularisée car les ontologies sont représentées en logiques de description, tandis que les alignements ne contiennent que des correspondances bien particulières reliant soit des concepts, soit des rôles, soit des individus de deux ontologies.

Syntaxe

Les correspondances en IDDL sont similaires aux axiomes des logiques de description, excepté qu'elles mettent en jeu des concepts, rôles ou individus de deux ontologies. J'utilise ici une notation similaire à celle des passerelles (*bridge rules*) de DDL, de façon à identifier de quelle ontologie provient un concept, un rôle ou un individu. Si un terme E appartient à une ontologie i, alors on écrit $i\!:\!E$. Les 6 types de correspondances possibles entre les ontologies i et j sont :

Definition 8.2.1 (Correspondance) *Une* correspondance *entre les ontologies i et j est l'une des formules suivantes :*

– $i\!:\!C \xleftrightarrow{\sqsubseteq} j\!:\!D$ *est une subsomption de concepts inter-ontologies ($\sqsubseteq_C$) ;*

– $i\!:\!R \overset{\sqsubseteq}{\longleftrightarrow} j\!:\!S$ *est une subsomption de rôles inter-ontologies* ($\sqsubseteq_R$) *;*
– $i\!:\!C \overset{\perp}{\longleftrightarrow} j\!:\!D$ *est une exclusion de concepts inter-ontologies* ($\perp_C$) *;*
– $i\!:\!R \overset{\perp}{\longleftrightarrow} j\!:\!S$ *est une exclusion de rôles inter-ontologies* ($\perp_R$) *;*
– $i\!:\!a \overset{\in}{\longleftrightarrow} j\!:\!C$ *est une appartenance d'individu inter-ontologies* ($\in$) *;*
– $i\!:\!a \overset{=}{\longleftrightarrow} j\!:\!b$ *identité d'individus inter-ontologies* ($=$)*.*

Un alignement d'ontologie en IDDL est un ensemble de ces correspondances. Accompagnés d'un ensemble d'ontologies, ils forment les composants d'un réseau d'ontologies alignées en IDDL.

Definition 8.2.2 (Réseau d'ontologies alignées) *Un* réseau d'ontologies alignées *est un couple* $\langle \mathbf{O}, \mathbf{A} \rangle$ *telle que* $\mathbf{O}$ *est un ensemble d'ontologies et* $\mathbf{A} = (A_{ij})_{i,j \in \mathbf{O}}$ *est une famille d'alignements reliant les ontologies de* $\mathbf{O}$. [1]

De façon générale, n'importe quelle logique de description peut être utilisée localement en IDDLet n'importe lesquelles des six correspondances peuvent relier ces ontologies. Or, de même que chaque constructeur de logique de description porte un nom, souvent sous la forme d'une lettre majuscule, les types de correspondances sont identifiés par des symboles permettant de décrire des restriction du formalisme. On voit ces symboles entre parenthèses dans la définition 8.2.1. On définit alors une restriction de IDDL en ajoutant entre parenthèses la liste des types de correspondances autorisés. Par exemple, IDDL($\sqsubseteq_C, \perp_C$) indique que seules les correspondances entre concepts sont autorisées. Pour définir une procédure de raisonnement, on n'étudiera que des restrictions de IDDL.

Sémantique

De même que dans le cas général du chapitre 4, la sémantique des réseaux d'ontologies alignées en IDDL dépend des sémantiques locales. Ces dernières, bien que toutes correspondent à des logiques de description, peuvent être variées. Ainsi, un réseau en IDDL peut comprendre un mélange d'ontologies en $\mathcal{ALC}$, $\mathcal{SHIQ}$ ou $\mathcal{SROIQ}$. Informellement, interpréter un réseau IDDL consiste à associer une interprétation standard en logique de description à chaque ontologie, puis à corréler les domaines d'interprétation grâce à la *fonction d'égalisation*. Il est à noter que la notion de fonction d'égalisation en IDDL est donné sous une forme différente que pour la sémantique générale, de façon à tenir compte de la restriction imposé par la logique distribuée concrète (voir la définition 4.3.2). En effet, la fonction d'égalisation en IDDL est telle que la composition d'une interprétation locale par la fonction d'égalisation est elle-même une interprétation locale.

Definition 8.2.3 (Fonction d'égalisation) *Étant donnée une famille d'interprétations locales* $\mathbf{I}$*, une fonction d'égalisation* $\boldsymbol{\varepsilon}$ *est une famille de fonctions indicées par* $\mathbf{I}$ *telle que pour tout* $I_i \in \mathbf{I}$*,* $\varepsilon_i : \Delta^{I_i} \to \Delta$ *où* Δ *est appelé le domaine global d'interprétation de* $\boldsymbol{\varepsilon}$*.*

Une interprétation distribuée assigne une interprétation standard de logique de description à chaque ontologie du réseau, ainsi qu'une fonction d'égalisation qui met en corrélation les connaissances locales dans un domaine d'interprétation global.

1. On notera constamment en caractère gras des familles (au sens mathématique du terme) d'éléments. Ainsi, $\mathbf{O}$ dénote la famille $(O_i)_{i \in I}$ où I est un ensemble d'indices. Aussi, $\mathbf{A} = (A_{ij})_{i,j \in \mathbf{O}}$ est indicé par des couples d'ontologies.

Definition 8.2.4 (Interprétation distribuée) *Soit $S = \langle \mathbf{O}, \mathbf{A} \rangle$ un réseau d'ontologies alignées. Une* interprétation distribuée *de S est une paire $\langle \mathbf{I}, \boldsymbol{\varepsilon} \rangle$ où $\mathbf{I}$ est une famille d'interprétations indicées par $\mathbf{O}$, $\boldsymbol{\varepsilon}$ est une fonction d'égalisation pour $\mathbf{I}$, tels que pour tout $i \in \mathbf{O}$, I_i interprète i et $\varepsilon_i : \Delta^{I_i} \to \Delta$.*

Rappelons qu'en logique de description, le domaine d'interprétation est défini par un simple ensemble non vide Δ, mais que la fonction d'interprétation est à valeurs dans $\Delta \cup 2^{\Delta} \cup 2^{\Delta \times \Delta}$. Or, selon la définition 4.3.1, une fonction d'égalisation doit associer à tous les éléments de cet ensemble une valeur dans un domaine d'interprétation global. On définit pour cela l'image de l'interprétation d'un concept ou d'un rôle de la façon suivante.

Definition 8.2.5 (Fonction d'égalisation étendue) *Étant donnée une interprétation distribuée $\langle \mathbf{I}, \boldsymbol{\varepsilon} \rangle$ d'un réseau d'ontologies S, la fonction ε_i est étendue à tout l'ensemble $\Delta^{I_i} \cup 2^{\Delta^{I_i}} \cup 2^{\Delta^{I_i} \times \Delta^{I_i}}$ en prenant, pour chaque concept C_i et chaque rôle R_i les images suivantes.*
- $\varepsilon_i(C_i^{I_i}) = \bigcup_{x \in C_i^{I_i}} \{\varepsilon_i(x)\}$ *et*
- $\varepsilon_i(R_i^{I_i}) = \bigcup_{\langle x,y \rangle \in R_i^{I_i}} \{\langle \varepsilon_i(x), \varepsilon_i(y) \rangle\}$.

On peut remarquer que la fonction d'égalisation en IDDL associe aux interprétations de concepts des sous-ensembles du domaine global, aux interprétations de rôles elle associe des relations ensemblistes sur ce domaine, et cette fonction préserve l'appartenance et l'inclusion. On verra dans la section 8.8 des variantes de cette sémantique où les fonctions ε_i sont plus ou moins contraintes.

Tandis que la notion de satisfiabilité locale est identique à celle des logiques de description, la satisfiabilité des correspondances met en jeu les fonctions d'égalisation.

Definition 8.2.6 (Satisfaction d'une correspondance) *Soit S un réseau d'ontologies alignées et i, j deux ontologies de S. Soit $\mathcal{I} = \langle \mathbf{I}, \boldsymbol{\varepsilon} \rangle$ une interprétation distribuée. On définit la satisfaction d'une correspondance c (ce que l'on note $\mathcal{I} \models_d c$) comme suit :*

$$
\begin{array}{lll}
\mathcal{I} \models_d i\!:\!C \xleftrightarrow{\sqsubseteq} j\!:\!D & \textit{si et seulement si} & \varepsilon_i(C^{I_i}) \subseteq \varepsilon_j(D^{I_j}) \\
\mathcal{I} \models_d i\!:\!R \xleftrightarrow{\sqsubseteq} j\!:\!S & \textit{si et seulement si} & \varepsilon_i(R^{I_i}) \subseteq \varepsilon_j(S^{I_j}) \\
\mathcal{I} \models_d i\!:\!C \xleftrightarrow{\perp} j\!:\!D & \textit{si et seulement si} & \varepsilon_i(C^{I_i}) \cap \varepsilon_j(D^{I_j}) = \emptyset \\
\mathcal{I} \models_d i\!:\!R \xleftrightarrow{\perp} j\!:\!S & \textit{si et seulement si} & \varepsilon_i(R^{I_i}) \cap \varepsilon_j(S^{I_j}) = \emptyset \\
\mathcal{I} \models_d i\!:\!a \xleftrightarrow{\in} j\!:\!C & \textit{si et seulement si} & \varepsilon_i(a^{I_i}) \in \varepsilon_j(C^{I_j}) \\
\mathcal{I} \models_d i\!:\!a \xleftrightarrow{=} j\!:\!b & \textit{si et seulement si} & \varepsilon_i(a^{I_i}) = \varepsilon_j(b^{I_j})
\end{array}
$$

En outre, pour toute formule locale $i : \phi$, $\mathcal{I} \models_d i : \phi$ si et seulement si $I_i \models \phi$. Autrement dit, la satisfaction locale implique la satisfaction globale. Une interprétation distribuée $\mathcal{I}$ satisfait un alignement A si et seulement si elle satisfait toutes les correspondances de A (ce que l'on note $\mathcal{I} \models_d A$) et elle satisfait une ontologie O_i si et seulement si elle satisfait tous les axiomes de O_i (ce que l'on note aussi $\mathcal{I} \models_d O_i$). Quand toutes les ontologies et les alignements sont satisfaits, le réseau d'ontologies est satisfait par l'interprétation distribuée. Auquel cas nous appelons cette interprétation un *modèle* du réseau.

Definition 8.2.7 (Modèle d'un réseau d'ontologies alignées) *Soit $S = \langle \mathbf{O}, \mathbf{A} \rangle$ un réseau d'ontologies alignées. Une interprétation distribuée $\mathcal{I}$ est un* modèle *de S (ce que l'on note $\mathcal{I} \models_d S$), si et seulement si :*

– *pour tout* $O_i \in \mathbf{O}$*,* $\mathcal{I} \models_d O_i$ *;*
– *pour tout* $A_{ij} \in \mathbf{A}$*,* $\mathcal{I} \models_d A_{ij}$.

L'ensemble des modèles d'un réseau d'ontologies alignées est noté $\mathrm{Mod}(S)$. Une formule α est une conséquence d'un réseau d'ontologies alignées S (noté $S \models_d \alpha$) si et seulement si $\forall \mathcal{M} \in \mathrm{Mod}(S), \mathcal{M} \models_d \alpha$. Cette sémantique en théorie des modèles offre un défi particulier aux infrastructures de raisonnement, que nous allons présenter dans la section suivante.

8.3 Raisonnement en $\mathrm{IDDL}(\sqsubseteq_C, \perp_C)$

Cette section vise à présenter une procédure de raisonnement pour vérifier si un réseau d'ontologies alignées $S = \langle \mathbf{O}, \mathbf{A} \rangle$ est cohérent ou non, dans le cas où seul des concepts sont mis en correspondances. Les correspondances de rôles seront partiellement prises en compte dans la section 8.5.

Le problème de la déduction $S \models_d \alpha$ peut-être réduit à déterminer la cohérence ou l'incohérence d'un réseau d'ontologies alignées dans le cas où α est soit une inclusion de concept local ($i{:}C \sqsubseteq D$), une correspondance de concept ($i{:}C \overset{\sqsubseteq}{\leftrightarrow} j{:}D$ ou $i{:}C \overset{\perp}{\leftrightarrow} j{:}D$) ou une assertion de la ABox (i: $C(a)$). La réduction de la déduction locale à un problème de cohérence de réseau d'ontologies alignées est effectuée de la même manière que la réduction de la déduction à un problème de cohérence d'une ontologie dans les logiques de description. De surcroît, la déduction de correspondance comme $S \models_d i : C \overset{\sqsubseteq}{\leftrightarrow} j : D$ est équivalente à l'incohérence du réseau $S \cup \{i : \{a\} \overset{\sqsubseteq}{\leftrightarrow} i : C\} \cup \{i : \{a\} \overset{\perp}{\leftrightarrow} j : D\}$, où a dénote le nom d'un nouvel individu ajouté à l'ontologie O_i.

Quand des ontologies sont corrélées avec des alignements, de nouvelles déductions peuvent survenir. En effet, les connaissances inter-ontologies interagissent avec les connaissances locales. En outre, les connaissances d'une ontologie peuvent influencer les connaissances d'une autre ontologie. Par ailleurs, les connaissances locales peuvent aussi induire des connaissances inter-ontologies (c'est-à-dire des alignements). Et finalement, des déductions peuvent être faites avec et au sujet des alignements seuls.

En fait, la difficulté du raisonnement en IDDL réside dans le fait de déterminer quelles connaissances se propagent des domaines locaux au domaine global, ou du domaine global aux domaines locaux. Je montrerai que, dans le cas restreint où seuls des correspondances de concepts sont autorisées, seules l'insatisfiabilité et la non-vacuité de concepts sont propagées.

Exemple 8.3.1 *Voici quelques exemples de propagation de connaissances.*

La subsomption se propage du niveau local au niveau global. *Par exemple, dans la figure 8.1.1, les concepts* $Region$ *et* $Country$ *sont présents localement dans l'ontologie* O_3 *et aussi au niveau des alignements. Puisque dans* O_3*, on a l'axiome* $Country \sqsubseteq Region$*, on peut en déduire au niveau des alignements que* $3 : Country \overset{\sqsubseteq}{\longleftrightarrow} 3 : Region$*. La réciproque n'est pas vraie, c'est-à-dire que la subsomption ne se propage pas du niveau global au niveau local.*

L'exclusion se propage du niveau global au niveau local. *Par exemple, dans la figure 8.1.1, les concepts* $Region$ *et* $Woman$ *sont présents localement dans l'ontologie* O_1 *et aussi au niveau des alignements. En utilisant les alignements seuls, on peut déduire que* $1 : Woman \overset{\perp}{\longleftrightarrow} 1 : Region$*. Ceci implique au niveau local que* $1 : Woman \sqsubseteq \neg Region$*. La réciproque n'est pas vraie, c'est-à-dire que l'exclusion ne se propage pas du niveau local au niveau global.*

L'exemple ci-dessus montre que le raisonnement sur des réseaux d'ontologies IDDL n'est pas trivial et les algorithmes existant pour raisonner sur des ontologies en logiques de description (par exemple, des algorithmes de tableaux) ne peuvent pas être directement utilisés. L'exemple suivant montre comment ces deux faits suffisent à propager des connaissances en chaîne d'une ontologie à une autre.

Exemple 8.3.2 *Soit un réseau d'ontologies alignées comprenant n ontologies $O_1, \ldots, O_n$ et $n-1$ alignements $A_1, \ldots, A_{n-1}$ tels que A_i aligne les ontologies O_i et O_{i+1}. Pour chaque $1 \leq i \leq n$, l'ontologie O_i contient les concepts X_i, Y_i, Z_i et l'alignement A_i contient les correspondances $i : Y_i \xleftrightarrow{\perp} i+1 : X_{i+1}$ et $i : Z_i \xleftrightarrow{\equiv} i+1 : Z_{i+1}$. L'ontologie O_1 contient l'axiome $Z_1 \sqsubseteq Y_1$. L'ontologie O_n contient un individu a et l'axiome $X_n \sqcap Z_n(a)$. Les ontologies O_i, pour $i \notin \{1, n\}$ contiennent un seul axiome $\neg X_i \sqsubseteq Y_i$. La figure 8.2 illustre la situation. L'axiome de O_1 permet de déduire que $S \models_d 1 : Z_1 \xleftrightarrow{\sqsubseteq} 1 : Y_1$. Avec l'alignement*

$\mathbf{O}_1$		$\mathbf{O}_2$			$\mathbf{O}_n$
$\boxed{Z_1 \sqsubseteq Y_1}$	$1{:}Y_1 \xleftrightarrow{\perp} 2{:}X_2$ $1{:}Z_1 \xleftrightarrow{\equiv} 2{:}Z_2$	$\boxed{\neg X_2 \sqsubseteq Y_2}$	$2{:}Y_2 \xleftrightarrow{\perp} 3{:}X_3$ $2{:}Z_2 \xleftrightarrow{\equiv} 3{:}Z_3$	$\ldots$	$\boxed{X_n \sqcap Z_n(a)}$

FIGURE 8.2 – Ce réseau d'ontologies alignées montre que la propagation des connaissances peut impliquer des axiomes de toutes les ontologies en même temps.

A_1, on en déduit que $S \models_d 2{:}Z_2 \xleftrightarrow{\perp} 2{:}X_2$. De cela, on déduit que $S \models_d 2{:}Z_2 \sqsubseteq \neg X_2$ puis, grâce à l'axiome de O_2 que $S \models_d 2{:}Z_2 \sqsubseteq Y_2$. On retrouve la situation précédente et l'on peut reproduire le raisonnement en chaîne, jusqu'à obtenir que $S \models_d n : Z_n \sqsubseteq \neg X_n$. Le réseau d'ontologies est donc incohérent. Cependant, si l'on retire ne serait-ce qu'un seul axiome ou une seule correspondance, le réseau d'ontologies redevient cohérent. Cela montre qu'il peut être nécessaire de tenir compte de toutes les connaissances du réseau pour déterminer la cohérence d'un réseau d'ontologies.

Le principe derrière l'algorithme est fondé sur le fait que les correspondances sont similaires à des axiomes et que les alignements ressemblent à des ontologies. En fait, un alignement représente une ontologie qui serait interprétée dans le domaine d'interprétation global (voir la définition 8.2.4). Dans cet algorithme, les alignements sont traduits en une ontologie qu'on appelle *l'ontologie d'alignement* (voir la définition 8.3.5).

Cependant, il n'est pas possible de traduire l'ensemble du réseau d'ontologies alignées en une seule ontologie, car il se peut que la combinaison des constructeurs des logiques de description utilisées rendent la logique résultante indécidable [2]. En outre, on souhaite au maximum garantir l'encapsulation des ontologies dans les systèmes locaux, c'est-à-dire interdire l'accès aux axiomes des ontologies pour le processus de vérification de cohérence globale.

La procédure décrite ici va donc faire appel à des systèmes de raisonnement locaux déjà existant, sans présupposition sur leur implémentation. On peut donc imaginer qu'à un nœud se trouve le moteur d'inférence Fact++, à un autre nœud le moteur Racer ou bien Pellet, voire même le système d'inférence Drago d'un pair plongé dans un réseau pair-à-pair de type DDL. Pour déterminer la cohérence du réseau, on utilise un système de raisonnement supplémentaire qui accède à l'ensemble des alignements, mais ne peut accéder au contenu des ontologies. En revanche, il peut communiquer avec les systèmes de raisonnement locaux en leur demandant si certains axiomes sont cohérents vis-à-vis de leur ontologie.

2. Voir à ce sujet les travaux sur les fusions de logiques de description [9], qui sont d'ailleurs à l'origine de $\mathcal{E}$-connection.

Le problème de cohérence revient alors à se demander quels axiomes doivent être transmis aux ontologies pour déterminer la cohérence. Ces axiomes doivent donc traduire la propagation des connaissances.

En réalité, si les correspondances sont restreintes aux seules subsomption et exclusion de concepts inter-ontologies, il suffit de ne propager que l'insatisfiabilitité et la non-vacuité de certains concepts bien définis. En effet, si un concept est localement interprété comme vide, alors son image via la fonction d'égalisation est vide aussi. De même, un ensemble non vide a son image par ε non vide aussi.

Malheureusement, comme on l'a vu dans l'exemple 8.3.2, on ne peut savoir ce qui doit être propagé en interrogeant les ontologies une par une. En effet, la satisfiabilité d'un concept peut être indéterminée dans une ontologie, mais déterminée par la combinaison des connaissances des ontologies et des alignements.

Or, la section suivante montre qu'il suffit de tester la satisfiabilité d'un nombre fini de concepts dans le réseau pour déterminer sa cohérence. Ainsi, si l'on peut forcer un ensemble de concepts à être non vides parmi cet ensemble fini de concepts, tel que toutes les ontologies et les alignements restent cohérents, alors on peut affirmer la cohérence du réseau d'ontologies. Puisque cet ensemble est fini, un test exhaustif fournit nécessairement la preuve de la cohérence.

Plus précisément, l'algorithme consiste à choisir ce qu'on appelle une configuration, c'est-à-dire la donnée de l'ensemble des concepts non vides parmi l'ensemble fini susdésigné. Puisqu'une configuration force la non-vacuité de certains concepts, elle impose d'ajouter des axiomes à l'ontologie d'alignement évoquée ci-dessus et oblige à tester la cohérence de cette nouvelle ontologie, puis à transmettre aux systèmes locaux des axiomes adéquats traduisant la propagation de connaissances. Si les ontologies locales étendues par ces axiomes sont toutes cohérentes, alors le réseau d'ontologies est cohérent. Sinon, on teste une autre configuration, jusqu'à en trouver une satisfaisante, ou bien jusqu'à épuisement des configurations si le réseau d'ontologies est incohérent.

On introduit dans ce qui suit la construction des ontologies étendues, élaborées à partir de $\mathbf{A}$ et de $\mathbf{O}$. On montrera que la cohérence d'un réseau IDDL S est équivalente à l'existence de telles ontologies étendues telles qu'elles sont cohérentes.

8.3.1 Configurations et ontologies étendues

Cette section présente les définitions formelles qui mèneront finalement à la construction des ontologies étendues évoquées plus haut. Une configuration détermine si certains concepts bien choisis dans un vocabulaire sont insatisfiables ou non-vides. Dans notre cas spécifique, le vocabulaire en question est défini grâce aux correspondances. Il sera établi qu'il est suffisant de ne considérer que les concepts apparaissant dans les correspondances lorsque l'on traite la propagation de connaissances en IDDL.

Les concepts apparaissant à gauche ou à droite des correspondances des alignements constituent le vocabulaire d'une ontologie d'alignement (voir la définition 8.3.5), appelé le *vocabulaire global*. Ce vocabulaire est constitué de *vocabulaires locaux* provenant des ontologies locales. Les définitions suivantes introduisent formellement la construction de ces éléments qui seront utilisés pour formuler les axiomes et assertions dans l'ontologies d'alignment et les ontologies étendues.

Definition 8.3.1 (Vocabulaires locaux) *Soit $S = \langle \mathbf{O}, \mathbf{A} \rangle$ un réseau d'ontologies alignées. On note $\mathscr{C}_i$ l'ensemble qui inclut le concept universel $\top$ et tous les concepts (qu'ils soient primitifs ou complexes) apparaissant en partie gauche d'une correspondance de A_{ij} ou en partie droite* *d'une correspondance de A_{ji}.*

Definition 8.3.2 (Vocabulaire global) *Soit $S = \langle \mathbf{O}, \mathbf{A} \rangle$ un réseau d'ontologies alignées. L'ensemble de noms de concepts globaux de S est $\mathscr{C} = \bigcup_{i \in \mathbf{O}} \{i : C \mid C \in \mathscr{C}_i\}$. Lorsque $w \subseteq \mathscr{C}_i$, $\widehat{w}$ dénote l'ensemble $\{i : C \mid C \in w\} \cup \{\top\}$ de noms de concepts globaux. Quand $W \subseteq \mathscr{C}$, on note $W|_i$ l'ensemble $\{C \in \mathscr{C}_i \mid i : C \in W\}$.*

Si l'on considère à nouveau l'exemple de la figure 8.1.1, les vocabulaires locaux sont

$$\mathscr{C}_1 = \{\mathit{Woman}, \mathit{Guyana}, \top\}$$

$$\mathscr{C}_2 = \{\mathit{Femme}, \top\}$$

$$\mathscr{C}_3 = \{\mathit{Region}, \mathit{SouthAmericanRegion}, \exists \mathit{partOf}.\{\mathit{France}\}, \top\}$$

et le vocabulaire global est $\mathscr{C} = \{1 : \mathit{Woman}, 1 : \mathit{Guyana}, 1 : \top, 2 : \mathit{Femme}, 2 : \top, 3 : \mathit{Region}, 3 : \mathit{SouthAmericanRegion}, 3 : \exists \mathit{partOf}.\{\mathit{France}\}, 3 : \top\}$.

On peut remarquer le concept $3 : \exists \mathit{partOf}.\{\mathit{France}\}$ qui, dans le vocabulaire global, est un concept atomique, bien qu'il soit localement composé. En fait, le vocabulaire global ne contient que des concepts atomiques.

Dans l'algorithme, on ne détermine pas directement la vacuité ou non des concepts globaux ou locaux. En fait, une configuration définit la vacuité de concepts particuliers, que l'on appelle les *atomes* du vocabulaire. Ce terme provient de [57] et l'ensemble des atomes forme la *décomposition atomique* du vocabulaire.

Definition 8.3.3 *Soit T un ensemble de concepts (primitifs ou complexes) contenant $\top$. Pour chaque sous-ensemble non vide $W \subseteq T$, le concept $C_W^T := (\bigsqcap_{X \in W} X \sqcap \bigsqcap_{X' \in T \setminus W} \neg X')$ est un atome vis-à-vis de T.*

De la définiton 8.3.3, il suit que tous les concepts C_W^T sont disjoints et leur union est équivalente au concept universel $\top$. Par conséquence, une interprétation du vocabulaire T permet d'associer aux concepts C_W^T une partition du domaine d'interprétation. La figure 8.3 illustre ces propriétés.

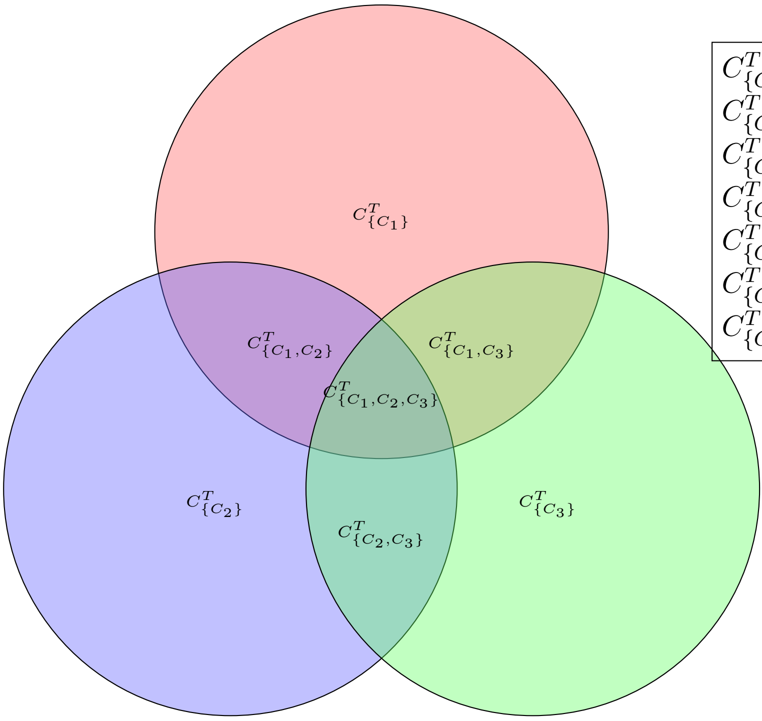
Quand il n'y a pas d'ambigüité, on notera C_W le concept $C_W^{\mathscr{C}}$ défini ci-dessus lorsque $\mathscr{C}$ est le vocabulaire global du réseau d'ontologies alignées S.

Exemple 8.3.3 *Si l'on choisit $T = \mathscr{C}_3$ avec $\mathscr{C}_3$ comme dans l'exemple ci-dessus et $W = \{\exists \mathit{partOf}.\{\mathit{France}\}, \top\}$, alors*

$$C_W^T = \exists \mathit{partOf}.\{\mathit{France}\} \sqcap \top \sqcap \neg \mathit{Region} \sqcap \neg \mathit{SouthAmericanRegion}$$

On peut remarquer que $\top$ est superflu dans cette conjonction, mais il n'en est pas de même pour $3 : \top$ lorsque l'on considère le vocabulaire global.

En s'appuyant sur les concepts C_W^T, on peut définir une relation d'équivalence sur l'ensemble des interprétations de T comme suit : deux interprétations appartiennent à classe d'équivalence si pour chaque sous-ensemble $W \subseteq \mathscr{C}$ les deux interprétations du concept $C_W^{\mathscr{C}}$ (introduit par la définition 8.3.3) sont soit toutes les deux vides soit toutes les deux non-vides. La notion de *configuration* définie ci-dessous représente une telle classe d'équivalence. Par commodité, une configuration n'indique que le sous-ensemble des atomes qui seront considérés comme non-vides, tandis que les autres seront considérés comme vides (ou insatisfiables).



$$
\begin{aligned}
C^T_{\{C_1,C_2,C_3\}} &= C_1 \sqcap C_2 \sqcap C_3\\
C^T_{\{C_1,C_2\}} &= C_1 \sqcap C_2 \sqcap \neg C_3\\
C^T_{\{C_1,C_3\}} &= C_1 \sqcap \neg C_2 \sqcap C_3\\
C^T_{\{C_1\}} &= C_1 \sqcap \neg C_2 \sqcap \neg C_3\\
C^T_{\{C_2,C_3\}} &= \neg C_1 \sqcap C_2 \sqcap C_3\\
C^T_{\{C_2\}} &= \neg C_1 \sqcap C_2 \sqcap \neg C_3\\
C^T_{\{C_3\}} &= \neg C_1 \sqcap \neg C_2 \sqcap C_3
\end{aligned}
$$

FIGURE 8.3 – Décomposition atomique du vocabulaire $T = \{C_1, C_2, C_3\}$. Dans ce diagramme, tous les atomes apparaissent comme non-vides, mais ce n'est pas nécessairement le cas. En revanche, il n'est pas possible que deux atomes se superposent.

Par conséquent, une configuration se décrit formellement comme un simple choix de sous-ensemble des atomes.

Definition 8.3.4 (Configuration Globale) *Soit S un réseau d'ontologies alignées avec un ensemble de noms de concepts globaux $\mathscr{C}$. Une* configuration globale *de S est un sous-ensemble Ω de $2^{\mathscr{C}}$.*

Les configurations sont essentielles parce que, comme nous le verrons, la cohérence d'un réseau d'ontologies peut être assimilée à l'existence d'une configuration appropriée. Ainsi, plutôt que de rechercher une fonction d'égalisation parmi l'infinité de fonctions possibles, il suffit d'explorer les configurations qui sont en nombre finies. Néanmoins, pour ce faire, il faut d'abord traduire les configurations sous forme d'axiomes et assertions qui expriment la non-vacuité et l'insatisfiabilité, respectivement .

Les définitions précédentes ont introduit les éléments nécessaires à la construction de la dite *ontologie d'alignement*. Cette ontologie "axiomatise" les alignements, qui représentent la connaissance inter-ontologies. Mis à part les axiomes exprimant les correspondances des alignements, une ontologie d'alignement inclut aussi des axiomes additionnels ou des assertions représentant la configuration globale.

Definition 8.3.5 (Ontologie d'alignement) *Soit $S = \langle \mathbf{O}, \mathbf{A} \rangle$ un réseau d'ontologies alignées. Soit Ω une configuration globale de S. L'*ontologie d'alignement *vis-à-vis de Ω est une ontologie $\widehat{\mathbf{A}}_\Omega$ définie comme suit :*

1. *pour tout $i, j \in \mathbf{O}$, si $i{:}C \overset{\sqsubseteq}{\leftrightarrow} j{:}D$ (respectivement $i{:}C \overset{\perp}{\leftrightarrow} j{:}D$) est une correspondance de concept dans $\mathbf{A}$ alors $i{:}C \sqsubseteq j{:}D$ (respectivement $i{:}C \sqsubseteq \neg j{:}D$) est un axiome de $\widehat{\mathbf{A}}_\Omega$;*

2. *pour tout* $W \in \Omega$, $C_W \equiv \{a_W\}$ *est un axiome de* $\widehat{\mathbf{A}}_\Omega$ *où* a_W *est un nouveau nom d'individu ;*
3. *pour tout* $W \notin \Omega$, $C_W \sqsubseteq \perp$ *est un axiome de* $\widehat{\mathbf{A}}_\Omega$.

Dans cette définition, on remarque que l'item 2 ne se contente pas d'affirmer la non-vacuité du concept C_W. On impose que le concept C_W ne contienne qu'un seul élément, mais la justification de ce choix impose d'examiner précisément la preuve du théorème qui va suivre. Plus précisément, le fait de caractériser entièrement le concept C_W permet d'éviter une possible ambigüité lorsque l'on choisit un élément quelconque de l'interprétation de C_W. Ce dernier point est crucial dans la preuve, que l'on trouve en annexe E.

Reconsidérons l'exemple 8.3.2. Si l'on choisit la configuration $\Omega = 2^{\mathscr{C}}$ pour construire l'ontologie d'alignement $\widehat{\mathbf{A}}$ selon la définition 8.3.5, alors $\widehat{\mathbf{A}}$ est incohérente.

La construction des configurations locales est très similaire à celle de la configuration globale exceptée que la compatibilité des configurations locales avec une configuration globale donnée doit être prise en compte. Cette compatibilité résulte de la sémantique des réseaux d'ontologies IDDL dépendant des fonctions d'égalisation, qui impose que si l'image d'un ensemble par une fonction d'égalisation est non vide alors cet ensemble doit être non vide.

Definition 8.3.6 (Configuration locale) *Soit* $S = \langle \mathbf{O}, \mathbf{A} \rangle$ *un réseau d'ontologies alignées. Soit* Ω *une configuration globale de* S*. Pour chaque* $O_i \in \mathbf{O}$*, on définit une* configuration locale *de* O_i *vis-à-vis de* Ω *comme un sous-ensemble* Ω^i *de* $2^{\mathscr{C}_i}$*. En outre, si* $w \in \Omega^i$ *alors il doit exister* $W \in \Omega$ *tel que* $\widehat{w} \subseteq W$.

Comme indiqué au début de cette section, la propagation des connaissances des alignements vers les ontologies locales est cruciale pour la construction des ontologies étendues préservant la cohérence du réseau IDDL. Une configuration globale Ω et une configuration locale Ω^i compatible avec Ω fournissent les éléments nécessaires pour définir une telle ontologie étendue. La définition suivante décrit comment propager les connaissances des alignements vers les ontologies locales via les configurations déterminées.

Definition 8.3.7 (Ontologie étendue) *Soit* $S = \langle \mathbf{O}, \mathbf{A} \rangle$ *un réseau d'ontologies alignées. Soit* Ω *une configuration globale pour* S*. Pour tout* $i \in \mathbf{O}$*, soit* Ω^i *une configuration locale vis-à-vis de* Ω*. L'*ontologie étendue $\widehat{O}_{\Omega^i}$ *vis-à-vis de* Ω^i *et de* Ω *est définie comme suit :*

1. $O_i \subseteq \widehat{O}_{\Omega^i}$ *;*
2. *pour tout* $w \in \Omega^i$, $C_w^{\mathscr{C}_i}(b_w)$ *est un axiome de* $\widehat{O}_{\Omega^i}$*, où* b_w *est un nouveau nom d'individu ;*
3. *pour tout* $w \notin \Omega^i$, $C_w^{\mathscr{C}_i} \sqsubseteq \perp$ *est un axiome de* $\widehat{O}_{\Omega^i}$ *;*
4. *pour tout* $W \in \Omega$ *et tout* $X \in W|_i$*, on définit un nouveau concept* C_W^X *pour l'ontologie* $\widehat{O}_{\Omega^i}$ *tel que :*
   (a) $C_W^X \sqsubseteq X \sqcap \bigsqcap_{X' \in \mathscr{C}_i \setminus W|_i} \neg X'$ *est un axiome de* $\widehat{O}_{\Omega^i}$ *;*
   (b) $C_W^X(b_W^X)$ *est un axiome de* $\widehat{O}_{\Omega^i}$ *avec* b_W^X *un nouveau nom d'individu dans* $\widehat{O}_{\Omega^i}$ *;*
   (c) $C_W^X \sqsubseteq C_W^\top$ *est un axiome de* $\widehat{O}_{\Omega^i}$ *;*
5. *pour tout* $W, W' \subseteq \mathscr{C}$ *tel que* $W \neq W'$, $C_W^\top \sqsubseteq \neg C_{W'}^\top$ *est un axiome de* $\widehat{O}_{\Omega^i}$.

Les items à partir de l'item 4 peuvent être vus comme la conséquence de la propagation des connaissances dues à la configuration globale. Comme on l'a remarqué auparavant, un sous-ensemble du domaine d'interprétation global est vide si et seulement si son image réciproque par la fonction d'égalisation est vide aussi. Cela se traduit par le fait que si un concept

$i\!:\!X$ de l'ontologie d'alignement est insatisfiable, alors il faut que son homologue local X dans l'ontologie O_i soit insatisfiable aussi. De même, si $i:X$ est non-vide (c'est-à-dire qu'il existe un individu dans le concept $i\!:\!X$), alors il doit exister un individu dans le concept local X.

Cependant, il n'est pas suffisant de propager l'insatisfiabilité ou la non-vacuité terme par terme. En effet, si l'on suppose que $(1\!:\!C)\sqcap\neg(1\!:\!D)$, alors il doit exister un individu local dans C *qui n'est pas dans* D. Ce point justifie l'item 4a. Plus précisément, le concept C_W^X dénote l'ensemble des individus qui sont dans le concept X mais ne se trouvent pas dans les concepts niés de C_W. L'item 4b exprime le fait que ce concept est non-vide, en nommant explicitement un nouvel individu pour forcer la non-vacuité du concept. D'après ces remarques, on constate donc que $C_W^\top$ doit normalement jouer le rôle de tous les individus qui ne sont pas dans les concepts niés de C_W. Mais pour être sûr que cela soit bien établi, on ajoute l'item 4c. Enfin, on a déjà fait remarquer que si $W \neq W'$ alors les concepts C_W et $C_{W'}$ de l'ontologie d'alignement sont disjoints. Par conséquent, leurs homologues locaux (ou, pourrait-on dire, leur antécédent) doivent être disjoints aussi. C'est pourquoi l'on ajoute l'item 5.

Exemple 8.3.4 *Soient deux ontologies O_1 et O_2. Elles sont alignées à l'aide des correspondances $1\!:\!Artiste \xleftrightarrow{\sqsubseteq} 2\!:\!Person$, $1\!:\!Film \xleftrightarrow{\sqsubseteq} 2\!:\!\top$ et $1\!:\!Livre \xleftrightarrow{\perp} 2\!:\!Person$. On choisit une configuration dans laquelle on trouve $W = \{1\!:\!Artiste, 1\!:\!\top, 2\!:\!\top, \top\}$. Cela signifie que l'on suppose que le concept*

$$C_W = (1\!:\!Artiste) \sqcap (1\!:\!\top) \sqcap \neg(1\!:\!Livre) \sqcap (2\!:\!Person) \sqcap \neg(2\!:\!\top) \sqcap \top$$

est non vide dans l'ontologie d'alignement.

Pour "propager" cette non-vacuité aux ontologies locales, on ajoute dans O_1 les concepts $C_W^{Artiste}$ et $C_W^\top$. On peut interpréter ces deux concepts comme des ensembles d'antécédents des éléments de C_W par une fonction d'égalisation. Une telle fonction préserve la vacuité et la non-vacuité. Or, il est explicitement indiqué que $C_W^{Artiste}$ et $C_W^\top$ sont non vides grâce aux individus ajoutés $b_W^{Artiste}$ et $b_W^\top$.

Le théorème suivant établit le résultat le plus important du présent chapitre. Il assure que la cohérence d'un réseau IDDL équivaut à l'existence d'une ontologie d'alignement et d'ontologies étendues qui préservent la sémantique du réseau IDDL.

Théorème 8.3.1 (Cohérence d'un réseau d'ontologies alignées) *Soit $S = \langle \mathbf{O}, \mathbf{A}\rangle$ un réseau d'ontologies alignées en IDDL($\sqsubseteq_C, \perp_C$). S est cohérent si et seulement s'il existe une configuration globale Ω pour S et une configuration locale Ω^i pour chaque $O_i \in \mathbf{O}$ vis-à-vis de Ω telles que l'ontologie d'alignement $\widehat{\mathbf{A}}_\Omega$ et les ontologies étendues $\{\widehat{O}_{\Omega^i}\}$ comme définis ci-dessus sont cohérentes.*

Preuve : L'intégralité de la preuve est donnée en annexe E, mais j'en présente ici quelques points clé pour en comprendre le principe. On subdivise la preuve en deux grandes parties, l'une pour l'implication directe, l'autre pour la réciproque.

Commençons par la réciproque. On suppose que les ontologies étendues existent et sont cohérentes. On cherche alors à définir un modèle du réseau. Les modèles locaux sont directement donnés par les modèles des ontologies étendues, mais il reste à construire la fonction d'égalisation. On choisit d'associer à tout élément e des domaines d'interprétation locaux, un élément du domaine d'interprétation de l'ontologie d'alignement correspondant à un individu nommé (a_W). Le choix de l'individu a_W est guidé par le fait que e appartient à l'interprétation d'un $C_W^\top$ ou non (voir le détail en annexe). On montre que ce choix établit un modèle du réseau d'ontologies en montrant d'abord que ε associe à tout élément e d'un domaine local Δ_i un individu a_W de l'ontologie d'alignement, tel que l'ensemble de concepts W comprend tous les concepts de $\mathscr{C}_i$ dont l'interprétation contient e. La vérification de la satisfaction du réseau d'ontologies se fait simplement en appliquant judicieusement ce résultat et par des manipulations d'ensembles un peu techniques mais simples.

Pour l'implication directe, on suppose que l'on dispose d'un modèle du réseau d'ontologies alignées. La preuve est organisée comme suit :

1. Construire les configurations en utilisant les modèles $\langle \mathcal{I}, \varepsilon \rangle$:
   (a) construire la configuration globale Ω ;
   (b) construire les configurations locales Ω^i pour tout $i \in \mathbf{O}$;
   (c) prouver que Ω^i est effectivement une configuration locale.
2. Prouver que $\widehat{\mathbf{A}}_\Omega$ est cohérente :
   (a) construire une interprétation $\mathcal{I}_A$ de $\widehat{\mathbf{A}}_\Omega$;
   (b) montrer que $\mathcal{I}_A \models \widehat{\mathbf{A}}_\Omega$.
3. Prouver que $\{\widehat{O}_{\Omega^i}\}$ est cohérent pour tout $i \in \mathbf{O}$:
   (a) construire une interprétation $\mathcal{I}_i$ de $\widehat{O}_{\Omega^i}$;
   (b) montrer que $\mathcal{I}_i \models \widehat{O}_{\Omega^i}$.

La construction des configurations est immédiate à partir du modèle du réseau d'ontologies. Pour la construction de l'interprétation de l'ontologie d'alignement, l'astuce consiste à utiliser la configuration elle-même comme domaine d'interprétation. Un concept global est alors interprété comme l'ensemble des W de la configuration qui contiennent le concept. Un individu a_W est simplement interprété par W. La construction d'une interprétation des ontologies étendues exige la démonstration d'un lemme intermédiaire, mais reste assez simple et directe. □

8.4 Un algorithme pour vérifier la cohérence

De la construction de l'ontologie d'alignement et des ontologies étendues introduites en section 8.3, on peut concevoir un algorithme (algorithme 8.4) pour raisonner sur un réseau d'ontologies IDDL S. Cet algorithme recherche des configurations valides à partir desquelles une ontologie d'alignement cohérente et des ontologies étendues cohérentes peuvent être construites. Si l'algorithme ne trouve aucune configuration satisfaisante alors S est incohérent. La correction de l'algorithme est assurée par le théorème 8.3.1.

Propriété 8.4.1 *Étant donnés c le nombre de concepts globaux dans $\mathscr{C}$ et N le nombre d'ontologies dans le réseau, le nombre d'appels à la vérification de cohérence locale pour l'algorithme 8.4 est borné par $N2^{(2^{c+1})}$. En outre, la taille des l'ontologies étendues est de l'ordre de $O(2^c)$.*

Preuve : Il y a autant de configurations globales que de sous-ensembles de $2^{\mathscr{C}}$, à savoir $2^{(2^c)}$. Pour toute configuration globale, toutes les configurations locales doivent être testées, pour chaque ontologie. Le nombre de configurations locales, pour une ontologie donnée dans le réseau, est inférieur à $2^{(2^c)}$ et il y a N ontologies à vérifier. Donc le nombre total de vérifications de cohérence est borné par $N2^{(2^c)} \cdot 2^{(2^c)} = N2^{(2^{c+1})}$. □

La propriété 8.4.1 montre que la complexité de l'algorithme 8.4 peut être déterminée à partir de la complexité des algorithmes de vérification de cohérence locaux. Quelle que soit la complexité de l'algorithme local, la complexité de la vérification de cohérence globale est au moins dans un ordre complexité comparable à une fonction double exponentielle de la taille des alignements. Dans le cas où l'algorithme local est lui-même doublement exponentiel (ce qui peut arriver avec certains algorithmes de tableaux sur des logiques de description très expressives), la vérification de cohérence globale par cet algorithme se retrouve dans une classe

Algorithm 1 Cohérence($\langle \mathbf{O}, \mathbf{A} \rangle$)

Require: $S = \langle \mathbf{O}, \mathbf{A} \rangle$ with $\mathbf{A} = \{A_{ij} \mid i, j \in \mathbf{O}\}$
Ensure: EstCohérent(S)

construire $\mathscr{C}$ selon la définition 8.3.2
for all ontologie $i \in \mathbf{O}$ **do**
  construire $\mathscr{C}_i$ selon la définition 8.3.1
end for
for all ensemble non vide $W \subseteq \mathscr{C}$ **do**
  définir C_W selon la définition 8.3.3
end for
for all ontologie $i \in \mathbf{O}$ **do**
  for all ensemble non vide $w \subseteq \mathscr{C}_i$ **do**
    définir $C_w^{\mathscr{C}_i}$ selon la définition 8.3.3
  end for
end for
for all configuration globale $\Omega \subseteq 2^{\mathscr{C}}$ **do**
  construire l'ontologie d'alignement $\widehat{\mathbf{A}}_\Omega$ à partir de Ω et $\mathbf{A}$ selon la définition 8.3.5
  if $\widehat{\mathbf{A}}_\Omega$ est cohérente **then**
    for all ontologie $i \in \mathbf{O}$ **do**
      soit $\widehat{\Omega}^i$ l'ensemble de toutes les configurations locales vis-à-vis de Ω
    end for
    for all famille de configurations locales $(\Omega^i)_{i \in \mathbf{J}}$ **do**
      for all ontologie $i \in \mathbf{O}$ **do**
        construire les ontologies étendues $\widehat{O}_{\Omega^i}$ à partir de Ω, Ω^i et O_i selon la définition 8.3.7
      end for
      if $\widehat{O}_{\Omega^i}$ est cohérente pour tout $i \in \mathbf{O}$ **then**
        return *OUI*
      end if
    end for
  end if
end for
return *NON*

de complexité triplement exponentielle, c'est-à-dire de l'ordre de $2^{(2^{(2^c)})}$. Ceci provient du fait que le raisonneur local est interrogé sur des données de taille exponentielle par rapport aux nombres de concepts globaux. En effet, l'algorithme soumet à ces raisonneurs $O(2^c)$ axiomes, ce qui accroit d'un exposant la complexité locale.

On peut tout de même nuancer ce défaut important par les propriétés intéressantes que nous avons garanties et par le fait qu'il s'agit là d'un algorithme naïf. Premièrement, la correction et la complétude de cet algorithme prouve que le formalisme IDDL est décidable dès lors que les logiques locales sont décidables. Deuxièmement, les logiques locales n'ont pas besoin d'être connues, ni les procédures de décision locales. Ainsi, les ontologies peuvent être encapsulées dans une interface qui n'expose que le résultat de la vérification de cohérence de l'ontologie étendue. D'ailleurs, l'expressivité des axiomes ajoutés aux ontologies étendues est restreinte à $\mathcal{ALC}$, ce qui la rend accessible à de nombreux raisonneurs de logiques de description. L'algorithme proposé offre donc aussi un avantage en termes de polyvalence.

Avant d'aborder de possibles optimisations en section 8.6, étudions d'abord le raisonnement lorsque l'on augmente l'expressivité du langage d'alignements. Dans la section qui suit, on ajoute l'exclusion de rôles aux correspondances autorisées.

8.5 Raisonnement en $\mathrm{IDDL}(\sqsubseteq_C, \perp_C, \sqsubseteq_R)$

Cette section explique les modifications à apporter à la procédure précédemment décrite pour qu'elle traite aussi la subsomption de rôles inter-ontologies. Le principe du raisonnement reste le même qu'auparavant, excepté que les configurations doivent être étendues pour prendre en compte les rôles impliqués dans les correspondances. Puisque la plupart des définitions nécessaires pour cette partie sont les mêmes ou presque que celles de la section précédente, on se contentera de mettre à jour les définitions modifiées, ou bien d'ajouter les nouvelles définitions.

D'abord, une nouvelle notion de vocabulaire doit être définie pour les rôles, localement et globalement.

Definition 8.5.1 (Vocabulaire local de rôles) *Soit $S = \langle \mathbf{O}, \mathbf{A} \rangle$ un réseau d'ontologies alignées. On note $\mathscr{R}_i$ l'ensemble qui inclut les rôles primitifs ou complexes apparaissant à gauche d'une correspondance de A_{ij} ou à droite d'une correspondance de A_{ji} de même que leur inverse (c'est-à-dire, $R \in \mathscr{R}_i \Leftrightarrow R^- \in \mathscr{R}_i$).*

Definition 8.5.2 (Vocabulaire global de rôles) *Soit $S = \langle \mathbf{O}, \mathbf{A} \rangle$ un réseau d'ontologies alignées . L'ensemble de noms de rôles globaux de S est $\mathscr{R} = \bigcup_{i \in \mathbf{O}} \{i\!:\!R \mid R \in \mathscr{R}_i\}$.*

Il faut maintenant définir un nouveau type de configurations qui servira à décrire une autre classe d'équivalence sur les interprétations, avec une condition sur les interprétations de rôles au lieu des concepts. Néanmoins, le traitement des rôles est assez différent de celui des concepts, parce qu'il y a des interactions entre les rôles et les concepts. De ce fait, il est nécessaire d'avoir un suivi de la satisfaisabilité des rôles en sus de la satisfaisabilité des concepts.

Plus précisément, on considère une configuration de concept Ω qui représente une partition du domaine d'interprétation. Puis, suivant la configuration, on définit la configuration de rôles comme une famille de relations sur Ω indexée par l'ensemble des rôles. En d'autres termes, une configuration de rôles détermine s'il existe une relation R entre deux ensembles dans la "partition" Ω.

Definition 8.5.3 (Configuration globale de rôles) *Soit S un réseau d'ontologies alignées avec un ensemble de noms de rôles globaux $\mathscr{R}$. Soit Ω une configuration globale pour S. Une* configuration globale de rôles *pour S vis-à-vis de Ω est un sous-ensemble Φ_Ω de $\Omega \times \Omega \times \mathscr{R}$.*

L'introduction de configuration de rôles mène à des contraintes supplémentaires sur l'ontologie d'alignement, que l'on ajoute à la définition 8.3.5.

Definition 8.5.4 (Ontologie d'alignement (révisée)) *Soit $S = \langle \mathbf{O}, \mathbf{A} \rangle$ un réseau d'ontologies alignées. Soit Ω une configuration globale pour S et soit Φ_Ω une configuration de rôles vis-à-vis Ω.* L'ontologie d'alignement *vis-à-vis de Ω et de Φ_Ω est une ontologie $\widehat{\mathbf{A}}_\Omega$ définie comme suit :*

1. *2. et 3. Voir la définition 8.3.5 ;*

4. *pour tout $i, j \in \mathbf{O}$, si $i\!:\!R \overset{\sqsubseteq}{\longleftrightarrow} j\!:\!S$ est une correspondance dans $\mathbf{A}$ alors $i\!:\!R \sqsubseteq j\!:\!S$ est un axiome d'inclusion de rôles dans $\widehat{\mathbf{A}}_\Omega$;*
5. *pour tout $\langle W, W', R \rangle \in \Phi_\Omega$, $\{a_W\} \sqsubseteq \exists R.\{a_{W'}\}$ est un axiome dans $\widehat{\mathbf{A}}_\Omega$;*
6. *pour tout $\langle W, W', R \rangle \notin \Phi_\Omega$, $\{a_W\} \sqsubseteq \forall R.\neg\{a_{W'}\}$ est un axiome dans $\widehat{\mathbf{A}}_\Omega$;*

Les axiomes introduits par les items 5 et 6 traduisent la sémantique de la configuration de rôles. Lorsqu'un triplet $\langle W, W', R\rangle$ appartient à la configuration, cela impose l'existence d'une connection par R des deux atomes W et W'. La configuration de rôles est donc bien plus complexe que la configuration (de concepts), puisqu'elle ne se contente pas d'identifier la vacuité ou non des rôles. Elle indique aussi entre quels atomes il existe une relation.

De même que pour les configurations locales, les configurations locales de rôles doivent satisfaire une certaine contrainte imposée par la configuration globale de rôles.

Definition 8.5.5 (Configuration locales de rôles) *Soit S un réseau d'ontologies alignées avec un ensemble de noms de rôles globaux $\mathscr{R}$. Soit Ω une configuration globale pour S, soit Φ_Ω une configuration globale de rôles vis-à-vis de Ω et Ω^i une configuration locale pour O_i vis-à-vis de Ω. Une* configuration locale de rôles *pour O_i vis-à-vis de Ω, Ω_i et Φ_Ω est un sous-ensemble Φ_{Ω^i} de $\Omega^i \times \Omega^i \times \mathscr{R}_i$ tel que $\langle w, w', R\rangle \in \Phi_{\Omega^i}$ implique qu'il existe $W, W' \in \Omega$ tels que $w \subseteq W|_i$, $w' \subseteq W'|_i$ et $\langle W, W', i\!:\!R\rangle \in \Phi_\Omega$.*

Les ontologies étendues sont encore enrichies d'axiomes mettant en jeu des rôles.

Definition 8.5.6 (Ontologie étendue (révisée)) *Soit $S = \langle \mathbf{O}, \mathbf{A}\rangle$ un réseau d'ontologies alignées. Soit Ω une configuration globale et Φ_Ω une configuration globale de rôles pour S. Pour tout $O_i \in \mathbf{O}$, soit Ω^i une configuration locale vis-à-vis de Ω et Φ_{Ω^i} une configuration locale de rôles vis-à-vis de Ω, Ω^i et Φ_Ω. L'*ontologie étendue *$\widehat{O}_{\Omega^i}$ vis-à-vis de Ω^i, Ω, Φ_Ω et de Φ_{Ω^i} est définie comme suit :*

1. *2. 3. 4. et 5. Voir la définition 8.3.7 ;*
6. *pour tout $\langle w, w', R\rangle \in \Phi_{\Omega^i}$, $(C_w^{\mathscr{C}_i} \sqcap \exists R.C_{w'}^{\mathscr{C}_i})(b_{w,w'}^R)$ est un axiome de $\widehat{O}_{\Omega^i}$, où $b_{w,w'}^R$ est un nouveau nom d'individu ;*
7. *pour tout $\langle w, w', R\rangle \notin \Phi_{\Omega^i}$, $C_w^{\mathscr{C}_i} \sqsubseteq \forall R.\neg C_{w'}^{\mathscr{C}_i}$ est un axiome de $\widehat{O}_{\Omega^i}$;*
8. *pour tout $W, W' \subseteq \Omega$ et tout $R \in \mathscr{R}_i$, on définit un nouveau nom de concept $C_{W,W'}^R$ pout l'ontologie $\widehat{O}_{\Omega^i}$ tel que :*
   (a) *$C_{W,W'}^R \sqsubseteq C_W^\top$ est un axiome de $\widehat{O}_{\Omega^i}$;*
   (b) *if $\langle W, W', i : R\rangle \in \Phi_\Omega$ alors $C_{W,W'}^R \sqsubseteq \exists R.C_{W',W}^{R^-}$ et $C_{W,W'}^R(\beta_{W,W'}^R)$ sont des axiomes de $\widehat{O}_{\Omega^i}$ avec $\beta_{W,W'}^R$ un nouveau nom d'individu ;*
   (c) *sinon, $C_{W,W'}^R \sqsubseteq \forall R.\neg C_{W',W}^{R^-}$ est un axiome de $\widehat{O}_{\Omega^i}$;*
9. *pour tout $R \in \mathscr{R}_i$, $\exists R.\top \sqsubseteq \bigsqcup_{W,W' \in \Omega} C_{W,W'}^R$ est un axiome de $\widehat{O}_{\Omega^i}$.*

Dans la définition précédente, l'item 6 signifie qu'un triplet $\langle w, w', R\rangle$ dans la configuration locale de rôles détermine l'existence d'une relation R entre une certaine instance du concept $C_w^{\mathscr{C}_i}$ et une certaine instance du concept $C_{w'}^{\mathscr{C}_i}$. Inversement, l'item 7 signifie que lorsqu'un triplet $\langle w, w', R\rangle$ n'est pas dans la configuration locale de rôles, alors les concepts $C_w^{\mathscr{C}_i}$ et $C_{w'}^{\mathscr{C}_i}$ ne sont pas reliés par R. L'item 8 ajoute un concept $C_{W,W'}^R$ qui représente l'ensemble des instances du concept local $C_W^\top$ qui ont leur homologue dans le concept global C_W et sont en relation par R avec une instance qui a son homologue dans $C_{W'}$. Enfin, l'item 9 assure que n'importe quelle instance impliquée dans une relation R doit appartenir à l'un des concepts $C_{W,W'}^R$ nouvellement introduits pour un certain W et un certain W'. Ce dernier item est important pour s'assurer que la structure des rôles est correctement propagée.

Théorème 8.5.1 (Cohérence d'un réseau d'ontologies alignées) *Soit $S = \langle \mathbf{O}, \mathbf{A}\rangle$ un réseau d'ontologies alignées. S est* cohérent *si et seulement si il existe une configuration globale Ω*

pour S, une configuration globale de rôles Φ_Ω vis-à-vis de Ω, des configurations locales Ω^i pour tout $O_i \in \mathbf{O}$ vis-à-vis de Ω et des configurations locales de rôles Φ_{Ω^i} vis-à-vis de Ω, Ω^i et Φ_Ω, telles que l'ontologie d'alignement $\widehat{\mathbf{A}}_\Omega$ et les ontologies locales étendues $\{\widehat{O}_{\Omega^i}\}$ comme définies aux définitions 8.5.4 et 8.5.6 sont cohérentes.

Preuve : L'intégralité de la preuve est donnée en annexe F. Je ne présente ici que certains points importants ajoutés par la subsomption de rôle.

Pour prouver l'implication réciproque, on doit d'abord prouver un lemme analogue à celui mentionné pour les concepts. Informellement, ce lemme indique que la fonction ε_i construite préserve la structure des rôles. Il devient alors très simple de vérifier la satisfaction du réseau d'ontologies.

La nouvelle version de la preuve de l'implication directe est organisée comme suit :

1. Construire les configurations selon les modèles $\langle \mathcal{I}, \varepsilon \rangle$:
   (a) construire la configuration globale Ω ; **(fait)**
   (b) construire les configurations locales Ω^i vis-à-vis de Ω ; **(fait)**
   (c) prouver que les Ω^i sont bien des configurations locales ; **(fait)**
   (d) construire la configuration globale de rôles Φ_Ω vis-à-vis de Ω ;
   (e) construire les configurations locales de rôles Φ_{Ω^i} vis-à-vis de Ω^i et de Φ_Ω ;
   (f) prouver que les Φ_{Ω^i} sont bien des configurations locales de rôles.
2. Montrer que $\widehat{\mathbf{A}}_\Omega$ est cohérente :
   (a) construire une interprétation $\mathcal{I}_A$ de $\widehat{\mathbf{A}}_\Omega$;
   (b) montrer que $\mathcal{I}_A$ est un modèle de $\widehat{\mathbf{A}}_\Omega$.
3. Montrer que $\{\widehat{O}_{\Omega^i}\}$ est cohérente pour tout $i \in \mathbf{O}$:
   (a) construire une interprétation $\mathcal{I}_i$ de $\widehat{O}_{\Omega^i}$;
   (b) montrer que $\mathcal{I}_i$ est un modèle de $\widehat{O}_{\Omega^i}$.

La construction des configurations de rôles est encore très directe. Pour construire l'interprétation de l'ontologie d'alignement, il faut maintenant indiquer quelle est l'interprétation des rôles. On fait correspondre à un rôle R l'ensemble des couples $\langle W, W' \rangle$ tels que $\langle W, W', R \rangle$ se trouve dans la configuration de rôle. La vérification de la satisfaction de l'ontologie d'alignement est alors très simple. La construction d'un modèle des ontologies étendues exige beaucoup plus de vérification, mais chacune reste simple. En particulier, la construction de l'interprétation fait intervenir des objets dont il faut prouver l'existence. Ces vérifications font l'objet de lemmes intermédiaires assez simples à démontrer. Une fois ces étapes préliminaires surmontées, le démonstration suit naturellement. □

Un algorithme exhaustif dans la lignée de l'algorithme 8.4 a une complexité plus importante puisqu'il faut aussi tester les configurations de rôles. Cependant, la classe de complexité reste la même, à savoir un nombre double exponentielle d'appels à des procédures de décision locales sur des données de taille exponentielle. Cette complexité très élevée est un sérieux problème pour l'utilisabilité de l'algorithme et, par conséquent, la section suivante explore quelques possibilités d'optimisation, en se limitant au cas de IDDL($\sqsubseteq_C, \perp_C$).

8.6 Optimiser l'algorithme

Cette section propose plusieurs optimisations simples qui réduisent significativement la complexité. Malheureusement, elles ne changent pas la classe de complexité, mais aide à s'approcher de l'utilisabilité dans les "situations favorables".

Le but est de réduire autant que possible le nombre de configurations qui doivent être considérées. Pour simplifier la discussion, on ne se concentre que sur les configurations de concepts globales. On envisage trois méthodes complémentaires pour améliorer l'algorithme.

8.6.1 En utilisant les correspondances

La première méthode utilise les correspondances pour réduire systématiquement le nombre de configurations possibles. En effet, on peut remarquer que si $i:C \overset{\sqsubseteq}{\leftrightarrow} j:D \in A_{ij}$, alors il n'est pas nécessaire d'inspecter les configurations contenant $W \subseteq \mathscr{C}$ tel que $C \in W$ et $D \notin W$ puisque $i:C \sqcap \neg D$ est nécessairement vide. De plus, si $i:C \overset{\perp}{\leftrightarrow} j:D \in A_{ij}$, alors il n'est pas nécessaire d'inspecter les configurations contenant $W \subseteq \mathscr{C}$ telles que $C, D \in W$ puisque $i{:}C \sqcap D$ est nécessairement vide.

Malheureusement, la complexité dans le pire des cas est à peine affectée. En effet, s'il y a n concepts $D_1, \ldots, D_n$ et un concept C tels que $i:C \overset{\sqsubseteq}{\leftrightarrow} j:D_k$ pour tout k, alors la non-vacuité de C implique la non-vacuité de tous les D_k, mais quand C est vide, les concepts D_k peuvent être indépendamment vides ou non-vides. Dans ce cas, il résulte que le nombre de configurations globales à explorer est égal à 2^{2^n+1}, sachant qu'il y a en tout $n+1$ concepts globaux.

Toutefois, dans les cas les plus favorables, la complexité décroit fortement. Par exemple, s'il y a n concepts $C_1, \ldots, C_n$ tels que $i:C_k \overset{\sqsubseteq}{\leftrightarrow} j:C_{k+1}$, alors la non-vacuité de n'importe quel C_k implique la non-vacuité de tous les C_l pour $l \geq k$. Dans ce cas de figure, le nombre de configurations à tester est simplement exponentiel par rapport à la taille des alignements.

Néanmoins, si intéressante soit-elle comparée à l'approche exhaustive, cette optimisation place toujours l'algorithme dans la classe double exponentielle. L'optimisation suivante est orthogonale à celle détaillée ci-avant et permet donc de s'éloigner plus souvent des cas les plus défavorables.

8.6.2 En utilisant des techniques de retour-arrière (*backtracking*)

Grâce au théorème 8.3.1, le problème de la vérification de cohérence à été reformulé en un problème de découverte de configuration. On peut remarquer que la configuration appropriée peut être déterminée pas à pas à l'aide d'un arbre de décision, en décidant si un sous-ensemble $W \subseteq \mathscr{C}$ donné est dans la configuration ou non. En fait, à chaque nœud de l'arbre on vérifie la cohérence dans le cas où C_W est vide, puis non vide si le résultat est incohérent. Dans le cas où C_W ne peut ni être vide, ni non-vide, l'algorithme effectue un “retour arrière” (*backtrack*) et essaie l'autre possibilité pour le sous-ensemble précédemment considéré. La figure 8.4 montre une partie d'un arbre de décision possible.

D'autres techniques de retour-arrière comme le saut arrière (*backjumping*) peuvent être utilisées. Avec une telle méthode et en l'absence d'autres optimisations, le nombre d'appels aux raisonneurs locaux peut être réduit à $O(2^n)$ dans les cas favorables, puisque 2^n est la taille d'une branche de l'arbre. Or, l'optimisation précédente permet de réduire le nombre de concepts à examiner, donc réduit la profondeur de l'arbre de recherche. Par exemple, dans le cas favorable donné dans la section 8.6.1, la profondeur est réduite à une taille proportionnelle à la taille des alignements. La dernière optimisation peut aussi servir à réduire la profondeur de l'arbre de recherche.

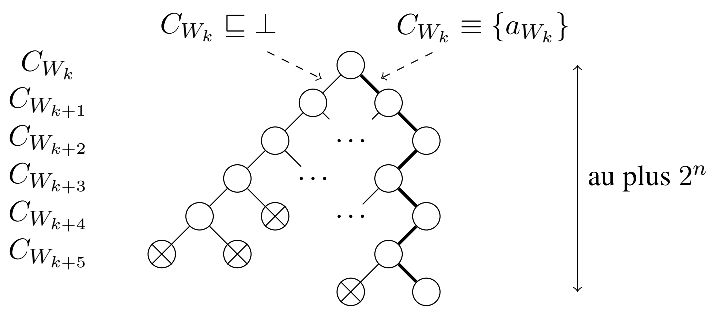


FIGURE 8.4 – À chaque nœud, la branche de gauche indique que le concept C_W est supposé vide, c'est-à-dire que l'axiome $C_W \sqsubseteq \bot$ est ajouté à l'ontologie, tandis que la branche de droite indique que le concept est non vide, c'est-à-dire que l'axiome $C_W \equiv \{a_W\}$ est ajouté avec un individu a_W non encore présent dans l'ontologie. Le chemin en trait épais indique une configuration possible pour le réseau d'ontologies.

8.6.3 Optimisation supplémentaire

Au cours du raisonnement, à un certain niveau de l'arbre de décision, il est possible qu'on ait pu prouver qu'un concept C est vide ($C \sqsubseteq \bot$). On peut alors aussi affirmer que toute conjonction de C avec d'autres concepts est vide. Plus précisément, pour tout $W \subseteq \mathscr{C}$ tel que $C \in W$, le réseau d'ontologies implique que $C_W \sqsubseteq \bot$, alors les configurations contenant W peuvent être éliminées. Ceci décroit encore l'espace de recherche. Par conséquent, on peut conjecturer qu'il existe des cas pratiques réalistes où le raisonnement avec cet algorithme (amélioré par les optimisations) peut être mené à bien, en dépit de la très forte complexité dans le pire des cas.

Néanmoins, ces optimisations ne suffisent toujours pas à traiter beaucoup de cas difficiles. Dans la section suivante, on étudie le raisonnement sur des variantes du formalisme. Dans un premier temps, on étudiera IDDL en affaiblissant l'expressivité des alignements, puis on verra le lien entre IDDLet d'autres formalismes distribués en changeant la définition des fonctions d'égalisation.

8.7 Réduire l'expressivité des alignements avec IDDL($\sqsubseteq_C, \sqsubseteq_R$)

La très forte complexité de l'algorithme de vérification de cohérence d'un réseau d'ontologies en IDDL vient du fait qu'une quantité double exponentielle de configurations de taille exponentielle. Ces configurations représentent la structure des modèles des ontologies d'alignement qui sont exprimées en une sous-logique de $\mathcal{ALC}$. On peut simplifier la structure des modèles en retirant l'exclusion de concepts et rôles inter-ontologiques du langage d'alignements. En l'absence de négation et de nominaux, les algorithmes de tableaux, par exemple dans [11], peuvent générer un modèle dont le domaine est réduit à un singleton pour une ontologie cohérente ne contenant que des axiomes entre concepts ou rôles primitifs. Ceci correspond à l'expressivité du langage d'alignements lorsqu'on lui retire l'exclusion et les nominaux. De ce fait, on peut réduire la complexité des configurations qui ne doivent représenter que des modèles réduits à un singleton. Par conséquent, pour déterminer si les conjonctions de concepts C_W sont vides ou non, il suffit de déterminer si l'un des concepts atomiques est vide ou égal au singleton. Ainsi, une configuration peut être réduite à un simple sous-ensemble du vocabulaire global, indiquant que les concepts choisis dans cet ensemble sont équivalents au singleton, tandis que les autres sont équivalents au concept vide.

On redéfinit alors les notions de configurations, de configurations de rôles, d'ontologie d'alignement et d'ontologies étendues d'une façon simplifiée suffisant à traiter complètement le problème de la cohérence d'un système IDDL($\sqsubseteq_C, \sqsubseteq_R$).

Definition 8.7.1 (Configuration simplifiée) *Soit S un réseau d'ontologies alignées avec un ensemble de noms de concepts globaux $\mathscr{C}$. Une* configuration simplifiée *de S est un sous-ensemble Ω de $\mathscr{C}$.*

Lorsqu'un concept C appartient à une configuration simplifiée, cela signifie implicitement qu'on suppose que ce concept est non-vide et même, plus précisément, égal à un singleton.

Definition 8.7.2 (Configuration simplifiée de rôles) *Soit S un réseau d'ontologies alignées avec un ensemble de noms de rôles globaux $\mathscr{R}$. Une* configuration de rôles simplifiée *de S est un sous-ensemble Φ de $\mathscr{R}$.*

Puisque le domaine d'interprétation global peut être réduit à un singleton, l'interprétation d'un rôle est soit vide, soit peut être ramenée à un simple couple $\langle x, x\rangle$, où x est le seul élément du domaine. Donc, lorsqu'un rôle appartient à une configuration de rôles simplifiée, cela signifie qu'il est supposé égal à l'identité sur ce singleton, sinon il est vide. On peut donc traduire ces suppositions comme des axiomes à introduire dans l'ontologie d'alignement simplifiée.

Remarque 8.7.1 *Il faut bien comprendre qu'un réseau d'ontologies alignées dans IDDL($\sqsubseteq_C$, $\sqsubseteq_R$)* n'indique pas *que le domaine global est réduit à un singleton. Seule la procédure de raisonnement le suppose en profitant du fait qu'il est toujours* possible *de trouver un modèle réduit à un singleton. Cette possibilité n'indique pas qu'il est* nécessaire *que le domaine global soit un singleton.*

Definition 8.7.3 (Ontologie d'alignement simplifiée) *Soit $S = \langle \mathbf{O}, \mathbf{A}\rangle$ un réseau d'ontologies alignées. Soit Ω une configuration simplifiée et Φ une configuration simplifiée de rôles. L'*ontologie d'alignement *vis-à-vis de Ω et Φ est une ontologie $\widehat{\mathbf{A}}_\Omega$ définie comme suit :*

1. *pour tout $i, j \in \mathbf{O}$, si $i{:}C \overset{\sqsubseteq}{\leftrightarrow} j{:}D$ (respectivement $i{:}R \overset{\sqsubseteq}{\leftrightarrow} j{:}Q$) est une correspondance de concepts (respectivement de rôles) dans $\mathbf{A}$ alors $i : C \sqsubseteq j : D$ (respectivement $i{:}R \sqsubseteq j{:}Q$) est un axiome de $\widehat{\mathbf{A}}_\Omega$;*
2. *pour tout $C \in \Omega$, $\top \sqsubseteq C$ est un axiome de $\widehat{\mathbf{A}}_\Omega$;*
3. *pour tout $W \notin \Omega$, $C \sqsubseteq \bot$ est un axiome de $\widehat{\mathbf{A}}_\Omega$;*
4. *pour tout $R \in \Omega$, $\top \equiv \exists R.\top$ est un axiome de $\widehat{\mathbf{A}}_\Omega$;*
5. *pour tout $W \notin \Omega$, $\exists R.\top \sqsubseteq \bot$ est un axiome de $\widehat{\mathbf{A}}_\Omega$;*
6. *$\top \sqsubseteq \{a\}$ est un axiome de $\widehat{\mathbf{A}}_\Omega$, où a est un individu n'apparaissant pas dans les alignements.*

Les axiomes 2, 3, 4 et 5 traduisent simplement les suppositions que l'on fait lorsque l'on choisit une configuration. Ces axiomes vont induire des conséquences sur les ontologies locales que l'on peut caractériser et introduire dans la notion d'ontologie étendue simplifiée.

Definition 8.7.4 (Ontologie étendue simplifiée) *Soit $S = \langle \mathbf{O}, \mathbf{A}\rangle$ un réseau d'ontologies alignées. Soit Ω une configuration simplifiée et Φ une configuration de rôles simplifiée. L'*ontologie étendue *$\widehat{O}^i_{\Omega,\Phi}$ vis-à-vis de Ω et Φ est définie comme suit :*

1. *$O_i \subseteq \widehat{O}^i_{\Omega,\Phi}$;*

2. *pour tout* $i\!:\!C \in \Omega$, $C(b_C)$ *est un axiome de* $\widehat{O}^i_{\Omega,\Phi}$ *avec* b_C *un nouveau nom d'individu dans* $\widehat{O}^i_{\Omega,\Phi}$ *;*
3. *pour tout* $i\!:\!C \notin \Omega$, $C \sqsubseteq \bot$ *est un axiome de* $\widehat{O}^i_{\Omega,\Phi}$ *;*
4. *pour tout* $i : R \in \Phi$, $R(\beta_R, \gamma_R)$ *est un axiome de* $\widehat{O}^i_{\Omega,\Phi}$, *avec* β_R *et* γ_R *deux nouveaux noms d'individu dans* $\widehat{O}^i_{\Omega,\Phi}$ *;*
5. *pour tout* $i\!:\!R \notin \Phi$, $R(\beta_R, \gamma_R)$ *est un axiome de* $\widehat{O}^i_{\Omega,\Phi}$.

On peut remarquer ici qu'il n'y a pas de configuration locale. La raison en est simple : dans leur version non simplifiée, les configurations représentent un choix d'atomes dans une décomposition atomique (voir la section 8.3.1). Or, les concepts de la décomposition atomique sur le vocabulaire global diffèrent de ceux d'une décomposition atomique sur un vocabulaire local. En revanche, dans sa version simplifiée, les concepts choisis dans la configuration se retrouvent à l'identique, modulo l'indice de l'ontologie ($i\!:\!C \in \mathscr{C} \Rightarrow C \in \mathscr{C}_i$).

Un théorème analogue au théorème 8.3.1 peut alors être formulé.

Théorème 8.7.1 *Soit* $S = \langle \mathbf{O}, \mathbf{A} \rangle$ *un réseau d'ontologies alignées en IDDL(*$\sqsubseteq_C, \sqsubseteq_R$*).* S *est* cohérent *si et seulement si il existe une configuration simplifiée* Ω *pour* S *et une configuration de rôles* Φ *telles que l'ontologie d'alignement simplifiée* $\widehat{\mathbf{A}}_{\Omega,\Phi}$ *et les ontologies locales étendues* $\{\widehat{O}^i_{\Omega,\Phi}\}$ *sont cohérentes.*

La preuve de ce théorème se fait en suivant le même schéma que pour les théorèmes précédents, mais toutes les vérifications sont très nettement simplifiées.

On peut également construire un algorithme élémentaire à partir de ce théorème, par un test exhaustif des onfigurations. Celui-ci doit alors appeler les raisonneurs locaux au plus $2^{|\mathscr{C}|}$ fois, en transmettant un nombre d'axiomes de taille comparable à la taille des configurations, soit une taille linéaire par rapport à la taille des alignements. Par ailleurs, en exploitant les techniques d'optimisation décrites dans la section 8.6, la complexité peut être réduite significativement dans de nombreux cas.

8.8 Variantes de IDDL

Dans cette section, on modifie légèrement la définition des fonctions d'égalisation. Il y a plusieurs raisons de faire cela. Premièrement, la définition 4.3.1 de fonction d'égalisation est plus générale que celle attribuée ici. En effet, la définition générale du chapitre 4 permet d'associer n'importe quel type d'entités à n'importe quel autre du domaine d'interprétation globale, tandis que la définition 8.2.3 de ce chapitre pose des contraintes sur l'interprétation des concepts et des rôles, à partir d'une fonction donnée entre domaine d'interprétation.

Notamment, la fonction d'interprétation en IDDL associe aux individus des individus et aux concepts vides des ensembles vides et inversement, des concepts non vides à des ensembles non vides. Pourtant, selon la définition générale du chapitre 4, on pourrait associer à des individus des ensembles et à des concepts non vides un ensemble vide.

Dans cette section, on se contentera de définir la fonction d'égalisation comme une *famille de relations* entre chaque domaine d'interprétation local et le domaine d'interprétation global. En utilisant une telle définition, on perd la possibilité d'utiliser les procédures décrites dans ce chapitre, mais on peut alors transformer un réseau d'ontologies IDDL en un réseau d'ontologies DDL.

Remarque 8.8.1 *Ici, on limitera le langage d'alignements aux seules correspondances de concepts ou de rôles.*

8.8.1 Fonctions d'égalisation et interprétation

On redéfinit maintenant les fonctions d'égalisation pour cette variante de IDDL. Ce changement ne modifie pas le fait que l'on affecte à chaque ontologie une interprétation locale, qui servira à définir la fonction d'égalisation. Pour éviter la confusion avec la fonction du formalisme IDDL original, on appellera cette nouvelle version de la fonction d'égalisation une *relation d'égalisation*.

Definition 8.8.1 (Relation d'égalisation) *Étant donnée une famille d'interprétations locales* $\mathbf{I}$*, une relation d'égalisation* $\boldsymbol{\varepsilon}$ *est une famille de relations indicées par* $\mathbf{I}$ *telle que pour tout* $I_i \in \mathbf{I}$*,* $\varepsilon_i \subseteq \Delta^{I_i} \times \Delta$ *où* Δ *est appelé le domaine global d'interprétation de* $\boldsymbol{\varepsilon}$*.*

Une interprétation distribuée est alors définie naturellement comme suit.

Definition 8.8.2 (Interprétation distribuée) *Soit* $S = \langle \mathbf{O}, \mathbf{A} \rangle$ *un réseau d'ontologies alignées. Une* interprétation distribuée *de* S *est un couple* $\langle \mathbf{I}, \boldsymbol{\varepsilon} \rangle$ *où* $\mathbf{I}$ *est une famille d'interprétations indicées par* $\mathbf{O}$*,* $\boldsymbol{\varepsilon}$ *est une fonction d'égalisation pour* $\mathbf{I}$*, tels que pour tout* $i \in \mathbf{O}$*,* I_i *interprète* i *et* $\varepsilon_i \subseteq \Delta^{I_i} \times \Delta$*.*

Pour rester cohérent avec la définition 4.3.1, il faut définir la fonction d'égalisation pour les interprétations d'individus, de concepts et de rôles.

Definition 8.8.3 (Fonction d'égalisation étendue) *Étant donnée une interprétation distribuée* $\langle \mathbf{I}, \boldsymbol{\varepsilon} \rangle$ *d'un réseau d'ontologies* S*, la relation* ε_i *est étendue en une fonction définie sur* $\Delta^{I_i} \cup 2^{\Delta^{I_i}} \cup 2^{\Delta^{I_i} \times \Delta^{I_i}}$ *en prenant, pour chaque individu* a_i*, chaque concept* C_i *et chaque rôle* R_i *les images suivantes.*

- $\varepsilon_i(a_i^{I_i}) = \{x \in \Delta \mid \langle a_i^{I_i}, x \rangle \in \varepsilon_i\}$,
- $\varepsilon_i(C_i^{I_i}) = \{x \in \Delta \mid \exists e \in C_i^{I_i}$ *tel que* $\langle e, x \rangle \in \varepsilon_i\}$ *et*
- $\varepsilon_i(R_i^{I_i}) = \{\langle x, y \rangle \in \Delta \times \Delta \mid \exists \langle e, f \rangle \in R_i^{I_i}$ *tel que* $\langle e, x \rangle \in \varepsilon_i$ *et* $\langle f, y \rangle \in \varepsilon_i\}$.

Les définitions de modèle d'une correspondance et modèle d'un réseau d'ontologies alignées sont alors identiques au cas de IDDL standard.

8.8.2 Transformation en DDL

Cette transformation consiste à traduire les correspondances en des axiomes, de façon similaire à l'ontologie d'alignement introduite en section 8.3.1. Ensuite, on ajoute des passerelles (*bridge rules*) entre les ontologies locales et cette ontologie d'alignement. Ces passerelles traduisent l'existence de la fonction d'égalisation.

Definition 8.8.4 (Transformation) *Soit* $S = \langle \mathbf{O}, \mathbf{A} \rangle$ *un réseau d'ontologies alignées. On définit le réseau d'ontologies et de passerelles en DDL* $\tau(S)$ *à partir de* S *de la façon suivante.*

- $\tau(S)$ *contient les ontologies de* $\mathbf{O}$*,*
- $\tau(S)$ *contient une ontologie* $\widehat{\mathbf{A}}$*, constituée des vocabulaires globaux* $\mathscr{C}$ *et* $\mathscr{R}$ *(définition 8.3.2 et définition 8.5.2) et des axiomes associés aux correspondances (comme en définition 8.3.5),*

– *les passerelles entre les ontologies de* $\mathbf{O}$ *sont vides, ainsi que les passerelles allant de* $\widehat{\mathbf{A}}$ *vers les ontologies de* $\mathbf{O}$*,*
– *pour tout concept ou rôle* $i : X$ *dans le vocabulaire de* $\widehat{\mathbf{A}}$*, on ajoute une passerelle* $i{:}X \xrightarrow{\equiv} a{:}(i{:}X)$ *reliant* $\mathbf{O}$ *à* $\widehat{\mathbf{A}}$.

Dans le dernier item, la lettre a identifie l'ontologie $\widehat{\mathbf{A}}$ et doit donc être différente de tous les indices des ontologies. Aussi, compte tenu de la construction du vocabulaire global, il faut distinguer $i : X$ se trouvant à gauche de la passerelle, qui identifie le concept ou rôle X (potentiellement complexe) de l'ontologie O_i, et le concept ou rôle *atomique* $i{:}X$ se trouvant à droite de la passerelle, qui est un élément du vocabulaire de $\widehat{\mathbf{A}}$.

Le théorème suivant établit le lien étroit entre cette variante de IDDL et le formalisme DDL.

Théorème 8.8.1 *Soit* S *un réseau d'ontologies alignées.* S *est cohérent si et seulement si* $\tau(S)$ *est cohérent.*

Preuve : On divise la preuve de ce théorème en deux parties correspondant aux deux implications de l'équivalence.

Implication directe : Soit $\langle \mathbf{I}, \varepsilon \rangle$ un modèle distribué de S avec relation d'égalisation. On construit alors un modèle de $\tau(S)$ comme suit.
– On affecte aux ontologies de $\mathbf{O}$ les modèles de $\mathbf{I}$.
– Pour tout $i : X$ du vocabulaire de $\widehat{\mathbf{A}}$, l'interprétation de $i : X$ est égale à $\varepsilon_i(X^{I_i})$ et le domaine d'interprétation de $\widehat{\mathbf{A}}$ est le domaine global.
– Pour tout i , la relation de domaines entre O_i et $\widehat{\mathbf{A}}$ est égale à ε_i.
– Toutes les autres relations de domaines sont vides.
Ceci constitue un modèle de $\tau(S)$, puisque les ontologies sont trivialement satisfaites, de même que chacune des passerelles.

Implication réciproque : Soit $\langle \mathbf{I} \cup \{I_A\}, \mathbf{r} \rangle$ un modèle DDL de $\tau(S)$, où I_A correspond à l'interprétation de $\widehat{\mathbf{A}}$. On construit alors un modèle distribué de S comme suit.
– On affecte aux ontologies de $\mathbf{O}$ les modèles de $\mathbf{I}$.
– Le domaine d'interprétation global est égal au domaine d'interprétation de I_A.
– La composante i de la relation d'égalisation est égale à la relation de domaine entre O_i et $\widehat{\mathbf{A}}$.
Pour vérifier que ceci constitue un modèle de S, il faut s'assurer que les correspondances sont satisfaites. Or, puisqu'on a un modèle de $\tau(S)$, la satisfaction des passerelles impliquent que pour toute correspondance $i : X \overset{*}{\leftrightarrow} j : Y$ (avec $* \in \{\sqsubseteq, \perp\}$), $\varepsilon_i(X) = (i : X)^{I_A}$ et $\varepsilon_j(Y) = (j{:}Y)^{I_A}$ et puisque I_A est un modèle de $\widehat{\mathbf{A}}$, si $* = \sqsubseteq$ alors $(i{:}X)^{I_A} \subseteq (j{:}Y)^{I_A}$ et si $* = \perp$ alors $(i{:}X)^{I_A} \cap (j{:}Y)^{I_A} = \emptyset$. □

Notons qu'il n'existe pas, a priori, de transformation inverse permettant de passer d'un réseau d'ontologies DDL à un réseau d'ontologies alignées en IDDL avec relation d'égalisation.

8.8.3 Autres variantes

On pourrait aussi étudier d'autres variantes, en faisant varier les contraintes sur la fonction d'égalisation. Par exemple, si l'on se donne des relations injectives à la place de relations ou de fonctions, on peut conjecturer que les réseaux d'ontologies peuvent être transformés en des systèmes modulaires P-DL.

Cette possibilité de choisir les propriétés de la fonction d'égalisation fait penser aux contraintes de domaines en DFOL [38]. Ces contraintes servent à imposer qu'une relation de domaine soit injective, surjective, totale, etc.

8.9 Bilan

Dans ce chapitre, j'ai présenté un formalisme pour exprimer des réseaux d'ontologies alignées en logique de description, en particularisant les relations au sein des correspondances. À ce formalisme est associée une sémantique formelle définie sur le modèle de la sémantique générique du chapitre 4.

On montre premièrement que la cohérence d'un réseau d'ontologies peut se réduire à la cohérence de plusieurs ontologies construites à partir des ontologies et des alignements du réseau, et d'une structure appelée configuration. On montre aussi que l'on peut se limiter à explorer un nombre finie de configurations pour déterminer la cohérence du réseau donc qu'il existe une procédure correcte et complète pour la vérification de cohérence. Cela implique la décidabilité du formalisme en fonction de la décidabilité des langages d'ontologies.

On décrit alors un algorithme pour la vérification de cohérence, d'abord naïf et inutilisable du fait de sa complexité. Des optimisations réduisent cette complexité de façon significative, mais ne parviennent pas à faire baisser la classe de complexité double exponentielle de la procédure. C'est pourquoi des variantes du formalisme sont explorées, notamment en réduisant l'expressivité des alignements, puis en changeant la définition des fonctions d'égalisation.

On peut émettre des critiques face à ce formalisme.

- La complexité très élevée ne passe pas à l'échelle ;
- l'exclusion de rôles n'est pas prise en compte ici ;
- le formalisme ne propose pas pour l'instant de constructeurs spécifiques au niveau des alignements, comme présenté dans le chapitre 6 ;
- une implémentation reste à concevoir afin de tester les performances de l'algorithme sur des cas pratiques ;

Chacune de ces critiques fournit une piste pour les travaux futurs au sujet de IDDL.

On peut aussi remarquer les avantages de cette approche. Notamment, l'encapsulation des raisonneurs locaux est une propriété intéressante pour la modularité des ontologies et pour la robustesse face à l'hétérogénéité des implémentations locales. D'autre part, l'algorithme n'exige pas que les systèmes de raisonnement locaux puissent traiter une logique de description complexe, mais ne pose pas non plus de limite supérieure à l'expressivité locale. Plus précisément, le langage minimal devant être traité par les raisonneurs locaux est une portion de $\mathcal{ALC}$.

En fait, toute logique utilisant des concepts, rôles et individus et capable d'interpréter les axiomes ajoutés à l'ontologie étendue, peut prendre la place d'un nœud dans un réseau d'ontologies en IDDL. Entre autre, une ontologie d'un tel réseau peut être un composant d'un réseau d'ontologies DDL ou encore P-DL, à condition qu'un raisonneur adapté soit implémenté localement pour ces logiques distribués. De même, un raisonneur global IDDL peut lui-même prendre la place d'un nœud local dans un autre réseau d'ontologies alignées. C'est ce que montre la figure 4.2 à la page 64, où une combinaison de sémantiques hybrides est exploitée dans un même réseau d'ontologies.

Ainsi, tandis que le chapitre 5 indiquait que la sémantique générale s'adapte bien à notre notion d'ontologies modulaires, ce chapitre montre en plus que le mécanisme de raisonnement lui-même, appliqué aux logiques de description, convient particulièrement bien à un système de modules reliés par des interfaces encapsulant une implémentation.

Conclusion

Résumé de la contribution

En m'interrogeant sur la sémantique des réseaux de connaissances, je me suis intéressé à modéliser la sémantique de réseaux d'ontologies alignées particuliers, dont on souhaite qu'ils soient constitués d'ontologies et d'alignements d'ontologies fortement indépendants et hétérogènes. J'ai établi au chapitre 3 que les formalismes logiques existant s'appliquent dans des contextes autres que ceux que j'ai voulu considérer. Notamment, ils ne permettent pas de représenter à la fois des connaissances hétérogènes, tout en assurant le principe de médiation pourtant essentiel à l'interopérabilité dans le type de réseaux envisagé. J'ai donc proposé au chapitre 4 un formalisme qui exploite au mieux les logiques locales déjà établies, les met en corrélation par un procédé original qui distingue d'un côté la représentation locale, propre à chaque nœud dans le réseau, et la représentation des connaissances inter-ontologies propre au médiateur.

Pour démontrer le bien fondé de ce choix, j'ai montré au chapitre 5 comment cette sémantique générique s'applique tout naturellement à un système de modules d'ontologies inspiré de l'ingénierie logicielle. En effet, si l'on considère qu'un module joue un rôle comparable à un médiateur vis-à-vis des modules qu'il importe, en particulier si ceux-ci nécessitent une mise en correspondance pour fonctionner ensemble alors le formalisme général du chapitre 4 permet de définir la sémantique des modules en fonction de la sémantique des modules importés.

Compte tenu de la séparation des sémantiques locales et de la sémantique globale, j'ai expliqué au chapitre 6 qu'il devient alors possible de découpler complètement le langage d'alignements des langages d'ontologies, et donc de favoriser une bonne médiation entre des systèmes ayant peut-être eux-même une faible expressivité. Cette caractéristique permet alors au médiateur d'effectuer un grand nombre d'opérations utiles à l'interopérabilité sans toutefois perturber ni même accéder aux informations locales. En augmentant la richesse des alignements, les opérations de transformation de requêtes ou d'instances ainsi que l'identification et unification de données s'en trouvent améliorées.

Aussi, tout en garantissant l'aspect distribué du formalisme, j'ai prouvé au chapitre 7 que cette sémantique permet la transitivité des alignements, et donc la réutilisation dans divers contextes d'alignements existants, sans passer par une réévaluation des relations sémantiques pouvant exister entre une nouvelle paire d'ontologies. Ce point particulier distingue le formalisme des autres approches sémantiques distribués. L'utilité de cette opération est démontrée non seulement par le seul fait qu'elle permet de construire des alignements sans passer par les algorithmes très complexes de découverte automatique de correspondances, mais aussi pour l'évolution et la mise à jour d'alignement et au sein de la modularité.

Enfin, alors que les chapitres précédents proposaient des propriétés formelles, le chapitre 8 décrit une procédure de vérification de la cohérence d'un réseau d'ontologies en logique de description. L'algorithme correct et complet souffre encore d'une trop forte complexité, mais il montre d'une part la décidabilité des réseaux d'ontologies alignées en fonction de la décidabilité des langages d'ontologies et d'autre part il introduit une méthode de vérification qui ne nécessite à aucun moment de connaitre l'implémentation concrète des moteurs d'inférences locaux. En cela, il est d'un grand intérêt pour la modularité des ontologies car il garantit l'encapsulation des systèmes de raisonnement locaux.

Critiques

Ceci étant dit, il n'est pas sans intérêt de poser un regard critique sur ce qui a été défini. Le principal inconvénient est, semble-t-il, la complexité du raisonnement distribué. Celui-ci intervient à de nombreux niveaux. Pour la modularité, les ontologies ne peuvent se passer d'un mécanisme de raisonnement exploitant l'ensemble des modules du réseau. D'un autre côté, la compilation des connaissances pourrait être une solution partielle à ce problème délicat.

Ensuite, pour le raisonnement sur le formalisme expressif, la complexité est d'autant plus élevée que les constructeurs sont riches. Toutefois, l'intérêt de ce langage se trouve dans sa capacité à définir des équivalences entre structures complexes qui faciliteront les transformations de requêtes ou d'instances. En cas de nécessité, il serait judicieux de restreindre les constructeurs de se langage pour en extraire un fragment traitable en pratique.

Le raisonnement distribué a son importance aussi en composition d'alignements. Néanmoins, puisque la composition est, par essence, incomplète, il semble plus acceptable de restreindre les déductions pour obtenir un mécanisme efficace en pratique. Les opérateurs que j'ai introduits ont de bonnes caractéristiques de ce point de vue.

Enfin, on a abordé au dernier chapitre différentes approches pour améliorer l'efficacité des mécanismes de raisonnement.

Perspectives

Les perspectives ouvertes par les travaux présentés ici peuvent être examinées au regard de chacune des applications décrites. En premier lieu, la modularisation doit offrir d'un côté des outils pour la gestion des modules, de leurs imports et des alignements. Ceci est indépendant de la sémantique et une implémentation est en court de développement dans le cadre d'un projet franco-brésilien. Pour y introduire une procédure de déduction conforme à la sémantique décrite au chapitre 5, en particulier pour modulariser des ontologies OWL, il faut avant tout développer des algorithmes, dans l'optique de ceux définis au chapitre 8. En outre, pour proposer au sein du système de modules des opérations de combinaison d'alignements, et notamment la composition, il faut développer les procédures de composition conformément à ce qui est décrit au chapitre 7. Ainsi, les perspectives du chapitre 5 ne font que renvoyer aux travaux des chapitres suivants.

Ensuite, pour exploiter la sémantique d'un langage expressif comme celui détaillé au chapitre 6, plusieurs pistes sont à envisager. D'un côté, on peut tenter d'élaborer des procédures de raisonnement distribué à l'instar du chapitre 8 pour un langage expressif. Mais l'intérêt d'un tel langage réside avant tout dans les possibilités de transformation de requêtes ou de données. Ces processus n'ont pas été approfondis dans le courant de ma thèse, mais offre de vastes possibilités de recherches. En outre, le langage expressif doit aussi être envisagé du point de vue de la composition. Ce point précis indique ainsi une perspective à envisager vis-à-vis du chapitre 7.

La composition est sans doute l'un des points les plus intéressants de cette sémantique, puisque celle-ci la différencie des autres formalismes distribués proposés auparavant. L'utilisation des techniques d'algèbre de relations montre qu'une implémentation efficace d'un opérateur de composition est réalisable et pourrait être intégré à l'API d'alignement, ainsi qu'au serveur d'alignement. Cette implémentation doit permettre d'une part d'effectuer des tests, notamment en profitant des divers alignements produits annuellement lors de l'évaluation des méthodes d'alignement automatiques [3]. Sur le plan théorique, avoir à disposition une méthode de composition d'alignements permet d'obtenir, potentiellement, plusieurs alignements différents selon les séquences d'alignements choisies. Dans le cas où les alignements doivent être fournis en temps réel, il n'est pas judicieux de calculer toutes les compositions. Par conséquent, il faut pouvoir choisir la meilleure séquence d'alignements ou parfois même construire l'alignement de toute pièce. Le choix de la meilleure stratégie en fonction de la qualité désirée et des ressources disponibles (en temps et en mémoire) est une importante voie d'investigation. En complément, la composition sémantique d'alignements expressifs est aussi à explorer, et particulièrement en tenant compte de la sémantique des ontologies elles-mêmes, mais celle-ci renvoie à nouveau à la nécessité de procédures de déduction distribuée.

Enfin, le dernier chapitre laisse entrevoir de nombreuses perspectives. L'optimisation de la procédure de décision présentée est cruciale pour son utilisation en pratique. Mais d'autres approches existent : le raisonnement peut être étudié en restreignant soit les langages de représentation locaux, soit les possibilités d'expression de correspondances. Il est aussi intéressant d'évaluer les implications de l'utilisation de fonctions d'égalisation ayant d'autres caractéristiques que celles d'IDDL. Enfin, bien entendu, il reste à implémenter les algorithmes obtenus de sorte que des tests puissent être effectués.

3. `http://oaei.ontologymatching.org/`

Bibliographie

[1] Nada ABDALLAH et François GOASDOUÉ : Systèmes pair-à-pair sémantiques et extension conservative d'une base de connaissances. *In* Bruno DEFUDE, éditeur : *Actes des 24ièmes Journées Bases de Données Avancées (BDA2008)*, 2008.

[2] Andrea ACCIARRI, Diego CALVANESE, Giuseppe de GIACOMO, Domenico LEMBO, Maurizio LENZERINI, Mattia PALMIERI et Riccardo ROSATI : QUONTO : QUerying ONTOlogies. *In* Manuela M. VELOSO et Subbarao KAMBHAMPATI, éditeurs : *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 1670–1671. AAAI Press / The MIT Press, 2005.

[3] Philippe ADJIMAN, Philippe CHATALIC, François GOASDOUÉ, Marie-Christine ROUSSET et Laurent SIMON : Scalability Study of Peer-to-Peer Consequence Finding. *In* Leslie Pack KAELBLING et Alessandro SAFFIOTTI, éditeurs : *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30-August 5, 2005*, pages 351–356. Professional Book Center, 2005.

[4] Philippe ADJIMAN, Philippe CHATALIC, François GOASDOUÉ, Marie-Christine ROUSSET et Laurent SIMON : Distributed Reasoning in a Peer-to-Peer Setting : Application to the Semantic Web. *Journal of Artificial Intelligence Research*, 25:269–314, 2006.

[5] Philippe ADJIMAN, François GOASDOUÉ et Marie-Christine ROUSSET : SomeRDFS in the Semantic Web. *Journal on Data Semantics*, 8:158–181, 2007.

[6] Suad ALAGIĆ et Philip A. BERNSTEIN : A Model Theory for Generic Schema Management. *In* Giorgio GHELLI et Gösta GRAHNE, éditeurs : *Database Programming Languages, 8th International Workshop, DBPL 2001, Frascati, Italy, September 8-10, 2001, Revised Papers*, volume 2397 de *Lecture Notes in Computer Science*, pages 228–246, Frascati, Italy, septembre 2001. Springer-Verlag. Revised Papers.

[7] James F. ALLEN : Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11):832–843, 1983.

[8] Franz BAADER, Diego CALVANESE, Deborah L. MCGUINNESS, Daniele NARDI et Peter F. PATEL-SCHNEIDER, éditeurs. *The Description Logic Handbook : Theory, Implementation, and Applications.* Cambridge University Press, 2003.

[9] Franz BAADER, Carsten LUTZ, Holger STURM et Frank WOLTER : Fusions of Description Logics and Abstract Description Systems. *Journal of Artificial Intelligence Research*, 16:1–58, 2002.

[10] Franz BAADER, Carsten LUTZ et Boontawee SUNTISRIVARAPORN : CEL—A Polynomial-time Reasoner for Life Science Ontologies. *In* Ulrich FURBACH et Natarajan SHANKAR, éditeurs : *Automated Reasoning, Third International Joint Conference, IJCAR 2006, Seattle, WA, USA, August 17-20, 2006, Proceedings*, volume 4130 de *Lecture Notes in Computer Science*, pages 287–291. Springer-Verlag, 2006.

[11] Franz BAADER et Ulrike SATTLER : Tableau Algorithms for Description Logics. *In* Roy DYCKHOFF, éditeur : *Automated Reasoning with Analytic Tableaux and Related Methods,*

International Conference, TABLEAUX 2000, St Andrews, Scotland, UK, July 3-7, 2000, Proceedings, volume 1847 de *Lecture Notes in Computer Science*, pages 1–18, St Andrews, Scotland, UK, 2000. Springer-Verlag.

[12] Jie BAO : *Representing and reasoning with modular ontologies*. Thèse de doctorat, Iowa State University, Ames, Iowa, 2007.

[13] Jie BAO, Doina CARAGEA et Vasant G. HONAVAR : On the Semantics of Linking and Importing in Modular Ontologies. *In* Isabel F. CRUZ, Stephan DECKER, Dean ALLEMANG, Christ PREIST, Daniel SCHWABE, Peter MIKA, Michael USCHOLD et Lora AROYO, éditeurs : *The Semantic Web - ISWC 2006, 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006, Proceedings*, volume 4273 de *Lecture Notes in Computer Science*, pages 72–86. Springer-Verlag, novembre 2006.

[14] Jie BAO, Giora SLUTZKI et Vasant G. HONAVAR : A Semantic Importing Approach to Knowledge Reuse from Multiple Ontologies. *In Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*, pages 1304–1309. AAAI Press, 2007.

[15] Brandon BENNETT, Amar ISLI et Anthony G. COHN : When does a composition table provide a complete and tractable proof procedure for a relational constraint language ? *In Proceedings of the IJCAI'97 Workshop on Spatial and Temporal Reasoning*, Nagoya, Japan, avril 24 1997.

[16] Philip A. BERNSTEIN, Todd J. GREEN, Sergey MELNIK et Alan NASH : Implementing mapping composition. *The Very Large Data Base Journal*, 17(2):333–353, 2008.

[17] Philip A. BERNSTEIN, Alon Y. HALEVY et Rachel A. POTTINGER : A vision for management of complex models. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, 29(4):55–63, 2000.

[18] Scott BOAG, Don CHAMBERLIN, Mary F. FERNÁNDEZ, Daniela FLORESCU, Jonathan ROBIE et Jérôme SIMÉON : XQuery 1.0 : An XML Query Language - W3C Recommendation 23 January 2007. Rapport technique, World Wide Web Consortium (W3C), janvier 23 2007.

[19] Marco CADOLI et Francesco M. DONINI : A Survey on Knowledge Compilation. *AI Communications*, 10(3–4):137–150, 1997.

[20] Diego CALVANESE, Giuseppe de GIACOMO et Maurizio LENZERINI : A Framework for Ontology Integration. *In* Isabel F. CRUZ, Stephan DECKER, Jérôme EUZENAT et Deborah L. MCGUINNESS, éditeurs : *The emerging semantic web*, pages 201–214. IOS Press, Amsterdam (NL), 2002.

[21] Diego CALVANESE, Giuseppe de GIACOMO et Maurizio LENZERINI : Description Logics for Information Integration. *In* Antonis C. KAKAS et Fariba SADRI, éditeurs : *Computational Logic : Logic Programming and Beyond, Essays in Honour of Robert A. Kowalski, Part II*, volume 2408 de *Lecture Notes in Computer Science*, pages 41–60, Amsterdam (NL), 2002. Springer-Verlag.

[22] Dan CONNOLLY, Frank van HARMELEN, Ian HORROCKS, Deborah L. MCGUINNESS, Peter F. PATEL-SCHNEIDER et Olga STEPÁNKOVÁ : Annotated DAML+OIL Ontology Markup, W3C Note 18 December 2001. Rapport technique, World Wide Web Consortium (W3C), décembre 18 2001.

[23] Bernardo CUENCA-GRAU, Ian HORROCKS, Yevgeny KAZAKOV et Ulrike SATTLER : A Logical Framework for Modularity of Ontologies. *In* Manuela M. VELOSO, éditeur : *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 298–303, 2007.

[24] Bernardo CUENCA-GRAU, Ian HORROCKS, Yevgeny KAZAKOV et Ulrike SATTLER : Modular Reuse of Ontologies : Theory and Practice. *In Journal of Artificial Intelligence Research*, volume 31, pages 273–318, février 2008.

[25] Bernardo CUENCA-GRAU et Oliver KUTZ : Modular Ontology Languages Revisisted. *In* Vasant G. HONAVAR, Tim FININ, Doina CARAGEA, Dunja MLADENIC et York SURE, éditeurs : *SWeCKa 2007 : Proceedings of the IJCAI-2007 Workshop on Semantic Web for Collaborative Knowledge Acquisition, Hyderabad, India, January 7, 2007*, 2007.

[26] Bernardo CUENCA-GRAU, Bijan PARSIA et Evren SIRIN : Combining OWL ontologies using $\mathcal{E}$-Connections. *Journal of Web Semantics*, 4(1):40–59, 2006.

[27] Randall DAVIS, Howard E. SHROBE et Peter SZOLOVITS : What Is a Knowledge Representation ? *Artificial Intelligence*, 14(1):17–33, 1993.

[28] Mike DEAN et Guus SCHREIBER : OWL Web Ontology Language Reference, W3C Recommendation 10 February 2004. Rapport technique, World Wide Web Consortium (W3C), février 10 2004.

[29] Jon DOYLE et Ramesh S. PATIL : Two Theses of Knowledge Representation : Language Restrictions, Taxonomic Classification, and the Utility of Representation Services. *Artificial Intelligence*, 48(3):261–297, 1991.

[30] Jérôme EUZENAT : An API for Ontology Alignment. *In* Frank van HARMELEN, Sheila MCILRAITH et Dimitri PLEXOUSAKIS, éditeurs : *The Semantic Web - ISWC 2004 : Third International Semantic Web Conference,Hiroshima, Japan, November 7-11, 2004. Proceedings*, volume 3298 de *Lecture Notes in Computer Science*, pages 698–712, Hiroshima, Japan, 2004. Springer-Verlag.

[31] Jérôme EUZENAT : Alignment infrastructure for ontology mediation and other applications. *In* Martin HEPP, Axel POLLERES, Frank van HARMELEN et Michael R. GENESERETH, éditeurs : *Actes 1st ICSOC international workshop on Mediation in semantic web services, Amsterdam (NL)*, volume 168 de *CEUR Workshop Proceedings*, pages 81–95. Sun SITE Central Europe (CEUR), décembre 2005.

[32] Jérôme EUZENAT, Adrian MOCAN et François SCHARFFE : Ontology Alignments – An Ontology Management Perspective. *In* Martin HEPP, Pieter De LEENHEER, Aldo de MOOR et York SURE, éditeurs : *Ontology Management*, volume 7 de *Semantic Web And Beyond Computing for Human Experience*, pages 177–206. Springer-Verlag, 2008.

[33] Jérôme EUZENAT, François SCHARFFE et Antoine ZIMMERMANN : Expressive alignment language and implementation. Deliverable D2.2.10, Knowledge Web NoE, 2007.

[34] Jérôme EUZENAT et Pavel SHVAIKO : *Ontology Matching*. Springer-Verlag, Heidelberg (DE), 2007.

[35] Jérôme EUZENAT, Antoine ZIMMERMANN et Fred FREITAS : Alignment-based modules for encapsulating ontologies. *In* Bernardo CUENCA-GRAU, Vasant G. HONAVAR, Anne SCHLICHT et Frank WOLTER, éditeurs : *Second International Workshop on Modular Ontologies (WOMO 2007)*, pages 32–45, juin 2007.

[36] Jérôme EUZENAT, Antoine ZIMMERMANN, Marta SABOU et Mathieu D'AQUIN : Matching ontologies for context. Deliverable D3.3.1, IST NeOn IP, 2007.

[37] Chiara GHIDINI : *A semantics for contextual reasoning : theory and two relevant applications*. Thèse de doctorat, Università degli Studi di Roma “La Sapienza”, 2008.

[38] Chiara GHIDINI et Luciano SERAFINI : Distributed First Order Logics. *In* Dov M. GABBAY et Maarten de RIJKE, éditeurs : *Frontiers of Combining Systems 2*, volume 7 de *Studies in Logic and Computation*, pages 121–139, Baldock, Hertfordshire, UK, 2000. Research Studies Press.

[39] Chiara GHIDINI et Luciano SERAFINI : Reconciling Concepts and Relations in Heterogeneous Ontologies. *In* York SURE et John DOMINGUE, éditeurs : *The Semantic Web : Research and Applications, 3rd European Semantic Web Conference, ESWC 2006, Budva, Montenegro, June 11-14, 2006, Proceedings*, volume 4011 de *Lecture Notes in Computer Science*, pages 50–64. Springer-Verlag, 2006.

[40] Chiara GHIDINI et Luciano SERAFINI : Mapping Properties of Heterogeneous Ontologies. *In* Peter HAASE, Vasant G. HONAVAR, Oliver KUTZ, York SURE et Andrei TAMILIN, éditeurs : *Proceedings of the 1st International Workshop on Modular Ontologies, WoMO'06, co-located with the International Semantic Web Conference, ISWC'06 November 5, 2006, Athens, Georgia, USA*, volume 232 de *CEUR Workshop Proceedings*. Sun SITE Central Europe (CEUR), 2007.

[41] Chiara GHIDINI, Luciano SERAFINI et Sergio TESSARIS : On Relating Heterogeneous Elements from Different Ontologies. *In Modeling and Using Context, 6th International and Interdisciplinary Conference, CONTEXT 2007, Roskilde, Denmark, August 20-24, 2007, Proceedings*, volume 4635 de *Lecture Notes in Computer Science*, pages 234–247. Springer-Verlag, 2007.

[42] Silvio GHILARDI, Carsten LUTZ et Frank WOLTER : Did I Damage My Ontology ? A Case for Conservative Extensions in Description Logics. *In* Patrick DOHERTY, John MYLOPOULOS et Christopher A. WELTY, éditeurs : *Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning, Lake District of the United Kingdom, June 2-5, 2006*, pages 187–197. AAAI Press, 2006.

[43] Fausto GIUNCHIGLIA : Contextual Reasoning. *Epistemologica*, 16:345–364, 1993.

[44] Fausto GIUNCHIGLIA et Pavel SHVAIKO : Semantic Matching. *In Proceedings of the IJCAI 2003 Workshop on Ontologies and Distributed Systems*, pages 139–146, 2003.

[45] Wilfrid HODGES : *A Shorter Model Theory*. Cambridge University Press, 1997.

[46] Ian HORROCKS, Peter F. PATEL-SCHNEIDER, Harold BOLEY, Said TABET, Benjamin GROSOF et Mike DEAN : SWRL : A Semantic Web Rule Language - Combining OWL and RuleML. Rapport technique, World Wide Web Consortium (W3C), mai 21 2004.

[47] Marcus KRACHT et Frank WOLTER : Properties of Independently Axiomatizable Bimodal Logics. *The Journal of Symbolic Logic*, 56(4):1469–1485, 1991.

[48] Oliver KUTZ, Carsten LUTZ, Frank WOLTER et Michael ZAKHARYASCHEV : $\mathcal{E}$-connections of abstract description systems. *Artificial Intelligence*, 156(1):1–73, 2004.

[49] Alexander MAEDCHE, Boris MOTIK, Nuno SILVA et Raphael VOLTZ : MAFRA - A MApping FRAmework for Distributed Ontologies. *In Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, 13th International Conference, EKAW 2002, Siguenza, Spain, October 1-4, 2002, Proceedings*, pages 235–250, London, UK, 2002. Springer-Verlag.

[50] Deborah L. MCGUINNESS et Frank van HARMELEN : OWL Web Ontology Language Overview W3C Recommendation 10 February 2004. Rapport technique, World Wide Web Consortium (W3C), février 10 2004.

[51] Christian MEILICKE, Heiner STUCKENSCHMIDT et Andrei TAMILIN : Improving Automatically Created Mappings using Logical Reasoning. *In* V. Richard BENJAMIN, Jérôme EUZENAT, Natasha Fridman NOY, Pavel SHVAIKO, Heiner STUCKENSCHMIDT et Michael USCHOLD, éditeurs : *Proceedings of the 1st International Workshop on Ontology Matching (OM-2006) Collocated with the 5th International Semantic Web Conference (ISWC-2006), Athens, Georgia, USA, November 5, 2006*, 2006.

[52] Alistair MILES et Dan BRICKLEY : SKOS Core Guide. Note, W3C, 2005.

[53] Alistair MILES et Dan BRICKLEY : SKOS Core Vocabulary. Note, World Wide Web Consortium (W3C), 2005.

[54] Boris MOTIK : *Reasoning in Description Logics using Resolution and Deductive Databases*. Thèse de doctorat, Universität Karlsruhe, Germany, janvier 2006.

[55] Alan NASH, Philip A. BERNSTEIN et Sergey MELNIK : Composition of mappings given by embedded dependencies. *In* Chen LI, éditeur : *Proceedings of the Twenty-fourth ACM*

SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 13-15, 2005, Baltimore, Maryland, USA, pages 172–183, Baltimore, MA, USA, 2005. ACM Press.

[56] Wolfgang NEJDL, Boris WOLF, Stephen STAAB et Julien TANE : EDUTELLA : Searching and Annotating Resources within an RDF-based P2P Network. *In* Martin FRANK, Natasha Fridman NOY et Stephen STAAB, éditeurs : *Proceedings of the WWW2002 International Workshop on the Semantic Web, Hawaii, May 7, 2002*, volume 55 de *CEUR Workshop Proceedings*. Sun SITE Central Europe (CEUR), 2002.

[57] Hans Jürgen OHLBACH et Jana KOEHLER : Modal Logics, Description Logics and Arithmetic Reasoning. *Artificial Intelligence*, 109(1–2):1–31, 1999.

[58] Peter F. PATEL-SCHNEIDER, Patrick HAYES et Ian HORROCKS : OWL Web Ontology Language Semantics and Abstract Syntax W3C Recommendation 10 February 2004. Rapport technique, World Wide Web Consortium (W3C), février 10 2004.

[59] Racer Systems GmbH & Co. KG. *RacerPro Reference Manual, Version 1.9*, décembre 2005.

[60] Yves RAIMOND, Samer ABDALLAH, Mark SANDLER et Frédérick GIASSON : The Music Ontology. *In Proceedings of the 8th International Conference on Music Information Retrieval*, San Mateo, California, 2007.

[61] David A. RANDELL, Zhan CUI et Anthony G. COHN : A Spatial Logic Based on Regions and Connection. *In* Bernhard NEBEL, Charles RICH et William SWARTOUT, éditeurs : *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR'92). Cambridge, MA, October 25-29, 1992*, pages 165–176, San Mateo, California, 1992. Morgan Kaufmann.

[62] François SCHARFFE et Jos de BRUIJN : A Language to Specify Mappings Between Ontologies. *In* Richard CHBEIR, Albert DIPANDA et Kokou YÉTONGNON, éditeurs : *Proceedings of the 1st International Conference on Signal-Image Technology and Internet-Based Systems, SITIS 2005, November 27 - December 1, 2005, Yaounde, Cameroon*, pages 267–271. Dicolor Press, 2005.

[63] Klaus SCHILD : Towards a Theory of Frames and Rules. Rapport technique, Berlin, Germany, 1989.

[64] Luciano SERAFINI et Andrei TAMILIN : DRAGO : Distributed Reasoning Architecture for the Semantic Web. *In* Asuncion GOMEZ-PEREZ et Jérôme EUZENAT, éditeurs : *The Semantic Web : Research and Applications, Second European Semantic Web Conference, ESWC 2005, Heraklion, Crete, Greece, May 29 - June 1, 2005, Proceedings*, volume 3532 de *Lecture Notes in Computer Science*, pages 361–376, Hersounisous (GR), mai 2005. Springer-Verlag.

[65] Luciano SERAFINI et Andrei TAMILIN : Instance Migration in Heterogeneous Ontology Environments. *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 et Philippe CUDRÉ-MAUROUX, éditeurs : *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007*, volume 4825 de *Lecture Notes in Computer Science*, pages 452–465. Springer-Verlag, 2007.

[66] Nuno SILVA et João ROCHA : Semi-Automatic Bridging for Knowledge based Interoperability. *In* Gabriele KOTSIS, David TANIAR, Stéphane BRESSAN, Ismail Khalil IBRAHIM et Salimah MOKHTAR, éditeurs : *iiWAS'2005 - The Seventh International Conference on Information Integrationand Web-based Applications Services, 19-21 September 2005, Kuala Lumpur, Malaysia*, volume 196 de *books@ocg.at*, pages 921–932. Austrian Computer Society, 2005.

[67] Evren SIRIN, Bijan PARSIA, Bernardo CUENCA-GRAU, Aditya KALYANPUR et Yarden KATZ : Pellet : A pratical OWL-DL reasoner. *Journal of Web Semantics*, 5(2):51–53, 2007.

[68] John F. SOWA : *Knowledge Representation : Logical, Philosophical, and Computational Foundations*. Brooks Cole Publishing Co., Pacific Grove, CA, 2000.

[69] Umberto STRACCIA : Reasoning within Fuzzy Description Logics. *Journal of Artificial Intelligence Research*, 14:137–166, 2001.

[70] Heiner STUCKENSCHMIDT et Michel KLEIN : Integrity and Change in Modular Ontologies. *In* Georg GOTTLOB et Toby WALSH, éditeurs : *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003*, pages 900–908. Morgan Kaufmann, août 2003.

[71] Alfred TARSKI : On the calculus of relations. *Journal of Symbolic Logic*, 6:73–89, 1941.

[72] Dmitry TSARKOV et Ian HORROCKS : *FaCT++* Description Logic Reasoner : System Description. *In* Ulrich FURBACH et Natarajan SHANKAR, éditeurs : *Automated Reasoning, Third International Joint Conference, IJCAR 2006, Seattle, WA, USA, August 17-20, 2006, Proceedings*, volume 4130 de *Lecture Notes in Computer Science*, pages 292–297. Springer-Verlag, 2006.

[73] Jeffrey D. ULLMAN : Information Integration Using Logical Views. *In* Foto N. AFRATI et Phokion G. KOLAITIS, éditeurs : *Database Theory - ICDT '97, 6th International Conference, Delphi, Greece, January 8-10, 1997, Proceedings*, volume 1186 de *Lecture Notes in Computer Science*, pages 19–40. Springer-Verlag, 1997.

[74] Michael USCHOLD et Christopher MENZEL : Semantic Integration & Interoperability Using RDF and OWL, W3C Editor's Draft 3 November 2005. Rapport technique, World Wide Web Consortium (W3C), novembre 3 2005.

[75] Christoph WEIDENBACH, Renate A. SCHMIDT, Thomas HILLENBRAND, Rostislav RUSEV et Dalibor TOPIC : System Description : SpassVersion 3.0. *In* Frank PFENNING, éditeur : *Automated Deduction - CADE-21, 21st International Conference on Automated Deduction, Bremen, Germany, July 17-20, 2007, Proceedings*, volume 4603 de *Lecture Notes in Computer Science*, pages 514–520. Springer-Verlag, 2007.

[76] Gio WIEDERHOLD : Mediators in the Architecture of Future Information Systems. *IEEE Computer*, 25(3):38–49, 1992.

[77] Antoine ZIMMERMANN : Integrated Distributed Description Logics. *In* Diego CALVANESE, Enrico FRANCONI, Volker HAARSLEV, Domenico LEMBO, Boris MOTIK, Sergio TESSARIS et Anni-Yasmin TURHAN, éditeurs : *Proceedings of the 20th International Workshop on Description Logics DL'07, June 8 - 10, 2007, Brixen/Bressanone, Italy*, pages 507–514. Bolzano University Press, juin 2007.

[78] Antoine ZIMMERMANN et Jérôme EUZENAT : Three Semantics for Distributed Systems and their Relations with Alignment Composition. *In* Isabel F. CRUZ, Stephan DECKER, Dean ALLEMANG, Christ PREIST, Daniel SCHWABE, Peter MIKA, Michael USCHOLD et Lora AROYO, éditeurs : *The Semantic Web - ISWC 2006, 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006, Proceedings*, volume 4273 de *Lecture Notes in Computer Science*, pages 16–29. Springer-Verlag, novembre 2006.

[79] Antoine ZIMMERMANN, Markus KRÖTZSCH, Jérôme EUZENAT et Pascal HITZLER : Formalizing ontology alignment and its operations with category theory. *In* Brandon BENNETT et Christiane D. FELLBAUM, éditeurs : *International Conference on Formal Ontology in Information Systems (FOIS 2006), Baltimore, Maryland (USA), November 9-11, 2006*, pages 277–288. IOS Press, novembre 2006.

Annexes

A La représentation de connaissances

A.1 Syntaxe d'un langage de représentation de connaissances

La syntaxe d'un langage de représentation des connaissances se caractérise par un ensemble de symboles et des règles de construction permettant d'obtenir des formules à partir de ces symboles et de constructeurs spécifiques au langage.

Definition A.1 (Symbole) *Soit L un langage de représentation de connaissances. L'ensemble des symboles de L est noté $\mathbf{Sym}_L$.*

Exemple A.1 *Les lettres de l'alphabet forment un ensemble de symboles. Les mots que l'on peut former avec les 26 lettres de l'alphabet forment aussi un ensemble possible de symboles pour un langage.*

Les symboles peuvent être "typés" ce qui permet de distinguer des symboles d'individus, des symboles de relations reliant les individus, etc.

Definition A.2 (Type) *Soit L un langage de représentation de connaissances. Les types de L appartiennent à un ensemble fini $\mathbf{Type}_L$.*

Exemple A.2 *En logique du premier ordre, on dispose des types* constante*,* fonction *et* prédicat*. En logique de description, les types sont les* concepts*, les* rôles *et les* individus *(parfois aussi les* types de données*).*

Une portion typée des symboles du langage est appelée une *signature*.

Definition A.3 (Signature) *Soit L un langage de représentation de connaissances. Une* signature *Σ est une famille d'ensembles $(\Sigma_t)_{t \in \mathbf{Type}_L}$ indicée par les types de L. On note $\mathbf{Sign}_L$ l'ensemble de toutes les signatures de L.*

Exemple A.3 *En logique du premier ordre, une signature est un triplet d'ensembles $\langle \mathcal{C}, \mathcal{F}, \mathcal{P} \rangle$ où $\mathcal{C}$ correspond aux constantes, $\mathcal{F}$ aux fonctions et $\mathcal{P}$ aux prédicats. En logique de description $\langle \mathcal{C}, \mathcal{R}, \mathcal{U} \rangle$ définit une signature lorsque $\mathcal{C}$ est un ensemble de concepts, $\mathcal{R}$ un ensemble de rôles et $\mathcal{U}$ un ensemble d'individus.*

Chaque signature permet de définir un ensemble de formules déterminé par le langage. Ces formules servent à énoncer des faits sur le monde.

Definition A.4 (Formule) *Soit L un langage de représentation de connaissances. Pour toute signature $\Sigma \in \mathbf{Sign}_L$, L définit un ensemble de formules $\mathbf{For}_L(\Sigma)$.*

Exemple A.4 *En logique du premier ordre, $\forall x \exists y. f(x) = 0 \Rightarrow P(x, y)$ est une formule. $\forall R.C \sqcap D \sqsubseteq \exists S.E$ est une formule de logique de description. Les règles de construction pour les logiques de description sont détaillées plus bas (section A.4).*

Le plus souvent, les règles de construction des formules d'un langage sont données à l'aide d'une grammaire formelle.

Un ensemble de formules d'une même signature constitue une ontologie. Ceci correspond à une description formelle d'un domaine de connaissances. Il s'agit de la représentation syntaxique d'un nœud de connaissances dans un réseau de connaissances interconnectées.

Definition A.5 (Ontologie) *Soit L un langage de représentation de connaissances. Une* ontologie *O de L est un couple $\langle \Sigma(O), A(O) \rangle$ où $\Sigma(O)$ est une signature de L et $A(O)$ est un sous-ensemble de formules de $\mathbf{For}_L(\Sigma)$. On appelle les formules de $A(O)$ des* axiomes *de O.*

Exemple A.5 *Définissons l'ontologie M décrivant des connaissances sur la musique. La signature de M contient des termes comme $Musicien$, $Instrument$, $Notes$, $Tempo$, $joue$, $compose$, etc. Parmi les axiomes de cette ontologie, on trouve "Un $Musicien$ $joue$ $d'un$ $Instrument$". Les mots "Un" et "$d'un$" sont des éléments syntaxiques superflus permettant de comprendre plus facilement la formule, mais ne font pas partie de la signature de l'ontologie. En logique de description on écrirait cela $Musicien \sqsubseteq \exists joue.Instrument$.*

A.2 Sémantique des langages de représentation de connaissances

La sémantique d'un langage de représentation des connaissances se caractérise par :
- une notion d'interprétation permettant d'associer aux symboles d'une signature des éléments d'un certain "univers" ou domaine d'interprétation,
- et des contraintes définissant la satisfaction d'une formule.

Definition A.6 (Interprétation) *Soit L un langage de représentation de connaissances. Soit Σ une signature de L. Une* interprétation *I est un couple $\langle \mathbf{I}, \Delta \rangle$ où Δ est une famille d'ensembles non vides $(\Delta_t)_{t \in \mathbf{Type}_L}$ indicée par les types de L, et $\mathbf{I}$ est une famille de fonctions de $I_t : \Sigma_t \to \Delta_t$ appelée* fonction d'interprétation. *Δ est appelée* domaine d'interprétation. *Une interprétation d'une ontologie O est une interprétation de la signature de O. On note $\mathbf{Int}_L(\Sigma)$ (respectivement $\mathbf{Int}_L(O)$) l'ensemble des interprétations de Σ (respectivement de O).*

Exemple A.6 *Étant donnée une signature $\Sigma = \langle \mathcal{C}, \mathcal{R}, \mathcal{U} \rangle$ en logique de description, une interprétation I de Σ associe à chaque concept $C \in \mathcal{C}$ un ensemble $C^I \subseteq \Delta$, à chaque rôle $R \in \mathcal{R}$ une relation $R^I \subseteq \Delta \times \Delta$ et à chaque individu $o \in \mathcal{U}$ un élément $o^I \in \Delta$.*

La seule notion d'interprétation ne suffit pas à elle seule pour raisonner sur des connaissances. En effet, elle ne permet que de relier les symboles à des éléments de l'univers de discours modélisé par le domaine d'interprétation. Ainsi, elle ne permet pas de déterminer ce qui est correct vis-à-vis d'une ontologie. La notion cruciale pour cela est celle de satisfaction. Elle sert à relier les formules aux interprétations qui satisfont ces formules, c'est-à-dire qui valident la connaissance exprimée par ces formules.

Definition A.7 (Relation de satisfaction) *Soit L un langage de représentation de connaissances. Pour chaque signature Σ de L, le langage définit une relation $\models_{L,\Sigma} \subseteq \mathbf{Int}_L(\Sigma) \times \mathbf{For}_L(\Sigma)$ entre les interprétations de Σ et les formules construites à partir de Σ. Lorsque $\langle I, f \rangle \in \models_{L,\Sigma}$, on note $I \models_{L,\Sigma} f$ et on dit que I satisfait f.*

Il est rare que l'on précise la signature en indice de la relation de satisfaction car elle ne pose généralement pas de problème d'ambigüité.

Exemple A.7 *Considérons les concepts C et D en logique de description et la formule $C \sqsubseteq D$. Elle est satisfaite par une interprétation I si et seulement l'interprétation de C par I est un sous-ensemble de l'interprétation de D par I. Plus formellement, $I \models_L C \sqsubseteq D \Leftrightarrow C^I \subseteq D^I$.*

On étend la notion de satisfaction aux ontologies de façon très naturelle. Une interprétation satisfaisant une ontologie est appelée modèle de l'ontologie.

Definition A.8 (Modèle d'une ontologie) *Soit L un langage de représentation de connaissances. Soit O une ontologie de L. Une interprétation I de O satisfait O si et seulement si I satisfait tous les axiomes de O. Dans ce cas, on note $I \models_L O$ et on dit que I est un modèle de O. Par ailleurs, on note $\mathbf{Mod}(O)$ l'ensemble de tous les modèles de O.*

Exemple A.8 *En reprenant l'exemple de l'ontologie de la musique, on peut interpréter le terme $\mathrm{Musicien}$ comme l'ensemble des personnes qui ont composé une musique dans leur vie, le terme $\mathrm{Instrument}$ comme l'ensemble des partitions et le terme joue comme la relation entre les personnes et les partitions qu'elles composent. Cette interprétation satisfait l'axiome.*

En revanche, en interprétant le terme $\mathrm{Musicien}$ comme l'ensemble de toutes les personnes, le terme $\mathrm{Instrument}$ comme l'ensemble des instruments de musiques et le terme joue comme la relation qui lie une personne aux instruments dont il sait jouer, l'axiome n'est plus vérifié puisqu'il y a des personnes qui ne savent jouer d'aucun instrument.

Ces définitions préliminaires permettent de caractériser les raisonnements que l'on peut effectuer sur le langage et les ontologies.

A.3 Raisonnement sur les langages de représentation de connaissances

Les deux notions essentielles de raisonnement logiques sont la cohérence d'une ontologie (*consistency*) et la conséquence sémantique (*entailment*).

Definition A.9 (Cohérence) *Soit L un langage de représentation de connaissances. Une ontologie O de L est cohérente si et seulement si il existe un modèle de O.*

Exemple A.9 *L'ontologie $O_1 = \{$Il n'y a pas de fumée sans feu.$\}$ est cohérente. En revanche, l'ontologie $O_2 = \{$Il n'y a pas de fumée sans feu.,L'azote liquide au contact de l'air produit de la fumée mais pas de feu.,Il existe de l'azote liquide au contact de l'air.$\}$ est incohérente. Plus formellement, on pourrait écrire $O_2 = \{$fumée$\Rightarrow$feu.,azote$\wedge$air$\Rightarrow \neg$feu$\wedge$fumée.,azote$\wedge$air.$\}$.*

Definition A.10 (Conséquence sémantique) *Soit L un langage de représentation de connaissances. Soit O une ontologie et f une formule de L. On dit que f est une conséquence sémantique de O lorsque tous les modèles de O satisfont f. Dans ce cas, on note $O \models_L f$.*

Exemple A.10 *Si dans l'ontologie de la musique on trouve un axiome indiquant que $Paul$ est $Musicien$ (en logique de description, on écrira $Musicien(Paul)$), la formule "il existe un $Instrument$" (qui ne s'exprime pas en logique de description) est une conséquence sémantique de l'ontologie.*

Dans certains cas, il est possible de réduire le problème de la conséquence sémantique ("f est-il conséquence de O ?") à celui de la cohérence d'une ontologie (plus exactement, de l'incohérence). En effet, il suffit que le langage de représentation de connaissance puisse exprimer la négation de formules. Si l'on note $\mathrm{non}(f)$ la négation d'une formule f, alors $O \models_L f$ équivalent à dire que $\langle \Sigma(O), A(O) \cup \{\mathrm{non}(f)\} \rangle$ est incohérent.

Déterminer la cohérence d'une ontologie et en déduire les conséquences sémantiques sont des enjeux majeurs de la représentation des connaissances. Disposer d'une sémantique formelle permet de poser ces questions formellement et, par suite, de construire des procédures automatiques pour y répondre. Néanmoins, il n'est pas toujours possible de produire de telles procédures dans le cas général. Seuls les langages décidables autorisent la construction de procédures répondant à ces questions quelle que soit l'ontologie.

Definition A.11 (Décidabilité) *Soit L un langage de représentation de connaissances. On dit que L est décidable si et seulement si pour toute ontologie O et toute formule f de $\mathbf{For}(\Sigma(O))L$, il existe un algorithme permettant de déterminer si f est une conséquence sémantique de O ou non.*

Exemple A.11 *La logique du première ordre est indécidable.*

Lorsque le langage est décidable, il existe un algorithme déterminant si une formule est conséquence d'une ontologie. Il est donc intéressant de savoir si un algorithme donné détermine bel et bien les conséquences d'une ontologie. Un algorithme est dit correct si tout ce qu'il permet de déduire est effectivement une conséquence sémantique et il est dit complet si toutes les conséquences sémantiques peuvent être déduites.

Definition A.12 (Correction et complétude) *Soit L un langage de représentation de connaissances. Un algorithme A retournant $\mathbf{VRAI}$ ou $\mathbf{FAUX}$ en fonction d'une ontologie O de L et d'une formule f de $\mathbf{for}(\Sigma(O))$ en entrée ($A(O, f) \in \{\mathbf{VRAI}, \mathbf{FAUX}\}$) est dit* correct *si et seulement pour tout O, f, $A(O, f) = \mathbf{VRAI}$ implique $O \models_L f$. Il est dit* complet *si et seulement pour tout O, f, $O \models_L f$ implique $A(O, f) = \mathbf{VRAI}$.*

Habituellement, on cherche à définir des algorithmes à la fois corrects et complets, à condition bien sûr que le langage soit décidable. En revanche, pour des raisons pratiques de complexité du raisonnement, on peut se contenter de la correction ou bien de la complétude si l'opération est suffisamment efficace.

A.4 Exemple : logiques de description

Les logiques de descriptions sont une famille de langages de représentations de connaissances. Ces logiques ont été introduites dans les années 80 dans le but de se donner un langage permettant de définir une terminologie d'une façon supposée plus "naturel" qu'en logique du premier ordre. Elles sont aussi moins expressives que la logique du premier ordre puisque la plupart de ces logiques forment un sous-ensemble décidable de la logique du premier ordre. Maintenant, il existe une grande variété de logiques de description dont l'expressivité et l'efficacité de raisonnement dépend de l'utilisation ou non de constructeurs spécifiques.

Les ontologies en logique de description modélisent les connaissances selon deux niveaux de description : le niveau terminologique (*TBox*) et le niveau factuel (*ABox*). Le niveau terminologique définit des concepts et des rôles représentant respectivement des ensembles d'entités du monde et des relations entre ces entités. Le niveau factuel, quant à lui, énonce des faits sur des individus représentant les entités elles-mêmes.

Niveau terminologique : Les concepts et rôles sont définis soit en les nommant (concepts ou rôles atomiques), soit en les construisant à l'aide d'autres concepts ou rôles et de constructeurs spécifiques. Les connaissances sur ces concepts et rôles sont décrites à l'aide d'axiomes indiquant la subsomption (de concepts ou de rôles) ou la transitivité d'un rôle. La table 8.1 donne une liste de constructeurs existant en logiques de description. Elle n'est pas exhaustive.

Termes atomiques	Syntaxe	Sémantique	
individu	a	$a^I \in \Delta^I$	
concept atomique	A	$A^I \subseteq \Delta^I$	
rôle atomique	R	$R^I \subseteq \Delta^I \times \Delta^I$	$(\mathcal{AL})$
Constructeur de concepts	Syntaxe	Sémantique	nom
concept universel	$\top$	$\top^I = \Delta^I$	$(\mathcal{AL})$
concept vide	$\bot$	$\bot^I = \emptyset$	$(\mathcal{AL})$
conjonction	$C \sqcap D$	$(C \sqcap D)^I = C^I \cap D^I$	$(\mathcal{AL})$
disjonction	$C \sqcup D$	$(C \sqcup D)^I = C^I \cup D^I$	$\mathcal{U}$
négation	$\neg C$	$\Delta^I \setminus C^I$	$\mathcal{C}$
restriction existentielle	$\exists R.C$	$\{x \mid \exists y.\langle x,y\rangle \in R^I \wedge y \in C^I\}$	$(\mathcal{AL})$
restriction de valeur	$\forall R.C$	$\{x \mid \forall y.\langle x,y\rangle \in R^I \Rightarrow y \in C^I\}$	$(\mathcal{AL})$
restrictions	$\leq nR$	$\{x \mid \sharp\{y.\langle x,y\rangle \in R^I\} \leq n\}$	$\mathcal{N}$
numériques	$\geq nR$	$\{x \mid \sharp\{y.\langle x,y\rangle \in R^I\} \geq n\}$	$\mathcal{N}$
nominaux	$\{a_1, \ldots, a_n\}$	$\{a_1^I, \ldots, a_n^I\}$	$\mathcal{O}$
Constructeur de rôles	Syntaxe	Sémantique	nom
conjonction de rôles	$R \sqcap S$	$(R \sqcap S)^I = R^I \cap S^I$	$\cdot^{\sqcap}$
disjonction de rôles	$R \sqcup S$	$(R \sqcup S)^I = R^I \cup S^I$	$\cdot^{\sqcup}$
complément de rôle	$\neg R$	$(\Delta^I \times \Delta^I) \setminus R^I$	$\cdot^{\neg}$
clôture transitive	R^+	transitive closure of R^I	$\cdot^{+}$
rôle inverse	R^-	$\{\langle y,x\rangle \mid \langle x,y\rangle \in R^I\}$	$\mathcal{I}$
composition de rôles	$R \circ S$	$\{\langle x,z\rangle \mid \exists y.\langle x,y\rangle \in R^I \wedge \langle y,z\rangle \in S^I\}$	$\cdot^{\circ}$
axiomes de TBox	Syntaxe	Contraintes d'interprétation	nom
subsomption	$C \sqsubseteq D$	$C^I \subseteq D^I$	$(\mathcal{AL})$
inclusion de rôles	$R \sqsubseteq S$	$R^I \subseteq S^I$	$\mathcal{H}$
transitivité de rôle	Trans(R)	$R^I = (R^+)^I$	$\mathcal{S}$
axiomes de ABox	Syntaxe	Contraintes d'interprétation	nom
appartenance à un concept	$C(a)$	$a^I \in C^I$	$(\mathcal{AL})$
appartenance à un rôle	$R(a_1, a_2)$	$\langle a_1^I, a_2^I\rangle \in R^I$	$(\mathcal{AL})$
idendité	$a_1 = a_2$	$a_1^I = a_2^I$	?

TABLE 8.1 – Syntaxe et sémantique des constructeurs de logiques de description.

Une logique de description est un langage utilisant une partie de ces constructeurs. On nomme généralement une logique de description en prenant comme base une logique simple comme $\mathcal{AL}$ et en lui accolant le nom des constructeurs qu'on autorise. Par exemple, $\mathcal{ALCHI}^{\sqcap,\neg}$ autorise les constructeurs d'$\mathcal{AL}$ ainsi que la négation de concept, la hiérarchie de rôles, les rôles inverses, la conjonction de rôles et le complément de rôle. Il existe d'autres noms de logiques simples comme $\mathcal{EL}$, $\mathcal{SI}$, DL-Lite, etc. dont certaines imposent des restrictions supplémentaires.

Niveau factuel : Ce niveau n'introduit que des noms d'individus et des propriétés sur ces individus, en exploitant les termes du niveau terminologique. Les axiomes de la ABox sont appelés *assertions*.

Les définitions générales données en section 2.2 peuvent être réécrites de la manière suivante pour les logiques de description.

Definition A.13 (Signature) *Soit LD une logique de description. Une signature de LD est un triplet $\langle \mathcal{C}, \mathcal{R}, \mathcal{U} \rangle$ tel que $\mathcal{C}$ est un ensemble de concepts atomiques, $\mathcal{R}$ un ensemble de rôle atomique et $\mathcal{U}$ un ensemble d'individus.*

Definition A.14 (Formule) *Soit LD une logique de description. Une formule de LD pour une signature Σ est l'un des axiomes de TBox ou de ABox indiqués dans la table 8.1 et autorisés par le langage, où C et D sont soit des concepts atomiques, soit des concepts complexes obtenus en utilisant les constructeurs du langage ; R et S sont soit des rôles atomiques, soit des rôles construits à l'aide des constructeurs de rôles ; a, a_1 et a_2 sont des individus.*

Definition A.15 (Ontologie) *Soit LD une logique de description. Une ontologie de LD est une paire $\langle \Sigma, F \rangle$ telle que Σ est une signature et $F = \langle T, A \rangle$ contient des axiomes de TBox T et des assertions de ABox A.*

Definition A.16 (Interprétation) *Soit LD une logique de description et $\Sigma = \langle \mathcal{C}, \mathcal{R}, \mathcal{U} \rangle$ une signature de LD. Une interprétation de Σ est une paire $\langle I, \Delta \rangle$ où Δ est un ensemble non vide et I un triplet de fonctions $I_{\mathcal{C}} : \mathcal{C} \rightarrow 2^{\Delta}$, $I_{\mathcal{R}} : \mathcal{R} \rightarrow 2^{\Delta \times \Delta}$ et $I_{\mathcal{U}} : \mathcal{U} \rightarrow \Delta$. On étend la fonction d'interprétation aux concepts complexes en appliquant récursivement les règles de la table 8.1.*

Definition A.17 (Satisfaction) *Soit LD une logique de description et Σ une signature de LD. Soit I une interprétation de Σ et f une formule de* $\mathbf{For}(\Sigma)$. *I satisfait f si et seulement la contrainte d'interprétation associé à la formule dans la table 8.1 est satisfaite.*

Le langage OWL DL est une instance particulière des logiques de description, plus précisément $\mathcal{SHION}(D)$, et sa syntaxe abstraite et sa sémantique sont détaillée de façon plus spécifique dans [58].

B Sémantique des alignements expressifs

Syntaxe Abstraite	Interprétation
Litéraux (v)	$\mathbb{D}$
"v"^^d	$L2V(D(d))(v)$
"v"	"v"
Valeurs (V)	$2^{\Delta\times(\Delta\cup\mathbb{D})}$
v	$\Delta \times \{v^{\mathcal{I}_L}\}$
i	$\mathcal{D} \times \{i^{\mathcal{I}}\}$
R	$R^{\mathcal{I}}$
P	$P^{\mathcal{I}}$
transf$(V_1, \ldots, V_n)$	$\{\langle x, h_{\text{transf}}(y_1, \ldots, y_n)\rangle \mid \langle x, y_1\rangle \in V_1^{\mathcal{I}_V} \wedge \cdots \wedge \langle x, y_n\rangle \in V_n^{\mathcal{I}_V}\}$
Restrictions (K)	2^{Δ}
$Q\ cp\ V$	$\{x \in \Delta \mid \exists y, y' \in \Delta, \langle x, y\rangle \in Q^{\mathcal{I}} \wedge \langle x, y'\rangle \in V^{\mathcal{I}_V} \wedge g_{\text{cp}}(y, y')\}$
$P\ cp\ d$	$\{x \in \Delta \mid \forall y \in \Delta, \langle x, y\rangle \in P^{\mathcal{I}} \Rightarrow g_{\text{cp}}(y, V(D(d)))\}$
$Q\ cp\ n$	$\{x \in \Delta \mid g_{\text{cp}}(\text{Card}_{\mathcal{I}}(x, Q), n)\}$
Classes (C)	2^{Δ}
c	$c^{\mathcal{I}}$
$C \sqcup C'$	$C^{\mathcal{I}} \cup C'^{\mathcal{I}}$
$C \sqcap C'$	$C^{\mathcal{I}} \cap C'^{\mathcal{I}}$
$\neg\mathcal{C}$	$\Delta \setminus C^{\mathcal{I}}$
$\exists K$	$K^{\mathcal{I}}$
Relations (R)	$2^{\Delta\times\Delta}$
r	$r^{\mathcal{I}}$
self	$\{\langle x, x\rangle \mid x \in \Delta\}$
$R \sqcup R'$	$R^{\mathcal{I}} \cup R'^{\mathcal{I}}$
$R \sqcap R'$	$R^{\mathcal{I}} \cap R'^{\mathcal{I}}$
$\neg R$	$\Delta \times \Delta \setminus R^{\mathcal{I}}$
$R \circ R'$	$\{\langle x, y\rangle \mid \exists y \in \Delta, \langle x, z\rangle \in R^{\mathcal{I}} \wedge \langle y, z\rangle \in R'^{\mathcal{I}}$
dom(C)	$\{\langle x, y\rangle \in \Delta \times \Delta \mid x \in C^{\mathcal{I}}\}$
range(C)	$\{\langle x, y\rangle \in \Delta \times \Delta \mid y \in C^{\mathcal{I}}\}$
inv(R)	$\{\langle x, y\rangle \mid \langle y, x\rangle \in R^{\mathcal{I}}\}$
sym(R)	$R^{\mathcal{I}} \cup \text{inv}(R)^{\mathcal{I}}$
trans(R)	$R^{\mathcal{I}} \cup \{\langle x, z\rangle \mid \exists y \in \Delta, \langle x, y\rangle \in R^{\mathcal{I}} \wedge \langle y, z\rangle \in \text{trans}(R)^{\mathcal{I}}\}$
refl(R)	$R^{\mathcal{I}} \cup \text{self}^{\mathcal{I}}$
Propriétés (P)	$2^{\Delta\times\mathbb{D}}$
p	$p^{\mathcal{I}_U}$
$P \sqcup P'$	$P^{\mathcal{I}} \cup P'^{\mathcal{I}}$
$P \sqcap P'$	$P^{\mathcal{I}} \cap P'^{\mathcal{I}}$
$\neg P$	$\Delta \times \mathbb{D} \setminus P^{\mathcal{I}}$
$R.P$	$\{\langle x, z\rangle\Delta \times \mathbb{D} \mid \exists y \in \Delta, \langle x, y\rangle \in R^{\mathcal{I}} \wedge \langle y, z\rangle \in P^{\mathcal{I}}$
dom(C)	$\{\langle x, y\rangle \in \Delta \times \mathbb{D} \mid x \in C^{\mathcal{I}}\}$
range(d)	$\{\langle x, y\rangle \in \Delta \times \mathbb{D} \mid y \in V(D(d))\}$

TABLE 8.2: Syntaxe abstraite et sémantique.

C Table des opérateurs

Type	Id	Origine	explication
numérique	add	XQuery	Renvoie la somme arithmétique du premier et du second arguments.
numérique	subtract	XQuery	Renvoie la différence arithmétique du premier argument moins le second argument.
numérique	multiply	XQuery	Renvoie le produit arithmétique du premier argument par le second argument.
numérique	divide	XQuery	Renvoie le quotient arithmétique du premier argument sur le second argument.
numérique	integer-divide	XQuery	Renvoie la partie entière du quotient arithmétique du premier argument sur le second argument.
numérique	mod	XQuery	Renvoie le modulo du quotient arithmétique du premier argument sur le second argument.
numérique	pow		Renvoie le premier argument élevé à la puissance du second argument.
numérique	unary-minus	XQuery	Renvoie l'argument avec son signe inversé.
chaîne	concat	XQuery	Renvoie la concaténation du premier et du deuxième argument.
chaîne	substring	XQuery	Renvoie la sous-chaine du premier argument commençant à la position dénotée par le deuxième argument et finissant à celle dénotée par le troisième argument.
chaîne	length	XQuery	Renvoie l'entier correspondand au nombre de caractères de la chaîne en argument.
chaîne	normalize-space	XQuery	Renvoie une chaîne obtenue par normalisation des espaces de la chaîne donnée en argument.
chaîne	upper-case	XQuery	Renvoie la chaîne contenant les lettres capitales correspondant à la chaîne en argument.
chaîne	lower-case	XQuery	Renvoie la chaîne contenant les lettres de bas de casse correspondant) la chaîne en argument.
chaîne	translate	XPath/XQuery	Renvoie la chaîne correspondant au premier argument dans lequel ses occurrences de caractères contenus dans le deuxième argument sont remplacées par les caractères à la position correspondante dans la chaîne en troisième argument.
chaîne	replace	XQuery	Renvoie la chaîne correspondant au premier argument dans lequel les sous-chaînes s'accordant avec l'expression régulière du deuxième argument sont remplacées par la chaîne dans le troisième argument.
chaîne	tokenize	XQuery	Renvoie une suite de chaînes dont les valeurs sont des sous-chaînes du premier argument séparées par des sous-chaînes qui s'accordent avec l'expression régulière du second argument.
uri	resolveURI	XQuery	Renvoie l'URI reference résolue de l'argument.
collection	concatenate	XQuery	Renvoie la concaténation des éléments de la liste donnée en argument.
collection	intersection		Renvoie une liste contenant les éléments se trouvant à la fois dans le premier argument et dans le second argument.
collection	union		Renvoie une liste contenant les éléments se trouvant dans l'un ou l'autre des arguments.
collection	difference		Renvoie une liste contenant les éléments du premier argument qui ne sont pas dans le second argument.
integer	length	Lisp	Renvoie le nombre d'éléments dans la liste donnée en argument.

TABLE 8.3: Opérateurs.

D Table des comparateurs

Type	Id	Origine	explication
tous	equal	XQuery	Satisfait si et seulement si le premier argument et le second argument sont les même.
tous	not-equal	SWRL	La négation de equal.
ordonné	less-than	XQuery	Satisfait si et seulement si le premier argument et le second argument sont tous les deux d'un même type implémenter et le premier argument est inférieur au second argument selon un ordre spécifique au type (partiel ou total), dans le cas où cet ordre est bien défini pour ce type. L'ordre pour les littéraux non typés est défini par la relation d'ordre des chaînes imposée par la marque de la langue (*language tag*) lorsqu'elles est la même pour les deux chaînes (ou toute deux manquante) et sont incomparables autrement.
ordonné	less-than-or-equal	SWRL	Équivaut à la disjonction de equal et de less-than (equal $\wedge$ less-than).
ordonné	greater-than	XQuery	Comme less-than, mais avec la relation d'ordre inverse.
ordonné	greater-than-or-equal	SWRL	Comme less-than-or-equal, mais en utilisant la relation d'ordre inverse.
chaîne	contains	XQuery	Satisfait si et seulement si le premier argument contient une sous-chaîne égale au second argument (sensible à la casse)
chaîne	starts-with	XQuery	Satisfait si et seulement si le premier argument commence avec les caractères du second argument.
chaîne	ends-with	XQuery	Satisfait si et seulement si le premier argument fini par les caractères du second argument.
chaîne	matches	XQuery	Satisfait si et seulement si le premier argument s'accorde avec l'expression régulière dans le second argument.
collection	contains	XQuery	Satisfait si et seulement si le premier argument contient le second argument
collection	includes	XQuery	Satisfait si et seulement si le premier argument contient tous les éléments du second argument.
collection	includes-strictly	XQuery	Satisfait si et seulement si le premier argument contient tous les éléments du second argument et au moins un autre élément.
collection	empty		Satisfait si et seulement si le premier argument est une liste vide.

TABLE 8.4: Comparateurs.

E Preuve du théorème 8.3.1

Théorème E.1 (Cohérence d'un réseau d'ontologies alignées) *Soit $S = \langle \mathbf{O}, \mathbf{A} \rangle$ un réseau d'ontologies alignées. S est cohérent si et seulement si il existe une configuration globale Ω pour S et une configuration locale Ω^i pour chaque $O_i \in \mathbf{O}$ vis-à-vis de Ω telles que l'ontologie d'alignement $\widehat{\mathbf{A}}_\Omega$ et les ontologies étendues $\{\widehat{O}_{\Omega^i}\}$ comme définies ci-dessus sont cohérentes.*

Preuve : La preuve de ce théorème est plutôt complexe, donc elle sera séparée en sections et sous-sections, avec des lemmes intermédiaires. Puisque le théorème déclare une équivalence (si et seulement si) nous séparons la preuve en deux grandes étapes, une pour l'implique "si" (c'est-à-dire $\Leftarrow$) dans la section E.1 et une pour l'implication inverse "seulement si" (c'est-à-dire $\Rightarrow$) en section E.2.

E.1 Si $\Leftarrow$

Dans cette partie, on suppose l'existence d'une configuration globale Ω et de configurations locales Ω^i telles que $\widehat{\mathbf{A}}_\Omega$ et $\widehat{O}_{\Omega^i}$ sont cohérentes. On prouve que S est cohérent dans ce cas en énonçant l'existence d'un modèle $\mathcal{I}_A$ de $\widehat{\mathbf{A}}_\Omega$ et des modèles $\mathcal{I}_i$ de $\widehat{O}_{\Omega^i}$ par le biais desquels on construira un modèle distribué de S. Plus précisément, les modèles de $\widehat{O}_{\Omega^i}$ sont aussi des modèles de O_i. Donc construire un modèle de S revient à construire une fonction d'égalisation $\boldsymbol{\varepsilon}$ pour $\mathbf{I} = (\mathcal{I}_i)_{i \in \mathbf{O}}$ telle que $\langle \mathbf{I}, \boldsymbol{\varepsilon} \rangle \models S$. La démonstration se déroule en trois étapes : d'abord, on construit la fonction d'égalisation ε_i en utilisant $\mathcal{I}_A$ et $\mathcal{I}_i$; puis, on démontre un lemme important qui aide à prouver que l'on a construit un modèle S ; enfin, la preuve effective de la satisfaction du réseau d'ontologies S est décomposée en preuves de la satisfaction de chaque type de correspondances.

Afin de faciliter les notations dans la preuve, on définit la fonction $\Theta : \Delta_i \rightarrow 2^{\mathscr{C}_i}$ qui associe à chaque élément $e \in \Delta_i$ l'ensemble des concepts de $\mathscr{C}_i$ ayant une interprétation contenant e (c'est-à-dire $\Theta(e) = \{X \in \mathscr{C}_i \mid e \in \mathcal{I}_i(X)\}$). Puisque $\top \in \mathscr{C}_i$ l'ensemble $\Theta(e)$ n'est jamais vide. Ceci implique que $C^{\mathscr{C}_i}_{\Theta(e)} \sqsubseteq \bot \notin \widehat{O}_{\Omega^i}$, c'est-à-dire $\Theta(e) \in \Omega^i$.

Construction de la fonction d'égalisation.

Selon l'hypothèse de cette section, il existe un modèle $\langle \Delta_A, .^{\mathcal{I}_A} \rangle$ de $\widehat{\mathbf{A}}_\Omega$ et un modèle $\langle \Delta_i, .^{\mathcal{I}_i} \rangle$ de $\widehat{O}_{\Omega^i}$ pour tout $i \in \mathbf{O}$.

On définit maintenant pour tout i une fonction ε_i de Δ_i vers Δ_A. Soit $e \in \Delta_i$.

D'après la définition 8.3.6 et la définition de $\Theta(e)$, il existe $W \in \Omega$ tel que $\widehat{\Theta(e)} \subseteq W$, c'est-à-dire $\Theta(e) \subseteq W|_i$. **(*)**

Pour définir l'image de e par ε_i, on distingue les deux cas suivants.

1. S'il existe $W' \in \Omega$ et $e \in \mathcal{I}_i(C^\top_{W'})$ alors on définit $\varepsilon_i(e) = \mathcal{I}_A(a_{W'})$ avec $C^\top_{W'}$ introduit par l'item 4 de la définition 8.3.7.

2. Sinon, on définit $\varepsilon_i(e) = \mathcal{I}_A(a_W)$ (où W a été introduit par **(*)**).

Ceci donne une définition valide et non-ambigüe de la fonction ε_i, puisque si $e \in \mathcal{I}_i(C^\top_{W'})$ et $e \in \mathcal{I}_i(C^\top_{W''})$ alors $W' = W''$ du fait de l'item 5 de la définition 8.3.7. La figure 8.5 illustre cette définition.

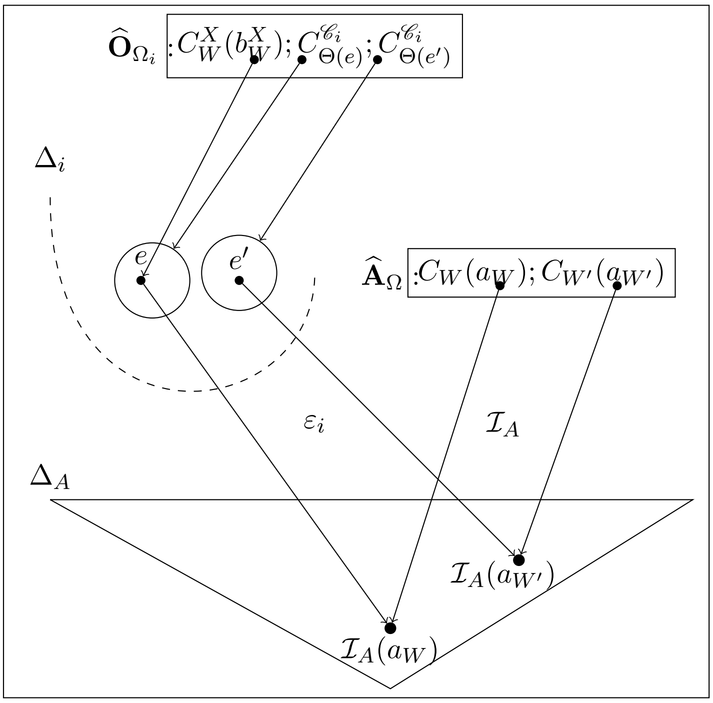


FIGURE 8.5 – Construction de ε_i à partir de $\mathcal{I}_i$ et $\mathcal{I}_A$. Pour tout $e = \mathcal{I}_i(b_W^X)$ ou $e \in \mathcal{I}_i(C_{\Theta(e)}^{\mathscr{C}_i})$ on peut trouver un individu a_W dans $\widehat{\mathbf{A}}_\Omega$ tel que $\varepsilon_i(e)$ est affectée à $\mathcal{I}_A(a_W)$.

Lemme auxiliaire.

Le lemme suivant assure que tout élément e d'un domaine d'interprétation local Δ_i peut être associé à un ensemble de concepts globaux $W \in \Omega$ tel que $\Theta(e) \subseteq W|_i$ et pour lequel $\varepsilon_i(e) = \mathcal{I}_A(a_W)$. Moins formellement, ce lemme sert à montrer que la structure locale d'inclusion de classes ou d'appartenance à une classe est préservée dans l'ontologie d'alignement.

Lemme E.1 *Pour tout $e \in \Delta_i$, il existe $W \in \Omega$ tel que $\Theta(e) \subseteq W|_i$ et $\varepsilon_i(e) = \mathcal{I}_A(a_W)$.*

Preuve : Pour prouver le lemme E.1, on considère les deux cas de la définition de ε_i :

1. S'il existe $W' \in \Omega$ et $e \in \mathcal{I}_i(C_{W'}^{\top})$ alors on a $\varepsilon_i(e) = \mathcal{I}_A(a_{W'})$. Ainsi,

$$e \in \bigcap_{X \in \mathscr{C}_i \setminus W'|_i} \Delta_i \setminus \mathcal{I}_i(X)$$

   Cela implique que $\Theta(e) \subseteq W'|_i$ (sinon, e serait dans $\mathcal{I}_i(\neg X)$ pour un certain $X \in \Theta(e)$).
2. Sinon, la propriété est évidemment vérifiée.

□

Satisfaction du réseau d'ontologies.

Puisque O_i est inclus dans $\widehat{O}_{\Omega^i}$ pour tout i, les modèles $\mathcal{I}_i$ de $\widehat{O}_{\Omega^i}$ sont aussi des modèles de O_i. Il ne reste donc qu'à montrer que les conditions de la définition 8.2.6 sont satisfaites avec les fonctions $\{\mathcal{I}_i\}$ et $\{\varepsilon_i\}$. Puisque les concepts peuvent être nominaux, l'appartenance inter-ontologies et l'identité inter-ontologies peuvent être traduites en subsomption de concept inter-ontologies et en équivalence de concept inter-ontologies. Ces preuves sont quelque peu techniques mais relativement directes, à condition d'utiliser convenablement le lemme E.1.

Subsomption de concepts inter-ontologies $(i\!:\!C \overset{\sqsubseteq}{\leftrightarrow} j\!:\!D \in A_{ij} \Longrightarrow \varepsilon_i(\mathcal{I}_i(C)) \subseteq \varepsilon_j(\mathcal{I}_j(D)))$. Supposons que $i:C \overset{\sqsubseteq}{\leftrightarrow} j:D \in A_{ij}$. Soit $x \in \varepsilon_i(\mathcal{I}_i(C))$. Il existe $e \in \mathcal{I}_i(C)$ tel que $\varepsilon_i(e) = x$. Par le lemme E.1, on sait qu'il existe $W \in \Omega$ tel que $\Theta(e) \subseteq W|_i$ et $\varepsilon_i(e) = \mathcal{I}_A(a_W)$. En outre, puisque $\mathcal{I}_A$ est un modèle de $\widehat{\mathbf{A}}_\Omega$ et que $i:C \sqsubseteq j:D \in \widehat{\mathbf{A}}_\Omega$ et $C_W \equiv \{a_W\} \in \widehat{\mathbf{A}}_\Omega$ (selon la définition 8.3.5) d'où $\mathcal{I}_A(a_W) \in \mathcal{I}_A(j:D)$. Par construction de C_W et du fait que $C_W \equiv \{a_W\}$ est dans $\widehat{\mathbf{A}}_\Omega$, on a $j:D \in W$ (sinon, $\neg j:D$ doit être inclus dans C_W, ce qui contredit $\mathcal{I}_A(a_W) \in \mathcal{I}_A(j\!:\!D)$). De la définition 8.3.7, on a $\widehat{O}_{\Omega^j} \models (D \sqcap \bigsqcap_{X' \in \mathscr{C}_j \setminus W|_j} \neg X')(b_W^D)$. Ceci implique $\mathcal{I}_j(b_W^D) \in \mathcal{I}_j(D)$. Aussi, $\widehat{O}_{\Omega^j} \models C_W^\top(b_W^D)$. De la définition de ε_j on obtient $\varepsilon_j(\mathcal{I}_j(b_W^D)) = \mathcal{I}_A(a_W)$. Ce qui implique que $x = \varepsilon_j(\mathcal{I}_j(b_W^D))$ et comme $\mathcal{I}_j(b_W^D) \in \mathcal{I}_j(D)$, $x \in \varepsilon_j(\mathcal{I}_j(D))$. Par conséquent, $\varepsilon_i(\mathcal{I}_i(C)) \subseteq \varepsilon_j(\mathcal{I}_j(D))$.

Exclusion de concepts inter-ontologies $(i\!:\!C \overset{\perp}{\leftrightarrow} j\!:\!D \in A_{ij} \Longrightarrow \varepsilon_i(\mathcal{I}_i(C)) \cap \varepsilon_j(\mathcal{I}_j(D)) = \emptyset)$. Supposons que $i:C \overset{\perp}{\leftrightarrow} j:D \in A_{ij}$. Soit $x \in \varepsilon_i(\mathcal{I}_i(C))$. Il existe $e \in \mathcal{I}_i(C)$ tel que $\varepsilon_i(e) = x$. D'après le lemme E.1, il s'ensuit qu'il existe $W \in \Omega$ tel que $\Theta(e) \subseteq W|_i$ et $\varepsilon_i(e) = \mathcal{I}_A(a_W)$. De surcroît, étant donné que $\mathcal{I}_A$ est un modèle pour $\widehat{\mathbf{A}}_\Omega$ et que $i:C \sqsubseteq \neg j:D \in \widehat{\mathbf{A}}_\Omega$ et $C_W \equiv \{a_W\} \in \widehat{\mathbf{A}}_\Omega$, on obtient $\mathcal{I}_A(a_W) \notin \mathcal{I}_A(j\!:\!D)$. **(**)**

Par ailleurs, pour tout $f \in \mathcal{I}_j(D)$, d'après le lemme E.1, il existe $W' \in \Omega$ tel que $\varepsilon_j(f) = \mathcal{I}_A(a_{W'})$ et $\Theta(f) \subseteq W'|_j$, donc $\mathcal{I}_A(a_{W'}) \in \mathcal{I}_A(j:D)$. De cela et de **(**)**, on obtient $\mathcal{I}_A(a_W) \neq \mathcal{I}_A(a_{W'})$. Ainsi, $\varepsilon_i(e) \neq \varepsilon_j(f)$ pour tout $f \in \mathcal{I}_j(D)$. Il en résulte que $\varepsilon_i(\mathcal{I}_i(C)) \cap \varepsilon_j(\mathcal{I}_j(D)) = \emptyset$.

Ainsi, l'interprétation distribuée $\langle(\mathcal{I}_i), (\varepsilon_i)\rangle$ satisfait le réseau d'ontologies S. Ce qui établit que S est cohérent.

E.2 Seulement si ⇒

Dans cette section, on suppose que S est cohérent. Il en découle qu'il existe un modèle distribué $\langle \mathbf{I}, \boldsymbol{\varepsilon}\rangle$ de S, avec $\mathbf{I} = (\langle \Delta_i, \cdot^{\mathcal{I}_i}\rangle)_{i \in \mathbf{O}}$ et $\boldsymbol{\varepsilon} = (\varepsilon_i : \Delta_i \to \Delta_{\boldsymbol{\varepsilon}})_{i \in \mathbf{O}}$. Il faut montrer qu'il existe une configuration globale Ω et une configuration locale Ω^i pour tout $i \in \mathbf{O}$, telles que $\widehat{\mathbf{A}}_\Omega$ et $\{\widehat{O}_{\Omega^i}\}$ sont cohérents.

La preuve est organisée comme suit :

1. Construire les configurations en utilisant les modèles $\langle \mathcal{I}, \boldsymbol{\varepsilon}\rangle$:
   (a) construire la configuration globale Ω ;
   (b) construire les configurations locales Ω^i pour tout $i \in \mathbf{O}$;
   (c) prouver que Ω^i est effectivement une configuration locale.
2. Prouver que $\widehat{\mathbf{A}}_\Omega$ est cohérente :
   (a) construire une interprétation $\mathcal{I}_A$ de $\widehat{\mathbf{A}}_\Omega$;
   (b) montrer que $\mathcal{I}_A \models \widehat{\mathbf{A}}_\Omega$.
3. Prouver que $\{\widehat{O}_{\Omega^i}\}$ est cohérent pour tout $i \in \mathbf{O}$:
   (a) construire une interprétation $\mathcal{I}_i$ de $\widehat{O}_{\Omega^i}$;
   (b) montrer que $\mathcal{I}_i \models \widehat{O}_{\Omega^i}$.

Avant de procéder à la preuve, on définit d'abord deux fonctions auxiliaires qui amélioreront la lisibilité de ce qui suit. La fonction φ associe un concept vocabulaire global $\mathscr{C}$ à un sous-ensemble du domaine global $\Delta_{\boldsymbol{\varepsilon}}$.

$$\begin{aligned} \varphi :\ & \mathscr{C} \longrightarrow 2^{\Delta_{\boldsymbol{\varepsilon}}} \\ & i\!:\!C \longmapsto \varepsilon_i(\mathcal{I}_i(C)) \end{aligned}$$

La fonction $\bar{\varphi}$ associe un concept C de $\mathscr{C}$ au complément de $\varphi(C)$ dans $\Delta_{\boldsymbol{\varepsilon}}$.

Construction des configurations.

On construit d'abord la configuration globale puis les configurations locales. Les configurations locales doivent satisfaire une contrainte supplémentaire vis-à-vis de la configuration globale. Par conséquent, une preuve de la validité de la construction suit immédiatement la configuration locale.

Configuration globale. Pour tout $W \subseteq \mathscr{C}$, $W \in \Omega$ si et seulement si $\varphi(C_W) \neq \emptyset$.

Configurations locales. Pour tout $w \in \mathscr{C}_i$, $w \in \Omega^i$ si et seulement si $\mathcal{I}_i(C_w^{\mathscr{C}_i}) \neq \emptyset$.

Lemme E.2 Ω^i *est une configuration locale vis-à-vis de* Ω*.*

Preuve : Supposons que $w \in \Omega^i$. Il s'ensuit que $\bigcap\limits_{X \in \widehat{w}} \varphi(X) \neq \emptyset$, par l'enchainement d'implications suivant :

$$\begin{aligned}
& w \in \Omega^i \\
\Longrightarrow & \mathcal{I}_i(C_w^{\mathscr{C}_i}) \neq \emptyset \\
\Longrightarrow & \bigcap_{X \in w} \mathcal{I}_i(X) \neq \emptyset \\
\Longrightarrow & \varepsilon_i(\bigcap_{X \in w} \mathcal{I}_i(X)) \neq \emptyset \\
\Longrightarrow & \bigcap_{X \in w} \varepsilon_i(\mathcal{I}_i(X)) \neq \emptyset \\
\Longrightarrow & \bigcap_{X \in \widehat{w}} \varphi(X) \neq \emptyset
\end{aligned}$$

Il existe un élément e dans $\bigcap\limits_{X \in \widehat{w}} \varphi(X)$. Nommons W l'ensemble $\widehat{W} \cup \{X \in \mathscr{C} \backslash \widehat{w} \mid e \in \varphi(X)\}$. Par définition, $\widehat{w} \subseteq W$ et $e \in \varphi(C_W)$. D'après la construction de Ω, cela assure que $W \in \Omega$ ce que établit la preuve. $\square$

Cohérence de $\widehat{\mathbf{A}}_\Omega$.

Afin de prouver la cohérence de $\widehat{\mathbf{A}}_\Omega$, on construit d'abord une interprétation de l'ontologie d'alignement. Deux propriétés importantes déterminent la construction de cette interprétation : 1) l'interprétation globale des concepts non vides C_W forment une partition du domaine d'interprétation global. Plus formellement, $\{\varphi(C_W) \mid W \subseteq \mathscr{C} \wedge \varphi(C_W) \neq \emptyset\}$ est une partition de $\Delta_{\boldsymbol{\varepsilon}}$; 2) à cause des particularités des axiomes de $\widehat{\mathbf{A}}_\Omega$, il est impossible de différentier deux éléments dans le même ensemble $\varphi(C_W)$. De ce fait, il suffit de ne disposer que d'autant d'éléments dans le domaine d'interprétation qu'il y a de concepts non vides C_W. Par conséquent, le domaine d'interprétation peut être n'importe quel ensemble de la taille d'Ω, comme par exemple Ω lui-même.

Construction d'une interprétation de l'ontologie d'alignement. On définit une interprétation $\langle \Delta_A, .^{\mathcal{I}_A} \rangle$ de $\widehat{\mathbf{A}}_\Omega$ de la façon suivante.

1. $\Delta_A = \Omega$;
2. Pour tout $X \in \mathscr{C}$ on définit $\mathcal{I}_A(X) = \{W \in \Omega \mid X \in W\}$;
3. Pour tout a_W ajouté par la définition 8.3.5 à $\widehat{\mathbf{A}}_\Omega$ par l'assertion $C_W \equiv \{a_W\}$, on définit $\mathcal{I}_A(a_W) = W$.

Dans la preuve qui suit, l'ensemble Ω a deux fonctions. En effet, Ω représente à la fois le domaine d'interprétation de l'ontologie d'alignement et la configuration globale de S (c'est-à-dire un ensemble d'ensembles de concepts).

Vérification de la satisfaction des axiomes. Les axiomes de $\widehat{\mathbf{A}}_\Omega$ sont tous introduits par les trois items de la définition 8.3.5. Cette preuve suit le même ordre que la définition pour démontrer que $\mathcal{I}_A$ les satisfaits.

1. Supposons qu'il existe une subsomption de concepts inter-ontologies $i:C \overset{\sqsubseteq}{\leftrightarrow} j:D \in A_{ij}$. Alors, $i:C \sqsubseteq j:D \in \widehat{\mathbf{A}}_\Omega$ est un axiome de $\widehat{\mathbf{A}}_\Omega$. Soit $W \in \mathcal{I}_A(i:C)$. Par construction de $\mathcal{I}_A$, $W \in \Omega$ et $i:C \in W$. Si l'on suppose que $i:D \notin W$, alors $\varphi(i:C) \cap \bar{\varphi}(i:D) \neq \emptyset$ (d'après la construction de Ω). Donc $\varepsilon_i(\mathcal{I}_i(C)) \cap \Delta_{\boldsymbol{\varepsilon}} \setminus \varepsilon_i(\mathcal{I}_i(D)) \neq \emptyset$. Ceci contredit le fait que $\langle \mathbf{I}, \boldsymbol{\varepsilon} \rangle \models i:C \overset{\sqsubseteq}{\leftrightarrow} j:D \in A_{ij}$. Ainsi, $i:D \in W$, donc $W \in \mathcal{I}_A(i:D)$ et plus généralement, $\mathcal{I}_A(i:C) \subseteq \mathcal{I}_A(i:D)$.

   De même, si $i:C \overset{\perp}{\leftrightarrow} j:D$ est une exclusion de concepts inter-ontologies dans A_{ij}, alors $i:C \sqsubseteq \neg j:D \in \widehat{\mathbf{A}}_\Omega$. Soit $W \in \mathcal{I}_A(i:C)$. Si l'on suppose que $i:D \in W$, alors $\varphi(i:C) \cap \varphi(i:D) \neq \emptyset$ (d'après la construction de Ω). Donc $\varepsilon_i(\mathcal{I}_i(C)) \cap \varepsilon_i(\mathcal{I}_i(D)) \neq \emptyset$. Ceci contredit le fait que $\langle \mathbf{I}, \boldsymbol{\varepsilon} \rangle \models i:C \overset{\perp}{\leftrightarrow} j:D \in A_{ij}$. Ainsi, $i:D \notin W$ et plus généralement, $\mathcal{I}_A(i:C) \subseteq \Omega \setminus \mathcal{I}_A(i:D)$.
2. Si $W \in \Omega$, alors $C_W \equiv \{a_W\} \in \widehat{\mathbf{A}}_\Omega$. Par construction de $\mathcal{I}_A$,

$$\begin{aligned}\mathcal{I}_A(C_W) &= \bigcap_{X \in W} \mathcal{I}_A(X) \cap \bigcap_{X \in \mathscr{C} \setminus W} \Omega \setminus \mathcal{I}_A(X) \\ &= \bigcap_{X \in W} \{\omega \in \Omega \mid X \in \omega\} \cap \bigcap_{X \in \mathscr{C} \setminus W} \Omega \setminus \{\omega \in \Omega \mid X \in \omega\} \qquad \textbf{(*)}\end{aligned}$$

   Assez manifestement, pour tout $X \in W$, $W \in \{\omega \in \Omega \mid X \in \omega\}$. En outre, pour tout $X \in \mathscr{C} \setminus W$, $W \notin \{\omega \in \Omega \mid X \in \omega\}$. C'est pourquoi $\mathcal{I}_A(a_W) = W \in \mathcal{I}_A(C_W)$. À présent, soit $W' \in \mathcal{I}_A(C_W)$. Selon **(*)**, il est évident que pour tout $X \in W$, $X \in W'$ et pour tout $X \in \mathscr{C} \setminus W$, $X \notin W'$. Par conséquent, $W = W'$. De cela et de $\mathcal{I}_A(a_W) = W \in \mathcal{I}_A(C_W)$, on en conclut que $\{\mathcal{I}_A(a_W)\} = \mathcal{I}_A(C_W)$.
3. Si $W \in \mathscr{C} \setminus \Omega$, alors $C_W \sqsubseteq \bot \in \widehat{\mathbf{A}}_\Omega$, ce qui signifie que $\varphi(C_W) = \emptyset$ par construction de Ω. De ce qui a été prouvé à l'item précédent, on déduit que si $W' \in \mathcal{I}_A(C_W)$ alors nécessairement, $W' = W$. Mais cela ne se peut car $\mathcal{I}_A \subseteq \Omega$ et $W \notin \Omega$. Donc $\mathcal{I}_A(C_W) = \emptyset$.

D'où $\mathcal{I}_A \models \widehat{\mathbf{A}}_\Omega$.

Cohérence de $\widehat{O}_{\Omega^i}$.

Afin de construire une interprétation valide de $\widehat{O}_{\Omega^i}$, on établit d'abord un résultat préliminaire.

Lemme E.3 *Si* $W \in \Omega$ *et* $X \in W|_i$*, alors il existe* $e \in \mathcal{I}_i(X) \cap \bigcap_{X' \in \mathscr{C}_i \setminus W|_i} \Delta_i \setminus \mathcal{I}_i(X')$ *tel que* $\varepsilon_i(e) \in \varphi(C_W)$.

Preuve : Supposons que $W \in \Omega$ et $X \in W|_i$. Alors, par construction de Ω, $\varphi(C_W) \neq \emptyset$. Donc il existe $x \in \varphi(C_W)$. Il s'ensuit que $x \in \varphi(i : X) \cap \bigcap_{X' \in \mathscr{C}_i \setminus W|_i} \bar{\varphi}(i : X')$. Donc il existe $e \in \mathcal{I}_i(X)$ tel que $\varepsilon_i(e) = x$. Par ailleurs, pour tout $X' \in \mathscr{C}_i \setminus W|_i$, $\varepsilon_i(e) \in \bar{\varphi}(i : X')$, donc $e \notin \mathcal{I}_i(X')$. Par conséquent, $e \in \mathcal{I}_i(X) \cap \bigcap_{X' \in \mathscr{C}_i \setminus W|_i} \Delta_i \setminus \mathcal{I}_i(X')$ et $\varepsilon_i(e) \in \varphi(C_W)$, ce qu'établit la preuve. □

Construction d'une interprétation de l'ontologie étendue. On définit une interprétation $\langle \Delta'_i, \cdot^{\mathcal{I}'_i} \rangle$ pour chaque $\widehat{O}_{\Omega^i}$ de la façon suivante.

1. $\Delta'_i := \Delta_i$;
2. $\mathcal{I}'_i(C) := \mathcal{I}_i(C)$ pour tout nom de concept C dans O_i et ainsi, dans $\widehat{O}_{\Omega^i}$ aussi ;
3. $\mathcal{I}'_i(R) := \mathcal{I}_i(R)$ pour tout nom de rôle R dans O_i ;
4. $\mathcal{I}'_i(o) := \mathcal{I}_i(o)$ pour tout nom d'individu o dans O_i ;
5. pour tout individu b_w ajouté par la définition 8.3.7 à $\widehat{O}_{\Omega^i}$ par l'assertion $C_w^{\mathscr{C}_i}(b_w)$, on choisit $\mathcal{I}_i(b_w)$ dans $\mathcal{I}_i(C_w^{\mathscr{C}_i})$ (qui existe, par construction de Ω^i) ;
6. pour tout concept C_W^X ajouté par la définition 8.3.7 à $\widehat{O}_{\Omega^i}$, on définit $\mathcal{I}_i(C_W^X) = \{e \in \mathcal{I}_i(X) \cap \bigcap_{X' \in \mathscr{C}_i \setminus W|_i} \Delta_i \setminus \mathcal{I}_i(X) \mid \varepsilon_i(e) \in \varphi(C_W)\}$.
7. pour tout individu b_W^X ajouté par la définition 8.3.7 à $\widehat{O}_{\Omega^i}$ par l'assertion $C_W^X(b_W^X)$, on choisit $\mathcal{I}_i(b_W^X)$ dans $\mathcal{I}'_i(C_W^X)$ (qui existe d'après le lemme E.3).

Vérification de la satisfaction des axiomes. Les axiomes existants dans l'ontologie O_i sont évidemment satisfaits par $\mathcal{I}'_i$. Ici, on démontre que les nouveaux axiomes ajoutés par le biais de Ω^i sont satisfaits, puis l'on montre que les axiomes ajoutés par le biais de la configuration globale sont aussi satisfaits.

1. *Satisfaction de* $C_w^{\mathscr{C}_i}(b_w)$. Soit $w \in \Omega^i$. Par construction de $\mathcal{I}'_i$, $\mathcal{I}'_i(b_w) \in \mathcal{I}_i(C_w^{\mathscr{C}})$. Donc $\mathcal{I}'_i \models C_w^{\mathscr{C}_i}(b_w)$.
2. *Satisfaction de* $C_w^{\mathscr{C}_i} \sqsubseteq \bot$. Soit $w \in \mathscr{C}_i \setminus \Omega^i$. Par construction de Ω^i, $\mathcal{I}_i(C_w^{\mathscr{C}_i}) = \emptyset$. Donc $\mathcal{I}'_i \models C_w^{\mathscr{C}_i} \sqsubseteq \bot$.
3. *Satisfaction de* $C_W^X \sqsubseteq X \sqcap \bigsqcap_{X' \in \mathscr{C}_i \setminus W|_i} X'$. Ceci est trivial du fait de la construction de $\mathcal{I}'_i$.
4. *Satisfaction de* $C_W^X(b_W^X)$. Ceci est évident grâce au lemme E.3 et à la construction de $\mathcal{I}'_i$.
5. *Satisfaction de* $C_W^X \sqsubseteq C_W^\top$. La définition de $\mathcal{I}'_i$ assure que $\mathcal{I}'_i(C_W^X) \sqsubseteq \mathcal{I}'_i(C_W^\top)$.
6. *Satisfaction de* $C_W^\top \sqsubseteq \neg C_{W'}^\top$. Pour tout $e \in \mathcal{I}'_i(C_W^\top)$, $\varepsilon_i(e) \in \varphi(C_W)$ et pour tout $f \in \mathcal{I}'_i(C_{W'}^\top)$, $\varepsilon_i(f) \in \varphi(C_{W'})$. Or, $\varphi(C_W)$ et $\varphi(C_{W'})$ sont des ensembles disjoints, donc $\varepsilon_i(e) \neq \varepsilon_i(f)$. D'où $e \neq f$ et, plus généralement, $\mathcal{I}'_i(C_W^\top) \cap \mathcal{I}'_i(C_{W'}^\top)$.

Les interprétations $\mathcal{I}'_i$ sont donc des modèles des ontologies respectives $\widehat{O}_{\Omega^i}$. □

F Preuve du théorème 8.5.1

Théorème F.1 (Cohérence d'un réseau d'ontologies alignées) *Soit $S = \langle \mathbf{O}, \mathbf{A} \rangle$ un réseau d'ontologies alignées. S est* cohérent *si et seulement si il existe une configuration globale Ω pour S, une configuration globale de rôles Φ_Ω vis-à-vis de Ω, des configurations locales Ω^i pour tout $O_i \in \mathbf{O}$ vis-à-vis de Ω et des configurations locales de rôles Φ_{Ω^i} vis-à-vis de Ω, Ω^i et Φ_Ω, telles que l'ontologie d'alignement $\widehat{\mathbf{A}}_\Omega$ et les ontologies locales étendues $\{\widehat{O}_{\Omega^i}\}$ comme définies aux définitions 8.5.4 et 8.5.6 sont cohérentes.*

Preuve : La preuve de cette nouvelle version du théorème a de nombreux points communs avec la précédente version, donc je ne présenterai dans cette section que les modifications ou ajouts nécessaires pour traiter les correspondances de rôles. En outre, le plan de cette preuve est presque identique au précédent.

F.1 Si $\Leftarrow$

Dans cette section, le seul changement se trouve dans la nécessité de satisfaire les correspondances de rôles. Les autres définitions et constructions sont identiques, en particulier la fonction d'égalisation est définie comme précédemment. Cependant, afin de démontrer la satisfaction des subsomptions de rôles inter-ontologies, il convient de prouver un lemme supplémentaire, qui peut être considéré comme le pendant du lemme E.1 vis-à-vis des rôles.

Lemme auxiliaire.

Informellement, ce lemme peut être formulé comme suit : la fonction ε_i construite précédemment préserve la structure des rôles.

Lemme F.1 *Pour tout $R \in \mathscr{R}_i$, $\varepsilon_i(\mathcal{I}_i(R)) = \{\langle \mathcal{I}_A(a_W), \mathcal{I}_A(a_{W'}) \rangle \mid \langle W, W', i\!:\!R \rangle \in \Phi_\Omega\}$.*

Preuve : La preuve de ce lemme est décomposée en deux étapes, correspondant à chacune des inclusions des deux ensembles mis en jeu.

($\subseteq$) : Soit $(x, x') \in \varepsilon_i(\mathcal{I}_i(R))$. Il existe $(e, e') \in \mathcal{I}_i(R)$ tel que $x = \varepsilon_i(e)$ et $x' = \varepsilon_i(e')$. Par suite, $e \in \mathcal{I}_i(\exists R.\top)$. Donc, d'après l'item 9 de la définition 8.5.6 et, puisque $\mathcal{I}_i$ est un modèle de $\widehat{O}_{\Omega^i}$, il existe $W, W' \in \Omega$ tel que $e \in \mathcal{I}_i(C^R_{W,W'})$ et semblablement, $e' \in \mathcal{I}_i(C^{R^-}_{W',W})$. Ceci implique que $\widehat{O}_{\Omega^i} \not\models C^R_{W,W'} \sqsubseteq \forall R \neg C^{R^-}_{W',W}$, ce qui signifie que $\langle W, W', R \rangle \in \Phi_\Omega$. Aussi, de par l'item 8a de la définition 8.5.6, $e \in \mathcal{I}_i(C^\top_W)$ et $e' \in \mathcal{I}_i(C^\top_{W'})$, donc $\varepsilon_i(e) = \mathcal{I}_A(a_W)$ et $\varepsilon_i(e') = \mathcal{I}_A(a_{W'})$. Ce qui établit la première inclusion.

($\supseteq$) : Soit $\langle W, W', i\!:\!R \rangle \in \Phi_\Omega$. Selon l'item 8b de la définition 8.5.6, on a $C^R_{W,W'}(\beta^R_{W,W'})$ et $C^R_{W,W'} \sqsubseteq \exists R.C^{R^-}_{W',W}$, en vertu de quoi il existe $x \in \mathcal{I}_i(C^{R^-}_{W',W})$ tel que $\langle \mathcal{I}_i(\beta^R_{W,W'}), x \rangle \in \mathcal{I}_i(R)$. Par ailleurs, de par l'item 8a de la définition 8.5.6, on a $\mathcal{I}_i(\beta^R_{W,W'}) \in \mathcal{I}_i(C^\top_W)$ et $x \in \mathcal{I}_i(C^\top_{W'})$, donc, par construction de ε_i, $\varepsilon_i(\mathcal{I}_i(\beta^R_{W,W'})) = \mathcal{I}_A(a_W)$ et $\varepsilon_i(x) = \mathcal{I}_A(a_{W'})$. C'est pourquoi, $\langle \mathcal{I}_A(a_W), \mathcal{I}_A(a_{W'}) \rangle \in \varepsilon_i(\mathcal{I}_i(R))$. Ce qui établit l'autre inclusion. $\square$

Satisfaction du réseau d'ontologies.

La preuve de la satisfaction des correspondances de concepts reste inchangée avec les nouvelles définitions. Par conséquent, on se contentera de vérifier la satisfaction de la subsomption de rôles inter-ontologies.

Subsomption de rôles inter-ontologies $(i\!:\!R \overset{\sqsubseteq}{\Leftrightarrow} j\!:\!S \in A_{ij} \Longrightarrow \varepsilon_i(\mathcal{I}_i(R)) \subseteq \varepsilon_j(\mathcal{I}_j(S)))$**.** Supposons que $i : R \overset{\sqsubseteq}{\Leftrightarrow} j : S \in A_{ij}$. Soit $\langle x, x' \rangle \in \varepsilon_i(\mathcal{I}_i(R))$. D'après le lemme F.1, il apparait qu'il existe $W, W' \in \Omega$ tels que $x = \mathcal{I}_A(a_W)$, $x' = \mathcal{I}_A(a_{W'})$ et $\langle W, W', i : R \rangle \in \Phi_\Omega$. Par définition de $\widehat{\mathbf{A}}_\Omega$, on en déduit que $\langle \mathcal{I}_A(a_W), \mathcal{I}_A(a_{W'}) \rangle \in \mathcal{I}_A(i : R)$. Aussi, du fait de la présence de la subsomption de rôles $i : R \overset{\sqsubseteq}{\Leftrightarrow} j : S$, l'axiome $i : R \sqsubseteq j : S$ est satisfait par $\mathcal{I}_A$. Ainsi, $\langle \mathcal{I}_A(a_W), \mathcal{I}_A(a_{W'}) \rangle \in \mathcal{I}_A(j : S)$ et donc $\langle W, W', j : S \rangle \in \Phi_\Omega$. D'où, grâce au lemme F.1, on obtient que $\langle \mathcal{I}_A(a_W), \mathcal{I}_A(a_{W'}) \rangle \in \varepsilon_j(\mathcal{I}_j(S))$. De ce fait $\langle x, x' \rangle \in \varepsilon_j(\mathcal{I}_j(S))$ et plus généralement, $\varepsilon_i(\mathcal{I}_i(R)) \subseteq \varepsilon_j(\mathcal{I}_j(S))$.

F.2 Seulement si ⇒

La nouvelle version de cette partie de la preuve est organisée comme suit :

1. Construire les configurations seln les modèles $\langle \mathcal{I}, \varepsilon \rangle$:
   (a) construire la configuration globale Ω ; **(fait)**
   (b) construire les configurations locales Ω^i vis-à-vis de Ω ; **(fait)**
   (c) prouver que les Ω^i sont bien des configurations locales ; **(fait)**
   (d) construire la configuration globale de rôles Φ_Ω vis-à-vis de Ω ;
   (e) construire les configurations locales de rôles Φ_{Ω^i} vis-à-vis de Ω^i et de Φ_Ω ;
   (f) prouver que les Φ_{Ω^i} sont bien des configurations locales de rôles.
2. Montrer que $\widehat{\mathbf{A}}_\Omega$ est cohérente :
   (a) construire une interprétation $\mathcal{I}_A$ de $\widehat{\mathbf{A}}_\Omega$;
   (b) montrer que $\mathcal{I}_A$ est un modèle de $\widehat{\mathbf{A}}_\Omega$.
3. Montrer que $\{\widehat{O}_{\Omega^i}\}$ est cohérente pour tout $i \in \mathbf{O}$:
   (a) construire une interprétation $\mathcal{I}_i$ de $\widehat{O}_{\Omega^i}$;
   (b) montrer que $\mathcal{I}_i$ est un modèle de $\widehat{O}_{\Omega^i}$.

Avant de présenter cette partie de la preuve, il convient d'étendre la fonction φ afin de traiter aussi l'interprétation des rôles.

$$\begin{aligned} \varphi :\ \mathscr{C} \cup \mathscr{R} &\longrightarrow 2^{\Delta_\varepsilon} \cup 2^{\Delta_\varepsilon \times \Delta_\varepsilon} & & \\ i\!:\!C &\longmapsto \varepsilon_i(\mathcal{I}_i(C)) & & \text{si } C \in \mathscr{C}_i \\ i\!:\!R &\longmapsto \varepsilon_i(\mathcal{I}_i(R)) & & \text{si } R \in \mathscr{R}_i \end{aligned}$$

Cela aidera à simplifier les notations.

Construction des configurations.

Les configurations locales et globales (de concept) peuvent être définies exactement comme dans la section E.2. La construction des configurations de rôles utilise les mêmes notations que précédemment.

Configuration globale de rôles. Pour tous $W, W' \in \Omega$ et tout $R \in \mathscr{R}$, $\langle W, W', R\rangle \in \Omega$ si et seulement si

$$\varphi(R) \cap (\varphi(C_W) \times \varphi(C_{W'})) \neq \emptyset$$

Configurations locales de rôles. Pour tous $w, w' \in \Omega_i$ et tout $R \in \mathscr{R}_i$, $\langle w, w', R\rangle \in \Phi_{\Omega^i}$ si et seulement si

$$\mathcal{I}_i(R) \cap \left(\mathcal{I}_i(C_w^{\mathscr{C}_i}) \times \mathcal{I}_i(C_{w'}^{\mathscr{C}_i})\right) \neq \emptyset$$

les configurations locales de rôles doivent satisfaire une contrainte spécifique, donc il faut prouver que la construction ci-dessus définit bien une configuration locale de rôles.

Lemme F.2 Φ_{Ω^i} *est une configuration locale de rôles vis-à-vis de* Ω^i *et de* Φ_Ω.

Preuve : Supposons que $\langle w, w', R\rangle \in \Phi_{\Omega^i}$. Cela implique que

$$\varphi(i{:}R) \cap \left(\bigcap_{X \in \widehat{w}} \varphi(X) \times \bigcap_{X' \in \widehat{w'}} \varphi(X')\right) \neq \emptyset$$

en suivant la séquence d'implication suivante :

$$\begin{aligned}
&\langle w, w', R\rangle \in \Phi_{\Omega^i}\\
\Longrightarrow &\mathcal{I}_i(R) \cap (\varphi(C_W) \times \varphi(C_{W'})) \neq \emptyset\\
\Longrightarrow &\mathcal{I}_i(R) \cap \left(\bigcap_{X \in w} \mathcal{I}_i(X) \times \bigcap_{X' \in w'} \mathcal{I}_i(X')\right) \neq \emptyset\\
\Longrightarrow &\varepsilon_i(\mathcal{I}_i(R)) \cap \left(\bigcap_{X \in w} \varepsilon_i(\mathcal{I}_i(X)) \times \bigcap_{X' \in w'} \varepsilon_i(\mathcal{I}_i(X'))\right) \neq \emptyset\\
\Longrightarrow &\varphi(i{:}R) \cap \left(\bigcap_{X \in \widehat{w}} \varphi(X) \times \bigcap_{X' \in \widehat{w'}} \varphi(X')\right) \neq \emptyset
\end{aligned}$$

Il existe donc une paire $\langle e, e'\rangle$ in $\varphi(i : R) \cap \left(\bigcap_{X \in \widehat{w}} \varphi(X) \times \bigcap_{X' \in \widehat{w'}} \varphi(X')\right)$. Nommons W l'ensemble $\widehat{w} \cup \{X \in \mathscr{C} \setminus \widehat{w} \mid e \in \varphi(X)\}$ et W' l'ensemble $\widehat{w'} \cup \{X' \in \mathscr{C} \setminus \widehat{w'} \mid e' \in \varphi(X')\}$. Par définition, $\widehat{w} \subseteq W$, $\widehat{w'} \subseteq W'$, $e \in \varphi(C_W)$ et $e' \in \varphi(C_{W'})$. Ceci prouve que $\varphi(R) \cap (\varphi(C_W) \times \varphi(C_{W'})) \neq \emptyset$. D'après la construction de Φ_Ω, cela assure que $\langle W, W', i : R\rangle \in \Omega$ ce qui établit la preuve. $\square$

Cohérence de $\widehat{\mathrm{A}}_\Omega$.

Construction d'une interprétation de l'ontologie d'alignement. On met à jour la construction précédente en ajoutant l'interprétation des rôles.

4. Pour tout $R \in \mathscr{R}$ on définit $\mathcal{I}_A(R) = \{\langle W, W'\rangle \in \Omega \times \Omega \mid \langle W, W', R\rangle \in \Phi_\Omega\}$;

Vérification de la satisfaction des axiomes. De même que dans la section F.1, il n'est pas utile de revérifier la satisfaction des axiomes déjà présents dans la première version. Donc, voici les seules preuves nécessaires.

4. Supposons qu'il existe une correspondance de rôles $i:R \overset{\sqsubseteq}{\leftrightarrow} j:S \in A_{ij}$. Ainsi, $i:R \sqsubseteq j:S \in \widehat{\mathbf{A}}_\Omega$ est un axiome de $\widehat{\mathbf{A}}_\Omega$. Soit $\langle W, W'\rangle \in \mathcal{I}_A(i:R)$. Par construction de $\mathcal{I}_A$, $\langle W, W', i:R\rangle \in \Phi_\Omega$. Donc par construction de Φ_Ω, $\varphi(i:S) \cap (\varphi(C_W) \times \varphi(C_{W'})) \neq \emptyset$. Mais comme $\langle \mathbf{I}, \boldsymbol{\varepsilon}\rangle \models i:R \overset{\sqsubseteq}{\leftrightarrow} j:S$, $\varphi(i:R) \subseteq \varphi(j:S)$. Donc, $\varphi(j:R) \cap (\varphi(C_W) \times \varphi(C_{W'})) \neq \emptyset$. D'après quoi, $\langle W, W', j:S\rangle \in \Phi_\Omega$, d'où par définition de $\mathcal{I}_A$, $\langle W, W'\rangle \in \mathcal{I}_A(j:S)$.
5. Si $\langle W, W', R\rangle \in \Phi_\Omega$ alors $\{a_W\} \sqsubseteq \exists R.\{a_{W'}\} \in \widehat{\mathbf{A}}_\Omega$. Il a été prouvé ci-dessus que $\langle W, W', R\rangle \in \Phi_\Omega$ implique nécessairement que $\langle W, W'\rangle \in \mathcal{I}_A(R)$ et puisque $\mathcal{I}_A(a_W) = W$ et $\mathcal{I}_A(a_{W'}) = a_{W'}$, il vient que $\mathcal{I}_A(a_W) \in \mathcal{I}_A(\exists R.\{a_{W'}\})$.
6. Si $\langle W, W', R\rangle \notin \Phi_\Omega$ alors $\{a_W\} \sqsubseteq \forall R.\neg\{a_{W'}\} \in \widehat{\mathbf{A}}_\Omega$. Quand ni W ni W' ne sont dans Ω, l'axiome est évidemment satisfait par $\mathcal{I}_A$. Sinon, $\mathcal{I}_A(a_W) = W$ et $\mathcal{I}_A(a_{w'})) = W'$, et $\langle W, W'\rangle \notin \mathcal{I}_A(R)$ par construction de $\mathcal{I}_A$. Ce qui implique clairement que $\mathcal{I}_A \models \{a_W\} \sqsubseteq \forall R.\neg\{a_{W'}\}$.

Cohérence de $\widehat{O}_{\Omega^i}$.

La preuve de la cohérence de $\widehat{O}_{\Omega^i}$ fait appel à deux résultats supplémentaires. Ils assurent la non-vacuité des ensembles qui serviront à déterminer l'interprétation de $\beta^R_{W,W'}$ et de $b^R_{w,w'}$, respectivement.

Le lemme suivant sera utilisé pour définir une interprétation de l'individu $\beta^R_{W,W'}$.

Lemme F.3 *Si $\langle W, W', i:R\rangle \in \Phi_\Omega$ alors il existe*

$$\langle e, e'\rangle \in \left(\bigcap_{X \in \mathscr{C}_i \setminus W|_i} \Delta_i \setminus \mathcal{I}_i(X)\right) \times \left(\bigcap_{X' \in \mathscr{C}_i \setminus W'|_i} \Delta_i \setminus \mathcal{I}_i(X')\right)$$

tel que $\langle e, e'\rangle \in \mathcal{I}_i(R)$.

Avant de procéder à la preuve, il faut donner un sens à l'ensemble $\bigcap_{X \in \emptyset} \Delta_i \setminus \mathcal{I}_i(X)$. On admettra la convention stipulant que cet ensemble est égal à Δ_i et on supposera aussi que l'ensemble $\bigcap_{X \in \mathscr{C}_i \setminus W|_i} \Delta_{\boldsymbol{\varepsilon}} \setminus \varphi(i:X)$ est égal à $\Delta_{\boldsymbol{\varepsilon}}$.

Preuve : Supposons que $\langle W, W', i:R\rangle \in \Phi_\Omega$. On peut élaborer les implications suivantes :

$$\begin{aligned}
&\langle W, W', i:R\rangle \in \Phi_\Omega \\
\Longrightarrow &\varphi(i:R) \cap (\varphi(C_W) \times \varphi(C_{W'})) \neq \emptyset \\
\Longrightarrow &\varphi(i:R) \cap \left(\bigcap_{X \in \mathscr{C} \setminus W} \bar{\varphi}(X) \times \bigcap_{X' \in \mathscr{C} \setminus W'} \bar{\varphi}(X')\right) \neq \emptyset \\
\Longrightarrow &\varepsilon_i(\mathcal{I}_i(R)) \cap \left(\bigcap_{X \in \mathscr{C}_i \setminus W|_i} \Delta_{\boldsymbol{\varepsilon}} \setminus \varphi(i:X) \times \bigcap_{X' \in \mathscr{C}_i \setminus W'|_i} \Delta_{\boldsymbol{\varepsilon}} \setminus \varphi(i:X')\right) \neq \emptyset \\
\Longrightarrow &\mathcal{I}_i(R) \cap \left(\bigcap_{X \in \mathscr{C}_i \setminus W|_i} \Delta_i \setminus \mathcal{I}_i(X) \times \bigcap_{X' \in \mathscr{C}_i \setminus W'|_i} \Delta_i \setminus \mathcal{I}_i(X')\right) \neq \emptyset
\end{aligned}$$

□

Ce dernier lemme sera utilisé pour définir une interprétation des individus $b^R_{w,w'}$.

Lemme F.4 *Si* $\langle w, w', R\rangle \in \Phi_{\Omega^i}$ *alors* $\mathcal{I}_i(C_w^{\mathscr{C}_i}) \cap \mathcal{I}_i(\exists R.C_{w'}^{\mathscr{C}_i}) \neq \emptyset$.

Preuve : Par contruction de Φ_{Ω^i},

$$\begin{aligned}
&\langle w, w', R\rangle \in \Phi_{\Omega^i}\\
\Longrightarrow &\mathcal{I}_i(R) \cap \Big(\mathcal{I}_i(C_w^{\mathscr{C}_i}) \times \mathcal{I}_i(C_{w'}^{\mathscr{C}_i})\Big) \neq \emptyset\\
\Longrightarrow &\mathcal{I}_i(C_w^{\mathscr{C}_i}) \cap \mathcal{I}_i(\exists R.C_{w'}^{\mathscr{C}_i}) \neq \emptyset
\end{aligned}$$

□

Construction d'une interprétation des ontologies étendues. Une interprétation de $\widehat{O}_{\Omega^i}$ est construire en étendant la construction entamée en section E.2.

8. pour tout $b^R_{w,w'}$ ajouté par la définition 8.5.6 à $\widehat{O}_{\Omega^i}$ avec l'assertion $(C_w^{\mathscr{C}_i} \sqcap \exists R.C_{w'}^{\mathscr{C}_i})(b^R_{w,w'})$, on choisit $\mathcal{I}'_i(b^R_{w,w'}) \in \mathcal{I}_i(C_w^{\mathscr{C}_i}) \cap \mathcal{I}_i(\exists R.C_{w'}^{\mathscr{C}_i})$ (qui existe, conformément au lemme F.4) ;
9. pour tout $C^R_{W,W'}$ ajouté par la définition 8.5.6 à $\widehat{O}_{\Omega^i}$, on définit $\mathcal{I}'_i(C^R_{W,W'})$ comme l'ensemble

$$\{e \in \bigcap_{X \in \mathscr{C}_i \setminus W|_i} \Delta_i \setminus \mathcal{I}_i(X) \mid \exists e' \in \bigcap_{X' \in \mathscr{C}_i \setminus W'|_i} \Delta_i \setminus \mathcal{I}_i(X') \text{ s.t. } \langle e, e'\rangle \in \mathcal{I}_i(R)\}$$

10. pour tout $\beta^R_{W,W'}$ ajouté par la définition 8.5.6 à $\widehat{O}_{\Omega^i}$ avec l'assertion $C^R_{W,W'}(\beta^R_{W,W'})$, on choisit $\mathcal{I}'_i(\beta^R_{W,W'})$ dans $\mathcal{I}'_i(C^R_{W,W'})$ (qui existe selon le lemme F.3).

Vérification de la satisfaction des axiomes. Il suffit de présenter les preuves de la satisfaction des nouveaux axiomes.

7. *Satisfaction de* $(C_w^{\mathscr{C}_i} \sqcap \exists R.C_{w'}^{\mathscr{C}_i})(b_w)$. Soit $\langle w, w', R\rangle \in \Phi_{\Omega^i}$. Par le lemme F.4 et la construction de $\mathcal{I}'_i$, on a $\mathcal{I}'_i(b_w) \in \mathcal{I}'_i(C_w^{\mathscr{C}_i} \sqcap \exists R.C_{w'}^{\mathscr{C}_i})$.
8. *Satisfaction de* $C_w^{\mathscr{C}_i} \sqsubseteq \forall R.\neg C_{w'}^{\mathscr{C}_i}$. Soit $\langle w, w', R\rangle \notin \Phi_{\Omega^i}$. Ceci est une conséquence directe de la construction de Φ_{Ω^i}.
9. Soit $W, W' \in \Omega$ et $R \in \mathscr{R}_i$.
   (a) *Satisfaction de* $C^R_{W,W'} \sqsubseteq C^\top_W$. Par construction de $\mathcal{I}'_i$,

   $$\mathcal{I}'_i(C^R_{W,W'}) \subseteq \bigcap_{X \in \mathscr{C}_i \setminus w} (\Delta_i \setminus \mathcal{I}_i(X)) = \mathcal{I}'_i(C^\top_W)$$

   (b) *Satisfaction de* $C^R_{W,W'} \sqsubseteq \exists R.C^{R^-}_{W',W}$ *et de* $C^R_{W,W'}(\beta^R_{W,W'})$. Supposons que $\langle W, W', i : R\rangle \in \Phi_\Omega$. Soit $e \in \mathcal{I}'_i(C^R_{W,W'})$. La construction de $\mathcal{I}'_i$ implique qu'il existe $e' \in \bigcap_{X' \in \mathscr{C}_i \setminus W'|_i} \Delta_i \setminus \mathcal{I}_i(X')$ tel que $\langle e, e'\rangle \in \mathcal{I}_i(R)$. Mais alors, e' est tel qu'il existe $e \in \bigcap_{X \in \mathscr{C}_i \setminus W|_i} \Delta_i \setminus \mathcal{I}_i(X)$ avec $\langle e, e'\rangle \in \mathcal{I}'_i(R^-)$. Donc $e' \in \mathcal{I}'_i(C^{R^-}_{W',W})$. Ainsi, $e \in \mathcal{I}'(\exists R.C^{R^-}_{W',W})$. Plus généralement, $\mathcal{I}'_i(C^R_{W,W'}) \subseteq \mathcal{I}'_i(\exists R.C^{R^-}_{W',W})$. La satisfaction de $C^R_{W,W'}(\beta^R_{W,W'})$ est une conséquence directe du lemme F.3 et de la construction de $\mathcal{I}'_i$.

   (c) *Satisfaction de* $C^R_{W,W'} \sqsubseteq \forall R.\neg C^{R^-}_{W,W'}$. Supposons que $\langle W, W', i : R\rangle \notin \Phi_\Omega$. Donc $\varphi(R) \cap (\varphi(C_W) \times \varphi(C_{W'})) = \emptyset$. Considérons $\langle e, e'\rangle \in \mathcal{I}'_i(R)$ tel que $e \in \mathcal{I}'_i(C^R_{W,W'})$ et $e' \in \mathcal{I}'_i(C^{R^-}_{W',W})$. Puisque $\mathcal{I}'_i \models C^R_{W,W'} \sqsubseteq C^\top_W$ (démontré ci-dessus), $e \in \mathcal{I}'_i(C^{\top}_{W})$ et $\mathcal{I}'_i \models C^{R^-}_{W',W} \sqsubseteq C^{\top}_{W'}$ implique que $e' \in \mathcal{I}'_i(C^{\top}_{W'})$. Donc, par construction de $\mathcal{I}'_i$, $\langle \varepsilon_i(e), \varepsilon_i(e') \rangle \in \varphi(i : R) \cap (\varphi(C_W) \times \varphi(C_{W'}))$. Cela contredit la supposition initiale. Par conséquent, il n'y a pas de tels e et e'.

10. *Satisfaction de* $\exists R.\top \sqsubseteq \bigsqcup_{W,W' \in \Omega} C^R_{W,W'}$. Soit $e \in \mathcal{I}'_i(\exists R.\top)$, ce qui signifie qu'il existe $e' \in \Delta_i$ tel que $\langle e, e' \rangle \in \mathcal{I}'_i(R)$. En utilisant la fonction Θ définie dans la section précédente ($\Theta : \Delta_i \to 2^{\mathscr{C}}$ tel que pour tout $e \in \Delta_i$, $\Theta(e) = \{X \in \mathscr{C}_i \mid e \in \mathcal{I}_i(X)\}$), on a $e \in \mathcal{I}_i(C^{\mathscr{C}_i}_{\Theta(e)})$ et $e' \in \mathcal{I}_i(C^{\mathscr{C}_i}_{\Theta(e')})$. Donc $\langle e, e' \rangle \in \mathcal{I}_i(R) \cap \Big(\mathcal{I}_i(C^{\mathscr{C}_i}_{w}) \times \mathcal{I}_i(C^{\mathscr{C}_i}_{w'})\Big)$. Ceci implique que $\langle \Theta(e), \Theta(e'), R \rangle \Phi_{\Omega^i}$. Par définition de Φ_{Ω^i}, il existe $W, W' \in \Omega$ tel que $\Theta(e) \subseteq W|_i$ et $\Theta(e') \subseteq W'|_i$ et $\langle W, W', i : R \rangle \in \Phi_\Omega$. Par construction de $\mathcal{I}'_i$ et de ce qui précède, on en déduit que $e \in \mathcal{I}'_i(C^R_{W,W'})$. Donc finalement, $\mathcal{I}' \models \exists R.\top \sqsubseteq \bigsqcup_{W,W' \in \Omega} C^R_{W,W'}$.

$\square$

Résumé : L'objectif de cette thèse est de modéliser la sémantique d'un ensemble des connaissances produites indépendamment les unes des autres, mais mises en correspondances. On suppose que ces connaissances forment un réseau où à chaque nœud se trouve une ontologie, reliée aux autres par des correspondances formant des alignements d'ontologies. De telles structures peuvent se trouver sur le Web ou les réseaux pair-à-pair sémantiques. Afin de favoriser l'utilisation d'ontologies indépendantes et pré-existantes, je définis une sémantique formelle exploitant le principe de médiation. Celle-ci définit l'interprétation d'un réseau d'ontologies alignées en affectant à chaque ontologie une interprétation locale selon sa propre logique, et en y ajoutant un domaine d'interprétation supplémentaire (dit global) dans lequel viennent se projeter les domaines locaux par le biais d'une fonction dite d'égalisation. Ce domaine sert de médiateur et c'est dans celui-ci que les alignements viennent poser leurs contraintes pour définir la satisfaction d'un réseau d'ontologies. Quatre applications sont présentées : la sémantique des ontologies modulaires, la sémantique d'un langage d'alignement expressif, la composition d'alignements d'ontologies et le raisonnement distribué. Cette sémantique s'applique aux ontologies modulaires en traitant un module comme médiateur vis-à-vis des modules qu'il importe. Ainsi la sémantique des modules importés (locaux) se distingue de celle du module importateur (global). La distinction entre niveau local et global permet de donner une sémantique à un langage d'alignement expressif, indépendamment de l'expressivité du langage d'alignement. Un opérateur de composition d'alignements fondé sur les algèbres de relation est défini de façon cohérente avec la sémantique du réseau d'ontologie. Enfin, en se restreignant au cas des logiques de description alignées par des correspondances simples, on défini une procédure de vérification de la cohérence d'un réseau d'ontologies alignées. Cette procédure est correcte et complète, et exploite des systèmes de raisonnement locaux existant sans en connaître l'implémentation explicite.

Mots-clés : Web sémantique, représentation de connaissances, alignement d'ontologies, médiation

Abstract : This thesis aims at modelling the semantics of a set of knowledge bases produced independently from each others, but put in correspondence. We assume that this knowledge forms a network where each node is represented as an ontology, linked to the others by correspondences in the form of an ontology alignment. Such structures exist on the Web or semantic peer-to-peer networks. In order to promote the use of independent, pre-existing ontologies, I define a formal semantics which exploit the principle of mediation. It defines the interpretation of a network of aligned ontologies by assigning a local interpretration to each ontology, according to its own logic, and by adding a supplementary domain of interpretation (the global domain) in which the local domains are mapped via a so-called equalising function. This domain can be seen as a mediator in which alignments pose the constraints which serve to define the satisfaction of a network of ontologies. This semantics is applied in four cases : the semantics of modular ontologies ; the semantics of an expressive alignment language ; the composition of ontology alignments ; and distributed reasoning. Such semantics applies to modular ontologies by dealing with a module as if it were a mediator with respect to the modules it imports. Thus, the semantics of imported modules (local) stay distinct from the semantics of importing modules (global). With this distinction between local and global level, the semantics of alignments can differ from the semantics of ontologies, in particular in term of expressivity. An ontology alignment composition operator based on relation algebras is defined as a sound reasoning operation with respect ot the semantics of the network of ontologies. Finally, when restricted to description logics and simple correspondences, we can define a consistency checking procedure for a network of aligned ontologies. It is sound and complete, and exploit existing local reasoner while not imposing any specific implementation.

Keywords : Semantic Web, knowledge representation, ontology alignment, mediation