UNIVERSITÉ D'ARTOIS

ÉCOLE DOCTORALE SCIENCES TECHNOLOGIES ET SANTÉ N°585

On Merging of Open-Domain Ontologies

DOCTORAL THESIS

presented and publicly defended on December 6th, 2022

in partial fulfillment of requirements for the degree of

Doctor of Philosophy of Artois University

in the subject of Computer Science

by

Truong Thanh MA

Doctoral Committee

President:	Salem Benferhat	Professor, CNRS & Artois University
Reviewers:	Eduardo Fermé	Professor, University of Madeira
	Odile Papini	Professor, Aix-Marseille University
Examiners:	Fatiha Sais	Professor, Paris-Saclay University
	Nicolas Schwind	Research Scientist, AIST, Japan
Supervisors:	Zied Bouraoui	Assoc. Professor, CNRS & Artois University
	Sébastien Konieczny	Directeur de Recherche, CNRS
	Ivan José Varzinczak	Assoc. Professor, LIASD, Université Paris 8

CENTRE DE RECHERCHE EN INFORMATIQUE DE LENS – CNRS UMR 8188
Université d'Artois, rue Jean Souvraz, S.P. 18 F-62307, Lens Cedex France
Secrétariat : +33 (0)3 21 79 17 23
http://www.cril.univ-artois.fr

Mise en page avec memcril (B. Mazure, CRIL) et thloria (D. Roegel, LORIA).

Acknowledgements

Suppose someone asked me about my Ph.D. journey, I would quickly respond that it has been an exciting, colorful, and challenging expedition. Therein, I might learn about culture, people, language, knowledge, and even how to conquer ourselves. Indeed, after accumulating experience and knowledge, my aspirations and ambitions were becoming more and more fulfilled, and a new horizon gradually opened. And presently, the curtain of "doctoral student" has gradually closed. The baggage of skills and knowledge has been enlarged, filled, and weighted more heavily. The future research orientation was also more straightforward and clear. Hence, before I blow out the "Ph.D. Student" candle and light a "Doctor" candle, I would like to send my sincere thanks from the bottom of my heart to . . .

First of all, I would like to thank *Centre de Recherche en Informatique de Lens* (CRIL for short), the *Université d'Artois*, and the region of *Nord-Pas-de-Calais* for sponsoring my PhD studies. In particular, I send a thousand big thanks to Artois University for providing the fund for my Ph.D. student life. In addition, I would like to express my gratitude to members of the *CRIL laboratory*, in particular, for providing me with an office and various essential resources in order to complete my thesis. Therein, I had the opportunity to meet and exchange ideas with talented scientists who inspired me to push toward my full potential.

I sincerely thank *Prof. Eduardo Fermé* and *Prof. Odile Papini*, who accepted to review this dissertation. To be honest, I am extremely happy and fortunate to receive the comments and reviews to improve the content of the thesis. Moreover, I would also like to thank *Fatiha Sais* for agreeing to join my jury and providing her significant comments.

Above all, complete and indispensable gratitude goes to those I will never forget throughout my life as my thesis supervisors. Indeed, I would like to give my respect, thankfulness, appreciation, and admiration to my honorable supervisors, including *Prof. Sébastien Konieczny*, *Assoc. Prof. Ivan Varzinczak*, and *Assoc. Prof. Zied Bouraoui*. They trust me to undertake this thesis topic under their dedicated instructions and straightforward direction. "Solidarity, daring to face challenges, good process management, and enthusiasm for work" are what I have learned from these proficient researchers. It will also be the significant "characteristics" and "weapons" leading to the success of these thesis contributions. Once again, my most heartfelt thanks and best wishes to my dearest ones.

Moreover, I would also like to express my deep gratitude to a respectable professor, *Salem Benferhat*. One might say that he was my first stepping stone along the research route. Moreover, he brought me to this hexagon-shaped nation (France) and carried me into CRIL. To be honest, if the god of luck had not smiled at me for meeting with *Prof. Benferhat*, perhaps my dream of becoming a Ph.D. in computer science might still be in question.

My appreciation, admiration, and gratitude will be indispensable in the name of *Nicolas Schwind*. Although I only worked with him for over a year, a large amount of knowledge and skills have been transmitted. We are half a world away from each other, but he always sends me sympathetic sharing, high motivation, enthusiastic advice, and enunciated instructions all the time.

Unfortunately, during my Ph.D. journey, it was the time of the outbreak of the COVID-19 pandemic. Indeed, having friends around for help and encouragement was a blessing to me. By the way, I would also like to express my sincere thanks to my international friends (*Hugues Wattez, Anasse Chafik, Alexis de Colnet, Ryma Boumazouza, Marie Miceli, Amel Hidouri, Romain Wallon, Thibault Falque*, and others), and collaborators for keeping me in balance and for engaging in wonderful conversations with me about science and otherwise.

A family is a solid place to receive love and encouragement. I always cherish the statement that: "wherever we go, we will return, where there is always warmth, protection, and never-paid support". Here, I would like to thank my beloved family, including my parents, sisters, and brothers, for their fondness, passion, adoration, and support. Indeed, having a warm and welcoming place back in the

home country always makes life easier.

From the bottom of my heart, I would like to express special and huge thanks directly to my wife (*Ngoc-Trang Tran*), whom I have had a deep affection for during this journey. She has offered me much motivation and encouragement to get through the COVID '19 pandemic and the most difficult times during the past three years. A thousand enormous thanks to this dear wife who cared for my parents.

Finally, the spiritual foods from my closest friends on the other side of Asia, such as *Van-Duong Nguyen, DATA (Duc-Anh Ta), Thai-Bao Pham, The-Anh Luu, and Tan-Thong Nguyen*, are extremely important to finish this thesis. I would like to thank these best "brothers". They always share and support me when encountering difficulties on my Ph.D. road. One of the most memorable moments in my Ph.D. life was chickenpox (varicella) and COVID'19. I overcame these life-threatening crises thanks to my close Vietnamese friends and supervisors' encouragement. To end this acknowledgment, I would like to borrow a quote from *Maya Angelou* to guide my direction.

"**You cannot use up creativity. The more you use, the more you have.**"

I dedicate this dissertation to
my future life,
my family,
my wife,
my supervisors,
and a large number of dear friends.
You have all been my greatest cheerleaders.

Contents

List of Figures viii

List of Tables ix

Chapter 1 Introduction 1

1.1 General Context of Thesis . . . 1
1.2 Motivations and Contributions . . . 3
1.3 Organization of the Thesis . . . 5

Chapter 2 Knowledge Representation and Ontologies

2.1 Knowledge Representation . . . 8
2.2 Logic-Based Languages . . . 9
2.2.1 Propositional Logic . . . 10
2.2.2 First-Order Logic . . . 13
2.3 Description Logics . . . 14
2.3.1 Notions . . . 15
2.3.2 Description Logic $\mathcal{ALC}$ . . . 20
2.3.3 Lightweight Description Logic $\mathcal{EL}$ . . . 22
2.3.4 The Extended $\mathcal{EL}$ Family . . . 23
2.4 Ontology Languages . . . 26
2.4.1 What are Ontology Languages? . . . 26
2.4.2 Connection between Ontology Languages and Description Logics . . . 28
2.5 Conclusion . . . 32

Chapter 3 Belief Change

3.1 Background . . . 35
3.1.1 Formal Preliminaries . . . 36
3.1.2 Notions of Aggregation Function . . . 37
3.2 Belief Revision . . . 38
3.2.1 Belief Revision within Propositional Logic . . . 38

3.2.2 Belief Revision in Description Logic . . . 40
3.3 Belief Merging . . . 41
3.3.1 Family of Model-based Operators . . . 44
3.3.2 Family of Syntax-based Operators . . . 49
3.3.3 DA^2 Operators . . . 51
3.4 Ontology Merging . . . 52
3.5 Inconsistency Handling . . . 54
3.5.1 Inconsistency Handling in Propositional Logic . . . 55
3.5.2 Inconsistency Handling in Ontologies . . . 57
3.6 Summary . . . 59

Chapter 4 Model-Based Merging for Exploiting Open-Domain Ontologies

4.1 Context and Motivation . . . 60
4.2 Semantic-Based Ontology Merging . . . 62
4.2.1 Model-Based Framework for Merging Ontologies . . . 62
4.2.2 Generating Possible Merging Solutions . . . 63
4.2.3 Selection of the Best Interpretation . . . 66
4.2.4 Expressing the Result of Merging . . . 69
4.3 Postulates . . . 73
4.4 Experimental Evaluation . . . 73
4.4.1 Mapping Process . . . 74
4.4.2 Quantitative Results . . . 75
4.4.3 Qualitative Results . . . 77
4.5 Concluding Remarks . . . 79

Chapter 5 Region-Based Merging of Open-Domain Terminological Knowledge

5.1 Context and Motivation . . . 80
5.2 Foundations . . . 82
5.2.1 Description Logic $\mathcal{EL}_\bot$ . . . 82
5.2.2 Region Connection Calculus . . . 83
5.2.3 Qualitative Constraint Networks . . . 83
5.3 Merging Framework Description . . . 86
5.4 Translating Terminological Knowledge into QCNs . . . 87
5.5 QCN Merging . . . 88
5.5.1 Computing a Distance between a Basic Relation and a Profile of Constraints . . . 88
5.5.2 Using the Distance to Build a Merged Consistent QCN . . . 90
5.6 Selecting a Representative Scenario of the Merged QCN . . . 92

5.7 Translating the Representative Scenario into a Terminological Knowledge 95
5.8 Concluding Remarks 96

Chapter 6 Tree Edit Distance Based Ontology Merging Evaluation Framework
6.1 Context and Motivation 97
6.2 Background 99
6.3 Ontology Merging Evaluation Framework 101
6.4 Building Ontology Profile 102
6.5 Ontology Merging Operators 105
6.6 Experimental Evaluation 105
6.6.1 Description of Implementation 106
6.6.2 Quantitative Results 107
6.6.3 Qualitative Results 109
6.7 Concluding Remarks 111

Chapter 7 Conclusion and Discussion
7.1 Summary 113
7.2 Open Questions and Future Directions 114

Appendix A First Appendix
A.1 Proofs of Chapter 4 117
A.2 Proofs of Chapter 5 118

Appendix B Second Appendix
B.1 Application and Experiments of Chapter 4 124
B.2 Implementation of Chapter 5 126
B.3 Implementation of Chapter 6 127

Bibliography

List of Figures

2.1 Overview of knowledge representation. . . . 10
2.2 Complexity of description logics. . . . 21
2.3 Family of description logic $\mathcal{EL}$. . . . 24
2.4 Kinds of ontology. . . . 28
2.5 Family of the OWL 2 profile. . . . 30
2.6 The expressiveness between logic-based languages. . . . 31

3.1 The relation of belief merging, update, and revision. . . . 36
3.2 Ontology merging and ontology alignment. . . . 54

4.1 General framework of the merging process. . . . 63
4.2 The computation time of the merging process (seconds). . . . 76
4.3 Construction and merged outcome of $Procedure$, $Technique$, and $Process$. . . . 78
4.4 Structure and merged result of $Sentence$, $Statement$, and $Propositions$. . . . 78
4.5 Hierarchy and merged outcome of $Money$, $Cash$, and $Currency$. . . . 79

5.1 The idea of the contribution in Chapter 5. . . . 81
5.2 Relations of RCC-8 and RCC-5. . . . 84
5.3 An example of the QCN. . . . 85
5.4 Framework of ontology merging via QCN merging. . . . 86
5.5 QCN profile of Example 5.1. . . . 89
5.6 A conceptual neighborhood graph of *RCC-5*. . . . 89
5.7 The QCNs iteratively generated by our algorithm. . . . 92
5.8 The consistent scenarios of the merged QCN. . . . 93

6.1 General idea of the ontology merging operator evaluation framework. . . . 98
6.2 Framework of merging noisy ontologies. . . . 101
6.3 A part of a practical "conference" example. . . . 103
6.4 Structure of $ExtremityPart$ in the human ontology. . . . 110
6.5 Structure of $Conference_Contribution$ in the conference ontology. . . . 111

B.1 Interface of knowledge base merging application. . . . 124
B.2 Interface of merged result notification. . . . 125
B.3 Merged outcome of the case of $Gravel$, $Rock$, $Stone$ . . . 125
B.4 Implementation of input practical ontology sources. . . . 126
B.5 Merged result of the case of $Text$, $Book$, $Document$, and $Paper$. . . . 126
B.6 Result of generating noisy ontologies. . . . 127
B.7 Computation time of generating noisy ontologies of the cmt ontology. . . . 127

List of Tables

2.1 Name and symbol of connectives. . . . 11
2.2 A simple transparent translation between FOL and DL. . . . 15
2.3 Syntax and semantics of description logics. . . . 18
2.4 Syntax and semantics of $\mathcal{ALC}$. . . . 21
2.5 Syntax and semantics of $\mathcal{EL}$. . . . 23
2.6 Variants and complexities of the $\mathcal{EL}$ family. . . . 26
2.7 DAML+OIL constructors. . . . 29
2.8 Abstract syntax and DL syntax of OWL. . . . 32

3.1 Maximum belief merging operator Δ^{Max}. . . . 46
3.2 Sum belief merging operator Δ^{Σ}. . . . 47
3.3 Leximax belief merging operator Δ^{GMax}. . . . 49
3.4 Comparison between ontology integration and ontology merging. . . . 53
3.5 Systems of merging and matching ontologies. . . . 55

4.1 The collected results of the combinations and all interpretations. . . . 66
4.2 Distances between interpretations and ontologies of $\mathcal{P}_1$. . . . 69
4.3 The number of concepts, axioms, isa relations of three ontology sources. . . . 74
4.4 The number of synchronized concepts for each level. . . . 75
4.5 Example of mappings and concept name synchronization. . . . 76
4.6 The experimental results with SUMO (S), Wikidata (W) and BabelNet(B). . . . 77

5.1 Syntax and semantics of $\mathcal{EL}_{\bot}$. . . . 83
5.2 The composition table of RCC-5. . . . 84
5.3 Syntax and semantics of *RCC-5*, $v_C, v_D \in V$. . . . 85
5.4 Distances between relations from $\mathcal{B}$ and the profile of constraints $\mathscr{E}_{CD}$, for $v_C, v_D \in V$. . . . 90

6.1 Information of practical ontology sources. . . . 106
6.2 Parameters and computation time. . . . 108
6.3 Measurement of the distance (TED) to evaluate merging operators. . . . 110

Chapter 1

Introduction

1.1 General Context of Thesis

When we discuss Artificial Intelligence (AI for short), a sketched picture drawn immediately in the brain is intelligent platforms, innovative applications, and AI devices, such as controlling robots, chatbots, automated question-answering systems, recommendation systems, and others. One emerged question is "*How can computers be so intelligent?*". Indeed, in order for computers to comprehend in the same manner as humans, we must impart a certain amount of valuable and certain knowledge to them. We can call it training by humans. It can be algorithms, datasets, frameworks, knowledge, and others. In practice, there are different forms of knowledge. We can instruct and teach a person (human) with pictures, sounds, or natural language, but we cannot teach a computer to think like a human. They will not be able to understand any of the knowledge without going through encryption. Indeed, computers always work with binary. All signals transmitted and received on the computer are processed in binary form. Several scientists have encoded knowledge from various formats into digital signals that computers can read and analyze [LLHX19, CWY19]. Namely, we can encode knowledge, such as images [LLHX19], audios [CWY19], movies [KMS18], and statements into numerical form. Here we call them "knowledge bases".

A knowledge base can be expressed in both structured and unstructured formats (even in semi-structured formats) [OCK+22]. Usually, natural languages are present in an unstructured form, such as documents, emails, text, and others. Unstructured knowledge is difficult to deconstruct because it has no predefined knowledge model. In some research, many scientists have deformed the unstructured manner into a structured fashion to leverage the power of inference of those structures [RT22, YJH+21, DRH20]. Indeed, structured knowledge foundations can often be reasoned in terms of logic. They will have constraints and knowledge representations based on the chosen presentation language's grammar (syntax) and semantics. Moreover, structured knowledge of concepts and properties has frequently been in various fields, such as machine learning (ML for short) [RSS19], natural language processing (NLP for short) [Nav18], information retrieval (IR for short) [ZSWY19] and semantic web [GHL+19]. Therein, ontologies are one of the foundations for representing hierarchically structured knowledge. They are one of the most efficient means of communicating and sharing knowledge between people and machines. Thanks to their shared vocabulary, they provide a common understanding of a given domain. Furthermore, they can be used (1) as a query model for knowledge sources or (2) as a basis for integrating heterogeneous resources. For this study, we will use a structured knowledge base for our investigation. The presentation of this kind of knowledge must be cohesive and logical. In order to be more precise, ontology will be the subject of this thesis.

In practice, we will have a large number of disparate sources of knowledge in the same discipline,

e.g., different hospitals will provide diverse knowledge on treating ailments, or many supermarkets will issue a large number of bills throughout the same day. Naturally, each knowledge source will be initialized and deployed differently to adapt to its builders' actual situation and knowledge base. They will always be correct and reliable with regard to the conditions, location, entity, and other factors of their specific situation. As a result, knowledge will differ depending on the context; for example, we can understand the word $Rock$ as a type of music, but we can also imagine that $Rock$ is similar to a stone. From another perspective, when presenting a problem, we will have completely different ways of expressing it. We can occasionally agree on the same point of view, but sometimes our views are contradictory. Therefore, different knowledge bases will have various ways of representing knowledge. e.g., $Thanh$ tells that a $Work$ has many $Tasks$, whereas $Thanh$'s wife says that $Work$ and $Task$ are completely distinct. Otherwise, $Thanh$'s sister states that a $Task$ is split into many different $Work$. From $Thanh$'s example, it is not straightforward for us to find a global (common) and consistent statement in Thanh's family with many subjective opinions. Indeed, it is also a common practical problem that must be solved. Suppose we switch to a broader perspective with a knowledge base instead of a few statements similar to the above example. In that case, it will be an intriguing topic to address the difficulties facing businesses, organizations, and companies in the real world. Let us consider an example as follows:

- *We assume that MTT Company owns the main hospital in Arras and two-member-branch hospitals in Lens City and Avion City. These three hospitals have been operating for over five years, using independent knowledge sources and distinct technical divisions. Recently, MTT Company plans to establish a centralized platform and to integrate three technical departments into a single knowledge operation center. Here, the knowledge base structures of each hospital are entirely distinct. In particular, pharmacological names are likewise distinguishable. Therefore, some questions arise: Does it matter what occurs during the merging (combining) of the knowledge? How do we obtain a solid, faithful knowledge base when many sources are merged?.*

Intuitively, since the three knowledge bases from the three hospitals are separate, it is difficult to reconstruct the knowledge so that the sources are not lost and it is synchronized. Namely, it can also lead to disagreements between many sources; e.g., in the case of the same sickness, different physicians will prescribe and provide treatment differently. Pharmacological interactions and contradictions in this situation may arise if medications are grouped and merged. In other words, several conflicts will appear if we merge the three knowledge bases together.

In general, the problem of "mergers" is not unique, nor is it difficult to recognize and seek out these in practice. For instance, we can effortlessly see the consolidation of departments or units of a company or the amalgamation of many small companies into a corporation. Intuitively, it is evident that the challenge of merging several knowledge sources is always associated with reality. It will be crucial and significant in various situations and periods for most businesses or organizations (large or small groups). In practice, the merging issue is not only to confine a specific domain but also to appear in any topic (open-domain). For instance, in the same dance movements, a book says that *left-hand* is in the orientation of 30^o. In contrast, another document states that the *left-hand* raises at shoulder level. In another example, Europeans assert that dragons have wings, but Asians claim that dragons have none.

Solving the knowledge merging problems would be an interesting and challenging topic for the above issues. Hence, multiple scientists have focused on this area and proposed several solutions. Therein, some solutions concentrate on finding and uncovering common concepts (the same semantic) among sources [PP18, ABM15], i.e., $holidays$ and $vacations$. Others focus on handling inconsistencies and contractions when merging knowledge sources [Hun02, ZXLVDB14]. Usually, the source itself will be consistent, reasonable, and faithful. Nevertheless, conflicts will arise when we update (modify) a few pieces of knowledge in that source or consolidate multiple sources. Consequently, the merging [KP02, KPP11, BBL13], update [TPPvdS05, GPZ88], and revision[GRP09, QD09, KPP13] operators

were also born to address these issues. After all, their ultimate goal is to synthesize knowledge from various sources to obtain a source that is suitable, reasonable, plausible, consistent, and best reflects the input knowledge.

In this thesis, we pay attention to solving problems in knowledge merging. More specifically, we crucially exploit structured knowledge bases, i.e., ontologies. The frameworks and the merging operators will be proposed for this work to handle the arising challenges in the merging process. Our expected outcome is a merged ontology that is consistent and faithful. In the next section, we present our contributions' context and picture.

1.2 Motivations and Contributions

From the problems pointed out in the introduction and the desire to solve real-life problems in many fields related to knowledge base merging, one of our strong motivations is to explore and investigate structured knowledge from many sources. In this thesis, our primary orientation is **the merging of structured knowledge encoded in ontologies**, which is a challenging problem on its own. There are two potential approaches related to knowledge merging, including: *ontology integration techniques* and *leveraging the merging theory*.

- The first one exploits the correspondences of the concepts, lexical structures, and semantics in the ontology sources [CR16]. Said differently, the approach studies *matching, alignment, and mapping methods*. Ontology merging and alignment/matching have also attracted much attention in the literature. Several relevant pieces of research in this aspect have been proposed [Mei15, ZZ16, LGM+17, THT18]. For instance, Raunich and Rahm [RR11] proposed an automatic approach to merging large taxonomies such as product catalogs or web directories. Their approach studies an "equivalence" matching between a source and target taxonomy to merge them. Another ontology merging approach, proposed by Rangel [RACA18], is to discover semantic similarity measures and conduct an exhaustive search of the closest concepts. The author uses the mapping and preference to the target taxonomy. Namely, the author has proposed an algorithm that preserves instances of input taxonomies and the structure of the target taxonomy. Notably, this thesis assumes that all knowledge encoded by the different sources is already aligned and mapped to each other. Namely, we suppose that our input sources will share the same terminology. Hence, ontology alignment (or matching) is the process of determining correspondences between terminologies of ontologies. Also, ontology *merging* aims to combine two (or more) ontologies having the same terminology while handling conflicting statements.

- The second approach is to use the theory of belief change for ontology merging or description logic (DL) merging. It mostly concentrates on handling conflicts when merging sources. Therein, belief merging has a firm foundation in the propositional logic setting. Several belief merging operators have been proposed [Rev97, LM99, KP02]. Moreover, many DLs are more expressive than propositional logic but less expressive than first-order logic [Smu68, RS17]. Hence, the research on description logic merging is in the neighborhood of this work. Several relevant DLs merging studies are proposed [BBPW19, WWJQ12, BLSW02]. Typically, the authors propose a reasoning merging method underlying specific DLs (i.e., $DL\text{-}Lite$, $\mathcal{FL}$, $\mathcal{ALC}$, and others). This work follows the belief merging theory principles to develop frameworks for merging ontologies.

From the above two approaches, for this thesis, we emphasize that our work is different from mapping and matching. We will use the mapping to synchronize the concept names, and then our merging method will take effect. In other words, the mapping process will be conducted before merging. Here, we will

concentrate mainly on developing effective strategies, frameworks, and operators for merging ontologies where disagreements and conflicts occur. The *belief merging* platform [KP02] of the second approach is what motivates our proposal for new ontology merging operators.

Naturally, merging knowledge (or statements) provided by several sources often leads to inconsistencies. Here, we concentrate crucially on handling two kinds of conflicts in this thesis as follows:

- Let us take account of an example as follows: The SUMO ontology claims that a $Creek$ is disjointed from a $River$ while Wikipedia claims that every $Creek$ is a $River$. We call this case *a logical conflict* when merging.

- Another problem is that sources represent the knowledge from different perspectives, which is a conflict in terms of semantics, i.e., a source states that every $Text$ is a $Book$ while a second source says that every $Book$ is a Text. Observantly, this case is not a logical conflict with two opposing views. The merged result can be an equivalence between $Text$ and $Book$. However, we avoid flattening the hierarchies of ontology. We say this one is *a semantic conflict*.

Indeed, the goal of merging, as it is classically studied, is to define operators that take as input the knowledge provided by the different sources and produce as output a base that is consistent and synthesizes the best of the sources while dealing with inconsistencies. It turns out that merging open-domain knowledge is a particularly challenging task. The following are several study questions derived from the aforementioned challenges, motivations, and objectives.

Research questions: Thus, several research questions arise related to ontology merging, including:

1. How to resolve *conflicts in terms of semantics* or *logical conflicts* when merging many open-domain ontology sources?

2. The realization that ontology can be represented in the conceptual space [BCCAS20]. Moreover, we are also inspired by Forrest Peter's *"From ontology to topology in the theory of region"* [For96]. Hence, we think some questions: May we merge knowledge in a setting/framework more expressive than the original? May we translate knowledge into another space? i.e., topological space. How do we merge knowledge and deal with inconsistencies in that space?

3. How can we evaluate the quality of the existing merging operators?

Contributions: To answer and solve the above questions, we had three main contributions as follows:

1. A first work is **Model-based Merging of Open-Domain Ontologies** joint work with Zied Bouraoui, Sébastien Konieczny, and Ivan Varzinczak. It is published in the proceedings of the 33^{rd} IEEE International Conference on Tools with Artificial Intelligence (ICTAI'20) [BKMV20b] and the JIAF'20 conference [BKMV20a] in 2020.

   Brief description: We take a motivation from the theoretical foundations of propositional belief merging [Rev97, KP02] to propose a solution of model-based merging of ontology (using interpretations);

2. A second work is **Region-Based Merging of Open-Domain Terminological Knowledge** joint works with Zied Bouraoui, Sébastien Konieczny, Nicolas Schwind, and Ivan Varzinczak. It is published in proceedings of the 18^{th} International Conference on Principles of Knowledge Representation and Reasoning (KR'22) in 2022. [BKM+22].

Brief description: We take inspiration from merging Qualitative Constraint Networks (QCNs) [CKMS10, CKMS09] and leverage the Region Connection Calculus (RCC-5) [PT97] formalism to propose a method for merging open-domain terminological knowledge (simply called ontologies).

3. A third work is **Tree Edit Distance Based Ontology Merging Evaluation Framework** joint works with Zied Bouraoui, Sébastien Konieczny, and Ivan Varzinczak. It is published in proceedings of the 15^{th} International Conference on Knowledge Science, Engineering and Management (KSEM'22) in 2022 [BKMV22].

   Brief description: We proposed a framework to evaluate the quality of ontology-based merging operators. To this end, we take inspiration from Tree Edit Distance (TED) [ZSS92] to create noisy ontologies and measure the distance between these ontologies and the original one to perform an evaluation.

Here, the first and second studies have addressed the first two questions. Question 3 was done using an ontology merging evaluation framework in the third contribution.

1.3 Organization of the Thesis

This thesis is organized into seven chapters, including an introduction chapter, a conclusion chapter, two chapters on the scientific basis used in the thesis, and three chapters contributing to the thesis. In addition to this chapter, which is an introduction, the other chapters are as follows:

- In Chapter 2, we present an overview of several well-known knowledge representations. Therein, we provide a painting of logic-based languages. In particular, we discuss the notion of propositional logic, first-order logic, and description logic. These languages provide a solid foundation for the representation of structured knowledge. More details, we will provide pieces of knowledge concerning the structure and semantics of the language.

  Moreover, we will go straight to the ontology languages, which are the main object of this thesis. Since this dissertation will focus mainly on ontologies encoded under the description logic foundation, we will provide a connection between them. In particular, the description logic $\mathcal{EL}$ is an essential consideration in this chapter because this dissertation pays attention to ontologies implemented under the $\mathcal{EL}$ family's foundation. Moreover, we provide several fragments of the $\mathcal{EL}$ family and applicable transformation rules in detail. We take advantage of these rules to transform from complex forms to normal forms for our contributions.

- In Chapter 3, we formally introduce the notion of belief change. Research on this background has mainly concentrated on dealing with problems occurring in logical languages, as discussed in Chapter 2. In short, belief change deals with the addition, deletion, and combination of pieces of information and, more generally, with the revision, updating, and fusion of knowledge bases that lead to contradictions. Here, we draw a condensed picture of belief revision, merging, and inconsistency handling. In particular, we provide how to compute the distance between knowledge bases. In addition, this chapter will also mention the operators and their postulates.

  Especially since the primary goal of the thesis is ontology merging, we then introduce some methods of integration that are close to ontology merging in the web semantic community. It includes ontology mapping, ontology matching, and alignment. These integration techniques are also known to be close to and can be used for ontology merging. Therefore, we present the difference between them and provide some successful applications in this respect. Note that our research will go far away from mapping and matching. Rather, we will use matching as the pre-processing

step. In other words, we will use matching to obtain the same concepts between sources before implementing our proposed method to solve the conflicts.

- The work in Chapter 4 is our first contribution to ontology merging. Inspired by Konieczny et al.'s *belief merging operators* for propositional logic [KP02, Kon00, KPP11], we proposed a model-based merging method for open-domain ontology [BKMV20b]. In this work, we focus on dealing with conflicts in terms of semantics, i.e., every "Document" is a "Text". At the same time, another says that all "Text" is a "Document" (see more details in Section 4.2). Therefore, we propose a framework to solve the semantic conflicts. Namely, a brief description of our proposal is as follows: we shall enumerate all possible ways to express knowledge. Then, these possible ways will be expressed in terms of interpretations (i.e., all possible interpretations). Next, we use the merging belief theory to measure the satisfaction of each solution (interpretation) regarding the ontology profile. And then, we select the closest interpretations to the ontology sources (the minimum distances) by an algorithm. Finally, the most plausible interpretation case will be selected to represent the knowledge contained in the profile as a semantic merging result. Simultaneously, we translate this result into an ontology as a corresponding syntactic merging outcome. Moreover, we also provide the postulates of our merging operator.

- In Chapter 5, we have proposed an ontology merging method through merging Qualitative Constraint Networks (QCNs for short) [BKM$^+$22]. Here, instead of merging the ontology directly, we borrow a more expressive framework to merge knowledge. In particular, we transfer the knowledge base from the current space into another space. Then, we conduct a merging procedure in that more expressive space. Finally, we translate the merged knowledge back to the initial space. Indeed, we provide a faithful two-way translation to transfer knowledge back and forth between two spaces. Specifically, we build *a bridge* between Description logic $\mathcal{EL}$ [BBL05] and RCC-5 [PT97, TBL09]. We translate the $\mathcal{EL}$ lightweight ontology into the QCNs [CBGG97], and back. The conflict problems will be solved in the RCC-5 space. We propose a new algorithm to select the scenario of consistent QCNs and take advantage of the facts (instances) to obtain the final merged result. Note that we add the "bottom" concept into the $\mathcal{EL}$ so that the knowledge can appear in the logical conflicts, e.g., a source states that every $Fruit$ is a $Vegetable$. In contrast, another source says that the concept $Fruit$ is disjointed with the concept $Vegetable$. Hence, in this work, we focus on the description logic $\mathcal{EL}_\bot$ [Kri20]. Finally, we briefly discuss the exciting issues with this approach.

- In Chapter 6, we propose a framework to evaluate the quality of ontology-based merging operators. We take inspiration from Tree Edit Distance (TED for short) [ZSS92] to create noise trees and measure the distance between these trees for evaluating the models. In particular, the idea is to create noisy ontologies (NoiOn(s) for short) from an original ontology (OriOn for short) and then merge them. The noise reduction will be a significant point in comparing and assessing the merging operators. Our main goal is to evaluate the two merging operators proposed. We present some experiments with several practical ontologies to have a distance measure. For this work, we implement the framework and provide some experiments with the practical ontologies. Furthermore, we provide some useful information to estimate the quality of our merging operators.

- In the last chapter, we summarize this thesis's contributions. Moreover, we will also discuss the advantages and disadvantages of the contributions. Finally, a perspective on the future direction is also discussed in this chapter. Concerning future research, in particular, we provide not only directions related to the works we presented but also suggestions that lie outside the field of "ontology

merging" and close to the domain of graph theory and the ability of reasoning to apply machine learning to the symbolic AI aspect.

Chapter 2

Knowledge Representation and Ontologies

Knowledge Representation and Reasoning (KRR for short) is central to the grand challenge of AI, which is to comprehend the nature of intelligence and cognition so thoroughly that computers may be programmed to demonstrate human-like skills. KRR aims to develop computer systems that can also reason about world representations. Indeed, engines can understand in a manner analogous to how humans reason. Any component of the natural world or any hypothetical system about which one intends to express knowledge for computational purposes can serve as the domain of interest.

Knowledge in practice could be decomposed into numerous subcategories, i.e., procedural knowledge [GL86], conceptual knowledge [Car86], and factual knowledge [Ung68]. Many forms of knowledge can be characterized or represented using pictures, knowledge, assertions, arguments, constraints, rules, and various other formats. For this work, we will primarily take account of knowledge bases based on inference and reasoning. In other words, we place our primary emphasis on languages capable of logically representing the practical application of knowledge, such as propositional logic, first-order logic, description logic, and others. In this chapter, we provide each language's notions, syntax, and semantics, which are the "launchers" to develop our research. In order to pinpoint the place of our work and offer context for the logical languages that will follow, we will first introduce the concept of knowledge representation.

2.1 Knowledge Representation

Knowledge representation (KR for short) [LN92, MPP20] is the sub-field of AI that focuses on the issues of expressing, maintaining, and manipulating knowledge in a specific subject area. Fundamentally, it is an investigation of the manner in which the beliefs, intents, and judgements of an intelligent agent may be conveyed in a format that is appropriate for automated reasoning. One of the primary purposes of knowledge representation includes modeling intelligent behavior for an agent. In other words, knowledge representation is the field of study concerned with using formal symbols to represent a collection of propositions believed by some putative agent. Foundational KR research has two main goals: the first is to provide the *theoretical foundations* on which we can build systems that are useful, comprehensible, faithful, and reliable; the second is to provide the *logical and computational* foundations of knowledge representation formalism and reasoning processes [BL04, LN92].

There are several techniques of representing knowledge such as:

1. Logical representation [Sow99]: is a language with specific, unambiguous representational principles that deal with propositions. It outlines some crucial communication standards and reflects a conclusion based on several factors. Additionally, the syntax and semantics are explicitly stated,

facilitating sound inference. Syntax and semantics would be used to translate each statement into logic.

2. Semantic Network Presentation [Leh92]: serves as an alternate method to predicate logic for knowledge representation. We can express knowledge in the form of graphical networks via semantic networks. This network comprises nodes, which stand for individual objects, and arcs, which depict the interconnections between those individual objects. In addition, it connects the many representations of the object and organizes them into categories.

3. Frame Representation [FK85]: A frame is a record-like structure containing a set of properties and values that characterize a world object. This artificially intelligent knowledge structure subdivides knowledge into substructures by depicting stereotypical circumstances. It is a collection of slots and slot values of any size or type. Facets are the names and values assigned to slots.

4. Production Rules [DBS77]: The agent examines the conditions in the production rules. If the terms are fulfilled, the rule is triggered, and the corresponding action is executed. The rule's condition specifies which rule might well be applied to a certain case. In contrast, the action portion carries out the relevant problem-solving measures. This entire process of action is known as the recognize-act cycle.

Moreover, logic is relevant to knowledge representation and reasoning. Logic studies entailment relations, including languages, truth conditions, and inference procedures. Not unexpectedly, we rely mainly on the methods and technologies of formal symbolic logic. Specifically, we will use a widely popular logical language, the predicate calculus, or as it is frequently known, the language of first-order logic, as our preliminary knowledge representation language (FOL) [Smu68]. We note that first-order logic is another way of knowledge representation in AI and is also an extension of propositional logic [BM93]. Hence, we provide a brief description of propositional logic and first-order logic in the next section.

To understand clearly the path of knowledge representation for this thesis, an overview picture will be provided in Figure 2.1 [Hoe09, LN92]. It starts with the logical approach to building an intelligent application. Then, the knowledge is represented in the framework of description logics (DLs) [Rud11] and stored in the description logic ontology. In this work, we focus on discovering ontologies encoded in DLs. Hence, we present it as follows:

$$\text{Propositional Logic} \rightarrow \text{First Order Logic} \rightarrow \text{Description Logic} \rightarrow \text{DL ontology}.$$

Now, we introduce a brief description of logic languages that provide a stable platform for expressing structural knowledge, which is the emphasis of this thesis in the next section.

2.2 Logic-Based Languages

The expression of beliefs and the reasoning about these beliefs (e.g., the deduction of new beliefs) requires using a particular logic defined by a certain syntax and proper semantics. Syntax defines the set of logic symbols (or vocabulary, or lexicon) as well as grammar, i.e., a set of rules on the correct way to combine these symbols, thus characterizing the formulas of logic, i.e., the language. Semantics gives meaning to any formula in the language, which is the basis of reasoning about formulas. A programming language is an example of a logical language since it has a particular syntax and semantics. Moreover, the study of logic has fascinated humans. Logic is a tool that enables us to be precise about inference and semantics, ensuring that correct conclusions are drawn from legitimate premises.

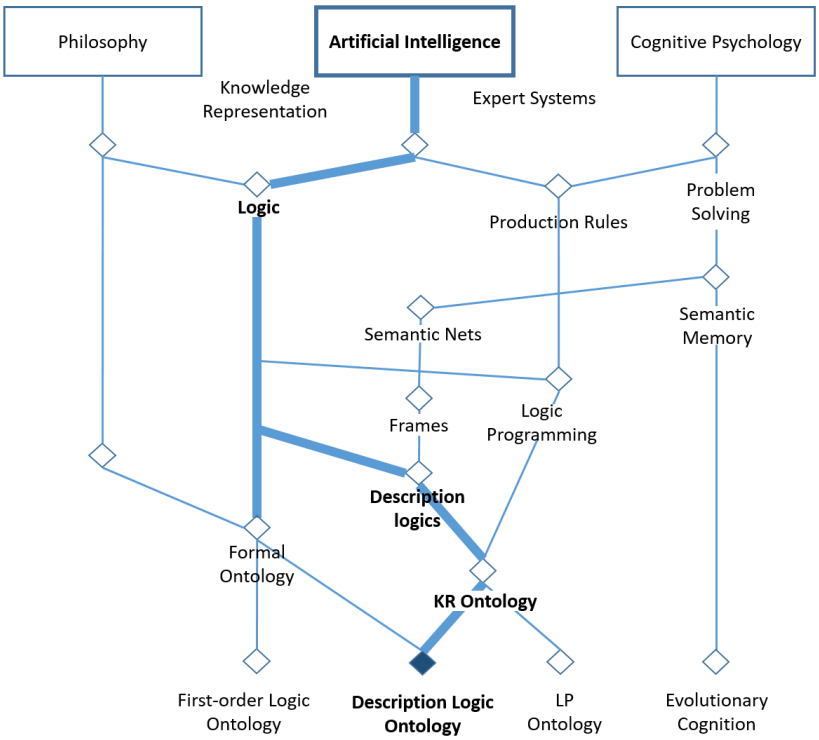


Figure 2.1: Overview of knowledge representation.

A spectrum of practical problems can easily be expressed and solved within the formalism of propositional logic, e.g., the issue of the satisfiability of formulas [GU89, BS92], that of the compilation of formulas [CMM01, DM02], the extraction of minimal inconsistent subsets of formulas [HK08, MLJ11, Kni03] to cite only a few examples. Therein, the problem of belief merging is no exception [KP02, DEKP20, CPRW16], which is why in this section, we will take into account the formalism of propositional logic. In this portion, we will introduce two popular forms of logic, such as propositional logic and first-order logic, to direct the target to description logic and ontology.

2.2.1 Propositional Logic

Propositional logic [Pos74, BM93, Luk20] represents ways to combine true statements by means of connectives to produce other true statements. Moreover, a propositional variable, or propositional atom, or propositional symbol, is also a Boolean variable representing a binary proposition and has two values (i.e., True or False). These two values are called truth values. We denote S as a finite set of propositional variables.

To form the formulas of the language and thus express knowledge, the propositional variables are combined using logical connectors. Some connectives provided by propositional logic are shown in Table 2.1. Here we say that a language L is the propositional language over a finite set S of propositional symbols, generated from the constant symbols **True, False** (corresponding $\top$ and $\bot$) and the unary connective $\neg$ and the binary connectives (remaining ones, see Table 2.1).

We write propositional symbols $a, b, x, y, \ldots$ and formulas will be denoted by Φ or Greek letter $\alpha, \beta, \ldots$ A belief base is a finite set of formulas denoted by $\mathcal{K}$. The set of the belief bases is denoted by $\mathcal{P}$, i.e., $\mathcal{P} = \{\mathcal{K}_1, \ldots, \mathcal{K}_n\}$. Here, a sentence or a statement must comply with this language's grammar. Now, we provide a formal definition of the grammar of the propositional logic language.

Name	Symbol
negative	$\neg$
and	$\wedge$
or	$\vee$
conditional	$\rightarrow$
bi-conditional	$\leftrightarrow$

Table 2.1: Name and symbol of connectives.

Definition 2.1 (Expression and Grammar) *The expression of propositional logic language is defined by the following grammar:*

$$\begin{aligned}\langle formula\rangle \stackrel{\text{def}}{=} & \neg\langle formula\rangle \\ & \mid (\langle formula\rangle \wedge \langle formula\rangle) \\ & \mid (\langle formula\rangle \vee \langle formula\rangle) \\ & \mid (\langle formula\rangle \rightarrow \langle formula\rangle) \\ & \mid (\langle formula\rangle \leftrightarrow \langle formula\rangle) \\ & \langle symbol\rangle\end{aligned}$$

where $\langle symbol\rangle$ are any letters.

Example 2.1 *Let x, y, z be variables or symbols. A formula is as follows:*

$$((x \wedge \neg y) \vee (x \rightarrow z))$$

Now, we provide the notion of interpretation in propositional logic. We write that an interpretation in propositional logic, denoted by $\mathcal{I}$, is a mapping : $\mathrm{P} \rightarrow \{\bot, \top\}$. For a proposition $X \in \mathrm{P}$, $X^{\mathcal{I}} = \bot$ represents that the proposition is **False**, and $X^{\mathcal{I}} = \top$ says that the proposition is **True**. The set of all interpretations over L is denoted by $\mathcal{W}$. The semantics of a formula in the interpretation $\mathcal{I}$ is defined inductively as follows, for all formulas α, β:

- $\mathcal{I}(\top)$ = True;
- $\mathcal{I}(\bot)$ = False;
- $\mathcal{I}(\neg\alpha)$ = True if and only if $\mathcal{I}(\alpha)$ = False;
- $\mathcal{I}(\alpha \wedge \beta)$ = True if and only if $\mathcal{I}(\alpha)$ = True and $\mathcal{I}(\beta)$ = True;
- $\mathcal{I}(\alpha \vee \beta)$ = False if and only if $\mathcal{I}(\alpha)$ = False and $\mathcal{I}(\beta)$ = False;
- $\mathcal{I}(\alpha \rightarrow \beta)$ = False if and only if $\mathcal{I}(\alpha)$ = True and $\mathcal{I}(\beta)$ = False; and
- $\mathcal{I}(\alpha \leftrightarrow \beta)$ if and only if $\mathcal{I}(\alpha) = \mathcal{I}(\beta)$.

Moreover, we also introduce *logic deduction*, *equivalence*, and *satisfaction* of propositional logic. Let α and β be two formulas. We say that β is entailed by α denoted $\alpha \models \beta$ if for any interpretation $\mathcal{I}$ that is a model of α (denoted by $\mathcal{I} \models \alpha$) and is also a model of β (denoted by $\mathcal{I} \models \beta$). Next, we denote that α and β are said to be equivalent, noted $\alpha \equiv \beta$ if and only if $\alpha \models \beta$ and $\beta \models \alpha$. We define the satisfaction relation between interpretations and formulas as follows:

- Given x is a proposition, $\mathcal{I} \models x$ iff $\mathcal{I}(x) = \top$;
- Given α is a formula, $\mathcal{I} \models \alpha$ iff $\mathcal{I}(\alpha) = \top$; $\mathcal{I} \models \neg\alpha$ iff $\mathcal{I} \not\models \alpha$;
- Given α and β are two propositional formulas, we have
  1. $\mathcal{I} \models \alpha \wedge \beta$ iff $\mathcal{I} \models \alpha$ and $\mathcal{I} \models \beta$;
  2. $\mathcal{I} \models \alpha \vee \beta$ iff $\mathcal{I} \models \alpha$ or $\mathcal{I} \models \beta$ or both;
  3. $\mathcal{I} \models \alpha \rightarrow \beta$ iff $\mathcal{I} \models \neg\alpha$ or $\mathcal{I} \models \beta$ or both; and
  4. $\mathcal{I} \models \alpha \leftrightarrow \beta$ iff $\mathcal{I} \models \alpha \rightarrow \beta$ or $\mathcal{I} \models \beta \rightarrow \alpha$.

Furthermore, when we take account of a belief base $\mathcal{K}$, some notions are as follows:

- $\mathcal{I}$ is called a model of $\mathcal{K}$, denoted by $\mathcal{I} \models \mathcal{K}$ iff $\forall\alpha \in \mathcal{K}$, $\alpha^{\mathcal{I}} = \top$. $Mod(\mathcal{K})$ denotes the set of models of $\mathcal{K}$.
- α is a consequence of $\mathcal{K}$, denoted by $\mathcal{K} \models \alpha$ iff every model of $\mathcal{K}$ is a model of α.
- $\mathcal{K}$ is *consistent* iff it has at least a model ($Mod(\mathcal{K}) \neq \emptyset$).

Theorem 2.1 ([DEGV01, JL74]) *The complexity of propositional logic programming is* $P - Complete$

For the complexity, bounded arithmetic and propositional logic [K+95] are closely interrelated and have several explicit and implicit connections to the computational complexity theory around P versus $NP - Problem$.

In general, propositional logic is less expressive and flexible than other logics (i.e., first-order logic and modal logic) [Smu68, Bar77, VHLP08, Che80a, CZ97, BCM+07]. We take account of the following example to illustrate some difficulties in representing knowledge in propositional logic.

Example 2.2 *Let us take account of two examples as follows:*

1. *We take account of the standard example of a valid deductive argument as follows:*

   For all x*, if* x *is* $male$ *then* x *is* $mortal$ *(or all* men *are* $mortal$*);*

   $Socrate$ *is a* Man*; Therefore,* $Socrate$ *is* $mortal$*.*

   Here, we call x *is* $male$ *presented by* $H(x)$ *and* x *is* $mortel$ *presented by* $M(x)$*. So, we can express as follows:*

   $$(H(x) \rightarrow M(x)) \wedge H(Socrate) \rightarrow M(Socrate)$$

   We cannot prove its validity in propositional logic. This clearly is an invalid argument. Its form is: $p, q \rightarrow r$*. A substitution instance of* r *can be* $False$ *when the substitution of both* p *and* q *are* $True$*. By the methods of propositional logic, the argument is invalid. It has limitations in propositional logic.*

2. *An example related to* $number$ *is as follows:*

   if n is a natural number, n+1 is also a natural number

   we can express this sentence in propositional logic as follows:

   $$n_1 \rightarrow n_2$$

$$n_2 \rightarrow n_3$$

$$\ldots$$

However, if there are an infinite number of propositional formulas, how can we encode all of them? Indeed, we need a bound on the natural number to conduct them, but it will not cover all knowledge of that sentence. Hence, propositional logic does not allow for this.

In the above examples, propositional logic is not perfectly adequate to express across all bases. Therefore, in the next section, we will mention first-order logic, which is more expressive.

2.2.2 First-Order Logic

First-order logic (FOL for short) [Smu68] can be seen as an extension of propositional logic. As shown above, propositional logic has no internal structure for the atomic formulas. Normally, they are propositional variables that are either $True$ or $False$, whereas the atomic formulas of FOL are *predicates*. Another important point is that the concept in FOL is *quantification*. The ability to assert that a certain property holds *for all* elements or that it holds *for some* elements.

The syntax of FOL is defined as a *signature*. A signature σ includes a set of constant symbols, a set of function symbols, and a set of predicate symbols. Note that we often refer to predicates as *relations*. Typically, we use letters such as c, d, x, y to denote constant symbols, f, g, h to denote function symbols, and P, Q, R to denote predicate symbols. Here, *terms* of FOL are used to reference objects. Terms are defined recursively as a constant, a variable, and an n-arity function symbol applied to n terms.

Definition 2.2 (Grammar of FOL) *A description of grammar for a first-order logic formula is as follows:*

Logical connectives: $\rightarrow|\leftrightarrow|\wedge\mid\vee\mid\neg$*;*
Quantifier: $\forall\mid\exists$*;*
Constant: *any string (without* $variable$*,* $predicate$ *and* $function$*);*
Variable: *any string, (without* $constant$*,* $predicate$ *and* $function$*);*
Predicate *any string, (without* $constant$*,* $variable$ *and* $function$*);*
Function *any string, (without* $constant$*,* $variable$ *and* $predicate$*);*
Term: $Function(Term, \ldots, Term) \mid Constant \mid Variable$*;*
Primitive Formula: $Predicate(Term, \ldots, Term)$*;*
Formula: $\top \mid \bot \mid \forall x \mid \exists y \mid x \rightarrow y \mid x \leftrightarrow y \mid x \wedge y \mid x \vee y \mid \neg x$.

Example 2.3 *Several examples to illustrate natural language representation by FOL are as follows:*

- *Statement: "Not all birds can fly". It is expressed as:* $\exists x Bird(x) \wedge \neg Fly(x)$ *or* $\neg(\forall x\ Bird(x) \rightarrow Fly(x)))$*;*
- *Statement: "If someone can solve the complex problem, then Thanh can do it". It is represented:* $(\exists x\ Solve(x, problem)) \rightarrow Solve(Thanh, problem)$*; and*
- *Statement: "Thanh love all people who do not love themselves". We write as follows:* $(\forall x\ Person(x) \wedge \neg Loves(x, x)) \rightarrow Loves(Thanh, x)$.

Regarding the semantics of FOL, a FOL model is a tuple $\langle \mathcal{D}, \mathcal{I} \rangle$, where $\mathcal{D}$ is a non-empty domain of objects, and $\mathcal{I}$ is an interpretation function. The domain $\mathcal{D}$ is a set of objects and can be finite, infinite, or even uncountable. Here, we say that the interpretation function $\mathcal{I}$ assigns an interpretation to each constant, function, and predicate symbol. Hence, we have several conditions as follows:

- If c is a constant, $\mathcal{I}(c)$ is a object in $\mathcal{D}$;
- If f is a function, $\mathcal{I}(f)$ is a total function from $\mathcal{D}^n$ to $\mathcal{D}$;
- If p is a predicate with $n > 0$, $\mathcal{I}(p)$ is a subset of $\mathcal{D}^n$; and
- If p is a predicate of arity 0, $\mathcal{I}(p) = True \mid False$.

Example 2.4 *Let us consider a statement: "All men drink coffee". Then, we can encode in FOL as follows:*

$$\forall x\ man(x) \rightarrow drink(x, coffee)$$

A model for this sentence might be the following:

$$\mathcal{D} = \{Thanh, Alexis, Marry\}$$

$$\mathcal{I}(Thanh) = Thanh$$

$$\mathcal{I}(drink) = \{\langle Thanh, coffee\rangle, \langle Alexis, coffee\rangle\}$$

$$\mathcal{I}(man) = \{Thanh, Alexis\}$$

To represent the knowledge base in this thesis, we will present a fragment of FOL, which is more expressive than propositional logic but less expressive than FOL [BCM+07]. This fragment will be the foundation for our objective (i.e., ontology). Particularly, we present description logic in the next section.

2.3 Description Logics

Description logics (DLs, for short) [BLSW02, BCM+07, BHLS17] are a family of knowledge representation languages corresponding to decidable fragments of first-order logic (FOL)[1] using only unary and binary predicates. While the absence of higher-arity predicates may seem to be a significant constraint, it turns out that unary and binary predicates (classes and properties) play a significant role in modeling. Indeed, DLs provide the basis of the OWL Web Ontology Language [Hor05, HKP+09], a W3C ontology language for the Semantic Web [MFSH03, GPC02], and RDF & RDFS [CWWM06]. In the next section, we will present these notions in detail.

Structurally, a knowledge base underlying the foundation of description logic consists of terminological and assertional parts. In other words, a DL knowledge base (KB) has two parts: a $TBox$ that contains general knowledge about the application domain and an $ABox$ that contains facts about particular individuals. The $TBox$ can be viewed as an ontology, which provides a conceptual model for the knowledge stored in the $ABox$. The terminological part (a.k.a $TBox$) represents *concepts* and *roles* with their relationships. Namely, a DL system is then characterized by four aspects [DGL96]:

1. The set of constructs constituting the language used for building the concepts and the roles mentioned in the $TBox$ and in the $ABox$;
2. The kind of assertions that may appear in $TBox$;
3. The kind of assertions that may appear in $ABox$; and
4. The inference mechanisms provided for reasoning on the knowledge bases expressible in the system.

[1] Although there are some DLs which are not decidable and not fragments of FOL.

Name	First-Order Logic (FOL)	Description Logic (DL)
Concept	$A(x)$	A
Concept Assertion	$C(a)$	$C(a)$ or $a : C$
Role Assertion	$r(a, b)$	$r(a, b)$ or $(a, b) : r$
Similar and Equivalence	$A \equiv B$	$A \equiv B$
Negation	$\neg C(x)$	$\neg C$
Conjunction	$C(x) \wedge D(x)$	$C \sqcap D$
Disjunction	$C(x) \vee D(x)$	$C \sqcup D$
Concept Subsumption	$\forall x(C(x) \rightarrow D(x))$	$C \sqsubseteq D$
Role Subsumption	$\forall x, \forall y(r(x, y) \rightarrow s(x, y))$	$r \sqsubseteq s$
Existential Restriction	$\exists y(r(x, y) \wedge C(y))$	$\exists r.C$
Universal Restriction	$\forall y(r(x, y) \rightarrow C(y))$	$\forall r.C$
Transitive Role	$\forall x \forall y \forall z(r(x, y) \wedge r(y, z) \rightarrow r(x, z))$	$r \circ r \sqsubseteq r$

Table 2.2: A simple transparent translation between FOL and DL.

In general, the key reasoning problems taken into account by the DL community are *satisfiability, subsumption, and instance checking*. In particular, checking satisfiability is essential for identifying modeling errors. In contrast, instance checking and subsumption are used to identify $TBox$ axioms and $ABox$ assertions that follow from the knowledge of the KB.

As with first-order logic, DL *semantics* are formal and well-defined. As a result, the expressive knowledge is unambiguous, i.e., it is entirely evident what it signifies based on its semantics. Therefore, given an appropriate calculus, it is feasible to generate logical conclusions from an ontology mechanically. This process is called *"deductive reasoning"*. Otherwise, *entailments* are the terms used in DLs for logical inferences. Entailments are the explicitization of implicit relationships through reasoning.

Before presenting formal conceptions and definitions of DLs, we provide a straightforward translation between FOL and DL, as shown in Table 2.2.

2.3.1 Notions

Let us formally define the notions that will be used in the whole thesis. Let N_C, N_R, and N_I be pairwise disjoint sets of concepts, roles, and individual names, respectively. We call *term* any element in $N_C \cup N_R \cup N_I$. A concept, A, is called atomic if $A \in N_C$. A role r is denoted as atomic if $r \in N_R$. Syntax and semantics of description logics are shown in Table 2.3 [BCM+07]. A concept C (or role r) is called *complex* if $C \notin N_C$ (or $r \notin N_R$) and C (or r) is an expression built from $N_C \cup N_R \cup N_I$ according to syntax rules, called constructors, of DLs. In $\mathcal{ALC}$, complex concepts are built using negation $\neg$, conjunction $\sqcap$, disjunction $\sqcup$, existential restrictions $\exists$, and universal restrictions $\forall$, from the atomic concepts N_C that include concept names, top $\top$ and bottom $\bot$. Let us take into account an example of complex concepts as follows:

Example 2.5 *The following C and D are complex concepts expressed by the atomic concepts A, B and the atomic role r:*

$$C = A \sqcap \exists r.(B \sqcap \exists r.\top)$$

$$D = \neg A \sqcup \forall r.(B \sqcap \exists r.A)$$

A $TBox$ encodes relations between concepts via *concept inclusions* (CIs) and relations between *role inclusions*. We formally define a terminological box as follows:

Definition 2.3 ($TBox$) *A $TBox$ $\mathcal{T}$ is a finite set of general concept inclusions and role inclusions. Some notions is defined as follows:*

- A general concept inclusion *(GCI) (a.k.a. concept subsumption) is an expression of the form $C \sqsubseteq D$, where C and D are concepts. A concept C is said to be subsumed by a concept D. We call a GCI of the form $A \sqsubseteq B$, where $A, B \in N_C$, an atomic concept subsumption;*
- A role inclusion *(RI) is an expression of the form $r \sqsubseteq s$, where r and s are complex roles. Moreover, we call a* role inclusion *of the form $r \sqsubseteq s$, where $r, s \in N_R$, an atomic role subsumption; and*
- *$C \equiv D$ is the abbreviation for $\{C \sqsubseteq D, D \sqsubseteq C\}$. Regarding role subsumption, $r \equiv s$ if and only if $\{r \sqsubseteq s, s \sqsubseteq r\}$. We call that a concept C (or role r) is* equivalent *to a concept D (or role r).*

Example 2.6 (Person example) *Let us consider an example of a concept definition with a statement: "a mother is a female (person) that has some children". We can represent it in DL as follows:*

$$Mother \equiv Female \sqcap \exists hasChild.Person$$

and every $Female$ and every $Male$ are a $Person$

$$Male \sqsubseteq Person, Female \sqsubseteq Person$$

In addition, an $ABox$ encodes facts about individuals in the domain via concepts and role *assertions*. We provide a formal definition as follows:

Definition 2.4 ($ABox$) *An $ABox$, denoted by $\mathcal{A}$, is a finite set of* concept assertions *and* role assertions *where*

- A concept assertion *(CA) is an expression of the form $C(a)$, where $a \in N_I$, C is a concept; and*
- A role assertion *(RA) is an expression of the form $r(a, b)$, where $a, b \in N_I$ and r is a role.*

Example 2.7 (Continue to "Person example")

$$Female(Marry), Female(Alice), Male(Tom), Person(Tom), hasChild(Marry, Tom)$$

Note that $GCIs$, RIs, CAs, and RAs are called *axioms*. A set of all axioms is called a DL knowledge base, including the union of a $TBox$ and $ABox$. In this thesis, we focus crucially on representing the knowledge base encoded in DL, so-called by *a DL ontology*. Now, we define a DL ontology.

Definition 2.5 (DL ontology, DL knowledege base) *Given a $TBox$ $\mathcal{T}$ and $ABox$ $\mathcal{A}$, we call $\mathcal{O} \stackrel{\text{def}}{=} \mathcal{T} \cup \mathcal{A}$ as* a DL ontology. Moreover, we also denote $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ as the knowledge base corresponding to a DL ontology $\mathcal{O}$. *Now, we write $\mathcal{P}$ as the set of knowledge bases (or DL ontologies).*

Moreover, an ontology expresses a set of constraints determining what is possible and not impossible in the domain. The semantics of DLs rely on the semantics of *interpretations*. We provide a formal definition of interpretation.

Definition 2.6 (Interpretation) *An interpretation, denoted by $\mathcal{I}$, is $\mathcal{I} \stackrel{\text{def}}{=} \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, where $\Delta^{\mathcal{I}}$ is non-empty set (called* domain*) and $\cdot^{\mathcal{I}}$ is a function (called* interpretation function*) which maps*

- *every concept name (CN) $A \in N_C$ to a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$;*

- *every role name (RN)* $r \in N_R$ *to a binary relation* $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$*; and*
- *every individual name* $a \in N_I$ *to an element* $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$.

An interpretation that satisfies the constraints given by the axioms of the ontology is called its *model*. Now, we define what satisfaction is and what a model is.

Definition 2.7 (Model, Satisfiability) *Let* Φ *be an axiom and* $\Phi \in \mathcal{O}$*. An interpretation* $\mathcal{I}$ *satisfies*

- *a general concept inclusion* $C \sqsubseteq D$ *if* $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$*;*
- *a role inclusion* $r \sqsubseteq s$ *if* $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$*;*
- *a concept assertion* $C(a)$ *if* $a^{\mathcal{I}} \in C^{\mathcal{I}}$*;*
- *a role role conclusion* $r(a, b)$ *if* $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$*; and*
- *a consistent ontology* $\mathcal{O}$ *if* $\mathcal{I} \models \Phi, \forall \Phi \in \mathcal{O}$.

An interpretation $\mathcal{I}$ *that satisfies* $\mathcal{O}$ *is called* a model of $\mathcal{O}$.

Note that we denote $\models$ as satisfaction between $\mathcal{I}$ and axioms, i.e., $\mathcal{I} \models \Phi$. Here, for ease of presentation, we extend how to represent an interpretation $\mathcal{I}$ as a set contained the interpretation of concepts and roles in **the form of angle brackets** $\mathcal{I} = \langle \ldots \rangle$, i.e., $\mathcal{I} = \langle A^{\mathcal{I}} = \{a\}, B^{\mathcal{I}} = \{a, b, c\}, C^{\mathcal{I}} = \{a, c\}, r^{\mathcal{I}} = \{(a, b), (b, c)\}\rangle$. Intuitively, a model can draw a "world" that allows it to exist according to the knowledge expressed by the ontology.

Example 2.8 *Let us consider the ontology* $\mathcal{O} = \{A \sqsubseteq B\}$ *which encodes that "every* A *is a* B*". We assume that there are two interpretations* $\mathcal{I}_1, \mathcal{I}_2$*. we have*

- *The interpretation* $\mathcal{I}_1$ *is a model of* $\mathcal{O}$ *if we have* $\Delta^{\mathcal{I}_1} = \{a\}$ *and* $\mathcal{I}_1 = \langle A^{\mathcal{I}_1} = \{a\}, B^{\mathcal{I}_1} = \{a\}\rangle$*;*
- *Otherwise,* $\mathcal{I}_2$ *is not a model of* $\mathcal{O}$ *if we have* $\Delta^{\mathcal{I}_2} = \{a\}$ *and* $\mathcal{I}_2 = \langle A^{\mathcal{I}_2} = \emptyset, B^{\mathcal{I}_2} = \{a\}\rangle$.

Furthermore, we also denote $\models$ as the semantics of interpretations that determines the *entailment relation*.

Definition 2.8 (Entailment) *Let* Φ *be an axiom. The entailment relation is denoted by* $\models$*. We have the entailment relations as follows:*

- $\mathcal{O} \models \mathcal{O}'$ *if all models of* $\mathcal{O}$ *are models of* $\mathcal{O}'$*;*
- $\mathcal{O} \equiv \mathcal{O}'$ *is for* $\mathcal{O} \models \mathcal{O}'$ *and* $\mathcal{O}' \models \mathcal{O}$*;*
- $\mathcal{O} \models \Phi$ *is for* $\mathcal{O} \models \{\Phi\}$*;*
- $\Phi \models \Phi'$ *is for* $\{\Phi\} \models \{\Phi'\}$*,* $\Phi \equiv \Phi'$ *is for* $\{\Phi\} \equiv \{\Phi'\}$*; and*
- *An individual* a *is called an instance of a concept* C *if* $\mathcal{O} \models C(a)$.

The important point is that the models can have arbitrary cardinality in DLs, but the domain of interest is finite. Here we also define "fixed-domain models" [GRS16, RS17] to restrict reasoning to models of finite domain size (called *finite model reasoning*). We will use this notion for our contribution (see Chapter 4) to be suited for practical application. Formally, we define "fixed-domain semantics" as follows:

Name	Syntax	Semantics
Individuals:		
Individual name	a	$a^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
Roles:		
Atomic role	r	$r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
Inverse role	r^{-}	$\{(x, y) \mid (y, x) \in r^{\mathcal{I}}\}$
Role composition	$r \circ s$	$\{(a, c) \mid \exists b\,(a, b) \in r^{\mathcal{I}} \wedge (b, c) \in s^{\mathcal{I}}\}$
Role Negation	$\exists \neg r.C$	$\{x \mid \exists y \in \Delta^{\mathcal{I}} : (x, y) \notin r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
Role Union	$\exists r \cup s.C$	$\{x \mid \exists y \in \Delta^{\mathcal{I}} : (x, y) \in r^{\mathcal{I}} \cup s^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
Concepts:		
Atomic Concept	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
Top Concept	$\top$	$\Delta^{\mathcal{I}}$
Bottom Concept	$\bot$	$\emptyset$
Concept	C	$C^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
Concept Negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
Concept Conjunction (Intersection)	$C_1 \sqcap C_2$	$C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$
Concept Disjunction (Union)	$C_1 \sqcup C_2$	$C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$
Existential Restriction	$\exists r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}}, (x, y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
Universal (Value) Restriction	$\forall r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \forall y \in \Delta^{\mathcal{I}}, (x, y) \in r^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$
At-most restriction	$\leq r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \mid \{y \in \Delta^{\mathcal{I}} \mid (x, y) \in r^{\mathcal{I}}\} \mid \leq n\}$
At-least restriction	$\geq r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \mid \{y \in \Delta^{\mathcal{I}} \mid (x, y) \in r^{\mathcal{I}}\} \mid \geq n\}$
Local reflexivity	$\exists r.Self$	$\{x \mid (x, x) \in r^{\mathcal{I}}\}$
Nomimal	$\{a\}$	$\{a^{\mathcal{I}}\}$

Table 2.3: Syntax and semantics of description logics.

Definition 2.9 (Fixed-Domain Semantics [GRS16]) *Let us denote that a fixed domain is an non-empty finite set $\Delta \subseteq N_I$. An interpretation $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \mathcal{I} \rangle$ is denoted to be Δ-fixed if $\Delta^{\mathcal{I}} = \Delta$ and $a^{\mathcal{I}} = a$ for all $a \in \Delta$. Accordingly, let $\mathcal{K}$ be a knowledge base, an interpretation $\mathcal{I}$ is a Δ-model of $\mathcal{K}$ if $\mathcal{I}$ is a Δ-fixed interpretation and $\mathcal{I} \models \mathcal{K}$. A knowledge base $\mathcal{K}$ is called Δ-satisfiable if it has a Δ-model. We also say that $\mathcal{K}$ Δ-entails an axiom Φ (denoted by $\mathcal{K} \models_{\Delta} \Phi$) if every Δ-model of $\mathcal{K}$ is also a model of Φ.*

Fixed-domain semantics might be considered an extension of finite-model reasoning's restrictions [GRS16]. This restriction affords us not only the computational complexity benefit but also, in some situations, potentially more comprehensible models of a knowledge base [RS17]. A recalled Wolpertinger reasoner[2] has been developed to support these reasoning tasks. It performs satisfiability checking and model enumeration.

Now, we define that two concepts are disjoint in all models denoted by *a disjointness axiom*

Definition 2.10 (Disjointness axiom) *Let C and D be concepts. A disjointness axiom is in the form of $C \sqsubseteq \neg D$ or $C \sqcap D \sqsubseteq \bot$.*

Note that, it can occur an inconsistency for $\mathcal{O}$ with disjointness axioms. It is also known as a logical conflict. Now, we provide a formal definition of the logical conflict.

Definition 2.11 (Logical conflict) *Let $A, B \in N_C$, with $A \neq B$ and $\mathcal{P} = \{\mathcal{O}_1, \ldots, \mathcal{O}_n\}$. We say that $\mathcal{P}$ has a logical conflict if and only if $\mathcal{O}_i \models A \sqsubseteq B$ and ($\mathcal{O}_j \models A \sqcap B \sqsubseteq \bot$ or $\mathcal{O}_j \models A \sqsubseteq \neg B$) where $\mathcal{O}_i, \mathcal{O}_j \in \mathcal{P}$.*

[2] https://github.com/wolpertinger-reasoner/Wolpertinger

Moreover, we also introduce the notions of incoherence and inconsistent in DL as follows:

Definition 2.12 (Unsatisfiable concept [FHP+06, BQL07]) *A named concept C in DL is unsatisfiable if and only if $C^{\mathcal{I}} = \emptyset$ for each interpretation $\mathcal{I}$.*

Definition 2.13 (Incoherent knowledge base, Incoherent ontology [BQL07]) *A knowledge base $\mathcal{K}$ is incoherent iff there exists an unsatisfiable concept in $\mathcal{K}$.*

Definition 2.14 (Inconsistent ontology) *A knowledge base $\mathcal{K}$ a.k.a. an ontology $\mathcal{O}$ is inconsistent iff it has no model.*

Example 2.9 *Let us consider several examples related to (in)consistent and (in)coherent ontology. We have:*

(1) $\mathcal{O}_{E_1} = \{C_1 \sqsubseteq \neg C_2, C_1(a), C_2(a)\}$ is an example of an inconsistent but coherent ontology*, in which the two disjoint concepts C_1 and C_2 share an instance a;*

(2) $\mathcal{O}_{E_2} = \{C_1 \sqsubseteq \neg C_2, C_3 \sqsubseteq C_1, C_3 \sqsubseteq C_2, C_1(a), C_2(b)\}$ is an example of a consistent but incoherent ontology*, in which the two disjoint concepts C_1 and C_2 share a sub-concept C_3;*

(3) $\mathcal{O}_{E_3} = \{C_1 \sqsubseteq \neg C_2, C_3 \sqsubseteq C_1, C_3 \sqsubseteq C_2, C_3(a)\}$ is an example of an inconsistent and incoherent ontology*, in which the two disjoint concepts C_1 and C_2 share a sub-concept C_3, which has an instance a; and*

(4) $\mathcal{O}_{E_4} = \{C_1 \sqsubseteq \neg C_2, C_1(a), C_2(a)\}$ is an example of an inconsistent but coherent $TBox$*, in which the two disjoint concepts C_1 and C_2 share a sub-concept which is a nominal a.*

Description logic languages Description logic (DL for short) languages are a family of knowledge representation languages that can be used to represent the knowledge of an application domain in a structured and formally well-understood way. It takes into account both the structure of a DL knowledge base and the reasoning services connected with it. The Description logic language family is the best-understood collection of formalism in all knowledge representations. The computational space has been meticulously mapped out. Moreover, many systems were developed to test various inference computing methods and were employed in several applications. Typically, each description logic has a unique specification for a particular domain. Several description logics exist for expressing knowledge, a casual naming convention, and a general overview of the operators executed. The fundamental description logic languages are:

- $\mathcal{AL}$: Attributive language is the base language that allows atomic negation (negation of concept names that do not appear on the left-hand side of axioms), concept intersection, universal restrictions, and existential quantification. It is a very well-known DL that corresponds to multi-modal logic [BCM+07];
- $\mathcal{FL}$: Frame-based description is a language which allows concept intersection, universal restrictions, limited existential quantification, role restriction [CF90];
- $\mathcal{EL}$: Existential language that allows concept intersection and existential restrictions (of full existential quantification) [BBL05]; and
- $\mathcal{SHIQ}$: It is a very expressive DL which constitutes the basis of the OWL family of DLs adopted as standard languages for ontology specification in the Semantic Web [GHLS08].

These languages can be extended to support different constructs, i.e., $\mathcal{ALC}$ departs from $\mathcal{AL}$ with the complement of any concepts allowed, not just atomic concepts. In similarity, $\mathcal{EL}^{++}$ is an extension of $\mathcal{EL}$ that has been added to some components such as nominal, concrete domain, bottom concept (negation), and others (see more details in Section 2.3.4).

Complexity measures Complexity theory is a crucial tool for calculating the cost of the optimal method for the worst case of a given issue. Therefore, it is beneficial to assess the difficulty of the problems for which there are presently no polynomial-time algorithms that solve them. A decision problem is a problem with only two potential solutions: Yes and No. Here, let L be a language and M be a Turing machine with stopping states of Yes or No. We say that M decides L if, for each word $x \in L$, the computations of M on input x terminate with the stop state Yes within several steps. In general, there are different complexity measures for solving conflicts of knowledge bases depending on what parameters of the problem are regarded as the input. Now, we provide some classes of complexity as follows:

- AC^0: They are problems that can be solved by a uniform family of circuits of polynomial-size;
- P: A decision problem solves in polynomial time in the size of input;
- $P\text{-}complete$: A decision problem is $P\text{-}complete$ (complete for the complexity class P) if it is in P and every problem in P can be reduced to it by an appropriate reduction;
- NP: A decision problem solves solved in non-deterministic polynomial time;
- EXP: A decision problem solves in exponential time (or in polynomial time with an NP oracle);
- $CoNP$: They are problems whose complement is in NP;
- $CoNP\text{-}complete$: $coNP\text{-}complete$ are the hardest problems in $co\text{-}NP$;
- $NP\text{-}hard$: These are the problems that are at least as hard as the $NP\text{-}complete$ problems. Note that $NP\text{-}hard$ problems do not have to be in NP, and they do not have to be decision problems;
- $NP\text{-}complete$: A problem that is NP and $NP\text{-}hard$ are $NP\text{-}complete$;
- $NEXPTime$: A decision problem solves in non-deterministic polynomial time with an NP oracle; and
- $2NEXPime$: It is a problem whose complement is in $NEXPTime$.

The classes are related as follows: $P \subseteq NP \subseteq EXP \subseteq NEXPTime \subseteq 2NEXPTime$. The complexity of some pieces of description logics are shown in Figure 2.2 [Ros11, KRH07, KP17]. Note that this thesis only concentrates on the description logic $\mathcal{EL}$. In the next section, we mention description logic $\mathcal{ALC}$ and $\mathcal{EL}$'s family in detail.

2.3.2 Description Logic $\mathcal{ALC}$

In this section we recall the definitions of the description logic $\mathcal{ALC}$ [SSS91, Var18, BCM+07], some well known notions like satisfiability, subsumption and acyclic $TBoxes$. An ontology or knowledge base in $\mathcal{ALC}$ is a set of axiom Φ_i defined over the triple (N_C, N_R, N_I) as well.

For $\mathcal{ALC}$, concept satisfiability is PSpace-complete [SSS91]. Regarding theories that use acyclic terminologies, the complexity is the same (see the complexity of Baader et al.[BH91]). These theories mainly comprise concept inclusions (CIs), where the left-hand side is atomic and represents a term's partial definitions. No term is permitted to be defined in theory, either directly or indirectly, in terms of another term. Concept satisfiability is $ExpTime$ in the presence of theories with generic CIs. (see Picture 2.2). Now, we present a formal definition of $\mathcal{ALC}$'s grammar as follows:

Definition 2.15 (Grammar of $\mathcal{ALC}$) *An $\mathcal{ALC}$ terminology is defined by the following formation rules:*

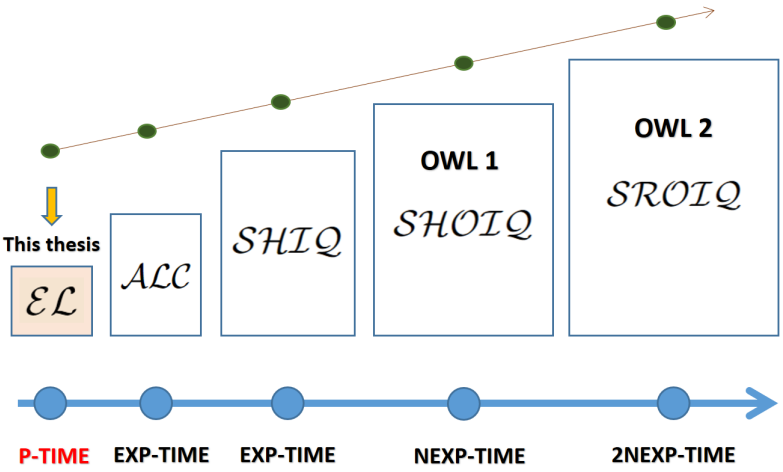


Figure 2.2: Complexity of description logics.

Name	Syntax	Semantics
Top Concept	$\top$	$\Delta^{\mathcal{I}}$
Bottom Concept	$\bot$	$\emptyset$
Concept	C	$C^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
Concept Negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
Concept Conjunction	$C_1 \sqcap C_2$	$C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$
Concept Disjunction	$C_1 \sqcup C_2$	$C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$
Existential Restriction	$\exists r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}}, (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
Universal Restriction	$\forall r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \forall y \in \Delta^{\mathcal{I}}, (x,y) \in r^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$

Table 2.4: Syntax and semantics of $\mathcal{ALC}$.

- *Axioms are of the form:*

$$C \sqsubseteq D \mid C \equiv D$$

  where C and D are concept expressions.

- *Concept expressions are of the form:*

$$CN \mid \top \mid \bot \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \exists r.C \mid \forall r.C$$

  where CN is a concept name, C and D are concept expressions and r is role expression.

- *Role expressions are of the form RN, where RN is a role name.*

The structures of syntax, as well as semantics in $\mathcal{ALC}$, are shown in Table 2.4. Regarding reasoning in $\mathcal{ALC}$, we have several types of reasons in $\mathcal{ALC}$ as follows:

- Instance checking: $\mathcal{K} \models C(a)$ or $\mathcal{K} \models r(a,b)$;
- Subsumption checking: $\mathcal{K} \models C \sqsubseteq D$;
- Equivalence checking: $\mathcal{K} \models C \equiv D$;

- Consistency (satisfiability) checking: $\mathcal{K} \not\models \top \sqsubseteq \bot$;
- Concept Satisfibility $\mathcal{K} \not\models C \sqsubseteq \bot$; and
- Disjoint Concepts: $\mathcal{K} \models C \sqcap D \sqsubseteq \bot$ or $\mathcal{K} \models C \sqsubseteq \neg D$.

where $\mathcal{K}$ is a knowledge base, C, D are two concepts, r is a role, and a, b are individual names. Here, the length of a concept C (denoted by $|C|$) is the number of sub-concepts of C (which is in the same order of size as the number of symbols in C). For a $TBox$ $\mathcal{T}$, $|\mathcal{T}|$ denotes the sum of all C, for C, the left or right-hand side of an inclusion axiom or a concept definition in $\mathcal{T}$.

Example 2.10 (Example for $\mathcal{ALC}$) *An example related to $Family$ to demonstrate about $TBox$ and $ABox$ for $\mathcal{ALC}$ is as follows:*

1. *$TBox$:*
   - $Person \sqsubseteq Animal \sqcap Biped$
   - $Woman \equiv Person \sqcap Female$
   - $Man \equiv Person \sqcap \neg Woman$
   - $Mother \equiv Woman \sqcap \exists ParentOf.Person$
   - $Father \equiv Man \sqcap \exists ParentOf.Person$
   - $Parent \equiv Mother \sqcup Father$
   - $GrandMother \equiv Woman \sqcap \exists ParentOf.Parent$
   - $MotherWithoutSon \equiv Mother \sqcap \forall ParentOf.\neg Son$
2. *$ABox$:*
   - $Woman(Marry), Woman(Alice), Man(David)$
   - $GrandMother(Alice)$
   - $(Person \sqcap Son)(John)$

Since we will present semantics in depth with another $\mathcal{ALC}$ fragment in the following section, we hence will not be discussing much semantics for $\mathcal{ALC}$. Instead, we simply provide a column in Table 2.4. In general, several fragments of $\mathcal{ALC}$, such as logic in the $\mathcal{FL}$, $\mathcal{EL}$, or $DL\text{-}Lite$ families, are well-understood. Next, we will provide the syntax and semantics of description logic $\mathcal{EL}$ in detail in the next section. This setting is the primary logic language for the contributions of this thesis.

2.3.3 Lightweight Description Logic $\mathcal{EL}$

$\mathcal{EL}$ is a family of lightweight DLs, which underlies the Ontology Web Language profile OWL2-EL, which is taken into account as one of the central representation formalisms to express terminological knowledge [BBL05, Sun09]. The main ingredients of DLs are individuals, concepts, and roles, which correspond at the semantic level to objects, sets of objects, and binary relations between objects. More formally, let N_C, N_R, N_I be three pairwise disjoint sets where N_C denotes a set of atomic concepts, N_R denotes a set of atomic relations (roles), and N_I denotes a set of individuals.

In this dissertation, we take account of $\mathcal{EL}$ concept expressions [Kri20] which are built according to the following grammar:

$$C ::= \quad \top \mid N_C \mid C \sqcap C \mid \exists r.C \text{ where } r \in N_R.$$

Let $C, D \in N_C$, $a, b \in N_I$, and $r \in N_R$. An $\mathcal{EL}$ ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ (a.k.a. knowledge base) comprises two components, the $TBox$ and $ABox$. The $TBox$ consists of a set of General Concept Inclusion (GCI) axioms of the form $C \sqsubseteq D$, meaning that C is more specific than D or simply C is subsumed by D. The $ABox$ is a finite set of assertions on individual objects of the form $C(a)$ or $r(a, b)$.

The semantics is given in terms of interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, which consist of a non-empty interpretation domain $\Delta^{\mathcal{I}}$ and an interpretation function $\cdot^{\mathcal{I}}$ that maps each individual $a \in N_I$ into an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, each concept $A \in N_C$ into a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, and each role $r \in N_R$ into a subset $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.

Syntax	Semantics
$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
r	$r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
a	$a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
$\top$	$\Delta^{\mathcal{I}}$
$\exists r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}} s.t. (x, y) \in r^{\mathcal{I}}, y \in C^{\mathcal{I}}\}$

Table 2.5: Syntax and semantics of $\mathcal{EL}$.

A summary of the syntax and semantics of $\mathcal{EL}$ is shown in Table 2.5. An interpretation $\mathcal{I}$ is said to be a model of (or satisfies) an axiom Φ in the form of the left column in the table, denoted by $\mathcal{I} \models \Phi$, when the corresponding condition in the right column is satisfied. For instance, $\mathcal{I} \models C \sqsubseteq D$ if and only if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. Similarly, $\mathcal{I}$ satisfies a concept (resp. role) assertion, denoted by $\mathcal{I} \models C(a)$ (resp. $\mathcal{I} \models r(a, b)$), if $a^{\mathcal{I}} \in C^{\mathcal{I}}$ (resp. $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$). An interpretation $\mathcal{I}$ is a model of an ontology $\mathcal{O}$ if it satisfies all the axioms and assertions in $\mathcal{O}$. An ontology is said to be consistent if it has a model. Otherwise, it is inconsistent.

An axiom Φ is entailed by an ontology, denoted by $\mathcal{O} \models \Phi$, if Φ is satisfied by every model of $\mathcal{O}$. We say that C is subsumed by D w.r.t. an ontology $\mathcal{O}$ iff $\mathcal{O} \models C \sqsubseteq D$. Similarly, we say that a is an instance of C w.r.t. $\mathcal{O}$ if and only if $\mathcal{O} \models C(a)$.

In this thesis, we assume that the input ontologies are provided in a specific normal form, to which we apply completion rules for classification. We assume that each source is in the *normal form*. We define it as follows:

Definition 2.16 (Normal form of $\mathcal{EL}$) *An $\mathcal{EL}$ $TBox$ is in* normal form *if all concept inclusions have one of the following forms:*

$$A \sqsubseteq B, A \sqcap B \sqsubseteq C, A \sqsubseteq \exists r.B, \exists r.A \sqsubseteq B$$

where $A, B \in N_C$.

Now, we will present the fragments of $\mathcal{EL}$ family in the following section.

2.3.4 The Extended $\mathcal{EL}$ Family

Description logic $\mathcal{EL}$ family [BOPP20, BBL05, BPS07] is the most simple version of the DL family. It is extended to be suitable for practical applications. From the first version of $\mathcal{EL}$, many authors extended it by adding complex role inclusion axioms. In particular, $\mathcal{EL}^+$ allows for conjunction $\sqcap$, existential restrictions $\exists$, and role inclusion axioms $r_1 \circ \ldots \circ r_n \sqsubseteq s$. Many real-world ontologies, including SNOMED CT [SPSW01, D$^+$06], the Gene Ontology [ABB$^+$00], and Galen [RNC$^+$94] expressed in

$\mathcal{EL}^+$, use languages from the $\mathcal{EL}$ family, which underlies the OWL 2 EL profile of the Semantic Web standard ontology language.

An $\mathcal{EL}^+$ ontology or $TBox$ is a finite set of GCI and role inclusion axioms, whose syntax is shown in the top-left table of Figure 2.3. Note that it also uses the name $\mathcal{HL}$ for the sub-language of $\mathcal{EL}$ that does not allow for existential restrictions. This name is motivated by the fact that GCIs involving $\mathcal{HL}$ concepts are proposition Horn clauses [FBB+09, CH85].

The next extended version of $\mathcal{EL}^+$ is the version of $\mathcal{EL}^+_\bot$ [BPS07] or $\mathcal{EL}_\bot$ [Kri20]. This approach is expanded with the *bottom concept* $\bot$. It allows us to express that the concepts are separate, i.e., $Man \sqcap Woman \sqsubseteq \bot$. Normally, this fragment can appear logical conflicts, i.e., $Book \sqcap Text \sqsubseteq \bot, Book \sqsubseteq Text$. The syntax and semantics of $\mathcal{EL}^+_\bot$ are shown in the bottom-left table of Figure 2.3. In the definition of the semantics RIs, the symbol $\circ$ denotes the composition of binary relations.

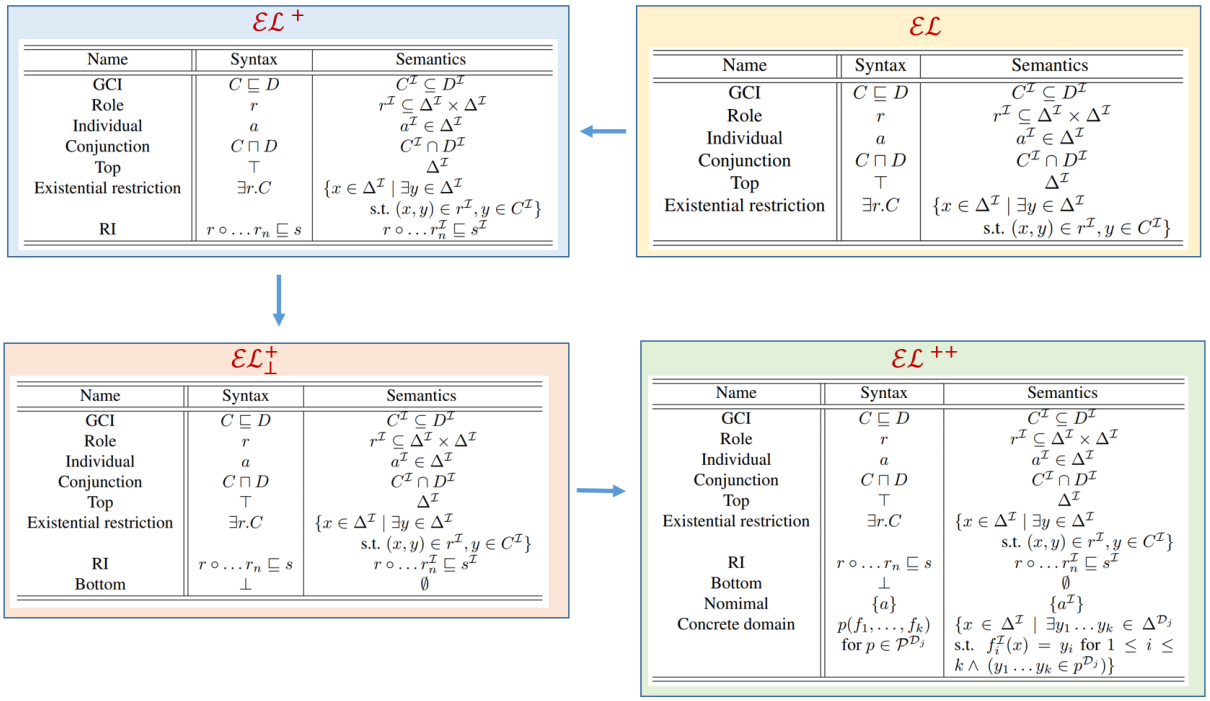


Figure 2.3: Family of description logic $\mathcal{EL}$.

At present, the full version of $\mathcal{EL}$ is $\mathcal{EL}^{++}$ [BBL05, BBL10]. It provides *concrete domains*, which permit reference to, e.g., strings, floats, and integers. Formally, a *concrete domain* $\mathcal{D}$ is a pair of $\langle \Delta^\mathcal{D}, \mathcal{P}^\mathcal{D} \rangle$, where $\Delta^\mathcal{D}$ is a domain and $\mathcal{P}^\mathcal{D}$ is a set of predicate names. Regarding the syntax and semantics of $\mathcal{EL}^{++}$, it is extended by adding two assertions, including *concept assertion* $C(a)$ and *role assertion* $r(a,b)$ shown in the bottom-right table of Figure 2.3. Instead of using $TBox$, $\mathcal{EL}^{++}$ introduces $CBox$, denoted by $\mathscr{C}$, which is a finite set of GCIs and RIs. The inference problems of DLs are defined as follows:

Definition 2.17 (Inference of $\mathcal{EL}^{++}$) *Let C, D be concepts, $\mathscr{C}$ be a $CBox$, and a be an individual. The inference problem is expressed as follows:*

- *C is* satisfiable *w.r.t $\mathscr{C}$ if $\mathcal{I} \models \mathscr{C}$ and $C^\mathcal{I} \neq \emptyset$;*
- *C* subsumes *D ($C \sqsubseteq_\mathscr{C} D$) w.r.t $\mathscr{C}$ if $C^\mathcal{I} \sqsubseteq D^\mathcal{I}$ and $\mathcal{I} \models \mathscr{C}$; and*

- a is an instance of C *in* $\mathcal{A}$ *w.r.t.* $\mathscr{C}$ *if* $\mathcal{A}$ *and* $\mathscr{C}$ *have a common model.*

Restricting the syntax to prevent conjunctions on the right and defining the semantics as usual in DLs seems to be the most natural way of handling provenance in DL languages with conjunctions. Since $\mathcal{EL}$ ontologies are often expressed in normal form [BLS05, BBL05, MET11], the main restriction in this language is the avoidance of qualified existential restrictions on the right-hand side. To normalize and complete an $\mathcal{EL}^{++}$ ontology, we first introduce the normal form of $\mathcal{EL}^{++}$.

Definition 2.18 (Normal form of $\mathcal{EL}^{++}$) *An $\mathcal{EL}^{++}$ $TBox$ is in* normal form *if all concept inclusions have one of the following forms:*

$$C \sqsubseteq D, C_1 \sqcap C_2 \sqsubseteq D, C \sqsubseteq \exists r.D, \exists r.C \sqsubseteq D$$

and all role inclusions are of the form

$$r \sqsubseteq s, r_1 \circ r_2 \sqsubseteq s$$

where $C, C_1, C_2 \in N_C$, $D \in N_C \cup \{\bot\}$.

Theorem 2.2 ([BBL05]) *Satisfiability, subsumption, $ABox$ consistency, and the instance problem in $\mathcal{EL}^{++}$ can be decided in polynomial time.*

Let $\mathcal{EL}^{\neg}$ be the extension of $\mathcal{EL}$ with negation.

Theorem 2.3 ([BBL05]) *In $\mathcal{EL}_{\bot}$ (or $\mathcal{EL}^{\neg}$), satisfiability and subsumption w.r.t. general $TBoxes$ is EXPTIME-complete.*

Lemma 2.1 ([BBL05]) *Subsumption w.r.t. CBoxes in $\mathcal{EL}^{++}$ can be reduced in linear time to subsumption w.r.t. normalized CBoxes in $\mathcal{EL}^{++}$.*

Now, we provide several normalization rules as follows:

Definition 2.19 (Normalization rules [BBL05, MET11]) *Let C, D, E be concepts. Normalization rules are as follows:*

NF_1: $C \equiv D \longrightarrow \{C \sqsubseteq D, D \sqsubseteq C\}$;

NF_2: $C \sqsubseteq D \longrightarrow \{C \sqsubseteq A, A \sqsubseteq D\}$;

NF_3: $B \sqsubseteq C \sqcap D \longrightarrow \{B \sqsubseteq C, B \sqsubseteq D\}$;

NF_4: $C \sqcap D \sqsubseteq E \longrightarrow \{D \sqsubseteq A, C \sqcap A \sqsubseteq E\}$;

NF_5: $\exists r.C \sqsubseteq D \longrightarrow C \sqsubseteq A, \exists r.A \sqsubseteq D$;

NF_6: $C \sqsubseteq \exists r.D \longrightarrow \{C \sqsubseteq \exists r.A, A \sqsubseteq D\}$; *and*

NF_7: $\exists r.C \sqsubseteq \exists s.D \longrightarrow \{\exists r.C \sqsubseteq A,\ A \sqsubseteq \exists s.D\}$.

A polynomial-time algorithm for deciding subsumption in $\mathcal{EL}^{++}$ in normal form is given in Baader et al.'s research [BBL05]. The main reasoning task taken into account in terminological ontologies is classification. It consists of computing all the entailed subsumptions (and equivalences) that hold between atomic concepts of an ontology and the concepts $\top$ or $\bot$. Such a procedure is described in Baader et al.'s work [BBL05], which first consists of transforming the ontology into a normal form using a set of rules and then performing a classification reasoning process using the set of inference (completion) rules (see [BBL05] for more details). The classification of $TBoxes$ is the computation of all subsumption relations between all named concepts of a $TBox$. For several extensions of $\mathcal{EL}$ classification can be performed in polynomial time [BBL05, BCM+07]. These classification algorithms typically proceed in three steps [MET11]:

1. Normalization of the $TBox$;
2. Applying completion rules to the completion graph; and
3. Reading off subsumptions relations from the saturated completion graph.

Some intractable extensions of the $\mathcal{EL}$ family and their complexities [HL08, KL07] are shown in Table 2.6. We select description logic $\mathcal{EL}$ to express the knowledge in this work because (1) its complexity is $NP\text{-}Time$, (2) it is suitable for practical ontologies, and (3) it is sufficient to represent the basic knowledge that exists in most practical ontologies. In the next section, we will discuss ontology languages that have some sort of connection to DL.

Extensions	Description	Complexity	
		Acyclic $TBoxes$	General $TBoxes$
$\mathcal{EL}$	A lightweight description logic $\mathcal{EL}$	in P	in $P\text{-}complete$
$\mathcal{EL}^{\neg A}$	$\mathcal{EL}$ and $\neg$ (atomic negation)	$coNP\text{-}Complete$	$coNP\text{-}Complete$
$\mathcal{EL}^{C \sqcup D}$	$\mathcal{EL}$ and $\cup$ (disjunction)	$coNP\text{-}Complete$	$coNP\text{-}Complete$
$\mathcal{ELI}^{f}$	$\mathcal{EL}$ and inverse roles and functional roles	in P	$P\text{-}Complete$
$\mathcal{EL}^{\exists r^{+}.C}$	$\mathcal{EL}$ and $\exists r^{+}.C$	$coNP\text{-}hard$	$coNP\text{-}hard$
$\mathcal{EL}^{\forall r.\perp}$, $\mathcal{EL}^{\forall r.C}$	$\mathcal{EL}$ and ($\forall r.\perp$ or $\forall r.C$)	$coNP\text{-}complete$	$coNP\text{-}complete$
$\mathcal{EL}^{\exists \neg r.C}$	$\mathcal{EL}$ and $\exists \neg r.C$	$coNP\text{-}hard$	$coNP\text{-}hard$
$\mathcal{EL}^{\leq kr}$	$\mathcal{EL}$ and $\leq kr$, fixed $k \geq 1$	$coNP\text{-}Complete$	$coNP\text{-}Complete$
$\mathcal{EL}^{\geq kr}$	$\mathcal{EL}$ and $\geq kr$, fixed $k \geq 2$	$coNP\text{-}Complete$	$coNP\text{-}Complete$

Table 2.6: Variants and complexities of the $\mathcal{EL}$ family.

2.4 Ontology Languages

The World Wide Web[3] (WWW for short) is a general term in the information technology field. Therein, the founders of the "Semantic Web" assert that the development of this aspect is revolutionary. Its purpose is to overcome an existing web restriction. The semantic web aims to develop tools that simplify the process for individuals to find knowledge using context instead of grammar alone. This chapter aims to provide an overview of several of the ontology languages that play a significant role in the semantic web and explain the formal underpinnings of these languages. A standardized, structured, and machine-readable conceptual schema is necessary to construct semantic web applications. Within the Semantic Web community, the dominating definition of an ontology is a formal explicit specification of a shared conceptualisation of a domain of interest [Gru93]. In other words, it is a machine-readable theory of the most fundamental concepts required to understand knowledge in any knowledge domain [Leg07]. In Gruber's view [Gru93], an ontology is the representation of the knowledge of a domain, where a set of concept/objects and their relationships is described by a vocabulary.

2.4.1 What are Ontology Languages?

Ontologies are recognized as one of the semantic web's basis technologies [Hor05, GPC02]. The semantic web community has developed a standardization of web ontology languages. An ontology language

[3] https://www.w3.org/.

is a way to describe, at an abstract level, what is true in the area of interest (a conceptual level). More precisely, an ontology language could express *constraints* over domains. The ontology includes the following components:

- **Concept:** It is also known as *class or entity or frame*. It denotes a collection of objects (e.g., $PhDStudent, Country$);
- **Relationship:** It is also known as *role or object property or association*. It expresses an association among concepts (e.g., $hasDance, isCapitalOf$);
- **Property:** It is also known as *data property or feature or attribute*. It represents an element of an ontology, namely a concept or a role. A property can be either atomic (i.e., integer, string) or structured (i.e., set, list);
- **Axiom**: is *statements* that they are $True$ in the area being explained. Normally, an axiom is a logical formula to represent constraints or rules (e.g., every $PhDStudent$ is a $Student$, Man and $Woman$ is disjoint);
- **Instances:** It is also known as *an individual* belonging to a concept (e.g., $PhDStudent(Thanh)$, it means that $Thanh$ is an instance of the $PhDStudent$ class); and
- **Facts:** It is known as *a relationship* between instances (e.g., $isCapitalOf(Paris, France)$, it means that $Paris$ is a capital of $France$).

To represent the ontology languages, let us take account of an example of courses and lecturers. We state "Thanh, is a lecturer, is teaching an Algorithm subject". In this sentence, there are many types of objects (i.e., Thanh, lecturer, course). In ontology languages, objects include classes (a.k.a. *concepts*) e.g., $lecturer$, $course$ and *individuals* (a.k.a. resources or instances), e.g., $Thanh$, $Algorithm$. In this context, a class may be viewed as a collection of objects known as the class's extension. Moreover, individuals that belong to a class are *instances* of that class. Moreover, an important point of classes is to *impose restrictions* on what can be stated. Now, we take account of two statements as follows: (1) $Algorithm$ is taught by $ComputerScience$, (2) $RoomD205$ is taught by $ThanhMa$.

Observantly, the first claim is non-sensible because a lecturer should only teach the courses. Therefore, it restricts the values of the object property "*is taught by*". Note that we restrict the *range* of the property in mathematical terms. In addition, the second statement is also non-sensible since $RoomD205$ is not a course. Again, this restricts the objects to which the property can be applied. In mathematical terms, we call this a restriction for the domain of the property. Furthermore, ontology structure is the hierarchy of class and properties.

- **Class Hierarchies**: Classes can establish relationships between them to create a graph (or tree) structure, i.e., A $professor$ is an $permanent\ staff$ of a laboratory;
- **Property Hierarchies**: A hierarchy of class relationships can be defined by *sub-properties*, i.e., $hasFather$ is a sub-property of $hasAncestor$.

Regarding the ontology languages, they are split into three main types of languages as follows: (1) **Graph-based languages:** Ontologies will be presented in the form of graphs or trees. They include: UML [GZK03], class diagrams [SY21], semantic networks [Leh92], conceptual graph [DKC05], and others; (2) **Frame-based languages:** Knowledge in ontologies will be expressed in the frame approach [KLW95, Gru95]. It consists of Open Knowledge Base Connectivity [CFF+98], Knowledge Machine[4]

[4] https://www.cs.utexas.edu/users/mfkb/RKF/km.html

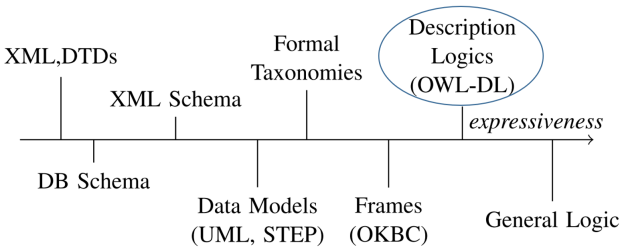


Figure 2.4: Kinds of ontology.

[CP03, KT18], others; (3) **Logic-based languages**: Ontologies will be represented underlying the foundation of logics. It can be First Order Logic [Smu68], Description logics [BBL05, BBL13], F-Logics [Fra96, BZ77], others. The different types [WLB12, Gru03] of ontologies are shown in Figure 2.4. Note that, in this thesis, we are interested in *logic-based languages* and OWL DL[5] [SPG+07, BVHH+04]. Namely, we use description logic as an ontology language. Next, we will present how to express knowledge in ontology, underlying the foundation of description logic..

2.4.2 Connection between Ontology Languages and Description Logics

One of the first proposals for a web-based ontology language was the simple HTML ontology extensions (SHOE for short [LSRH97, HH00]). It coincides with the development of RDF and RDFS [McB04, CWWM06]. SHOE allowed the semantic annotation of webpages, had an XML syntax, and was frame-based. Nevertheless, like the RDF schema, it did not have well-defined semantics. Thus, the significant point was that a web-based knowledge representation language had a standard inference. It is an important difference with highly expressive language, i.e., KIF [GL00]. Clearly, the language should strike a fine balance between the open nature and expansiveness of the web on the one hand and formal, well-defined semantics on the other. The DAML+OIL language of Connolly et al. ([MFSH03]) was a proposal for a semantic web knowledge representation language [Wel10, VHLP08]. It had a DL-style semantics and was a combination of two languages, including the *DARPA Agent Markup Language (for short DAML-ONT)* [MFSH03] and the *Ontology Inference Layer* [HFB+00]. DAML+OIL was the first knowledge representation language that combined the principles of knowledge sharing with the formal semantics and limited expressiveness of description logic, allowing for sound and complete reasoning [HS01, FVHH+01].

Furthermore, OWL [MVH+04, BVHH+04] extends the syntax and semantics of RDFS with a fair number of constructs and combines the description logic semantics of DAML+OIL with more rigorous semantic and syntactic compatibility regarding RDF and RDFS. A DAML+OIL ontology could be seen to correspond to a DL $TBox$ with a hierarchy, representing the domain in terms of *classes (corresponding to classes)* and properties (corresponding to roles). For example, to illustrate $Human \sqcap Female$, we can write it as follows:

```
<daml:Class>
    <daml:intersectionOf rdf:parseType="daml:collection">
    <daml:Class rdf:about="#Human"/>
    <daml:Class rdf:about="#Female"/>
    </daml:intersectionOf>
</daml:Class>
```

The syntax of DL and constructors of DAML+OIL are presented in Table 2.7.

[5] It will be presented in the next section

Constructors	DL Syntax	Example
$intersectionOf$	$C_1 \sqcap \ldots \sqcap C_n$	$Animal \sqcap Water$
$unionOf$	$C_1 \sqcup \ldots \sqcup C_n$	$Studio \sqcup Hotel$
$ComplementOf$	$\neg C$	$\neg Daughter$
$OneOf$	$\{x_1, \ldots, x_n\}$	$\{Thanh, Alexis\}$
$toClass$	$\forall r.C$	$\forall hasWork.Teacher$
$hasClass$	$\exists r.C$	$\exists hasCapital.Country$
$hasValue$	$\exists r.\{x\}$	$\exists hasCapital.\{France\}$
$minCardinalityQ$	$\geq nr.C$	$(\geq 2 hasChild.Teacher)$
$maxCardinalityQ$	$\leq nr.C$	$(\leq 3 hasNiece.Female)$
$inverseOf$	r^-	$hasParent^-$

Table 2.7: DAML+OIL constructors.

When taking an account of OWL [MVH+04, BVHH+04, Hor05], the various requirements for a web ontology language led to the specification of three *species*, each emphasising different language features: OWL Full, DL, and Lite [MVH+04]. It includes: (1) OWL Full is a semantic extension of RDFS that adds several knowledge representation primitives that were previously unavailable. It emphasizes expressiveness and compatibility with RDFS: the RDFS interpretation of an OWL full ontology is consistent with its OWL full entailment; (2) OWL DL is a syntactic subset of OWL Full but limits the semantics to the $\mathcal{SHOIN}(D)$ description logic. At the time, this language was the most expressive. Therefore, it was a decidable description logic for which efficient algorithms are known and use cases exist; and (3) OWL Lite's aim is to alleviate some of the problems of its bigger brother. It is a syntactic subset of OWL DL where the semantic designs are to be limited to the $\mathcal{SHIF}(D)$ description logic. Here, we have the following strict language inclusion relationship [GPC02, AFH05]:

$$\text{RDF/RDFS} \subset \text{OWL Lite} \subset \text{OWL DL} \subset \text{OWL FULL}$$

in which $\subset$ holds both on the syntactic and semantic sides.

In addition, the W3C developed OWL2 in 2007 as the OWL Web Ontology Language successor. One such language expands the fundamental OWL with some characteristics for which sufficient and efficient reasoning algorithms exist[6]. Efficient implementations of OWL2 were already available in standard reasoners such as $Pellet$ [SPG+07], $Fact++$ [TH06], $HermiT$ [GHM+14], $Racer$ [HM01].

The OWL 2 [HKP+09] includes several profiles. Some of these are well-known tractable subsets of the OWL 2 DL specification. The main motivation for providing these profiles is that many existing ontologies use only a particular subset of the language constructs available in DL. There are three types of OWL 2 Full, such as OWL 2 EL, OWL QL, and OWL 2 RL. A brief description, shown in Figure 2.5 [GHVD03], is as follows: (1) **OWL 2 EL**, the $\mathcal{EL}$ profile, based on the $\mathcal{EL}^{++}$[7] introduced by Baader et al. [BBL05], is an extension of the $\mathcal{EL}$ description logic. Its strength is the ability to reason in polytime on the ontologies with large $TBoxes$. We concentrate on this profile to merge the knowledge for this thesis; (2) **OWL 2 QL**, the QL profile of OWL 2 was developed to handle query answering on the ontologies, which contain a large number of $ABox$ assertions. Typically, $TBox$ of ontologies is uncomplicated. Therefore, the description logic of this profile is $DL\text{-}Lite$ [BK03] and (3) **OWL 2 RL**. The RL profile is based on so-called description logic programs [GHVD03], which is a subset of OWL DL 1.0 and the Horn profile of First Order Logic (FOL).

[6] See the website at http://www.w3.org/2007/OWL.

[7] presented in the above section.

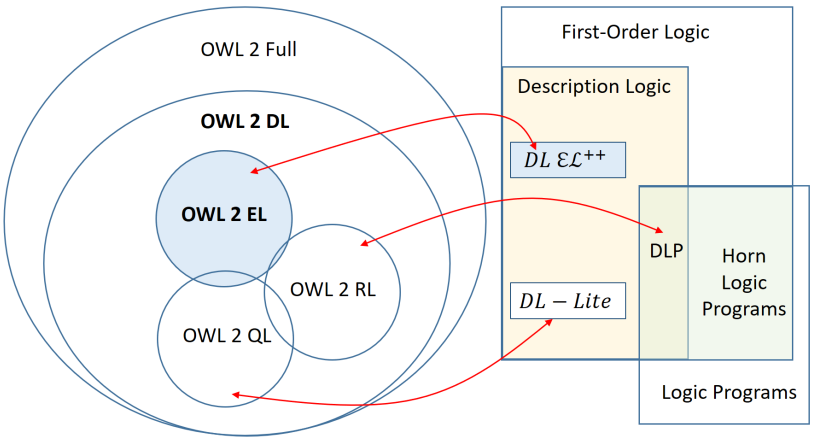


Figure 2.5: Family of the OWL 2 profile.

Example 2.11 *Now, we give some examples to illustrate abstract syntax of OWL*

1. *Cases of Class:*

```
Class(ex:Country partial owl:Thing)
Class(ex:Person partial owl:Thing)
Class(ex:Student partial owl:Person)
Class(ex:Vietnamese partial owl:Person)
       hasValue(ex:nationality ex:Vietnam)
```

2. *Cases of Data Property and Object Property:*

```
DatatypeProperty(ex:Age domain(ex:Person) range (xsd:integer))
DatatypeProperty(ex:Name domain(ex:Person) range (xsd:String))
ObjectProperty(ex:nationality domain(ex:Person) range (ex:Country))
ObjectProperty(ex:hasEthnicGroup domain(ex:Person) range (ex:EthnicGroup))
```

3. *Cases of Individual:*

```
Individual(ex:Vietnamese type(ex:Country))
Individual(ex:French type(ex:Country))
Individual(ex:Thanh type(ex:Vietnamese) value(ex:Age "32"^^xsd:integer))
```

A picture of the expressiveness[8] of the OWL languages and logic-based languages is presented in Figure 2.6. Let us provide examples of how simple ideas could be represented in First-Order Logic, Description Logic, and OWL.

Example 2.12 *Let us take account of two examples. First, we assume there is a statement as* "Electric Motors are Motors"*. Now, we can express them as follows:*

- *First-order logic:* $\forall x(ElectricMotor(x) \rightarrow Motor(x))$*;*
- *Description logic:* $ElectricMotor \sqsubseteq Motor$*; and*

[8] https://ontology-explained.com/2021/owl/

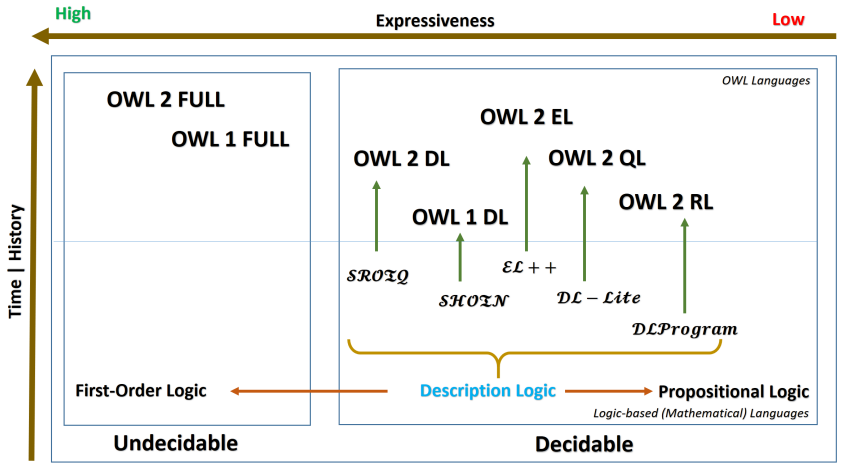


Figure 2.6: The expressiveness between logic-based languages.

- *OWL:*

```
<owl:Class rdf:about="http://example-thesisThanh#Motor"/>
<owl:Class rdf:about="http://example-thesisThanh#ElectricMotor"/>
  <rdfs:subClassOf rdf:resource="http://example-thesisThanh#Motor"/>
</owl:Class>
```

Second, we take into consideration that "The mother of the child's parents is the grandmother of the child"*. This claim can be expressed in the form of:*

- *First-order logic:* $\forall x, y(\exists z(motherOf(x, z) \wedge parentOf(z, y)) \rightarrow grandmotherOf(x, y))$*;*
- *Description logic :* $motherOf \circ parentOf \sqsubseteq grandmotherOf$*; and*
- *OWL:*

```
<owl:ObjectProperty rdf:about="http://example-thesisThanh#parentOf"/>
<owl:ObjectProperty rdf:about="http://example-thesisThanh#motherOf"/>
<owl:ObjectProperty rdf:about="http://example-thesisThanh#grandmotherOf"/>
  <owl:propertyChainAxiom rdf:parseType="Collection"/>
      <rdf:Description rdf:about="http://example-thesisThanh#motherOf"/>
      <rdf:Description rdf:about="http://example-thesisThanh#parentOf"/>
   </owl:propertyChainAxiom>
</owl:ObjectProperty>
```

OWL uses description-forming constructs in axioms that provide knowledge about classes, properties, and individuals. We present the abstract syntax of OWL and the DL syntax in Table 2.8. Therein, we denote A, dr, r, u, a, v are respectively class names (concept), data range, object properties (role), datatype properties (concrete role), individuals, and data values.

A formal semantics, familiar with the semantics provided for description logic, is provided for the style of using OWL. Full details on the theory can be found in the OWL semantics and Abstract syntax [PSHH04].

Furthermore, to develop applications related to ontologies, several tools and libraries are commonly used as follows: (1) **Protégé**[9]**:** It is a powerful tool to manipulate ontologies. In addition, the developer

[9] https://protege.stanford.edu/

Abstract syntax	DL syntax
$Class(A\ partial\ C_1, \ldots, C_n)$	$A \sqsubseteq C_1 \sqcap \ldots \sqcap C_n$
$Class(A\ complete\ C_1, \ldots, C_n)$	$A \equiv C_1 \sqcap \ldots \sqcap C_n$
$SubClassOf(C_1, C_2)$	$C_1 \sqsubseteq C_2$
$EquivalentClasses(C_1, \ldots, C_n)$	$C_1 \equiv \ldots \equiv C_n$
$DisjointClasses(C_1, \ldots C_n)$	$C_i \sqcap C_j \sqsubseteq \bot, i \neq j$
$ObjectProperty(r,\ super(r_1), \ldots, super(r_n))$	$r \sqsubseteq r_i, i \leq n$
$+domain(C_1), \ldots, domain(C_m)$	$\geq 1r \sqsubseteq C_1$
$+range(D_1), \ldots, range(D_t)$	$\top \sqsubseteq \forall r.D_1$
$DataTypeProperty(u,\ super(u_1), \ldots, super(u_n))$	$u \sqsubseteq u_i, i \leq n$
$+domain(C_1), \ldots, domain(C_n)$	$\geq 1u \sqsubseteq C_1$
$+range(D_1), \ldots, range(D_n)$	$\top \sqsubseteq \forall u.D_1$
$SubPropertyOf(r_1, r_2)$	$r_1 \sqsubseteq r_2$
$EquivalentProperies(r_1, \ldots, r_n)$	$r_1 \equiv \ldots \equiv r_n$
$Individual(a, type(C_1), \ldots, type(C_n))$	$a \in C_i$
$+value(r_1, a_1), \ldots, value(r_n, a_n)$	$(a_1, a_i) \in r$
$+value(u_1, v_1), \ldots, value(u_n, v_n)$	$(a_1, v_i) \in u$
$SameIndividual(o_1, \ldots, o_n)$	$\{o_1\} \equiv \ldots \equiv \{o_n\}$
$DifferentIndividuals(o_1, \ldots, o_n)$	$\{o_i\} \sqsubseteq \neg\{o_j\}, i \neq j$

Table 2.8: Abstract syntax and DL syntax of OWL.

provided a protege-OWL API, which is an open-source Java library for OWL and RDF(s); (2) **OWL API**[10]**:** It is a library written in Java and an API for OWL2. It provided a reasoner interface supported by many description logic reasoners; (3) **Jena API**[11]**:** It is a free and open-source Java framework for developing applications related to the Semantic Web and Linked Data. It provides an API for knowledge extraction and RDF graph manipulation; (4) **Owlready2**[12] It is a Python library for ontology-oriented programming. It could also load OWL 2.0 ontologies as Python objects, edit them, restore them, and reason about them using HermiT (included). Owlready2 enables direct OWL ontology access (contrary to the standard Java-based API). In this thesis, we use Owlready2 with practical ontologies since all experiments are implemented in the Python programming language.

2.5 Conclusion

In this part, we introduced the KRR, which is one of the solid foundations for expressing assertions, constraints, and pieces of knowledge. We have also located our research branch, background, and approach in this chapter. Specifically, we presented a complete overview from propositional logic to ontology languages and illustrative examples.

Our research crucially concentrates on the ontology language encoded underlying the family of the description logics, namely description logic $\mathcal{EL}$. The fact that our contributions are close to studies connected to proposition and first-order logic is an essential point to emphasize. In addition, some complexities of the $\mathcal{EL}$ family are also provided in this chapter. Moreover, we also provide a picture of ontology languages and a connection between ontology and logic languages. One point to note is

[10] http://owlapi.sourceforge.net/
[11] https://jena.apache.org/documentation/ontology/
[12] https://owlready2.readthedocs.io/en/v0.37/:

that the following sections will use the symbols and notions introduced in this chapter. Therefore, in order to simplify the presentation and ensure consistency, we refer to the same concepts, i.e., "$\mathcal{I}$" is an interpretation, and "$\mathcal{K}$" denotes a knowledge base.

We rely on logical languages' deductive and inductive strengths to propose and build our frameworks. In an actual application, constraints and assertions are usually imperfect and incomplete. In addition, most knowledge bases are represented by several distinct aspects, circumstances, and perspectives. Consequently, when we unify, update, and edit multiple pieces of knowledge, the logical representation of the point may have inconsistencies and conflicts. The following chapters will provide the strategies for addressing the issues and provide the necessary context for discovering and developing our solutions.

Chapter 3

Belief Change

Belief change is a topic of AI and knowledge science dedicated to information dynamics. Belief change can be understood as a process of comparing the multidimensional value of an old belief to that of a new belief and changing beliefs when the latter is more significant. The subject of how an agent's beliefs should change in response to new knowledge is a crucial challenge for agents and AI systems. Moreover, the belief change community has developed a novel and highly productive study strategy for this topic. The method is top-down in that it considers postulates but agent-centric in that it takes a subjective perspective on the phenomena of belief change [Neb98, FH11, DEKP20].

For this topic, we can take into account of an example as follows:

- *Thanh is applying dynamic reasoning to the fact that "chloroquine phosphate*[13]*" treats coronavirus (Covid). Initially, he refuses these facts entirely. Now, Thanh reads in the paper that a renowned French physician claims the truth of the issue. Given his skepticism, he is thereafter doubtful about the facts. Then, this assertion is neither accepted nor rejected. Since Thanh is cautious in nature, he requires further evidence before he can accept this assertion. After that, he read that Japanese physicians had successfully treated coronavirus using "chloroquine phosphate". He accepts this fact in the end.*

  During his confinement in Covid's period, Thanh has more time to read the news online, and he reads that the studies conducted so far have not yet determined whether a fact should be accepted or rejected. Hence, Thanh is uncertain and incredulous about this fact. Notably, if Thanh accepts this fact but is informed that it is wrong, he will not deny the facts and instead assume that the information is ambiguous and uncertain out of a sense of inertia.

Indeed, new knowledge will directly affect the current knowledge base. In some cases, it can appear contradictions and inconsistency [BBT15, HK08] when we update that novel knowledge. For example, most people say that birds can fly, but another source states the example of penguins, ostriches, and kiwis. This leads to conflicts and inconsistencies (Chapter 2.2). Therefore, several scientists paid much attention to this topic and proposed solutions to resolve the conflicts [KM91a, KPP13]. This chapter will provide formal definitions and methods for dealing with problems arising when we update or merge multiple sources of knowledge [KM02, KM04, HW19]. The notions, postulates, and operators presented in [KPP11] will be the basis for our contributions' inspiration.

[13] Chloroquine (also known as chloroquine phosphate) is an anti-malarial medicine

3.1 Background

The area of *belief change* is the study of how an agent may modify its beliefs in response to new information [FH18]. Common belief change operators are revision operators [KM91a, Gär88, AGM85, DP97, KPP01, KM91a, HPW07a], update operators [FUV83, KM91a], merging operators [KP98, KP02, KM05, HPW09]. The relation of belief merging, update and revision is shown in Figure 3.1 [KPP13]. In order to give an overview of these operators, we illustrate them by examples that operators can be supposed to model:

- *Update operators [FUV83, KM89, KM91b, DP97, CKP18]:* They aim to model the process of knowledge evolution in a change. Update operators can take account of point-wise revisions by looking at each model of the base instead of taking the base as a whole;

  Example 3.1 *Suppose we have a room with five books, $B_1, \ldots, B_5$. The books may be put on the table or not. We suppose there is a constraint that is either only B_1 is on the table or B_3, B_4, and B_5 are on the table. Then, a robot conducts the instruction (another constraint) to put all the books on the table or remove all the books from the table. For this case, if we use the belief update operators, we lead to the following belief base: all the books are on the table, or there are no books on the table.*

- *Revision operators [Gär88, AGM85, Pap92, CPPW14, HW19, SIK+19]*: They aim to model the process of accommodating a new piece of knowledge that is more reliable than the agent's current belief;

  Example 3.2 *Thanh and Peter are two close friends. Peter is an American volunteer teaching at the university where Thanh is a lecturer. Peter graduated in economics and taught marketing, accounting, and finance courses. Five years after Peter returned home, Thanh could not contact Peter and had no news. At that time, Thanh thought Peter would work at a bank or in corporate finance after returning to the US. After that, Thanh got acquainted with Tom, another volunteer teaching at Thanh's university. Surprisingly, Tom said Peter had a permanent position as a programmer at an IT company in California. For an observation, Tom provided a piece of new knowledge and revised Thanh's belief since Thanh thought that Peter would work in a company related to the economic field instead of as a programmer.*

- *Merging operators [Kon00, KP02, SPD06, HWP08, CPP12, CPRW16]* They aim to model the process of finding the beliefs associated with a group of agents. It is a collection of all beliefs from many sources. Merging operators can be considered extensions of revision operators to multiple belief bases.

  Example 3.3 *In a discussion about animals, Thanh said that crocodiles are terrestrial animals that often eat other animals. In contrast, Peter said crocodiles live in water, lay eggs, and have thick horny skin. If we merge these beliefs, it seems sensible to reach a joint agreement on the belief that crocodiles eat other animals, lay eggs, and have thick horny skin. However, there is some inconsistency about whether crocodiles are terrestrial or underwater animals. The belief merging operators are to solve these conflicts.*

A change operation uses a knowledge base (viewed as a belief base) and a piece of new knowledge (referred to as an input) to build a new knowledge base structure. Typically, a belief change issue may have several possible resolutions. Therefore, the specification of a change technique must adhere to constraints and provide postulates[14], which are primarily universal and others that may be purpose-

[14]Postulates are for characterizing the behaviors of an operator.

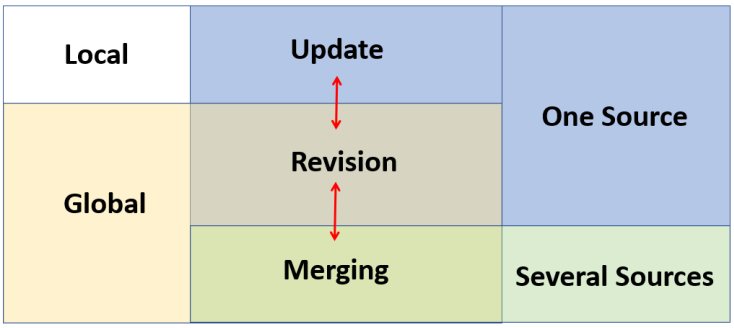


Figure 3.1: The relation of belief merging, update, and revision.

dependent. These principles will aid in selecting the appropriate knowledge and defining a family of "reasonable" belief transformation procedures. Several change principles are often seen as fundamental, as follows:

- *Minimal Change:* When accepting incoming data, it is vital to modify the original knowledge base as little as possible;
- *Consistency:* A change operation with a consistent input cannot yield an inconsistent output if it begins with a consistent knowledge base; and
- *Syntax irrelevance:* The development of a knowledge base should not have been contingent on its encoding.

Moreover, belief change theory aims to investigate and characterize the operators that allow an agent's beliefs to be modified in response to new knowledge. Such operators are required to design autonomous agents able to deal with unexpected problems. Revision and contraction operators are the most fundamental belief change operators. On the one hand, revision operators try to incorporate a formula (indicating new evidence) into an agent's beliefs while eliminating beliefs that contradict the additional formula. Belief change is represented at the knowledge level. The agents' beliefs are modeled by a set of formulas called *a belief set* or theory. Philosophers define a belief set as a collection of coherent, deductively closed logical propositions. A belief change operation bridges one belief set to another. Moreover, belief bases are not closed under logical consequence and are usually interpreted as basic beliefs from which additional beliefs (the belief set) can be derived. The AGM change operators are defined in terms of belief bases. Before introducing **belief revision** and **belief merging** in detail, we recall some general notions and provide formal definitions to use for the next sections.

3.1.1 Formal Preliminaries

First, we denote the set of all interpretations by $\mathcal{W}$. Given a formula α, $Mod(\alpha)$ denotes the set of all models of α. Particularly, $Mod(\alpha) \stackrel{\text{def}}{=} \{\mathcal{I} \in \mathcal{W} \mid \mathcal{I} \models \alpha\}$. We denote the classical consequence relation by $\models$. Moreover, a *belief base* $\mathcal{K}$ is a finite set of propositional formulae, which can be seen as the formula α, which is the conjunction of the formula of $\mathcal{K}$. The set of all belief bases is written by $\mathcal{P} = \{\mathcal{K}_1, \ldots, \mathcal{K}_n\}$. Note that the conjunction of the belief bases of $\mathcal{P}$ denote $\bigwedge \mathcal{P} = \{\mathcal{K}_1 \wedge \ldots \wedge \mathcal{K}_n\}$. The union of multi-set is denoted $\sqcup$, i.e., $\mathcal{K}_1 \sqcup \mathcal{K}_2$. Here, we say that $\mathcal{P}$ is consistent if and only if $\bigwedge \mathcal{P}$ is consistent. Moreover, we write $Mod(\mathcal{P})$ to denote $Mod(\bigwedge \mathcal{P})$ and say $\mathcal{I} \models \mathcal{P}$ for $\mathcal{I} \in Mod(\mathcal{P})$.

Here, we take account of a *belief* referred to as a piece of knowledge expressed in logical terms. A belief is a concept we regard as being true [Sch11, Arm73]. It is a declaration that something is real

or true, regardless of evidence or proof. When mentioning the *belief*, we clarify the difference between belief set and belief base. Namely, a belief set is a consistent, deductively closed set of logical sentences, while a belief base is a finite set of logical sentences. Since the belief set is "too big" and it is difficult and challenging to solve the problems explicitly on the practical side, the belief base will be more reasonable if we consider that this finite set can model an agent's knowledge. The elements of the belief set that are not part of the belief base are merely derived from the belief set, which is the logical closure of the belief base [Han98, DP98]. In general, for practical applications, we have to work on belief bases instead of belief sets because the implications of any given belief base may be an infinite (or an extremely large) set of propositions [FP01]. Remarkably, depending on the research community, researchers may use the terms "belief base" or "knowledge base" [QLB06, AABH18]. Since this study is concerned with ontology, we shall frequently refer to a knowledge base as a belief base. An equivalence of belief sets is $\mathcal{P}_1 \equiv \mathcal{P}_2$ if and only if $Mod(\mathcal{P}_1) = Mod(\mathcal{P}_2)$. Now, we provide several notions related to *order* and *pre-order* of belief sets as follows.

Definition 3.1 (Pre-order, order [Sch10]) *Let $\mathcal{K}$ be a belief set. A pre-order $\leq$ on $\mathcal{K}$ is a reflexive and transitive binary relation on $\mathcal{K}$. An order on $\mathcal{K}$ is an antisymmetric pre-order on $\mathcal{K}$. A strict order on $\mathcal{K}$ is an irreflexive and transitive binary relation on $\mathcal{K}$. The strict order $<$ induced by is defined $\forall x, y \in \mathcal{K}$ by $x < y$ if and only if $x \leq y$ and $y \not\leq x$.*

Furthermore, we also denote a binary relation $\leq$ as a pre-order if and only if it is a reflexive and transitive relation on $\mathcal{W}$. $\leq$ is *total* if, for all $\mathcal{I}, \mathcal{J} \in \mathcal{W}, \mathcal{I} \leq \mathcal{J}$ or $\mathcal{J} \leq \mathcal{I}$. We also denote $\mathcal{I} \equiv \mathcal{J}$ if and only if $\mathcal{I} \leq \mathcal{J}$ and $\mathcal{J} \leq \mathcal{I}$. Let $\mathcal{I}$ be an interpretation. We write $\mathcal{I} \in min(\mathcal{W}, \leq)$ if and only if $\mathcal{I} \in \mathcal{W}$ and $\forall \mathcal{J} \in \mathcal{W}, \mathcal{I} \leq \mathcal{J}$.

Definition 3.2 *Let $\mathcal{K}, \mathcal{K}'$ be two belief sets such that $\mathcal{K} \subseteq \mathcal{K}'$ and let $\leq$ be a total pre-order on $\mathcal{K}'$. We denote by $\min(\mathcal{K}, \leq)$ the set of elements of $\mathcal{K}$ which are minimal for the total pre-order $\leq$. Formally, we define as follows:*

$$min(\mathcal{K}, \leq) = \{x \in \mathcal{K} \mid \nexists y \in \mathcal{K}, y < x\}.$$

We will provide definitions of computing the distance between interpretations in the next sections. Now, we first present an aggregation function to support computing between models and belief bases.

3.1.2 Notions of Aggregation Function

For computing distance, an aggregation function that takes the distances between the models and the belief bases is calculated. It also aggregates the measures into a collective distance. Here, we write $\mathbb{R}$ as the set of real numbers. $\mathbb{R}_0^+$ denotes the set of positive real numbers, i.e., the set $\{h \in \mathbb{R} \mid h \geq 0\}$. Typically, an aggregation function [KM04, KP02] associates a value representing a list of numerical values according to certain criteria.

Definition 3.3 ([KM04]) *An aggregation function f associates a vector of positive real numbers with a positive real number and satisfies the following properties:*

$$\begin{cases} \text{if } x_1 \leq x'_1, \ldots, x_p \leq x'_p, \text{ then } f(x_1, \ldots, x_p) \leq f(x'_1, \ldots x'_p) & \text{(monotony)} \\ x_1 = \ldots = x_p = 0 \text{ if and only if } f(x_1, \ldots, x_p) = 0 & \text{(minimality)} \end{cases}$$

We present some additional properties for aggregate functions.

Definition 3.4 (Properties of the aggregation functions) *Let f and g be two aggregation functions. We say*

1. *f is* associative *if and only if* $f(f(x_1,\ldots,x_p),f(y_1,\ldots,y_p)) = f(x_1,\ldots,x_p,y_1,\ldots,y_q)$*;*
2. *f is* symmetric *if and only if for all permutation t from $\mathbb{R}^p$ to $\mathbb{R}^p$, p being a strictly natural integer positive, and* $f(x_1,\ldots,x_p) = f(t(x_1),\ldots,t(x_p))$*; and*
3. *f is strictly monotony if and only if it satisfies the condition as follows: if* $x_1 \leq x'_1,\ldots,x_p \leq x'_p$ *and* $\exists i, x_i \leq x'_i$*, then* $f(x_1,\ldots,x_p) < f(x'_1,\ldots,x'_p)$*.*

Several operators, i.e., revision operators and merging operators, have been suggested based on these pre-orders of interpretations [QD09, NPP03, CPPW14, HKP03, KP02, KPP11]. Therein, the majority of these techniques use the idea of "*distance*" between any interpretation and the belief base. Given that these operators are defined in a manner that is not dissimilar from that of the revision operator, which makes use of the *Hamming distance* [BKR02, Dal88a] between interpretations. Now, we formally define them as follows:

Definition 3.5 (Hamming Distance [Dal88a]) *Let $\mathcal{I},\mathcal{J}$ be two interpretations and S be a finite set of propositional variables (c.f. Section 2.2). The Hamming distance between $\mathcal{I}$ and $\mathcal{J}$, denoted by $dist_H(\mathcal{I},\mathcal{J})$, is the number of propositional letters that differ. The distance between $\mathcal{I}$ and $\mathcal{J}$ is defined as follows:*

$$dist_H(\mathcal{I},\mathcal{J}) = \mid \{x \in S \mid \mathcal{I}(x) \neq \mathcal{J}(x)\} \mid$$

We suppose that a distance between interpretations is a function $dist_H : \mathcal{W} \times \mathcal{W} \to \mathbb{N}$, where $\mathbb{N}$ is the set of natural numbers such that it satisfies the expected properties:

- $dist_H(\mathcal{I},\mathcal{J}) = dist_H(\mathcal{J},\mathcal{I})$;
- $dist_H(\mathcal{I},\mathcal{J}) = 0$ if $\mathcal{I} = \mathcal{J}$.

3.2 Belief Revision

Belief revision has been a research branch in several domains, including philosophy and artificial intelligence, for the last decades [FS19, Pap92, Pap01]. It concentrates on studying the epistemic change process when a new piece of knowledge has just been added. If the new knowledge is inconsistent with the original set, the new set has to be *revised*.

When mentioning belief revision, we have to refer directly to a set of postulates for the revision operator, called the AGM Postulates, produced by Alchourron, Gardenfors, and Makinson [Gär88, AGM85]. These postulates express rationality conditions that any revision operator has to satisfy. In the AGM framework [HKP03], epistemic states are closed under logical consequence (Cn) and are referred to as *belief sets*.

3.2.1 Belief Revision within Propositional Logic

Given two formulas, α, and β, we denoted $\alpha \circ \beta$ as the new belief base obtained by revising α with the new formula β. Two formulas, α and β, are equivalent ($\alpha \equiv \beta$) iff $Mod(\alpha) \equiv Mod(\beta)$. For a set ψ of formulas, $Cn(\psi)$ denotes the closures of ψ under the consequence operation (Cn). We denote that $\psi + \alpha = Cn(\psi \cup \{\alpha\})$ (expansion). In the AGM paradigm, the underlying logic is assumed to be classical propositional logic, and the beliefs are modeled by a theory called the belief set. Let ψ and α be formulas, revision of ψ by α is looking for the models of α the closest as possible of models of ψ. A belief set (or a theory ψ) is such that $\psi = \{\phi \mid \psi \models \phi\}$. Now, we provide some formal definitions [Pap00, Gär88, GM88, AGM85] as follows:

Definition 3.6 (Basic revision AGM postulates [Gär88]) *Let $\alpha, \beta \in L$. We denote an AGM revision operator $*$ as a function mapping a belief set ψ and a formula α to a new belief set $\psi * \alpha$, which satisfies the following properties:*

P_1: $\psi * \alpha = Cn(\psi * \alpha)$ *(closure);*

P_2: $\alpha \in \psi * \alpha$ *(success);*

P_3: $\psi * \alpha \subseteq \psi + \alpha$ *(inclusion);*

P_4: *If* $\neg\alpha \notin \psi$ *then* $\psi * \alpha = \psi + \alpha$ *(closure);*

P_5: $\psi * \alpha = L$ *only if* α *is unsatisfiable ((consistency preservation)) (syntax independence);*

P_6: *If* $\alpha \equiv \beta$ *then* $\psi * \alpha \equiv \psi * \beta$ *;*

P_7: $\psi * (\alpha \wedge \beta) \subseteq (\psi * \alpha) + \beta$ *(conjunctive inclusion); and*

P_8: *If* $\neg\beta \notin \psi * \alpha$ *then* $(\psi * \alpha) + \beta = \psi * (\alpha \wedge \beta)$ *(conjunctive vacuity).*

where $\psi + \alpha$ is the smallest deductively closed set of formulas containing both $\mathcal{K}$ and α and $\subseteq$ denotes the usual set containment.

The posulates is the following: P_1 represent that the result of revision is a theory and also is a closure. P_2 tells that the new information α is true in the new theory. P_3 says that revising by new information cannot add any belief that is not a consequence (Cn) of the new information and the theory. P_3 and P_4 state that when the new information does not contradict the initial theory, then the result of revision go down to the expansion ($+$). P_5 states that the only way to get an inconsistent theory by revision is to revise by the an inconsistent information. P_6 expresses the result of revision has to be indefedent from the syntactic encoding of the new information. P_7 and P_8 revising by the conjunction of two pieces of inforamtion amounts to a revision by the first one and an expansion by the second one whenever possible. While the postulates from P_1 to P_6 are the basic of revision operators, P_7 and P_8 are additional postulates.

Moreover, in order to characterize different proposed semantic operators, Katsuno and Mendelzon [KM91a] reformulated the AGM postulates (which are an adaptation of the AGM postulates to the finite case, i.e., to belief bases) as follows:

Definition 3.7 ([KM91a]) *Let ψ be a belief base and β be a new piece of belief (or a sentence). $\psi \circ \beta$ is a revision of ψ by β. An operator $\circ$ is said to be a KM revision operator if it satisfies the following postulates:*

R_1: $\psi \circ \beta \models \beta$;

R_2: *If* $\psi \wedge \beta$ *is consistent then* $\psi \circ \beta \equiv \psi \wedge \beta$;

R_3: *If* β *is consistent then* $\psi \circ \beta$ *is also consistent;*

R_4: *If* $\psi_1 \equiv \psi_2$ *and* $\beta_1 \equiv \beta_2$ *then* $\psi_1 \circ \beta_1 \equiv \psi_2 \circ \beta_2$;

R_5: $(\psi \circ \beta) \wedge \beta' \models \psi \circ (\beta \wedge \beta')$; *and*

R_6: *If* $(\psi \circ \beta) \wedge \beta'$ *is consistent then* $\psi \circ (\beta \wedge \phi) \models (\psi \circ \beta') \wedge \beta'$.

Therein, R_1 is a simple success criterion, stating that the revised belief set should entail the new knowledge β. R_2 requires that when the new knowledge does not contradict the initial beliefs, the revision should be performed by simply conjoining the new knowledge to the initial KB. R_3 requires a revised set of beliefs to be consistent if new knowledge is coming. R_4 says revision results should be independent of the syntax for both the KB and the new knowledge. Finally, R_5 and R_6 characterize the concept of minimal change relative to the set of models of ψ. However, they are much harder to motivate intuitively.

Definition 3.8 *Let ψ be a belief base. A* revision assignment *(denoted $\psi \mapsto \leq_\psi$) is a function which assigns to each ψ to a pre-order $\leq_\psi$ on interpretations $\mathcal{W}$. The revision assignments are faithful if and only if*

1. $\forall \varphi, \varphi' \in \mathcal{W},\ if\ \varphi, \varphi' \in Mod(\psi)\ then\ \varphi \simeq_\psi \varphi'$;
2. $\forall \varphi, \varphi' \in \mathcal{W},\ if\ \varphi \in Mod(\psi)\ and\ \varphi' \notin Mod(\psi)\ then\ \varphi <_\psi \varphi'$; *and*
3. $If\ Mod(\psi) = Mod(\psi')$ *then* $\leq_\psi = \leq_{\psi'}$.

Proposition 3.1 ([KM91a]) *Let ψ be a belief base and β be a new piece of belief. A revision operator $\circ$ satisfies conditions $R_1 - R_6$ iff there exists a faithful revision assignment that maps each ψ to a total pre-order $\leq_\psi$ such that:*

$$Mod(\psi \circ \beta) \stackrel{\text{def}}{=} \min(Mod(\beta), \leq_\psi)$$

3.2.2 Belief Revision in Description Logic

Applying AGM theory to description logic might be traced back to Nebel et al.'s research published in 1990 [Neb90, Neb89]. They mentioned five principles for belief base revision in a logical language. Moreover, they argued that the revision of a $TBox$ should only be applied to the definition of an atomic concept. Then, revision operators were defined to revise concept definitions that satisfied all the principles. From these perspectives, many pieces of research are related to applying belief revision theory to handle inconsistencies in ontology. For example, some works [FHP+06] analyzed the feasibility of applying AGM postulates for contraction to DLs. Their work concentrated on the *coherenct model*. In the basic AGM approach, a KB is a set of propositions, denoted by $\mathcal{K}_P$, closed under the operation Cn of logical consequence ($\mathcal{K}_P = Cn(\mathcal{K}_P)$). The operation of contraction can be formulated as a function mapping the pair $(\mathcal{K}_P, x)$ with $x \in L$ to a new KB $\mathcal{K}'_P$, denoted by $\mathcal{K}'_P = \mathcal{K}_P - x$. They reformulate AGM postulates into general logic. Now, we denote $-$ as a contraction operation, referring to the consistent removal of a piece of knowledge from a belief base.

Definition 3.9 ([FPA06]) *Let $\mathcal{K}_P$ be a belief set and $A_P, B_P \subseteq L$. The basic AGM postulates for contraction are as follows:*

K_1: *(Closure):* $Cn(\psi_P - A_P) = \mathcal{K}_P - A_P$;

K_2: *(Inclusion):* $\mathcal{K}_P - A_P \subseteq \mathcal{K}_P$;

K_3: *(Vacuity): If* $A_P \not\subseteq Cn(K_P)$, *then* $\mathcal{K}_P - A_P = \mathcal{K}$;

K_4: *(Success): If* $A_P \not\subseteq Cn(\emptyset)$, *then* $A_P \not\subseteq Cn(\mathcal{K}_P - A_P)$;

K_5: *(Preservation): If* $Cn(A_P) = Cn(B_P)$, *then* $\mathcal{K}_P - A_P = \mathcal{K}_P - B_P$; *and*

K_6: *(Recovery):* $\mathcal{K}_P \subseteq Cn((\mathcal{K}_P - A_P) \cup A_P)$.

Moreover, Qi et al. [QLB06] proposed a set of postulates for knowledge base revision in DLs based on reformulated AGM postulates. They defined a revision operator as a mapping from a pair of DL knowledge bases to a *disjunctive DL knowledge base*. A *disjunctive DL knowledge base* is denoted by $\mathcal{K}$. An interpretation is a model of $\mathcal{K}$ if and only if it is a model of DL knowledge bases. We use $Mod(\mathcal{K})$ to denote the set of all models of $\mathcal{K}$.

Definition 3.10 ([QLB06]) *Let $\mathcal{K}$ and $\mathcal{K}'$ be two knowledge bases. A set of postulates for knowledge base revision in DLs is as follows:*

G_1: $Mod(\mathcal{K} \circ \mathcal{K}') \subseteq Mod(\Phi)$ *for all* $\Phi \in \mathcal{K}'$;

G_2: *If* $Mod(\mathcal{K}) \cap Mod(\mathcal{K}') \neq \emptyset$, $Mod(\mathcal{K} \circ \mathcal{K}') = Mod(\mathcal{K}) \cap Mod(K')$;

G_3: *If* $\mathcal{K}'$ *is consistent, then* $Mod(\mathcal{K} \circ \mathcal{K}') \neq \emptyset$;

G_4: *If* $Mod(\mathcal{K}) = Mod(\mathcal{K}_1)$ *and* $Mod(\mathcal{K}') = Mod(\mathcal{K}_2)$, *then* $Mod(\mathcal{K} \circ \mathcal{K}') = Mod(\mathcal{K}_1 \circ \mathcal{K}_2)$;

G_5: $Mod(\mathcal{K} \circ \mathcal{K}_1) \cap Mod(\mathcal{K}_2) \subseteq Mod(\mathcal{K} \circ (\mathcal{K}_1 \cup \mathcal{K}_2))$; *and*

G_6: *If* $Mod(\mathcal{K} \circ \mathcal{K}_1) \cap Mod(\mathcal{K}_2)$ *is not empty, then* $Mod(\mathcal{K} \circ (\mathcal{K}_1 \cup \mathcal{K}_2)) \subseteq Mod(\mathcal{K} \circ \mathcal{K}_1) \cap Mod(\mathcal{K}_2)$.

Now, we show a formal definition of a faithful assignment.

Definition 3.11 ([QLB06]) *Let $\mathcal{K}$ and $\mathcal{K}'$ be DL knowledge bases and $\mathcal{W}$ be the set of all interpretations in DL, a total pre-order $\leq_{\mathcal{K}}$ on $\mathcal{W}$ is said to be a faithful assignment if the following conditions hold:*

1. *if $\mathcal{I}, \mathcal{I}' \models \mathcal{K}$, then $\mathcal{I} \leq \mathcal{I}'$ does not hold;*
2. *$\mathcal{I} \models \mathcal{K}$ and $\mathcal{I}' \not\models \mathcal{K}$, then $\mathcal{I} \leq_{\mathcal{K}} \mathcal{I}'$ holds; and*
3. *$\mathcal{K} \equiv \mathcal{K}'$, then $\leq_{\mathcal{K}} \equiv \leq_{\mathcal{K}'}$.*

Now, a formal proposition of the correspondence between the set of postulates and the faithful assignment.

Proposition 3.2 ([QLB06]) *A revision operator $\circ$ of DL knowledge base satisfies the postulates $G_1 - G_6$ iff for any DL belief base $\mathcal{K}$, there exists a faithful assignment $\leq_{\mathcal{K}}$ such that*

$$Mod(\mathcal{K} \circ \mathcal{K}') = min(Mod(\mathcal{K}'), \leq_{\mathcal{K}})$$

Moreover, several revision operators and postulates have proposed, such as [QLB06, QD09, WWT10, KPP13, CdLKM16, SIK+19].

3.3 Belief Merging

An approach close to the belief revision aspect is *belief merging*. However, the belief merging operator and the belief revision operator are different. An important point to distinguish between them [DEKP20, GRP09] is that, regarding belief revision, an agent is confronted with selecting among many alternatives to restore consistency. In contrast, belief merging recognizes the beliefs of a group from the individual members' assertions, which are not necessarily jointly consistent. When the claims are contradictory, these beliefs conflict and disagree. Alternatively stated, belief merging seeks to merge knowledge from

numerous sources. In contrast, belief revision research involves revoking knowledge to maintain consistency (some new pieces of evidence added to a belief base make it inconsistent). Belief merging operators have been defined as an extension of AGM belief revision theory [AGM85, KM91a, EKM15].

This section provides a logical definition of belief merging operators with integrity constraints (written IC). In particular, postulates of belief merging operators will be presented. These operators have to satisfy them to have good behavior concerning belief merging. Now, let us denote μ by a finite set of formulae of integrity constraints, $\mathcal{CB}$ by the set of the *consistent* belief set, and $\mathcal{P}$ by a profile. We consider operators $\Delta : \mathcal{P} \times \mathcal{CB} \leftrightarrow \mathcal{CB}$ with the integrity constraints written by $\Delta_\mu(\mathcal{P})$ instead of $\Delta(\mathcal{P}, \mu)$. A simple way to merge a profile $\mathcal{P}$ under the integrity constraints μ is the following [KP99]:

Definition 3.12 (IC [KP99, KP02]) *Let Δ be an IC merging operator and μ be an integrity constraint. The postulates that Δ satisfies are defined as follows:*

IC_1: $\Delta_\mu(\mathcal{P}) \models \mu$;

IC_2: *If μ is consistent, then* $\Delta_\mu(\mathcal{P})$ *is consistent;*

IC_3: *If* $\bigwedge \mathcal{P}$ *is consistent with* μ, *then* $\Delta_\mu(\mathcal{P}) = \bigwedge \mathcal{P} \wedge \mu$;

IC_4: *if* $\mathcal{P}_1 \equiv \mathcal{P}_2$ *and* $\mu_1 \equiv \mu_2$, *then* $\Delta_{\mu_1}(\mathcal{P}_1) \equiv \Delta_{\mu_1}(\mathcal{P}_1)$;

IC_5: *If* $\mathcal{K}_1 \models \mu$ *and* $\mathcal{K}_2 \models \mu$, *then* $\Delta_\mu(\mathcal{K}_1 \sqcup \mathcal{K}_2) \wedge \mathcal{K}_1 \not\models \bot \leftrightarrow \Delta_\mu(\mathcal{K}_1 \sqcup \mathcal{K}_2) \wedge \mathcal{K}_2 \not\models \bot$;

IC_6: *If* $\Delta_\mu(\mathcal{P}_1) \wedge \Delta_\mu(\mathcal{P}_2) \models \Delta_\mu(\mathcal{P}_1 \sqcup \mathcal{P}_2)$;

IC_7: *If* $\Delta_\mu(\mathcal{P}_1) \wedge \Delta_\mu(\mathcal{P}_2)$ *is consistent, then* $\Delta_\mu(\mathcal{P}_1 \sqcup \mathcal{P}_2) \models \Delta_\mu(\mathcal{P}_1) \wedge \Delta_\mu(\mathcal{P}_2)$;

IC_8: $\Delta_{\mu_1}(\mathcal{P}) \wedge \mu_2 \models \Delta_{\mu_1 \wedge \mu_2}(\mathcal{P})$;

IC_9: *If* $\Delta_{\mu_1}(\mathcal{P}) \wedge \mu_2$ *is consistent, then* $\Delta_{\mu_1 \wedge \mu_2}(\mathcal{P}) \models \Delta_{\mu_1}(\mathcal{P}) \wedge \mu_2$.

Now, we describe postulates for the IC merging operator. We can say that $\Delta_\mu(\mathcal{P})$ is a belief base close to the belief set $\mathcal{P}$ satisfying μ. The meaning of the postulates is as follows: (IC_1) gives that the result of the merging operator satisfies the integrity constraint μ. (IC_2) tells if the integrity constraints are consistent, the result of the merging operator will also be consistent. (IC_3) states that, if possible, the result of the merging is simply the conjunction of the belief bases with the integrity constraints. (IC_4) says if two belief bases are equivalent and two integrity constraint bases are also equivalent, the result of two mergings will be logically equivalent. (IC_5) is the fairness postulate. When merging two belief bases, merging operators must not give preference to one of them. (IC_6) represents if two groups $\mathcal{P}_1$ and $\mathcal{P}_2$ agree on some alternatives, these alternatives will be selected if we join those two groups. Note that (IC_5) and (IC_6) have a statement together that if one could find two sub-groups that agree on at least one alternative, the result of the global merging is those alternatives that two groups agree on. Finally, (IC_7) and (IC_8) are direct generalizations of the postulates (R_5) and (R_6) for revision.

Moreover, they also provided merging operators with majority and arbitration. They are defined as follows:

Definition 3.13 (Majority, Arbitration, Independent Majority (MI), Weakened MI (WMI)) *The postulates of majority [BM91] and arbitration [PM98] are as follows:*

Maj: $\exists n \Delta_\mu(\mathcal{P}_1 \sqcup \mathcal{P}_\mu^n) \models \Delta_\mu(\mathcal{P}_2)$;

$$Arb: \left.\begin{array}{l} \Delta_{\mu_1}(\mathcal{K}_1) \equiv \Delta_{\mu_2}(\mathcal{K}_2) \\ \Delta_{\mu_1 \equiv \neg\mu_2(\mathcal{K}_1 \sqcup \mathcal{K}_2)} \equiv (\mu_1 \equiv \neg\mu_2) \\ \mu_1 \not\models \mu_2 \\ \mu_2 \not\models \mu_1 \end{array}\right\} \Rightarrow \Delta_{\mu_1 \vee \mu_2}(\mathcal{K}_1 \sqcup \mathcal{K}_2) \equiv \Delta_1(\mathcal{K}_1).$$

In particular, (Maj) expresses the fact that for any profile $\mathcal{P}$, the same occurrence of $\mathcal{P}$ can appear enough times in a profile for all the models of the result of the fusion of this profile to be models of $\mathcal{P}$. In other words, the result of the merger represents the opinion of the majority of the group. (Arb) is more intuitively explained in terms of pre-orders.

Based on some of these additional postulates, Konieczny and Pino [KP99, Kon00] proposed three classes of merging operators:

Definition 3.14 (Majority Merging Operator) *A majority merging operator is an operator satisfying the postulates* (IC_1) - (IC_9) *and* (Maj).

Definition 3.15 (Arbitration Merging Operator) *An arbitration merging operator is an operator satisfying the postulates* (IC_1) - (IC_9) *and* (Arb).

Definition 3.16 (Quasi-Merging Operator) *An Quasi-Merging merging operator is an operator satisfying the postulates* (IC_1) - (IC_6), (IC_8), (IC_9), *and a weaken version of* IC_6 *as follows:*

$$IC_6' : \text{ if } \Delta_\mu(\mathcal{P}_1) \wedge \Delta_\mu(\mathcal{P}_2) \text{ is consistent, then } \Delta_\mu(\mathcal{P}_1 \sqcup \mathcal{P}_2) \models \Delta_\mu(\mathcal{P}_1) \vee \Delta_\mu(\mathcal{P}_2).$$

Here, we also introduce the notion of *syncretic assignment*, which is the foundation of belief merging operators.

Definition 3.17 ($\leq_\mathcal{P}$ for proposition logic) *A syncretic assignment is a function mapping each belief set* $\mathcal{P}$ *to a total pre-oder* $\leq_\mathcal{P}$ *over interpretations such that* $\forall\mathcal{P}, \mathcal{P}_1, \mathcal{P}_2$ *and* $\forall\mathcal{K}_1\, \mathcal{K}_2$, *the following conditions hold:*

1. *If* $\mathcal{I} \models \mathcal{P}$ *and* $\mathcal{J} \models \mathcal{P}$, *then* $\mathcal{I} \leq_\mathcal{P} \mathcal{J}$;
2. *If* $\mathcal{I} \models \mathcal{P}$ *and* $\mathcal{J} \not\models \mathcal{P}$, *then* $\mathcal{I} <_\mathcal{P} \mathcal{J}$;
3. *If* $\mathcal{P}_1 \equiv \mathcal{P}_2$, *then* $\leq_{\mathcal{P}_1} = \leq_{\mathcal{P}_2}$;
4. $\forall\mathcal{I} \models \mathcal{K}_1, \exists\mathcal{J} \models \mathcal{K}_2, \mathcal{J} \leq_{\mathcal{K}_1 \sqcup \mathcal{K}_2} \mathcal{I}$;
5. *If* $\mathcal{I} \leq_{\mathcal{P}_1} \mathcal{J}$ *and* $\mathcal{I} \leq_{\mathcal{P}_2} \mathcal{J}$, *then* $\mathcal{I} \leq_{\mathcal{P}_1 \sqcup \mathcal{P}_2} \mathcal{J}$;
6. *If* $\mathcal{I} <_{\mathcal{P}_1} \mathcal{J}$ *and* $\mathcal{I} \leq_{\mathcal{P}_2} \mathcal{J}$, *then* $\mathcal{I} <_{\mathcal{P}_1 \sqcup \mathcal{P}_2} \mathcal{J}$;
7. *If* $\mathcal{I} <_{\mathcal{P}_2} \mathcal{J}$, *then* $\exists I <_{\mathcal{P}_1 \sqcup \mathcal{P}_2^n} \mathcal{J}$ *(a majority syncretic assignment); and*
8. $\left.\begin{array}{l} \mathcal{I} <_{\mathcal{K}_1} \mathcal{J} \\ \mathcal{I} <_{\mathcal{K}_2} \mathcal{J}' \\ \mathcal{J} \equiv_{\mathcal{K}_1 \sqcup \mathcal{K}_2} \mathcal{J}' \end{array}\right\} \Rightarrow \mathcal{I} <_{\mathcal{K}_1 \cup \mathcal{K}_2} \mathcal{J}$(*a fair syncretic assignment).*

Here, the two initial conditions guarantee that the models of the knowledge base set correspond to the interpretations for the related pre-order $\leq$. The third requirement is that two equal knowledge base sets ($\equiv$) have similar pre-orders. Finally, according to the fourth requirement, there must be a model of the second belief basis that is at least as good as the first one when the two belief bases are merged. The

fifth condition says that if $\mathcal{I}$ is at least as plausible as $\mathcal{J}$ for $\mathcal{K}_1$; If $\mathcal{I}$ is at least as plausible as $\mathcal{J}$ for $\mathcal{K}_2$ and we merge the two sources, then $\mathcal{I}$ will be at least as plausible as $\mathcal{J}$. The sixth condition is a special case of the fifth condition with $<$ (a.k.a. more plausible than). The seventh condition is more strict than the fifth and sixth conditions, with a quorum n of repetitions. The eighth condition says that if $\mathcal{I}$ is more plausible than $\mathcal{J}$ for $\mathcal{K}_1$ and $\mathcal{I}$ is more plausible than $\mathcal{J}'$ for $\mathcal{K}_2$ and $\mathcal{J}$ and $\mathcal{J}'$ are equivalent for the $\mathcal{K}_1 \sqcup \mathcal{K}_2$, then $\mathcal{I}$ will be more plausible than $\mathcal{J}$ and $\mathcal{J}'$.

Now, we provide the following proposition for belief merging operators to obtain the expected merging models (results):

Proposition 3.3 *Let μ be an integrity constraint. An operator Δ is an IC belief merging operator iff there exists a syncretic assignment $\leq$ that maps from $\mathcal{P}$ to a total pre-order $\leq_{\mathcal{P}}$ such that*

$$Mod(\Delta_\mu(\mathcal{P})) \stackrel{\text{def}}{=} min(Mod(\mu), \leq_{\mathcal{P}})$$

Now, we will present families of merging operators in the next sections.

3.3.1 Family of Model-based Operators

Model-based merging operators are typically characterized by a distance between interpretations and an aggregation function. This sub-section presents three merging operators that satisfy the belief merging postulates mentioned [EKM07, EKM12, KP02]. The first one is not a merging operator but will demonstrate an approach to arbitration operators. The next operator is a majority merging operator, and the last one is an arbitration operator.

Before presenting the family of these operators, we explain how the distances are applied for merging. First of all, it will be Hamming's distance [Dal88b] to compute between two interpretations (c.f. Definition 3.5). Then, the distance between $\mathcal{I}$ and a knowledge base $\mathcal{K}$ is defined as follows:

Definition 3.18 *Let $\mathcal{I}$ be an interpretation and $\mathcal{K}$ be a belief base. The distance between $\mathcal{I}$ and $\mathcal{K}$ is*

$$dist(\mathcal{I}, \mathcal{K}) \stackrel{\text{def}}{=} min(dist_H(\mathcal{I}\,\mathcal{J}))$$

where $\mathcal{J} \in Mod(\mathcal{K})$.

The distance is exploited to associate any belief base $\mathcal{K}$ a pre-order $\leq$ on the interpretations, defined as follows:

Definition 3.19 *Let $\mathcal{I}, \mathcal{J} \in \mathcal{W}$ be two interpretations. We have*

$$\mathcal{I} \leq \mathcal{J} \text{ if and only if } dist(\mathcal{I}, \mathcal{K}) \leq dist(\mathcal{J}, \mathcal{K})$$

Next, a definition of the distance between two belief bases is as follows:

Definition 3.20 *Let $\mathcal{K}_1, \mathcal{K}_2$ be two knowledge bases. A distance between them is defined as follows:*

$$dist(\mathcal{K}_1, \mathcal{K}_2) \stackrel{\text{def}}{=} min(dist(\mathcal{I}, \mathcal{J}))$$

where $\mathcal{I} \in Mod(\mathcal{K}_1)$ and $\mathcal{J} \in Mod(\mathcal{K}_2)$.

Now, we present a family of belief merging operators and provide an example to illustrate these operators. An example is as follows:

Example 3.4 *First of all, we assume there are two sources of knowledge encoded in the propositional logic, such as* $\mathcal{I}_1 : (0, 1, \mathbf{0,1})$ *and* $\mathcal{I}_2 : (0, 1, \mathbf{1,0})$*. Then, the Hamming's distance* $dist(\mathcal{I}_1, \mathcal{I}_2) = 2$ *since the two interpretations are different on the third and fourth letter.*

Now, we will describe a common example: there are four members of Thanh's family, including the parents, one elder sister, and Thanh. To prepare for Thanh's wedding in July, the family conducted a discussion to collect opinions on selecting the food dishes for that wedding. As a result, the restaurant offers three kinds of food: fish, beef, and seafood. Note that the restaurant manager told us that the cost would increase if we selected two of the three items.

To encode this knowledge base, we denote F, B*, and* S *respectively as fish, beef, and seafood. Moreover, we also denote* C *as the cost increase.*

Now, let us consider four members of Thanh's family as knowledge bases $\mathcal{K}_i, i \in [1, 4]$*. We then have a profile (a.k.a. a set of belief bases)* $\mathcal{P} = \{\mathcal{K}_1, \mathcal{K}_2.\mathcal{K}_3, \mathcal{K}_4\}$*. Moreover, we have a constraint from the restaurant. The restaurant manager states that it will impact the cost if we select two (or more) foods. We can encode this constraint in the propositional logic as follows:*

$$\mu = ((F \wedge B) \vee (F \wedge S) \vee (B \wedge S)) \rightarrow C$$

In order not to increase costs, it will be modeled as follows:

$$Mod(\mu) = \mathcal{W} \setminus \{(0, 1, 1, 0), (1, 0, 1, 0), (1, 1, 0, 0), (1, 1, 1, 0)\}$$

In that discussion, they have different opinions, such as Parents choosing all kinds of foods; they can pay the money at any cost. Thanh would like to eat $Beef$ *and* $Seafood$ *but does not want to increase the cost. His sister just wants a vegetarian party. Here, let us denote Father as* $\mathcal{K}_1$*, Mother as* $\mathcal{K}_2$*, Thanh as* $\mathcal{K}_3$*, and his sister as* $\mathcal{K}_4$*.*

Now let us assume there are four propositional letters F, B, S*, and* C*. We model their statements as follows:*

$$Mod(\mathcal{K}_1) \equiv Mod(\mathcal{K}_2) = \{(F, B, S, C), (F, B, S, \neg C)\} = \{(1, 1, 1, 1), (1, 1, 1, 0)\};$$

$$Mod(\mathcal{K}_3) = \{(F, B, S, \neg C), (\neg, B, S, \neg C)\} = \{(1, 1, 1, 0), (0, 1, 1, 0)\};$$

$$Mod(\mathcal{K}_4) = \{\neg F, \neg B, \neg S, \neg C\} = \{(0, 0, 0, 0)\}.$$

∘ **Max Operator (Δ^{Max})** The first merging operator is the maximum operator, denoted by Δ^{Max}. It was introduced by Revesz [Rev97] related to an example of the model fitting operator. It is close to the minimax rule used in decision theory [SFL77]. This operator is close to the arbitration operator but does not satisfy all postulates. Therefore, we can call it *quasi-merging operators*.

Definition 3.21 *Let* $\mathcal{P}$ *be a profile and* $\mathcal{I}$ *be an interpretation. We define the maximum distance between* $\mathcal{I}$ *and* $\mathcal{P}$ *as follows:*

$$dist_{Max}(\mathcal{I}, \mathcal{P}) \stackrel{\text{def}}{=} \max_{\mathcal{K} \in \mathcal{P}}(dist(\mathcal{I}, \mathcal{K}))$$

Then, we have a pre-order as follows:

$$\mathcal{I} \leq_{\Phi}^{Max} \mathcal{J} \text{ iff } dist_{Max}(\mathcal{I}, \mathcal{P}) \leq dist_{Max}(\mathcal{J}, \mathcal{P})$$

Finally, the operator Δ_{Max} *is defined by:*

$$Mod(\Delta_{\mu}^{Max}(\mathcal{P})) \stackrel{\text{def}}{=} \min(Mod(\mu), \leq_{\mathcal{P}}^{Max})$$

The idea of this operator is to find the closest worlds to the overall knowledge base set.

Remark 3.1 $dist_{Max}(\mathcal{I}, \mathcal{P}_1 \sqcup \mathcal{P}_2) = \max\{dist_{Max}(\mathcal{I}, \mathcal{P}_1), dist_{Max}(\mathcal{I}, \mathcal{P}_2)\}$

Example 3.5 *From Example 3.4, we apply the maximum belief merging operator with a constraint μ, Δ_{μ}^{Max}. Now, we recall that Dalal's distance is used in this example. The computational results of distances are shown in Table 3.1: We have marked $*$ for the shortest distances in the last column.*

Id	Possible World	$\mathcal{K}_1$	$\mathcal{K}_2$	$\mathcal{K}_3$	$\mathcal{K}_4$	$dist_{Max}$	$*$
1	$(0,0,0,0)$	3	3	0	2	3	
2	$(0,0,0,1)$	3	3	1	3	3	
3	$(0,0,1,0)$	2	2	1	1	2	$*$
4	$(0,0,1,1)$	2	2	2	2	2	$*$
5	$(0,1,0,0)$	2	2	1	1	2	$*$
6	$(0,1,0,1)$	2	2	2	2	2	$*$
7	$(0,1,1,0)$	1	1	2	0	2	$*$
8	$(0,1,1,1)$	1	1	3	1	3	
9	$(1,0,0,0)$	2	2	1	2	2	$*$
10	$(1,0,0,1)$	2	2	2	3	3	
11	$(1,0,1,0)$	1	1	2	1	2	$*$
12	$(1,0,1,1)$	1	1	3	2	3	
13	$(1,1,0,0)$	1	1	2	1	2	$*$
14	$(1,1,0,1)$	1	1	3	2	3	
15	$(1,1,1,0)$	0	0	3	0	3	
16	$(1,1,1,1)$	0	0	4	1	4	

Table 3.1: Maximum belief merging operator Δ^{Max}.

Moreover, we also highlighted the red lines, which do not satisfy the constraints μ. Finally, the following interpretations, which are close to statements with a minimal distance highlighted in the green lines, will be selected:

$$Mod(\Delta_{\mu}^{Max}(\mathcal{P})) = \{(0,0,1,0), (0,0,1,1), (0,1,0,0), (0,1,0,1), (1,0,0,0)\}$$

or

$$Mod(\Delta_{\mu}^{Max}(\mathcal{P})) = \{(\neg F, \neg B, S, \neg C), (\neg F, \neg B, S, C),$$
$$(\neg F, B, \neg S, \neg C), (\neg F, B, \neg S, C), (F, \neg B, \neg S, \neg C)\}$$

From this result, we have two options:

1. *If we select that the cost does not increase ($\neg C$), the result will be:*

$$\{(\neg F, \neg B, \mathbf{S}, \neg C), (\neg F, \mathbf{B}, \neg S, \neg C), (\mathbf{F}, \neg B, \neg S, \neg C)\}$$

   It means that we should choose one of three food kinds.

2. *Otherwise, we allow the cost increases, the result will be: $\{(\neg F, \neg B, S, C), (\neg F, B, \neg S, C)\}$. It means that if we select $Seafood$ or $Beef$, the cost can increase.*

◦ **Sum Operator** (Δ^{Σ}) The second operator that we consider is the sum operator, denoted by Δ^{Σ}. It is also known as the majority merging operator [LM99]. This definition is similar to Borda's rule [Arr51, Gro04]. While the Borda voting rule aims to minimize the sum of ranks, the sum operator is to minimize the sum of distances. The idea of the sum operator is that the distance between $\mathcal{I}$ and $\mathcal{P}$ is the sum of the distance between $\mathcal{I}$ and the $\mathcal{K}$(s) in $\mathcal{P}$.

Definition 3.22 *Let $\mathcal{P}$ be a profile and $\mathcal{I}$ be an interpretation. We have the distance between $\mathcal{I}$ and $\mathcal{P}$ as follows:*

$$dist_{\Sigma}(\mathcal{I}, \mathcal{P}) \stackrel{\text{def}}{=} \Sigma_{\mathcal{K} \in \mathcal{P}} dist(\mathcal{I}, \mathcal{K})$$

Then, we defined a pre-order for Σ as follows:

$$\mathcal{I} \leq_{\Phi}^{\Sigma} \mathcal{J} \text{ iff } dist_{\Sigma}(\mathcal{I}, \mathcal{P}) \leq dist_{\Sigma}(\mathcal{J}, \mathcal{P})$$

Finally, the Σ operator Δ^{Σ} is defined:

$$Mod(\Delta_{\mu}^{\Sigma}(\mathcal{P})) \stackrel{\text{def}}{=} \min(Mod(\mu), \leq_{\mathcal{P}}^{\Sigma})$$

Proposition 3.4 ([KP02]) *Δ^{Σ} is an majority merging operator.*

Example 3.6 *From Example 3.4, we apply the sum belief merging operator with a constraint μ, Δ_{μ}^{Σ}. Similarly, we also use Dalal's distance to compute. Now, the computational results of distances are shown in Table 3.2: We also highlight the red lines that do not satisfy the constraints μ. Note that*

Id	Possible World	$\mathcal{K}_1$	$\mathcal{K}_2$	$\mathcal{K}_3$	$\mathcal{K}_4$	$dist_{\sum}$	$*$
1	$(0,0,0,0)$	3	3	0	2	8	
2	$(0,0,0,1)$	3	3	1	3	10	
3	$(0,0,1,0)$	2	2	1	1	6	
4	$(0,0,1,1)$	2	2	2	2	8	
5	$(0,1,0,0)$	2	2	1	1	6	
6	$(0,1,0,1)$	2	2	2	2	8	
7	$(0,1,1,0)$	1	1	2	0	4	$**$
8	$(0,1,1,1)$	1	1	3	1	6	
9	$(1,0,0,0)$	2	2	1	2	7	
10	$(1,0,0,1)$	2	2	2	3	9	
11	$(1,0,1,0)$	1	1	2	1	5	$***$
12	$(1,0,1,1)$	1	1	3	2	7	
13	$(1,1,0,0)$	1	1	2	1	5	$***$
14	$(1,1,0,1)$	1	1	3	2	7	
15	$(1,1,1,0)$	0	0	3	0	3	$*$
16	$(1,1,1,1)$	0	0	4	1	5	$***$

Table 3.2: Sum belief merging operator Δ^{Σ}.

we have marked $(*)$ for the shortest distances in the last column. However, it is not satisfied with the constraint μ. Then, we select the next shortest distance (i.e., 4) highlighted by $(**)$, but it does still not satisfy μ. Finally, we can select a result highlighted by $(***)$ with a distance of 5. The following interpretation, which is close to the statements with a minimal distance highlighted in the green line, will be selected:

$$Mod(\Delta_{\mu}^{\Sigma}(\mathcal{P})) = \{(1,1,1,1)\}$$

or

$$Mod(\Delta_{\mu}^{\Sigma}(\mathcal{P})) = \{(F, B, S, C)\}$$

This result means that if we take account of all the statements, we choose three good kinds and agree with increasing the cost.

∘ ***Leximax* Operator (Δ^{Gmax})** The third belief merging operator we mentioned is leximax operators or Gmax operators. The idea is to capture the arbitration behavior of Δ^{Max} without losing the properties of an IC belief merging operator. It is also known as a refinement of the Δ^{Max} operator.

Definition 3.23 *Let $\mathcal{P}$ be a profile ($\mathcal{P} = \{\mathcal{K}_1, \ldots, \mathcal{K}_n\}$). For each $\mathcal{I}$, we have the list $(d_1^{\mathcal{I}}, \ldots, d_n^{\mathcal{I}})$ of distances between $\mathcal{I}$ and $\mathcal{K}$, where $d_k^{\mathcal{I}} = dist(\mathcal{I}, \mathcal{K}_k)$. We also denote*

$$dist_{Gmax}(\mathcal{I}, \mathcal{P}) \stackrel{\text{def}}{=} \mathscr{L}_{\mathcal{P}}^{\mathcal{I}}$$

where we say $\mathscr{L}_{\mathcal{P}}^{\mathcal{I}}$ is a list $(d_1^{\mathcal{I}}, \ldots, d_n^{\mathcal{I}})$ sorting in descending order. Let $\leq_{Gmax}$ be the lexicographical order between sequences of integers. Then, we have a pre-order form $Gmax$ as follows:

$$\mathcal{I} \leq_{\Phi}^{Gmax} \mathcal{J} \text{ iff } dist_{Gmax}(\mathcal{I}, \mathcal{P}) \leq dist_{Max}(\mathcal{J}, \mathcal{P})$$

Finally, the operator Δ_{Max} is defined by:

$$Mod(\Delta_{\mu}^{Gmax}(\mathcal{P})) \stackrel{\text{def}}{=} \min(Mod(\mu), \leq_{\mathcal{P}}^{Gmax})$$

Now, we provide the proposition, remark, and lemma to prove that Δ^{GMax} has desirable properties and to support our proofs.

Proposition 3.5 ([KP02]) *Δ^{GMAx} is an majority arbitration operator.*

Remark 3.2 $\Delta_{\mu}^{Gmax}(\mathcal{P}) \models \Delta_{\mu}^{max}(\mathcal{P})$.

Definition 3.24 ([KP98]) *Let $\mathcal{X}_1$ and $\mathcal{X}_2$ be two lists of n numbers sorted in descending order. We denote $\mathcal{X}_1 \odot \mathcal{X}_2$ the list obtained by sorting in descending order the concatenation of $\mathcal{X}_1$ and $\mathcal{X}_2$.*

Lemma 3.1 ([KP98]) *Let $\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_1', \mathcal{X}_2'$ be four lists of integers sorted in descending order. If $\mathcal{X}_1 \leq_{GMax} \mathcal{X}_1'$ and $\mathcal{X}_2 \leq_{GMax} \mathcal{X}_2'$ then $\mathcal{X}_1 \odot \mathcal{X}_2 \leq_{GMax} \mathcal{X}_1' \odot \mathcal{X}_2'$.*

Lemma 3.2 ([KP98]) *Let $\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_1', \mathcal{X}_2'$ be four lists of integers sorted in descending order. If $\mathcal{X}_1 \leq_{GMax} \mathcal{X}_1'$ and $\mathcal{X}_2 <_{GMax} \mathcal{X}_2'$ then $\mathcal{X}_1 \odot \mathcal{X}_2 <_{GMax} \mathcal{X}_1' \odot \mathcal{X}_2'$.*

Example 3.7 *From Example 3.4, we apply the leximax belief merging operator with a constraint μ, Δ_{μ}^{Gmax}. Similarly, we also use Dalal's distance to compute. Now, the computational results of distances are shown in Table 3.3: We also highlight the red lines that do not satisfy the constraints μ. Note that we have marked $(*)$ for the shortest distances in the last column. However, the shortest one is unsatisfied with the constraint μ. Then, we select the next shortest distance (i.e., $(2, 1, 1, 0)$) highlighted by $(**)$, but it does still not satisfy μ. Finally, we can select a result highlighted by $(***)$ with the distance being $(2, 2, 1, 1)$. The following interpretation, which is close to statements with a minimal distance highlighted in the green line, will be selected:*

$$Mod(\Delta_{\mu}^{Gmax}(\mathcal{P})) = \{(0, 0, 1, 0), (0, 1, 0, 0)\}$$

or

$$Mod(\Delta_{\mu}^{Gmax}(\mathcal{P})) = \{(\neg F, \neg B, S, \neg C), (\neg F, B, \neg S, \neg C)\}$$

From this result, we also realize that Δ^{Gmax} is narrower than Δ^{Max}'s choices. We also say that the Δ^{Gmax} operator is sharper and more condensed than Δ^{Max}. The outcome shows that if we choose $Beef$ or $Seafood$, the price does not increase.

Id	Possible World	$\mathcal{K}_1$	$\mathcal{K}_2$	$\mathcal{K}_3$	$\mathcal{K}_4$	$dist_{Gmax}$	$*$
1	$(0,0,0,0)$	3	3	0	2	$(3,3,2,0)$	
2	$(0,0,0,1)$	3	3	1	3	$(3,3,3,1)$	
3	$(0,0,1,0)$	2	2	1	1	$(2,2,1,1)$	$***$
4	$(0,0,1,1)$	2	2	2	2	$(2,2,2,2)$	
5	$(0,1,0,0)$	2	2	1	1	$(2,2,1,1)$	$***$
6	$(0,1,0,1)$	2	2	2	2	$(2,2,2,2)$	
7	$(0,1,1,0)$	1	1	2	0	$(2,1,1,0)$	$*$
8	$(0,1,1,1)$	1	1	3	1	$(3,1,1,1)$	
9	$(1,0,0,0)$	2	2	1	2	$(2,2,2,1)$	
10	$(1,0,0,1)$	2	2	2	3	$(3,2,2,2)$	
11	$(1,0,1,0)$	1	1	2	1	$(2,1,1,1)$	$**$
12	$(1,0,1,1)$	1	1	3	2	$(3,2,1,1)$	
13	$(1,1,0,0)$	1	1	2	1	$(2,1,1,1)$	$**$
14	$(1,1,0,1)$	1	1	3	2	$(3,2,1,1)$	
15	$(1,1,1,0)$	0	0	3	0	$(3,0,0,0)$	
16	$(1,1,1,1)$	0	0	4	1	$(4,1,0,0)$	

Table 3.3: Leximax belief merging operator Δ^{GMax}.

3.3.2 Family of Syntax-based Operators

The syntax of the propositional formulae that make up the profile's belief base often determines the outcome delivered by an operator represented as being *syntax-based* [HPW07b, HPPS14].

Combination Operators

The initial syntax-based operators to be established in the research literature take into account the union of the profile's belief bases. Then, we choose maximally consistent subsets of formulae from the set produced as a consequence [BKM91, BDP99]. In the studies of Baral et al. [BKM91, BKMS91], these operators are distinguished from merging operators by the usage of the phrase "combination operator", which is the same name used to refer to both types of operators [Kon00]. The definitions that are presented here are from the research by Konieczny et al. [Kon00].

Definition 3.25 (Maximum consistent subsets [Kon00]) *Let μ be a propositional formula. For any set $\mathcal{K}$ of propositional formulas, $MAXCONS(\mathcal{K}, \mu)$ denotes the set of subsets of formulas of $\mathcal{K} \cup \{\mu\}$ which contain μ, which are maximal (for $\subseteq$) and consistent. Formally, $MAXCONS(\mathcal{K}, \mu)$ is the set of all sets M of formulas that verify the following three properties:*

1. $M \subseteq \mathcal{K} \cup \{\mu\}$;
2. $\mu \in M$; *and*
3. *If* $M \subset M' \subseteq \mathcal{K} \cup \{\mu\}$, *then* M' *is inconsistent.*

Example 3.8 *Let us consider an example with an inconsistent base as follows:*

$$\mathcal{K} = \{a \wedge b \wedge c, a \vee b \vee c, a \wedge \neg b, a \wedge \neg c\}$$

and a constraint $\mu = \top$. *There are two subsets in* $MAXCONS(\mathcal{K}, \top)$ *as follows:*

$$\{a \wedge b \wedge c, a \vee b \vee c, \top\}$$

$$\{a \vee b \vee c, a \wedge \neg b, a \wedge \neg c, \top\}$$

If we take account of $\mu = c$, *the result of* $MAXCONS(\mathcal{K}, c)$ *will be*

$$\{a \wedge b \wedge c, a \vee b \vee c, c\}$$

$$\{a \vee b \vee c, a \wedge \neg b, c\}$$

Definition 3.26 (Maximum consistent knowledge base [Kon00]) *Let* μ *be a propositional formula. For any* $\mathcal{P}$ *of propositional formulas,* $MAXCONS_{\cup}(\mathcal{P}, \mu)$ *denotes the set of sub-multisets of formulas of* $\mathcal{K} \sqcup \{\mu\}$ *that contain* μ, *which are maximal in terms of cardinality and consistent. Formally,* $MAXCONS_{\cup}(\mathcal{K}, \mu)$ *is the set of all multi-sets* M *of formulas that verify the following three properties:*

1. $M \sqsubseteq \mathcal{K} \cup \{\mu\}$;
2. $\mu \in M$; *and*
3. *For any multiset of formulas* M' *such that* $M' \sqsubseteq \mathcal{K} \sqcup \{\mu\}$ *and* $|M| < |M'|$, $\bigwedge M'$ *is inconsistent.*

Let $\mathcal{P} = \{\mathcal{K}_1, \ldots, \mathcal{K}_m\}$ *be a profile. We have* $MAXCONS(\mathcal{P}, \mu) = MAXCONS(\bigcup \mathcal{K}_i \mid \mathcal{K}_i \in \mathcal{P}, \mu)$ *and* $MAXCONS_{\cup}(\mathcal{P}, \mu) = MAXCONS_{\cup}(\bigsqcup \mathcal{K}_i \mid \mathcal{K}_i \in \mathcal{P}, \mu)$.

Example 3.9 *Let us consider an example with two bases* $\mathcal{K}_1 = \{a \wedge b \wedge c, a \vee b \vee c\}$, $\mathcal{K}_2 = \{a \wedge \neg b, a \wedge \neg c\}$, *and a constraint* $\mu = \top$. *For* $\mathcal{P} = \{\mathcal{K}_1 \mathcal{K}_2\}$, *we have* $MAXCONS(\mathcal{P}, c)$ *as follows:*

$$\{a \wedge b \wedge c, a \vee b \vee c, \top\}$$

$$\{a \vee b \vee c, a \wedge \neg b, a \wedge \neg c, \top\}$$

If we take account of $\mu = c$, *the result of two other maximal consistent will be*

$$\{a \wedge b \wedge c, a \vee b \vee c, c\}$$

$$\{a \vee b \vee c, a \wedge \neg b, c\}$$

Now, we take account of the maximal consistent for carnality. We have

$$MAXCONS_{\cup}(\mathcal{P}, \top) = \{a \vee b \vee c, a \wedge \neg b, a \wedge \neg c, \top\}$$

$$MAXCONS_{\cup}(\mathcal{P}, c) = \{\{a \wedge b \wedge c, a \vee b \vee c, c\}, \{a \vee b \vee c, a \wedge \neg b, c\}\}$$

Definition 3.27 (Combination operators [Kon00]) *Let* $\mathcal{P}$ *be a profile and* μ *be a propositional formula. We have*

1. $\Delta_{\mu}^{C_1}(\mathcal{P}) \stackrel{\text{def}}{=} \bigvee\{M \in MAXCONS(\mathcal{P}, \mu)\}$.
2. $\Delta_{\mu}^{C_3}(\mathcal{P}) \stackrel{\text{def}}{=} \bigvee\{M \mid M \in MAXCONS(\mathcal{P}, \top)$ *and* $M \cup \{\mu\}$ *is consistent*$\}$.
3. $\Delta_{\mu}^{C_4}(\mathcal{P}) \stackrel{\text{def}}{=} \bigvee\{M \in MAXCONS_{\cup}(\mathcal{P}, \mu)\}$.

4. $\Delta_\mu^{C_5}(\mathcal{P}) \stackrel{\text{def}}{=} \bigvee\{M \cup \{\mu\} \mid M \in MAXCONS(\mathcal{P}, \top)$ *and* $M \cup \{\mu\}$ *is consistent*$\}$ *if this set is non-empty,* $\Delta_\mu^{C_5}(\mathcal{P}) \stackrel{\text{def}}{=} \mu$ *otherwise.*

The operator $\Delta_\mu^{C_1}(\mathcal{P})$ considers the set of maximally consistent sets containing the constraint. $\Delta_\mu^{C_3}(\mathcal{P})$ considers the set of consistent maximal sets and retains those that are consistent with the constraint. $\Delta_\mu^{C_4}(\mathcal{P})$ considers the set of consistent maximal sets for cardinality that contain the constraint. $\Delta_\mu^{C_5}(\mathcal{P})$ is a variant of $\Delta_\mu^{C_3}(\mathcal{P})$, but verifies more logical properties (see proposition 9). In particular, $\Delta_\mu^{C_3}(\mathcal{P})$ can generate inconsistent merged bases, which is not the case for the other operators listed here. We do not find in this list of operators $\Delta_\mu^{C_2}(\mathcal{P})$ since the operator $\Delta_\mu^{C_2}(\mathcal{P})$ initially proposed by Baral, Kraus, and Minker is identical to $\Delta_\mu^{C_1}(\mathcal{P})$.

Example 3.10 *The following bases can be used to encode Thanh's, Alexis's, and Ryma's respective objectives: (respectively)*

$$\textit{Thanh: } (\neg a \wedge \neg b) \vee (a \wedge b \wedge c)$$

$$\textit{Alexis: } (\neg a \wedge \neg b \wedge c) \vee (a \wedge b \wedge \neg c)$$

$$\textit{Ryma: } (\neg a \wedge \neg b \wedge \neg c) \vee (a \wedge b \wedge \neg c)$$

Now, we assume that $\mu = \top$*. The maximal consistent subsets of the union of these bases are:*

$$\{(\neg a \wedge \neg b) \vee (a \wedge b \wedge c), (\neg a \wedge \neg b \wedge c) \vee (a \wedge b \wedge \neg c), \top\}.$$

$$\{(\neg a \wedge \neg b) \vee (a \wedge b \wedge c), (\neg a \wedge \neg b \wedge \neg c) \vee (a \wedge b \wedge \neg c), \top\}$$

$$\{(\neg a \wedge \neg b \wedge c) \vee (a \wedge b \wedge \neg c), (\neg a \wedge \neg b \wedge \neg c) \vee (a \wedge b \wedge \neg c), \top\}$$

Hence, for this example, we obtain:

$$\Delta_\mu^{C_1}(\mathcal{P}) \equiv \Delta_\mu^{C_3}(\mathcal{P}) \equiv \Delta_\mu^{C_4}(\mathcal{P}) \equiv \Delta_\mu^{C_5}(\mathcal{P}) \equiv (\neg a \wedge \neg b) \vee (a \wedge b \wedge \neg c)$$

The operators correspond to operators previously defined in [BKM91]. In addition, the logical properties of each of them have been studied in [Kon00]. We do not detail them here but simply note that none of these operators satisfy the syntax-independence postulate (IC3). It characterizes them as syntax-sensitive operators, i.e., syntax-based. On the other hand, all these operators verify the postulate of majority independence (MI) because they do not consider the distribution of knowledge within each source.

3.3.3 DA^2 Operators

Konieczny, Lang, and Marquis propose a general family of merging operators called DA^2 operators [KM04]. The set of *model-based* merging operators and certain *syntax-based* operators are both included in this class of operators. A DA^2 operator's parameters are determined by the distance ($dist$) between an interpretation and a propositional formula, as well as by two aggregation functions (f and g). Their definition is somewhat comparable to the definitions of the *model-based* operators $\sum$, Max, and $GMax$. Now, we provide a formal definition of the DA^2 operator:

Definition 3.28 (DA^2 operator [KM04]) *The distance* $dist_{DA^2}$ *between an interpretation and a propositional formula and the aggregation functions* f *and* g *are used to calculate a distance* $dist_f$ *between any interpretation* $\mathcal{I}$ *and any profile* $\mathcal{P}$*, defined by*

$$dist_{DA^2}(\mathcal{I}, \mathcal{P}) \stackrel{\text{def}}{=} f\{dist(\mathcal{I}, \mathcal{K}) \mid \mathcal{K} \in \mathcal{P}\}$$

where $dist(\mathcal{I}, \mathcal{K}) = g\{dist(\mathcal{I}, \Phi) \mid \Phi \in \mathcal{K}\}$. *Then, we define a pre-order* $\leq_{\mathcal{P}}^{dist,f,g}$ *as follows:*

$$\mathcal{I} \leq_{\mathcal{P}}^{dist,f,g} \mathcal{J} \text{ iff } dist_{DA^2}(\mathcal{I}, \mathcal{P}) \leq dist_{DA^2}(\mathcal{J}, \mathcal{P})$$

Finally, the DA^2 *operator* $\Delta^{dist,f,g}$ *is defined:*

$$Mod(\Delta_{\mu}^{dist,f,g}(\mathcal{P})) \stackrel{\text{def}}{=} \min(Mod(\mu), \leq_{\mathcal{P}}^{dist,f,g})$$

The fact that the aggregating function f is included in the family of operators known as DA^2 is what makes them a generalization of the family of operators known as *model-based* operators. Note that, when f corresponds to the function $\sum$ (either $\max$ or $GMax$), for instance, the operator DA^2 that corresponds to f coincides with the *model-based* operator $\sum$. (respectively, the operator $\max$, the operator $GMax$). It is possible to take into consideration conflicting belief bases by using the aggregation function known as g. According to the parameterization provided by the triplet $(dist, f, g)$, research by Konieczny et al. [KM04] was conducted on the logical features of these operators as well as a complexity study (about the inference issue). This study included a broad range of distances and aggregation functions.

Many other families of belief base merging operators have been studied in the literature. For example, Everaere, Konieczny, and Marquis examine two families of merging operators in their work [EKM10, EKM05]. The $quota$ operator family and the $GMin$ operator family [EKM10] are the families that are investigated. The operators of these two families are referred to as *disjunctive* because they imply the disjunction of the belief bases (provided that this disjunction is consistent). The interpretations retained in the fusion result are the models of the integrity constraints that satisfy *sufficient* knowledge bases.

3.4 Ontology Merging

Researchers have concentrated on merging knowledge bases from pragmatic and theoretical perspectives. In particular, they tried to handle the inconsistencies during the merging process. To the best of our collective knowledge, there are usually two research approaches for knowledge base merging.

- The first one is ontology integration research with mapping and alignment techniques [LT06, FBC13, OBYD21, Ala09, CS16]. They take advantage of mapping and alignment for ontology merging. In the semantic web, researchers concerning matching ontology [SE13] have established an ontology matching community[15] to discover this aspect. The ontology community has embraced the categorization of the ontology integration challenge [Ala09, BEF+06] into *matching, alignment, merging, and repairing knowledge*, with matching being a prerequisite for merging. Repairing can be included in the matching process or performed separately [OBYD21]. Typically, the principal strategies involve two steps: (i) first, the matching step, which reconciles differences by identifying semantic correspondences (primarily similarities) between different elements, using a similarity computation; then (ii) the merging step, which exploits the result of the matching step by merging or linking the matched elements to create a new unified view;

- The second one is to propose operators [KP02, KM04, KPP13] to handle the arising problems while merging. Normally, knowledge-merging research's main challenge is handling inconsistencies (or so-called conflicts). They remove the minimal amount of knowledge so that the remaining holds consistent [BBT15, HPW07b]. There are several kinds of changing knowledge, including: *merging [KPP11, PC19], revising [Gär88, HW19], and updating [FUV83].* Many researchers built and developed the operators and their behaviors [KP02, BK03].

[15] http://www.ontologymatching.org/

Moreover, ontology integration and merging terms are ambiguous. Some authors take both of them as synonyms, while others do not. To understand the differences between them, we provide a comparison shown in Table 3.4 [OYD21].

Information	Ontology Merging	Ontology Integration
Type	Simple Merge or Full Merge	Simple Merge, Full Merge, Matching, Alignment
Final Goal	To create a new ontology	to enrich and extend an ontology
Method	Collection or Aggregation	Inclusion or Assembling
Method Details	Aggregating ontologies to form a target ontology	Incorporating input ontologies (one by one) into a target ontology.
Input	Two or more ontologies	one target ontology + one or more sources.
Output	A newly target ontology, called the merged ontology	A new version of the expected goal ontology.

Table 3.4: Comparison between ontology integration and ontology merging.

In general, pragmatic approaches have just focused on the ontology mapping activity [CSH06, KS03, KS14]. In contrast, the theoretical approaches pay attention to handling the merging problem [PE11, KVS06]. Both of the approaches are interconnection and mutual support. Namely, the whole merging process might be as follows: First, we compute the similarity of concepts from different sources and then find the concept correspondence to map them to each other. After mapping the concepts obtained, the merging activity will be conducted. During the merging process, the concepts from all sources are copied into the output base to solve the problem. Note that *mapping process will be performed before merging sources*.

Now, we give a difference between merging and alignment as in Figure 3.2 and provide some notions related to ontology operations as follows:

- *Ontology merging* [RR11] is the process of creating a new single consistent ontology from two or more existing ontology sources related to the same domain. The new ontology will replace the input ontology sources;

- *Ontology mapping* [SE13, PDYP05] is a specification of the semantic equivalence correspondence between two or more ontologies. In other words, it is a formal expression that describes the semantic link between two or more concepts from two or more distinct ontologies;

- *Ontology alignment (matching)* [LT08, OK18] is the process or method of creating a consistent and coherent link between two or more ontologies by bringing them into a mutual agreement. In other words, it is A set of correspondences between two (or more) ontologies. These correspondences are called “Mappings”; and

- *Ontology annotation* [TML07] Ontology-based annotation refers to the process of creating metadata using ontology as a vocabulary.

Indeed, there are several mechanisms developed for merging and matching ontologies, such as ONIONs [GSG00], FCA-Merge [SM01], FCAMap-KG [CwCZ19], CANARD [THT18], PROMPT [NM03, NM00], RiMom [LZLT07], and others. We also provide detailed features regarding the merging and matching at Table 3.4.

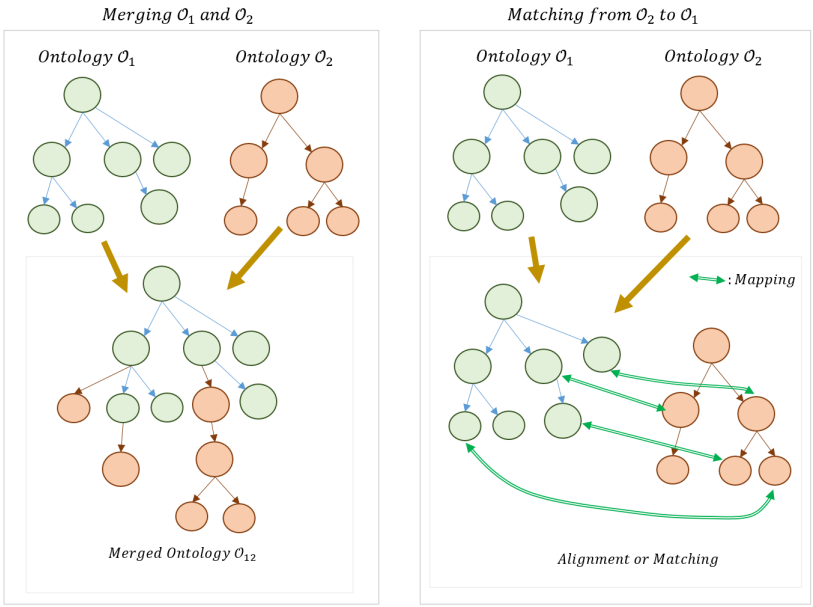


Figure 3.2: Ontology merging and ontology alignment.

Many researchers concentrate on handling the conflicts when merging the ontologies. They take advantage of the existing knowledge fundamentals to perform knowledge merging, i.e., belief merging [KP99, KP02, Cur10, DP94], belief revision [NPP03, DEKP20, Pap01, Pap92], social choice [Arr51, PE11, GRP09], and others. As presented in the above part, belief merging has a solid background. Several concrete merging operators have been proposed [Rev97, LM99, KP02, HPW07b, KPP11, HW19, MGC02]. For instance, Porello [PE11] studied aggregating and merging statements by summarizing a group's global choice using the available opinions and judgments. They also provided a formal framework based on social choice theory. They proposed several simple procedures (majority rules, quota rules). In addition, ontology and description logic originate from formal logic. Hence, the research of description logic merging is in the neighborhood of belief merging and voting theory. Several relevant DLs merging studies are proposed [BBPW19, BBLR14, WWJQ12, BLSW02]. Typically, the authors propose a reasoning merging method underlying specific DLs (i.e., $DL\text{-}Lite$, $\mathcal{FL}$, $\mathcal{ALC}$, and others). For example, the author groups of Benferhat et al. [BBLR14] have utilized $DL\text{-}Lite$ in their merging approach. They have determined the minimal subsets of assertions to resolve conflicts based on the inconsistency minimization principle. Otherwise, belief revision can be used to merge ontologies, i.e., Gomez and Simari [GS13] that used belief revision and defeasible logic programming to merge ontologies. They use a kernel revision operator to find the minimal subset and convert the subset into a set of defensible rules.

3.5 Inconsistency Handling

One of the most significant current issues in OWL ontology management is dealing with inconsistency [SGH20, HDRDA11, JBQ19]. Moreover, it is well-known that OWL and description logic's classical semantics are not inconsistent, i.e., they do not allow for taking account of a meaningful piece of knowledge in an inconsistency. This section provides two kinds of handling inconsistency: propositional logic and ontology.

Name	Operation	Algorithm	Input/Ouput	Technique
ONION	Merging	String matching algorithms	RDF file and plain text	SKAT[16].
FCA-Merge	Merging	Lexicon based	Two DL ontologies and a merged ontology	Concept lattice.
CARNARD	Complex Alignment, Matching	An algorithm based on Competency Questions for Alignment (CQAs)	Two ontologies and an alignment	SPARQL[17] and SEALs[18].
PROMT	Merging and Alignment	A novel algorithm for automatical ontology merging	Two DAML ontologies and a merged ontology	OKBC [CFF+98].
RiMOM	Matching	GAMO	RDF and OWL ontologies and an alignment result	OWL API[19].
HCONE	Merging and alignment	String matching algorithms	Two ontologies and a merging ontology	Neo classic and Description logic.
YAGO	Mapping	Quick sort algorithm	High data and Accuracy ontology	Lightweight Ontological Tech.
COM++	Mapping, Matching	SVM	XML, OWL files and an alignment	OWL API.

Table 3.5: Systems of merging and matching ontologies.

3.5.1 Inconsistency Handling in Propositional Logic

In classical logic, the notion of *inconsistency* is a binary concept. Either the formula is inconsistent or not. However, the conflict handling problem has been given much attention in the last years. When we merge or revise different sources, it might lead to inconsistency. Numerous strategies have been proposed and implemented to address this issue [KMB08, KMV15, RM70, BBC+16]. To understand the inconsistency or conflicts, we give the following example to illustrate them. Let us consider the following two classical logic knowledge base

$$\mathcal{K}_1 = \{a, \neg a, b, \neg b, c\} \qquad \mathcal{K}_2 = \{\neg b, a, c, a \rightarrow b\}$$

We can say that both $\mathcal{K}_1$ and $\mathcal{K}_2$ are inconsistent. Here, in $\mathcal{K}_1$, there are two apparent contradictions, i.e., $\{a, \neg a\}$ and $\{b, \neg b\}$ are conflicting formulas. In $\mathcal{K}_2$, the conflict is a bit more complex. These formulas are necessary to produce a contradiction, i.e., $\{\neg b, a, a \rightarrow b\}$. Furthermore, two formulas c in both $\mathcal{K}_1$ and $\mathcal{K}_2$ are not involved in a conflict.

Now, we provide several typical approaches to inconsistency handling. They include

- **Paraconsistent logics:** A paraconsistent logic is an attempt by a logical system to deal with contradictions in a discriminating way. A minimal condition for a suitable inference relation is not explosive. Such inference relationships are called paraconsistent. Moreover, it also consists of accepting inconsistency while capturing it using weak inference relations. There are two approaches: weakening logical connectors [BH95, Hun02, DKP03, PTC+17] and richer semantics [KM02, KMB08];

- **Repair inconsistency:** The idea is to find the set of maximally consistent subsets that restore the knowledge base's consistency. In particular, let $\mathcal{K}$ be an inconsistent knowledge base, we will try to find a subset $\mathcal{K}'$ such that: $\mathcal{K}' \subset \mathcal{K}$, and $\mathcal{K}'$ is consistent, and if $\mathcal{K}' \subset \mathcal{K}'' \subseteq \mathcal{K}$, then $\mathcal{K}''$ is inconsistent. Then, we can use $\mathcal{K}'$ to conduct inference. Many works related to this aspect are proposed [RM70, BDP93, BDP97, KM05, LM02];
- **Inconsistency analysis:** It might provide analysis and measure the inconsistency of a knowledge base to make an inference. Typically, they can use the inconsistency degree to do reasoning for managing the consistent subsets of formulas [HK05, HK10, GH11, HP14]; and
- **Argumentation framework:** It is a method for dealing with problematic knowledge and coming to a conclusion using formalized arguments. Typically, it involves selecting between contradictory statements. Specifically, when there is a collection of conflicting opinions, the framework will recognize and identify the positive and negative arguments to reach a certain conclusion. [KMV15, AC04, BB20].

For solving the inconsistency, we provide some notions as follows:

Definition 3.29 (Minimal inconsistent subset (MIS) [Sal21]) *A subset (sub-base) $\mathcal{K}'$ of $\mathcal{K}$ is said to be minimally inconsistent if and only if it satisfies the two following requirements:*

- $\mathcal{K}' \models \bot$*; and*
- $\forall \Phi \in \mathcal{K}', \mathcal{K}' - \{\Phi\} \not\models \bot$.

Definition 3.30 (Maximal consistent subset (MCS) [MSM+20, Sal21]) *A maximal consistent subset of a belief base $\mathcal{K}$ is a subset of $\mathcal{K}'$ of $\mathcal{K}$ such that $\mathcal{K}' \subset \mathcal{K}, \mathcal{K}' \models \bot$ and for each $\Phi \in \mathcal{K} \setminus \mathcal{K}', \mathcal{K} \cup \{\Phi\} \models \bot$.*

Moreover, we can also define the *free formula* as follows:

Definition 3.31 (Free formula [Sal21]) *Let $\mathcal{K}' \subseteq \mathcal{K}$. A formula Φ in a belief base $\mathcal{K}$ is said to be* free *in $\mathcal{K}$ if $\Phi \notin \mathcal{K}'$ for every $\mathcal{K}' \in MIS(\mathcal{K})$. We say $Free(\mathcal{K})$ is a set of free formulas in $\mathcal{K}$.*

For this definition, we can say that a formula Φ is *problematic* in $\mathcal{K}$ if Φ does not belong to $Free(\mathcal{K})$.

Furthermore, much of the work focuses on measuring inconsistency. We denote $\mathbb{R}^+_\infty$ as the set of positive real numbers stated with the greatest element denoted by ∞. We write that $INC(S)$ is the set of inconsistent formulas belonging to S, a set of formulas.

Definition 3.32 (Inconsistency measure [HK08, Thi19]) *An inconsistency measure is a mapping Mse: $\mathcal{K} \to \mathbb{R}^+_\infty$ such that, with every $\mathcal{K}', K'' \subseteq \mathcal{K}$ and $\alpha, \beta \in \mathcal{K}$, it satisfies the following properties:*

- ***Normalization:*** $0 \leq Mse(\mathcal{K}) \leq 1$*;*
- ***Free-formula independence:*** *If $\alpha \in Free(\mathcal{K})$ then $Mse(\mathcal{K}) = Mse(\mathcal{K} \setminus \{a\})$;*
- ***Consistency****: $Mse(\mathcal{K}) = 0$ iff $\mathcal{K} \not\models \bot$;*
- ***Monotonicity:*** *If $\mathcal{K}'' \subseteq \mathcal{K}'$ then $Mse(\mathcal{K}') \leq Mse(\mathcal{K}'')$;*
- ***Equivalence:*** *If $\alpha \equiv \beta$ then $Mse(\mathcal{K}' \cup \{\alpha\}) \equiv Mse(\mathcal{K}' \cup \{\beta\})$;*
- ***Tautology****: if $\top \models \emptyset$ then $Mse(\mathcal{K} \cup \{\alpha\}) = Mse(\mathcal{K})$; and*

- ***Dominance:*** *If* $\alpha \not\models \bot$ *and* $\alpha \models \beta$ *then* $Mse(\mathcal{K} \cup \{\alpha\} \leq Mse(\mathcal{K} \cup \{\beta\}))$.

Moreover, we provide several inconsistency measures from the literature as follows:

- [HK08] $Mse(\mathcal{K}) = 1$ if $\mathcal{K} \models \bot$, and $Mse(\mathcal{K}) = 0$ if $\mathcal{K} \not\models \bot$;
- [HK08] $Mse(\mathcal{K}) =\mid MIS(\mathcal{K}) \mid$;
- [GH11] $Mse(\mathcal{K}) =\mid MCS(\mathcal{K}) \mid + \Sigma(\alpha) - 1$, where $\Sigma(\mathcal{K}) = 1$ if $Inc(\mathcal{K}) \neq \emptyset$ and $\Sigma(\mathcal{K}) = 0$ if $Inc(\mathcal{K}) = \emptyset$;
- [GH11] $Mse(\mathcal{K}) =\mid \mathcal{K} \setminus Free(\mathcal{K}) \mid$; and
- [GH17] $Mse(\mathcal{K}) = \min(\max_{\Phi \in \mathcal{K}}(dist(Mod(\Phi), \mathcal{I}) \mid \mathcal{I} \in \mathcal{W}))$.

where $dist(S, \mathcal{I}) = \min\{dist(\mathcal{I}', \mathcal{I}) \mid \mathcal{I}' \in S\}$ in the case of $dist(\mathcal{I}, \mathcal{I}') =\mid \{x \in \mathcal{K} \mid \mathcal{I}(x) \neq \mathcal{I}'(x)\} \mid$.

Example 3.11 *Let us consider the knowledge base* $\mathcal{K}$ *as follows:*

$$\mathcal{K} = \{a \wedge c, b \wedge \neg c, \neg a \wedge \neg b\}$$

Here, $\mathcal{K}$ *has a single minimal inconsistent subset* $\{a \wedge c, b \wedge \neg c\}$ *and* $\neg a \vee \neg b$ *is a free formula. Since* Mse *satisfies* ***text-formal independence****, we obtain* $Mse(\mathcal{K}) = Mse(\mathcal{K} \setminus \{\neg a \vee \neg b\})$.

3.5.2 Inconsistency Handling in Ontologies

In the context of description logic, a knowledge base includes two components, i.e., $TBox$ and $ABox$ as presented in Section 2.3. Notably, incoherence and inconsistency are related. Incoherence is a potential cause of inconsistency, i.e., we suppose C is an unsatisfiable concept in $\mathcal{O}$. Now, if we add an instance a to C, the $\mathcal{O}$ will result in being inconsistent, e.g., $\mathcal{O} = \{A \sqsubseteq B, A \sqcap B \sqsubseteq \bot, A(a)\}$. In most work related to debugging ontologies, the incoherence problem is discussed at the terminology level, or $TBox$. At the same time, $ABox$ is considered irrelevant for incoherence.

First, we present some useful definitions of incoherence ($TBox$) and inconsistency ($ABox$).

Definition 3.33 (MUPS [CSAH06]) *Let* A *be a concept name which is unsatisfiable in a* $TBox$ $\mathcal{T}$*. A set* $\mathcal{T}' \subseteq \mathcal{T}$ *is* ***a minimal unsatisfiability preserving subset*** $TBox$ *of* $\mathcal{T}$ *if* A *is unsatisfiable in* $\mathcal{T}'$*, and* A *is satisfiable in* $\mathcal{T}'' \subset \mathcal{T}'$.

Definition 3.34 (MIPS [CSAH06]) *Let* $\mathcal{T}$ *be an incoherent* $TBox$*. A* $TBox$ $\mathcal{T}' \subseteq \mathcal{T}$ *is a minimal incoherence preserving subset* $TBox$ *(MIPS) of* $\mathcal{T}$ *if* $\mathcal{T}'$ *is incoherent, and* $\mathcal{T}'' \subset \mathcal{T}'$ *is coherent*

Typically, one manages the conflicts based on MIPS and MUPS. One can remove or modify the axioms such that $\mathcal{O}$ will be consistent and satisfiable [QHH+08, Lam19]. To illustrate this, we give an example as follows:

Example 3.12 *We suppose that we have an ontology that contains the following axioms (constraints)* Φ_i*:*

- $\Phi_1 = \begin{cases} Brain \sqsubseteq BodyPart \\ Brain \sqsubseteq CentralNervousSystem \\ Brain \sqsubseteq \forall IsRegionOf.HeadAndNeck \end{cases}$

- $\Phi_2 : CentralNervousSystem \sqsubseteq NervousSystem$
- $\Phi_3 : NervousSystem \sqcap BodyPart \sqsubseteq \bot$

Let us notice the above example. The ontology will be inconsistent because $Brain$ is a part of $BodyPart$ and $NervousSystem$ (according to Φ_1 and Φ_2), while Φ_3 states that $NervousSystem$ and $BodyPart$ are disjoint (according to Φ_3). To solve this conflict, we can delete any one of the axioms Φ_i (either Φ_1 or Φ_2 or Φ_3).

Regarding inconsistency, the methods for detecting inconsistencies rely on the idea of a selection function to identify the relevant axioms which produce the inconsistency. Algorithms to find *minimal inconsistent sub-ontologies* and *maximal consistent sub-ontologies* have been proposed and developed [CPSV06, HDRDA11]. Note that, to avoid inconsistency, we have to take account of both $ABox$ and $TBox$. Moreover, the studies also focus on the repair of axioms [TCG+18]. Let us consider an example as follows:

Example 3.13 *We suppose that we have an ontology related to education, denoted by $\mathcal{O}_E$, which has the following axioms Φ_i (terminological axioms and assertion axioms) as follows:*

- $\Phi_1 : Researcher \sqsubseteq Human$
- $\Phi_2 : PhDStudent \sqsubseteq Student$
- $\Phi_3 : Student \sqcap Researcher \sqsubseteq \bot$
- $\Phi_4 : Scientific Paper \sqsubseteq Publication$
- $\Phi_5\ PhDStudent(Thanh)$
- $\Phi_6\ Researcher(Peter)$

Here, if we suppose that a new assertion axiom is coming, which tells us that $Peter$ is a PhD student, i.e., $\Phi_7 : PhDStudent(Peter)$. This claim creates a conflict for the ontology $\mathcal{O}_E$ since $Peter$ is a $Researcher$ (Φ_6 says) and Φ_3 states $Student$ and $Researcher$ are separate. A possible solution is that we delete Φ_3. Another solution is to revise the statements by changing the axiom, i.e., we say that all students, except $Peter$, are not researchers. Namely, we can now weaken the terminology axiom as follows:

$$Student \sqcap \neg\{Peter\} \sqsubseteq \neg Researcher$$

where $\{Peter\}$ is nominal.

Next, we take account of another inconsistency, which is known as *inconsistency-tolerant semantics* (ITS). Now, we define some notions that are useful to explain the ITS.

Definition 3.35 *[LLR+10] We say that $\mathcal{A}$ is $\mathcal{T}$-consistent if $\mathcal{O}(\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle)$ is consistent, otherwise we write $\mathcal{T}$-inconsistent. A subset $\mathcal{A}' \subseteq \mathcal{A}$ is a minimal $\mathcal{T}$-inconsistent subset of $\mathcal{A}$ if $\mathcal{A}'$ is $\mathcal{T}$-inconsistent and $\mathcal{A}'' \subset \mathcal{A}'$ is $\mathcal{T}$-consistent.*

A standard way of dealing with inconsistency proceeds by first computing the set of maximal subsets of $\mathcal{A}$ that are $\mathcal{T} - consistent$. Note that a repair is often defined as a maximally consistent subset of assertions. Here, we distinguish between a maximal repair and a repair that is simply a consistent subset of assertions. This is analogous to the notions of maximal consistent sub-theory and consistent sub-theory in propositional logic. More formally, a maximal repair is defined as follows:

Definition 3.36 (Maximal repair [BBC21]) *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be an ontology or knowledge base. A subbase $\mathcal{R} \subseteq \mathcal{A}$ is a maximal repair if $\mathcal{K}$ is consistent and for $\mathcal{R}' \subseteq \mathcal{A}$. $\mathcal{R} \subset \mathcal{R}'$, $\langle \mathcal{T}, \mathcal{R}' \rangle$ is consistent. Therefore, if $\mathcal{K}$ is consistent, there is a single maximal repair $\mathcal{R} = \mathcal{A}$. We denote by MAR($\mathcal{A}$) as the set of maximal repairs of $\mathcal{A}$ w.r.t $\mathcal{T}$.*

We now recall the most well-known inconsistency-tolerant semantics proposed [ABC99, Bie12, LLR+10, Bie19]. We call q a conjunctive query. A query q is said to be:

- **AR:** A consistent consequence of $\mathcal{K}$ if $\forall \mathcal{R} \in \mathcal{R}(\mathcal{A})$, $\langle \mathcal{T}, \mathcal{R}' \rangle \models q$;
- **IAR**: an IAR consequence of $\mathcal{K}$ if $\langle \mathcal{T}, \bigcap_{\mathcal{R} \in \mathcal{R}(\mathcal{A})} \mathcal{R} \rangle \models q$;
- **CAR**: an CAR consequence of $\mathcal{K}$ if $\forall \mathcal{R} \in \mathcal{R}(Cl(\mathcal{A}))$, $\langle \mathcal{T}, \mathcal{R}' \rangle \models q$;
- **ICAR**: an ICAR consequence of $\mathcal{K}$ if $\langle \mathcal{T}, \bigcap_{\mathcal{R} \in \mathcal{R}(Cl(\mathcal{A}))} \mathcal{R} \rangle \models q$; and
- **ICR**: an ICR consequence of $\mathcal{K}$ if $\langle \mathcal{T}, \bigcap_{\mathcal{R} \in \mathcal{R}(\mathcal{A})} Cl(\mathcal{R}) \rangle \models q$.

where $Cl(\mathcal{A}) = \{\alpha \mid \langle \mathcal{T}, \mathcal{A} \rangle \models \alpha \wedge \text{terms}(\alpha) \subseteq \text{terms}(\mathcal{A})\}$.

3.6 Summary

In this chapter, we mentioned the belief change theory, which is the inspiration and fundamental for developing our methodologies. Here, we provided the notions, operators, and postulates related to this topic. Namely, we presented belief revision operators and their properties. Moreover, we also introduced belief merging operators that are the foundation and motivation for the contributions in this thesis. Naturally, conflicts and inconsistencies can appear when we merge many sources or revise a knowledge base. Hence, belief change will be one of the most consistent, faithful, and reasonable selections to handle these arising problems. Regardless of whether the input source is consistent, the final result is expected to be consistent and the best representation of the previous statements and the new claims.

Moreover, the merging, revision, and change operators for belief bases have successfully solved the problems concerning constraints and statements. Therefore, we take account of these operators to propose novel operators and take advantage of them for the ontology side. To the best of our knowledge, there are many investigations into propositional logic merging and much research on using matching and alignment to merge ontologies. However, a tiny number of studies apply belief change theory to ontology merging. Therefore, the emphasis of our study will be on examining ontology merging by using robust theoretical underpinnings already in place.

Most of the research related to ontology merging is closely related to mapping, alignments, and matching. However, this work is far from them. First, we will use mapping to find the same terminologies (for instance, holiday and vacation, employee and staff, chance and opportunity, and others). Then, after obtaining the same concepts, the merging operators will work on them. In the following sections, we will present our contributions based on the foundation of belief merging.

Chapter 4

Model-Based Merging for Exploiting Open-Domain Ontologies

In this chapter, we will develop a model-based merging operator for merging ontologies and discuss semantic conflicts (see Definition 4.1). To this end, we provide a merging ontology framework that takes inspiration from the model-based merging of Konieczny et al. [KP02, EKM12, EKM07]. In particular, our approach performs a distance calculation between the semantic results and the input sources to identify the final merged semantic result close to the inputs. Then, an IFSI algorithm (referring to Algorithm 1, page 70) is what we have come up with in order to obtain a unique semantic outcome from the set of merged semantic results. After that, we translated the merged semantic outcome to seek the merged syntactical result. In this chapter, we also provided an experiment to assess the computation time. Moreover, this chapter also provides some postulates on the merging operator.

In this study, we will work with ontologies encoded in the $\mathcal{EL}$ description logic setting (see more details in Section 2.3.3). Since we will be focusing on handling semantic conflicts, we use DL $\mathcal{EL}$ to guarantee that other conflicts do not occur. Before detailing the contribution of this chapter, we will provide a situation and motivation for undertaking this study.

4.1 Context and Motivation

As presented above, structured knowledge of concepts and properties is typically encoded using ontologies or knowledge graphs [KNPSQ20]. The difference between these two frameworks is that knowledge graphs are considerably less expressive than ontologies [HBC+21]. Therefore it can be thought of as a more straightforward form of ontologies[20]. An important point, however, is that many ontologies sharing the same knowledge are available. Several open-domain ontologies, such as SUMO, OpenCyc[21], Wikidata, Babelnet, and others, are available on the Web. Therein, they often express knowledge of a particular domain, such as food, sport, dance, and so on, using different terminology. Let us consider, for example, the concept of a Box, for which the SUMO ontology uses the concept name Box, BabelNet uses the concept name $Carton$ *(bn:00016327n)*, and WikiData uses $Parcel$ *(Q13107365)* as a concept name. This observation has led to several methods aimed at unifying and linking ontologies to each other. Therefore, ontology integration approaches, such as *ontology matching, mapping, and alignment* have emerged, e.g., [Euz17, dSRABE18, ZZ16, LGM+17, THT18] (see Section 3.4). In this work, we will not work with ontology integration approaches. We assume that our ontologies are matched to each

[20] In this chapter, we shall use ontologies to refer to knowledge graphs as well.
[21] http://sw.opencyc.org

other. The main goal of this chapter is to merge sources of knowledge. Hence, this assumption is a significant point for our approach.

For the merging problem, naturally, when expressing knowledge of a given domain and assuming that the same underlying signature (i.e., individual, concept, and role names are the same), many points of view are possible. Therefore, many types of conflicts may arise. By conflict, representing knowledge does not only refer to logical inconsistency but also conflicts in terms of semantics that might arise from the knowledge structured in the different statements. We call the fact that two sources represent the same concepts differently as *a semantic conflict*, i.e., a source states every $Book$ is a $Text$ while another says every $Text$ is a $Book$. Now, we take account of, for instance, the concept $Process$ as defined by the following three practical ontologies: SUMO, WikiData, and BabelNet.

- **Wikidata:** every $Procedure$ **is a** $Technique$, and every $Technique$ **is a** $Process$;
- **SUMO:** every $Technique$ **is a** $Procedure$, and every $Technique$ **is a** $Process$; and
- **Babelnet:** every $Technique$ and $Procedure$ **are a** $Process$.

In this example, the three ontologies are already matched to each other[22]. While there is no logical inconsistency between these ontologies, the emerging problem is that they structure the knowledge about the domain in different and potentially conflicting ways. Indeed, Wikidata says that every $Procedure$ is a $Technique$ while SUMO states that every $Technique$ is a $Procedure$. Then, one issue can occur whether we will select Wikidata's statements or SUMO's declaration for the merged result. We call the case a semantic conflict. Hence, ontology merging combines two or more ontology sources into a target ontology while solving semantic and logical conflicts between them. In this chapter, we focus on handling the conflicts that arise when the ontology sources express the knowledge in different statements (using the same terminology about a particular domain). In the foregoing example, one possible solution to deal with semantic conflicts would be to represent all the concepts as equivalent (e.g., $Procedure \equiv Technique$). However, we prefer to avoid this solution since it flattens out the hierarchy of the output ontology (i.e., the result of merging). Recall that the primary purpose of an ontology is to structure conceptual knowledge hierarchically. A possible solution is to take advantage of the majority of statements (i.e., constraints or axioms) to select the merged result. For instance, if five ontologies agree that every $Procedure$ is a $Technique$, whereas only one says that every $Technique$ is a $Procedure$, then the majority [BM91] will be followed, and the merged result will include the information that every $Procedure$ is a $Technique$.

In order to solve the problems above, we propose a method for ontology merging based on the solid theoretical foundations of propositional belief merging theory [Rev97, KP02, Kon00]. This proposal is close to *model-based* operators of the propositional logic. It has extensively been studied for the case of beliefs encoded in propositional languages. In particular, several postulates characterizing appropriate behavior of such merging operators have been proposed [KP02, Rev97] (see Section 3.3.1). In this work, we introduce ontology merging operators in terms of semantics. Noteworthy, we emphasize that these sources have already been matched before the merging process. An important point is that we use $\mathcal{EL}$ [BBL05] (c.f. Section 2.3.3), which is one of the basic description logics and is used by large-scale ontologies for practical applications, such as SNOMED CT[23], GO[24]. We select this setting because this work does not focus on bottom concepts ($\bot$), which can lead to logical conflicts. Instead, we only work on handling semantic conflicts [25].

[22] Details about the matching are described in Section 4.4.
[23] https://bioportal.bioontology.org/ontologies/SNOMEDCT
[24] http://geneontology.org/
[25] We present a formal definition in the next section.

4.2 Semantic-Based Ontology Merging

This section introduces an ontology merging process that computes distances between interpretations and input ontologies in the context of semantics merging. Therein, an algorithm to select a merged semantic result and several relevant proofs will also be provided. Finally, from the merged semantic result, we propose a method to translate from semantics to the syntactic structure of the ontology and then select the merged result based on the majority [BM91].

4.2.1 Model-Based Framework for Merging Ontologies

First, we denote a set of input merging ontologies as a profile $\mathcal{P} = \{\mathcal{O}_1, \ldots, \mathcal{O}_n\}$. Two profiles are said to be equivalent if their ontologies are the same. Formally, $\mathcal{P}_1 \equiv \mathcal{P}_2$ if and only if there is a bijection f from $\mathcal{P}_1 = \{\mathcal{O}_1, \ldots, \mathcal{O}_n\}$ to $\mathcal{P}_2 = \{\mathcal{O}'_1, \ldots, \mathcal{O}'_n\}$ such that $\mathcal{O}_i \equiv f(\mathcal{O}'_i)$. An axiom in the $\mathcal{O}$ is referred to as a claim or a statement from a source. Given a profile of ontologies describing the knowledge about a particular domain and using the same terminology (same concepts and role names), one can distinguish between two possible cases:

1. The case where the ontologies **agree** on the same claims (e.g., all ontologies claim that a concept A is subsumed by B);

2. The other case where there is **a disagreement** (or conflict) on how to express knowledge.

In the first case, there is no conflict, and the result of merging contains the same statements from the sources. In the second case, a solution that appropriately solves the conflicts is to enumerate all statements and select the most plausible ones on which the different sources agree. This case is semantic disagreement (e.g., $Technique$ is subsumed by $Procedure$ in the SUMO ontology, whereas Wikidata states that $Procedure$ is subsumed by $Technique$). Here, we call the disagreement as a semantic conflict. It is defined as follows:

Definition 4.1 (Semantic Conflict) *Let $\mathcal{P} = \{\mathcal{O}_1, \ldots, \mathcal{O}_n\}$ be a profile. We say $\mathcal{P}$ has a semantic conflict if there are concepts $A, B \in N_C$, with $A \neq B$, and ontologies $\mathcal{O}_i, \mathcal{O}_j \in \mathcal{P}$ such that $\mathcal{O}_i \models A \sqsubseteq B$ and $\mathcal{O}_i \not\models B \sqsubseteq A$ and $\mathcal{O}_j \models B \sqsubseteq A$ and $\mathcal{O}_j \not\models A \sqsubseteq B$.*

In this chapter, we follow a semantic approach for merging by first computing the set of claims close to the input ontologies. When this result is obtained, we compute an ontology from the set of selected claims that gives us the (global) merged result. Intuitively, given the signatures of the different ontologies, we enumerate all statements. These statements will be represented or transformed into a set of interpretations (e.g., a statement $A \sqsubseteq B$ represents into an interpretation $\mathcal{I} = \langle A^{\mathcal{I}} = \{a\}, B^{\mathcal{I}} = \{a, b\}\rangle$). Once these interpretations have been computed, we select the most plausible interpretations that satisfy the input sources. We propose an ontology merging framework to give a general picture of our merging process and how to deal with the existing conflicts. Our merging framework is summarized in Figure 4.1. Notice that our work goes beyond checking for logical inconsistencies that concentrate crucially on handling semantic conflicts.

The framework splits into three main components:

1. Applying an existing mapping process to generate a list of synchronized names;

2. Extracting the axioms with synchronized names *(i.e., rewrite all ontologies with the same signature)* from the sources;

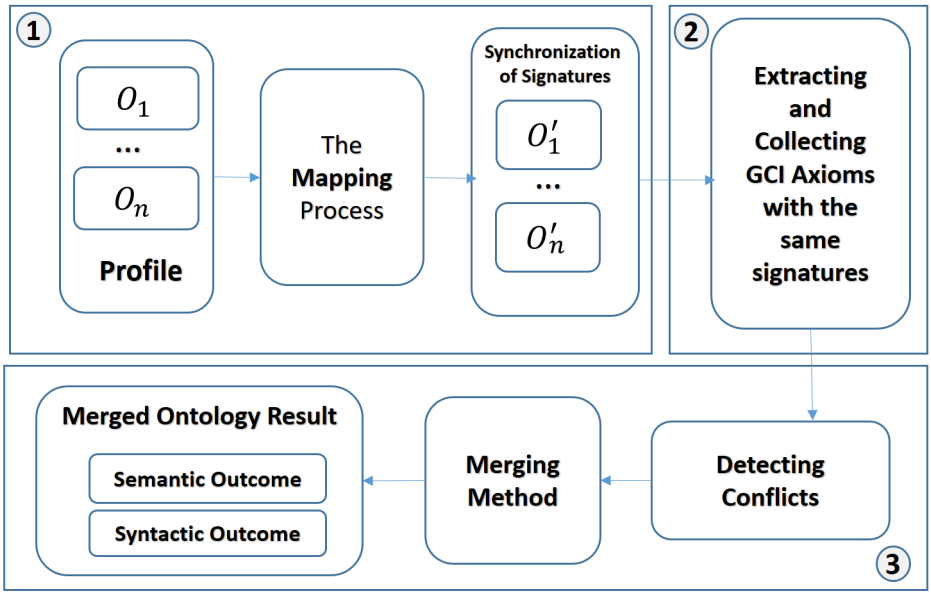


Figure 4.1: General framework of the merging process.

3. Applying the ontology merging method proposed in this chapter. The steps of the third part are the following:

   (i) Given the signatures of the different ontologies, we enumerate all possible ways to express knowledge *(therein, one possible way will be the result of merging)*;

   (ii) We express these possible statements semantically in terms of interpretations. Thus, we detect the semantic conflicts; and

   (iii) We apply belief merging theory to compute the satisfaction of interpretations w.r.t. the ontology profile.

4.2.2 Generating Possible Merging Solutions

Given the set of axioms, we focus on the axioms of the normal form [BBL05] (see Definition 2.16). Then, these rules are rewritten by a combination between the relations and the set of concept names, called *patterns*[26]. We define the patterns as follows:

Definition 4.2 (Pattern) *Let $A, B, C \in N_C$. A concept pattern is a (finite) collection of all axioms of a specific form built up from the set of atomic concepts N_C and the set of roles N_R. One of the pattern forms based on the normal form of $\mathcal{EL}$ (see Definition 2.16 and Section 2.3.4) includes:*

- $A \sqsubseteq B$ *(subsumption);*
- $A \sqcap B \sqsubseteq C$ *(intersection); and*
- $A \sqsubseteq \exists r.B$ *or* $\exists r.A \sqsubseteq B$ *(existential subsumption).*

With α we shall denote a set of concept patterns.

[26] We can see these patterns as second-order is-a relations, whose instances are the concepts from the ontologies.

The patterns correspond to the axioms in the $\mathcal{EL}$ normal form. Now, we collect combinations based on the set of concept patterns. Each combination is a set of patterns (or axioms) with at least two elements. A set of combinations is formally introduced as follows:

Definition 4.3 (Combination) *Let α be the set of concept patterns. The set of combinations, denoted by $\mathcal{H}$, is defined as follows:*

$$\mathcal{H} \stackrel{\text{def}}{=} \{X \mid X \subseteq \alpha \; and \; 2 \leq |X|\}$$

We use Definition 4.3 to generate all the possible combinations that can be expressed in the same terminology. To illustrate the generation of the combinations, let us consider three concepts N_C = {A, B, C}. We apply Definition 4.2 to combine N_C with the patterns to have the concept patterns α = {$A \sqsubseteq B$, $B \sqsubseteq C$, $A \sqcap B \sqsubseteq C$, $A \sqsubseteq \exists r.B$, $\exists r.A \sqsubseteq B$, $A \sqsubseteq C$, $C \sqsubseteq A$, $A \sqcap C \sqsubseteq B$, $B \sqcap C \sqsubseteq A$, $A \sqsubseteq \exists r.C$, $\exists r.A \sqsubseteq C$, $C \sqsubseteq B$, $B \sqsubseteq \exists r.A$, $B \sqsubseteq \exists r.C$, $\exists r.B \sqsubseteq A$, $\exists r.B \sqsubseteq C$, $\exists r.C \sqsubseteq A$, $\exists r.C \sqsubseteq B$ }. Finally, the result of the set of combinations *(use Definition 4.3)* is as follows: $\mathcal{H} = \{\{A \sqsubseteq B, A \sqsubseteq C\}, \{A \sqsubseteq B, B \sqsubseteq C\}, \{A \sqsubseteq B, C \sqsubseteq B\} \ldots\}$.

Note that these generated syntactic representations derive from the axioms in the $\mathcal{EL}$ normal form (see Definition 2.16). Hence, they will fully guarantee how to represent knowledge (axioms) in the DL $\mathcal{EL}$. Furthermore, the axiom number and the concept number of input sources are finite, and the generated syntaxes will depend on the concepts of the input sources. Therefore, we can use combinations as surrogates for the inputs to find the interpretations. In the sequel, we generate all the interpretations for the merging process based on the combination set collected.

Now, we will discuss the constraints and how to generate all possible interpretations. Here, we introduce a closure definition, which is the foundation of collecting all possible interpretations. The important point is that most real ontologies are not quite enough for the real $ABoxes$, which only have the $TBoxes$ normally. Hence, a pseudo-$ABox$ will be useful and significant in implementing and finding the merged result. Therefore, we assume a pseudo-$ABox$ in this work. Moreover, we take an inspiration from *fixed-domain semantics* for model enumeration [GRS16, RS17]. The *fixed-domain* semantics allows reasoning over an explicitly given finite domain. Indeed, we assume a version of the fixed-domain semantics is implemented for this work. Namely, the domain of interpretation is the set of individual names in such a way that each one of those individual names is mapped to itself. Here we assume the domain is merely finite (but of arbitrary, unknown size) or bounded-size. Then, we take account of the case in which the domain has *a priori known cardinality and uses a fixed domain*. This assumption guarantees that the generated interpretations (from syntactical representations) can be used as surrogates for all possible interpretations in the semantics. We emphasize that this work only carries on with interpretations of finite domains of a certain cardinality, i.e., the number of individual names in N_I (a fixed domain $\Delta \subseteq N_I$ presented in Definition 2.9). To ease the presentation for the rest of this work, we will write $\models$ as $\models_\Delta$ (see Definition 2.9) for the entailment on the fixed-domain (Δ-fix). Now, we first provide a pseudo-$ABox$ definition as:

Definition 4.4 *Let $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$. A pseudo-$ABox$ induced by $\mathcal{I}$, denoted $\mathcal{A}_{\mathcal{I}}$, is defined as:*

$$\mathcal{A}_{\mathcal{I}} \stackrel{\text{def}}{=} \{A(a) \mid a \in N_I \; and \; \mathcal{I} \models A(a)\} \cup \{r(a,b) \mid a, b \in N_I \; and \; \mathcal{I} \models r(a,b)\}$$

Here, we define $\langle \Phi, \mathcal{A}_{\mathcal{I}} \rangle \models A(a)$ (resp. $r(a,b)$) if $A(a) \in \mathcal{A}_{\mathcal{I}}$ and $\Phi \models A(a)$ (resp. $r(a,b)$). Now, we represent the closure definition to find out the full interpretations from the axioms:

Definition 4.5 (Closure) *Let Φ be an axiom, $B \in N_C$, $r \in N_R$, and $a, b \in N_I$, and let $\mathcal{A}_{\mathcal{I}}$ be a pseudo-$ABox$. With*

$$\mathcal{C}l_\Phi(\mathcal{A}_{\mathcal{I}}) \stackrel{\text{def}}{=} \{B(a) \mid \langle \Phi, \mathcal{A}_{\mathcal{I}} \rangle \models B(a)\} \cup \{r(a,b) \mid \langle \Phi, \mathcal{A}_{\mathcal{I}} \rangle \models r(a,b)\}$$

we denote the closure of $\mathcal{A}_{\mathcal{I}}$ w.r.t Φ.

An illustration to explain the closure is as follows: assuming we have an axiom $\Phi = A \sqsubseteq B$, and $\mathcal{A}_\mathcal{I}$ = $\{A(a), B(b), C(c)\}$ then the closure of Φ is $\mathcal{C}l_\Phi(\mathcal{A}_\mathcal{I}) = \{A(a), \mathbf{B(a)}, B(b), C(c)\}$. This step performs to generate interpretations based on the axioms generated. Now, we generate all possible interpretations using the closure and the set of combinations. The set of interpretations is defined as:

Definition 4.6 *Let $\mathcal{H}$ be a set of combinations and $\mathcal{A}_\mathcal{I}$ be the pseudo-ABox. The universe of interpretations, denoted by $\mathcal{U}$, is defined as:*

$$\mathcal{U} \stackrel{\text{def}}{=} \bigcup_{1 \leq i \leq |\mathcal{H}|} \mathcal{C}l_{\alpha_i}(\mathcal{A}_\mathcal{I})$$

where $\alpha = \{\Phi_1, \ldots, \Phi_n\} \in \mathcal{H}$ is a set of the axioms and

$$\mathcal{C}l_\alpha(\mathcal{A}_\mathcal{I}) \stackrel{\text{def}}{=} \{\mathcal{C}l_\Phi(\mathcal{A}_\mathcal{I}) \mid \Phi \in \alpha\}$$

We assume α= $\{C \sqsubseteq B, C \sqsubseteq A\}$ and $\mathcal{A}_\mathcal{I}$ = $\{A(a), B(b), C(c)\}$, then the closure of α is $\mathcal{C}l_\alpha(\mathcal{A}_\mathcal{I})$ = $\{A(a)$, $A(c)$, $B(b)$, $B(c)$, $C(c)\}$. Here, if $\mathcal{H}$=$\{\{A \sqsubseteq B, B \sqsubseteq C\}, \{A \sqsubseteq B, B \sqsubseteq C, C \sqsubseteq B\}\}$, we use Definitions 4.5 and 4.6 to aggregate the set of interpretations. The result of the set of all interpretations is as follows: $\mathcal{U}$ = $\{A(a)$, $B(a)$, $B(b)$, $C(a)$, $C(b)$, $C(c)\}$, $\{A(a)$, $B(a)$, $B(b)$, $B(c)$, $C(a)$, $C(b)$, $C(c)\}\}$.

For collecting interpretations, to ensure the correction of our merging method, we mention three constraint cases as follows:

1. The interpretation does not contain the empty set;
2. Each concept has at least one individual that belongs to it; and
3. The interpretation does not duplicate.

Here, for ease of presentation, we remind that we extend how to represent an interpretation $\mathcal{I}$ as a set contained the interpretation of concepts and roles in *the form of angle brackets* $\mathcal{I} = \langle \ldots \rangle$ (see Section 2.3.1), i.e., $\mathcal{I}$ = $\langle A^\mathcal{I}$ = $\{a\}$,$B^\mathcal{I}$ = $\{a, b, c\}$,$C^\mathcal{I}$ = $\{a, c\}, r^\mathcal{I} = \{(a, b), (b, c)\}\rangle$. More formally, we call the interpretations restricted by the constraints as "spurious" interpretations.

Definition 4.7 *Let $\mathcal{I}$ be an interpretation. We say $\mathcal{I}$ is spurious if one of the following three conditions holds:*

- *There is $A \in N_C$ s.t. $A^\mathcal{I} = \emptyset$; or*
- *There is $a \in N_I$ s.t. for every $A \in N_C, a^\mathcal{I} \notin A^\mathcal{I}$; or*
- *There is $\mathcal{I}' = (\Delta^{\mathcal{I}'}, \cdot^{\mathcal{I}'})$ s.t. $\Delta^\mathcal{I}$=$\Delta^{\mathcal{I}'}$ and $\cdot^\mathcal{I} = \cdot^{\mathcal{I}'}$.*

With $\mathcal{U}_S$ we denote the set of all spurious interpretations.

Finally, we generate "useful" interpretations by eliminating the "spurious" interpretations from the set of "universal" interpretations. The set of useful interpretations is defined as:

Definition 4.8 *Let $\mathcal{U}$ be the set of universal interpretations and $\mathcal{U}_S$ be the set of spurious interpretations. The set of useful interpretations, denoted by $\mathcal{W}$, is defined as follows:*

$$\mathcal{W} \stackrel{\text{def}}{=} \mathcal{U} \setminus \mathcal{U}_S$$

Combination Set ($\mathcal{H}$)	All useful Interpretations ($\mathcal{W}$)
1. $\{A \sqsubseteq B, A \sqsubseteq C\}$	1. $\langle A^{\mathcal{I}} = \{a\}, B^{\mathcal{I}} = \{a, b\}, C^{\mathcal{I}} = \{a, c\}\rangle$
2. $\{A \sqsubseteq B, C \sqsubseteq A\}$	2. $\langle A^{\mathcal{I}} = \{a, c\}, B^{\mathcal{I}} = \{a, b, c\}, C^{\mathcal{I}} = \{c\}\rangle$
3. $\{A \sqsubseteq B, A \sqsubseteq C, C \sqsubseteq A\}$	3. $\langle A^{\mathcal{I}} = \{a, c\}, B^{\mathcal{I}} = \{a, b, c\}, C^{\mathcal{I}} = \{a, c\}\rangle$
4. $\{A \sqcap C \sqsubseteq B\}$	4. $\langle A^{\mathcal{I}} = \{a\}, B^{\mathcal{I}} = \{a, b\}, C^{\mathcal{I}} = \{a, b, c\}\rangle$
5. $\{A \sqsubseteq B, A \sqsubseteq \exists r.C\}$	5. $\langle A^{\mathcal{I}} = \{a\}, B^{\mathcal{I}} = \{a, b, c\}, C^{\mathcal{I}} = \{c\}\rangle$
6. $\{A \sqsubseteq B, \exists r.A \sqsubseteq C\}$	6. $\langle A^{\mathcal{I}} = \{a\}, B^{\mathcal{I}} = \{a, b, c\}, C^{\mathcal{I}} = \{a, b, c\}\rangle$
7. $\{A \sqsubseteq B, B \sqsubseteq C\}$	7. $\langle A^{\mathcal{I}} = \{a, b\}, B^{\mathcal{I}} = \{b\}, C^{\mathcal{I}} = \{a, b, c\}\rangle$
8. $\{A \sqsubseteq B, C \sqsubseteq B\}$	8. $\langle A^{\mathcal{I}} = \{a, b, c\}, B^{\mathcal{I}} = \{b\}, C^{\mathcal{I}} = \{c\}\rangle$
9. $\{A \sqsubseteq B, B \sqsubseteq C, C \sqsubseteq B\}$	9. $\langle A^{\mathcal{I}} = \{a, b, c\}, B^{\mathcal{I}} = \{b\}, C^{\mathcal{I}} = \{a, b, c\}\rangle$
10. $\{A \sqcap B \sqsubseteq C\}$	10. $\langle A^{\mathcal{I}} = \{a, b\}, B^{\mathcal{I}} = \{b\}, C^{\mathcal{I}} = \{b, c\}\rangle$
11. $\{A \sqsubseteq B, B \sqsubseteq \exists.C\}$	11. $\langle A^{\mathcal{I}} = \{a, b, c\}, B^{\mathcal{I}} = \{b, c\}, C^{\mathcal{I}} = \{c\}\rangle$
12. $\{A \sqsubseteq B, \exists r.B \sqsubseteq C\}$	12. $\langle A^{\mathcal{I}} = \{a, b, c\}, B^{\mathcal{I}} = \{b, c\}, C^{\mathcal{I}} = \{b, c\}\rangle$
13. $\{B \sqsubseteq A, A \sqsubseteq C\}$	13. $\langle A^{\mathcal{I}} = \{a, b\}, B^{\mathcal{I}} = \{a, b\}, C^{\mathcal{I}} = \{a, b, c\}\rangle$
14. $\{B \sqsubseteq A, C \sqsubseteq A\}$	14. $\langle A^{\mathcal{I}} = \{a, b, c\}, B^{\mathcal{I}} = \{a, b, c\}, C^{\mathcal{I}} = \{c\}\rangle$
...	...

Table 4.1: The collected results of the combinations and all interpretations.

We illustrate the generation of the combinations and the set of all possible interpretations corresponding to a profile of ontologies in the following example:

Example 4.1 *We assume an example of a profile $\mathcal{P}_1$ of three ontologies $\{\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3\}$ in which three concepts A, B, C have the same names:*

$$\mathcal{P}_1 = \{\mathcal{O}_1 : \{C \sqsubseteq B, A \sqsubseteq B\}, \mathcal{O}_2 : \{A \sqsubseteq B, B \sqsubseteq C\}, \mathcal{O}_3 : \{A \sqsubseteq C, C \sqsubseteq B\}\}.$$

First of all, from the combination between the concept names $N_C^2 = \{A, B, C\}$ and four forms (corresponding to the concept patterns), we generate 18 patterns (α = $\{A \sqsubseteq B$, $B \sqsubseteq C$, $A \sqcap B \sqsubseteq C$, $A \sqsubseteq \exists r.B$, $\exists r.A \sqsubseteq B$, $A \sqsubseteq C$, $C \sqsubseteq A$, $A \sqcap C \sqsubseteq B$, $B \sqcap C \sqsubseteq A$, $A \sqsubseteq \exists r.C$, $\exists r.A \sqsubseteq C$, $C \sqsubseteq B$, $B \sqsubseteq \exists r.A$, $B \sqsubseteq \exists r.C$, $\exists r.B \sqsubseteq A$, $\exists r.B \sqsubseteq C$, $\exists r.C \sqsubseteq A$, $\exists r.C \sqsubseteq B$ $\}$.). Next step, we apply Definition 4.3 to collect the set of combinations. (e.g., $\{A \sqcap C \sqsubseteq B\}$). The set of combinations $\mathcal{H}$ is in column 2 of Table 4.1.

After the set of combinations $\mathcal{H}$ obtains, we collect all possible interpretations based on the initial interpretation $\langle A^{\mathcal{I}}$=$\{a\}$, $B^{\mathcal{I}}$=$\{b\}$, $C^{\mathcal{I}}$=$\{c\}$, $r^{\mathcal{I}}$=$(a, b)\rangle$. Applying Definition 4.6, the total number of collected interpretations is $(\Sigma_{k\in\mathcal{H}}|(N_I)_k|)$=104 interpretations. The set of combinations is in column 3 of Table 4.1. Notice that each interpretation $\mathcal{I}$ must satisfy Definition 4.7 and must contain the individuals a, b, c, i.e., $\langle A^{\mathcal{I}}$=$\{a, b, c\}$, $B^{\mathcal{I}}$=$\{b, c\}$, $C^{\mathcal{I}}$=$\{c\}$, The purpose of this work is to avoid losing any pieces of information.

After collecting the set of all possible interpretations, we introduce an ontology merging method based on the belief merging theory in the next section.

4.2.3 Selection of the Best Interpretation

In this section, we provide ontology merging operators and explain how to compute the distance between the interpretation $\mathcal{I}$ and the ontology $\mathcal{O}$. The aim of using a distance is to seek the interpretations satisfying the input ontologies. A model-based operator defines to select the interpretations which satisfies the ontologies.

First, we compute the distance between one interpretation and the ontology $\mathcal{O}$ through axioms $\Phi_i \in \mathcal{O}$ as follows:

Definition 4.9 (Distance) *Let $\mathcal{I}$ be an $\mathcal{EL}$ interpretation and let $\mathcal{O}$ be an ontology. With*

$$dist(\mathcal{I}, \mathcal{O}) \stackrel{\text{def}}{=} \mid \{\Phi_i \in \mathcal{O} \mid \mathcal{I} \not\models \Phi_i\} \mid$$

We denote the distance between $\mathcal{I}$ and $\mathcal{O}$.

The distance between $\mathcal{I}$ and $\mathcal{O}$ is the number of axioms in $\mathcal{O}$ that are not satisfied in $\mathcal{I}$. This is also known as the disagreement of $\mathcal{I}$ with $\mathcal{O}$. In this work, the disagreement compute at axiom level. Notice that, when calculating the distance of an interpretation to the ontology $\mathcal{O}$, we take into account both explicit and implicit knowledge. Normally, an implicit knowledge departs from explicit one. Implicit knowledge about concepts and individuals can be inferred automatically with the help of inference procedures. In particular, subsumption relationships between concepts and instance relationships between individuals and concepts. Moreover, if only explicit knowledge is assumed, then the distance of an interpretation to two equivalent ontologies could be different. Now, we take account of the following example: given $\mathcal{O}_1$=$\{A \sqsubseteq B, B \sqsubseteq C\}$ and $\mathcal{O}_2$ = $\{A \sqsubseteq B, B \sqsubseteq C, A \sqsubseteq C\}$. It is clear that the two ontologies are equivalent, but $\mathcal{O}_1$ contains fewer axioms than $\mathcal{O}_2$. Given an interpretation $\mathcal{I}$ = $\langle A^{\mathcal{I}} = \{a, b, c\}, \{B^{\mathcal{I}} = \{b\}, \{C^{\mathcal{I}} = \{c\}\rangle$, then $dist(\mathcal{I}, \mathcal{O}_1)$ = $dist(\mathcal{I}, \mathcal{O}_2)$ = 2 when applying the closure, but it is not the case if we only use explicit knowledge (i.e., $dist(\mathcal{I}, \mathcal{O}_1)$ = 2 is different from $dist(\mathcal{I}, \mathcal{O}_2)$ = 3).

In order to conduct the merging based on the distance, we start from Konieczny's studies [KP02, Kon00, KM04]. From these works, an aggregation function must be used to evaluate the distance between an interpretation and a profile. The operators are proposed to calculate the distances between interpretations and profiles. Hence, ontology merging operators are also based on the aggregation of these distances. The idea is to find the closet interpretation to the profile. Now, we call an aggregation function g [PKM14], then we define:

Definition 4.10 *An aggregation function g (i.e., $g \in \{\max, \Sigma, leximax\}$) associates a non-negative number to every finite tuple of non-negative numbers, such that:*

- *if $x \leq y$, then $g(x_1, \ldots, x, \ldots, x_n) \leq g(x_1, \ldots, y, \ldots, x_n)$;*
- *$g(x_1, \ldots, x_n) = 0$ iff $x_1 = \ldots = x_n = 0$; and*
- *for every non-negative number x, $g(x) = x$.*

In order to merge the ontologies in the profile $\mathcal{P}$, we define the distance from an interpretation to a profile as follows:

Definition 4.11 *Let $\mathcal{I}$ be an interpretation, let $\mathcal{P}$ be a profile, and g be an aggregation function. We define the distance $dist_g$ as:*

$$dist_g(\mathcal{I}, \mathcal{P}) \stackrel{\text{def}}{=} g(dist(\mathcal{I}, \mathcal{O}) \mid \mathcal{O} \in \mathcal{P})$$

where $g \in \{\max, \Sigma, leximax\}$.

In this chapter, we focus on the aggregation functions[27], that echoes to the most usual propositional merging operators. We have an assignment that maps each profile $\mathcal{P}$ to a pre-order $\leq_{\mathcal{P}}$ over interpretations (on $\mathcal{W}$), referring to the chapter of Konieczny et al. [KPP11]. When an interpretation $\mathcal{I}$ is a model of the $\mathcal{O}$, we provide a formal definition of the notion of syncretic assignment.

[27] We write $Gmax$ instead of $leximax$ like in [KP02].

Definition 4.12 *A syncretic assignment is a function mapping each a profile $\mathcal{P}$ to a pre-order $\leq_{\mathcal{P}}$ over interpretations $\mathcal{I}, \mathcal{J}$ such that for any profile $\mathcal{P}$, $\mathcal{P}_1$, $\mathcal{P}_2$ and for any ontologies $\mathcal{O}$, $\mathcal{O}'$ the following conditions hold:*

- *If $\mathcal{I} \models \mathcal{P}$ and $\mathcal{J} \models \mathcal{P}$, then $\mathcal{I} \simeq_{\mathcal{P}} \mathcal{J}$;*
- *If $\mathcal{I} \models \mathcal{P}$ and $\mathcal{J} \not\models \mathcal{P}$, then $\mathcal{I} <_{\mathcal{P}} \mathcal{J}$;*
- *If $\mathcal{P}_1 \equiv \mathcal{P}_2$, then $\leq_{\mathcal{P}_1} = \leq_{\mathcal{P}_2}$;*
- *For all $\mathcal{I}$ s.t. $\mathcal{I} \models \mathcal{O}$ and for some $\mathcal{J}$ s.t. $\mathcal{J} \models \mathcal{O}'$, then $\mathcal{I} \leq_{\mathcal{O} \cup \mathcal{O}'} \mathcal{J}$;*
- *If $\mathcal{I} \leq_{\mathcal{P}_1} \mathcal{J}$ and $\mathcal{I} \leq_{\mathcal{P}_2} \mathcal{J}$, then $\mathcal{I} \leq_{\mathcal{P}_1 \cup \mathcal{P}_2} \mathcal{J}$; and*
- *If $\mathcal{I} <_{\mathcal{P}_1} \mathcal{J}$ and $\mathcal{I} \leq_{\mathcal{P}_2} \mathcal{J}$, then $\mathcal{I} <_{\mathcal{P}_1 \cup \mathcal{P}_2} \mathcal{J}$.*

where we denote $\leq_{\mathcal{P}_1 \cup \mathcal{P}_2}$ as a pre-order over interpretations for a union set of $\mathcal{P}_1$ and $\mathcal{P}_2$. Similarly, we write $\leq_{\mathcal{O} \cup \mathcal{O}'}$ as a pre-order over interpretations for a union set of $\mathcal{O}'$ and $\mathcal{O}$.

Before representing the ontology merging operators, we denote $Mod(\Phi)$ as the axiom's model of ontology, i.e., $Mod(\Phi) \stackrel{\text{def}}{=} \{\mathcal{I} \in \mathcal{W} \mid \mathcal{I} \models \Phi\}$. Furthermore, we also denote $Mod(\mathcal{O})$ as the set of models of ontology $Mod(\mathcal{O}) \stackrel{\text{def}}{=} \{\mathcal{I} \in \mathcal{W} \mid \mathcal{I} \models \Phi \; for \; every \; \Phi \in \mathcal{O}\}$. We also denote $\leq^g_{\mathcal{P}}$ as a pre-order over interpretations for a profile $\mathcal{P}$ using an aggregation function g (i.e., $g \in \{\max, \Sigma, leximax\}$). Now, we define our ontology merging operators $\oplus_g$ as follows:

Definition 4.13 *Let $\mathcal{P}$ be a profile of ontologies, g be an aggregation function, and $\mathcal{I}, \mathcal{J} \in \mathcal{W}$. We let*

$$\mathcal{I} \leq^g_{\mathcal{P}} \mathcal{J} \text{ iff } dist_g(\mathcal{I}, \mathcal{P}) \leq dist_g(\mathcal{J}, \mathcal{P})$$

We then define the ontology merging operator $\oplus_g$ as

$$Mod(\oplus_g(\mathcal{P})) \stackrel{\text{def}}{=} \min(\mathcal{W}, \leq^g_{\mathcal{P}})$$

The result of the ontology merging operator can be a set of interpretations when there are many interpretations that are the same shortest distance. However, the expected ontology merging result will be one interpretation and one ontology. Hence, once the set of close interpretations using the operators is collected, we build an ontology that corresponds to these interpretations following the procedure described in the next section. One significant point is that the selected distance will be a flexible parameter so that users can customize the choice to accommodate the system's requirements. Now, we take account of the running example to demonstrate how the operator works.

Example 4.2 *Let us continue with Example 4.1. First of all, the calculation of the distance between an interpretation $\mathcal{I}$ and each ontology $\mathcal{O}$ is as follows: assuming we have the interpretation*

$$\mathcal{I}_1 = \langle A^{\mathcal{I}_1} = \{a, b, c\}, B^{\mathcal{I}_1} = \{b\}, C^{\mathcal{I}_1} = \{b, c\}, r^{\mathcal{I}_1} = (a, b)\rangle$$

We compute the distance between $\mathcal{I}_1$ and $\mathcal{P}_1$ as follows: $dist(\mathcal{I}_1, \mathcal{O}_1) = d^{\mathcal{I}_1}_{\mathcal{O}_1} = 2$ because an axiom of $\mathcal{O}_1$ is unsatisfied by $\mathcal{I}_1$ (namely $\mathcal{I}_1 \not\models A \sqsubseteq B$ and $\mathcal{I}_1 \not\models C \sqsubseteq B$); similarly, $d^{\mathcal{I}_1}_{\mathcal{O}_2} = 1$, since for $\mathcal{I}_1 \not\models A \sqsubseteq B$ and $\mathcal{I}_1 \models B \sqsubseteq C$; $d^{\mathcal{I}_1}_{\mathcal{O}_3} = 2$ due to $\mathcal{I}_1 \not\models A \sqsubseteq C$ and $\mathcal{I}_1 \not\models C \sqsubseteq B$. Accordingly, we have the distance between an interpretation and a profile by applying Definitions 4.11 and 4.13 as follows:

$$dist_{\max}(\mathcal{I}_1, \mathcal{P}_1) = max(d^{\mathcal{I}_1}_{\mathcal{O}_1}, d^{\mathcal{I}_1}_{\mathcal{O}_2}, d^{\mathcal{I}_1}_{\mathcal{O}_3}) = max(2, 1, 2) = 2;$$

All useful interpretations	$\mathcal{O}_1$	$\mathcal{O}_2$	$\mathcal{O}_3$	d_{max}	d_Σ	d_{Gmax}
$\langle A^\mathcal{I} = \{a\}, B^\mathcal{I} = \{a,b,c\}, C^\mathcal{I} = \{a,b,c\}\rangle$	0	0	0	0	0	(0,0,0)
$\langle A^\mathcal{I} = \{a,b\}, B^\mathcal{I} = \{a,b,c\}, C^\mathcal{I} = \{a,b,c\}\rangle$	0	0	0	0	0	(0,0,0)
$\langle A^\mathcal{I} = \{a,c\}, B^\mathcal{I} = \{a,b,c\}, C^\mathcal{I} = \{a,b,c\}\rangle$	0	0	0	0	0	(0,0,0)
$\langle A^\mathcal{I} = \{a,c\}, B^\mathcal{I} = \{a,b,c\}, C^\mathcal{I} = \{a,c\}\rangle$	0	1	0	1	1	(1,0,0)
$\langle A^\mathcal{I} = \{a\}, B^\mathcal{I} = \{a,b,c\}, C^\mathcal{I} = \{a,c\}\rangle$	0	1	0	1	1	(1,0,0)
...						
$\langle A^\mathcal{I} = \{a,b,c\}, B^\mathcal{I} = \{a,b,c\}, C^\mathcal{I} = \{a,c\}\rangle$	0	1	1	1	2	(1,1,0)
...						
$\langle A^\mathcal{I} = \{a\}, B^\mathcal{I} = \{a,b\}, C^\mathcal{I} = \{a,b,c\}\rangle$	1	1	1	1	3	(1,1,1)
$\langle A^\mathcal{I} = \{a\}, B^\mathcal{I} = \{a,b\}, C^\mathcal{I} = \{a,c\}\rangle$	1	1	1	1	3	(1,1,1)
...						
$\langle A^\mathcal{I} = \{a,b,c\}, B^\mathcal{I} = \{b,c\}, C^\mathcal{I} = \{c\}\rangle$	1	2	1	2	4	(2,1,1)
$\langle A^\mathcal{I} = \{a,b,c\}, B^\mathcal{I} = \{b\}, C^\mathcal{I} = \{b,c\}\rangle$	2	2	1	2	5	(2,2,1)
...						

Table 4.2: Distances between interpretations and ontologies of $\mathcal{P}_1$.

$$dist_\Sigma(\mathcal{I}_1, \mathcal{P}_1) = \sum_{i=1}^{3}(d_{\mathcal{O}_i}^{\mathcal{I}_1}) = (2+1+2) = 5;$$
$$dist_{Gmax}(\mathcal{I}_1, \mathcal{P}_1) = (2,2,1).$$

Regarding Example 4.1, the result of computing the distances is shown in Table 4.2 (where $d_{\max}$, d_Σ, and d_{Gmax} are $dist_{\max}$, $dist_\Sigma$, and $dist_{Gmax}$, respectively). We enumerate all possible interpretations and compute the distance between all possible interpretations and the ontologies $\mathcal{P} = \{\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3\}$. Moreover, we also apply the aggregation functions g for calculating the distance. From these outcomes, the three closest interpretations selected with the minimal distance (i.e., $dist_{\max} = 0$) include:

$$\begin{aligned} Mod(\oplus_g(\mathcal{P}_1)) = \{\ & \langle A^\mathcal{I} = \{a\}, B^\mathcal{I} = \{a,b,c\}, C^\mathcal{I} = \{a,b,c\}\rangle, \\ & \langle A^\mathcal{I} = \{a,b\}, B^\mathcal{I} = \{a,b,c\}, C^\mathcal{I} = \{a,b,c\}\rangle, \\ & \langle A^\mathcal{I} = \{a,c\}, B^\mathcal{I} = \{a,b,c\}, C^\mathcal{I} = \{a,b,c\}\rangle\ \}. \end{aligned}$$

From this result, we introduce the method to seek one interpretation and one ontology for representing the merging ontology result in the next section.

4.2.4 Expressing the Result of Merging

In the previous section, we have shown how to compute the set of interpretations representing the information close to the profile. The computed interpretations are known as the "intermediate" results because the final merged expected result is only an ontology. Hence, we extract the set of axioms held in those interpretations in order to obtain an ontology. Namely, we translate the merged results into an ontology. In addition, another reason is that the result should be one interpretation since we can not model disjunction in this framework. To be specific, the description logic $\mathcal{EL}$ does not allow a concept union $C \sqcup D$. To this end, we first build an algorithm called *Individuals Frequency in Selected-Interpretation* (IFSI for short) to seek the one representative interpretation. Then we translate the unique interpretation into the syntactic side.

Algorithm 1: IFSI (Individuals Frequency in Selected Interpretation)

input: κ: List of selected interpretations
output: $IFSI$: One Interpretation

```
1  begin
       // ν is to collect one interpretation
2      ν, IFSI ⟵ ∅
       // Collect all individuals into each concept for one interpretation ν.
3      foreach α ∈ κ do
4          foreach C, I ∈ α do
5              ν[C] ← ν[C] ∪ I
       // Filter out the frequent individuals in ν.
6      foreach C, ϑ ∈ ν do
           // Frequency function counts the frequency of individuals.
7          ℱ ← Frequency(ϑ)
8          τ ← |κ|
           // ℱ: e is an individual, num = count_A(e).
9          foreach e, num ∈ ℱ do
10             if num = τ then
11                 IFSI[C] ← IFSI[C] ∪ e
12     return IFSI
```

Determining the unique semantic merged result

We build an IFSI algorithm to determine the final semantic result. The algorithm's idea is to seek the frequency of individuals of each concept. It includes three main parts, such as

- We collect individuals from the selected interpretations;
- We put them into a unique interpretation for each specific concept; and
- We filter the most frequent instances.

An individual is collected if it appears in all the selected interpretations. We here call an individual that always exists in all the selected interpretations as the "reliable" individual. e.g., we assume there are $\mathcal{I}_1 = \langle \mathcal{A}^{\mathcal{I}} = \{a, b\}\rangle$ and $\mathcal{I}_2 = \langle \mathcal{A}^{\mathcal{I}} = \{a\}\rangle$, then, in this case, the individual a will be the most reliable individual. Notice that we only assume assertions of free open-domain ontologies, and we only focus on merging terminological knowledge. Hence, a pseudo-$ABox$ is then assumed in this chapter. Notably, although we only assume the method uses a pseudo-$ABox$, in the case of an existing real-$ABox$, our approach could also handle semantic conflicts. This improvement of the framework is left for future work. However, in the next sections, we show with the experiments that we obtain very sensible results even without this real $ABox$.

Here, $IFSI$ is a function name and also a call to Algorithm 1. Let us now define the full semantic merging process as follows:

Definition 4.14 *Let $\mathcal{P}$ be a profile of ontologies. $f_{\oplus}$ is a full semantic merging operator if $\oplus$ is an ontology merging operator and*

$$f_{\oplus}(\mathcal{P}) \stackrel{\text{def}}{=} IFSI(Mod(\oplus_g(\mathcal{P})))$$

Let us describe the IFSI algorithm *(Algorithm 1)* as follows: the input of the IFSI algorithm is the list of selected interpretations denoted as κ. The output is one "frequent" interpretation denoted as $IFSI$, in which the individuals frequently appear in the input source. This algorithm includes two main parts:

1. First *(line 2 to 5)*, we assume the individuals in each concept, include $a \in A$, $b \in B$, $c \in C$[28]. Let ν be "one variant interpretation" (including a concept and a list of individuals denoted by ϑ) and $\nu[C]$ be a list of individuals of C in ν, i.e., $\nu[C] = [a, b, c]$ or $\nu = \langle C^{\mathcal{I}} = [a, b, c]\rangle$. Here, in line 3, each $\mathcal{I}$ is an interpretation in κ; in the line 4, $Inst$ is a set of individuals and C is a corresponding concept, i.e., $Inst = \{a, b, c\}, Inst \in C^{\mathcal{I}}$, represented as $C^{\mathcal{I}} = \{a, b, c\}$. Then, at line 5, we collect the individuals into ν for each concept. For Example 4.2, we collect all the individuals of selected interpretations for each concept. The outcome is

$$\nu = \langle A^{\mathcal{I}} = [a, a, a, b, c],$$

$$B^{\mathcal{I}} = [a, a, a, b, b, b, c, c, c],$$

$$C^{\mathcal{I}} = [a, a, a, b, b, b, c, c, c]\rangle$$

   where the input will be $\{\langle A^{\mathcal{I}}{=}\{a\}, B^{\mathcal{I}} = \{a, b, c\}, C^{\mathcal{I}} = \{a, b, c\}\rangle; \langle A^{\mathcal{I}} = \{a, b\}, B^{\mathcal{I}}{=}\ \{a, b, c\}, C^{\mathcal{I}} = \{a, b, c\}\rangle; \langle A^{\mathcal{I}} = \{a, c\}, B^{\mathcal{I}}{=}\ \{a, b, c\}, C^{\mathcal{I}} = \{a, b, c\}\rangle\}$.

2. Second *(line 6 to 11)*, we compute the frequency of the individuals in each concept. Let $count_{\vartheta}(a)$ be the number of occurrences of a in ϑ. Then, the $Frequency$ function is defined as

$$Frequency(\vartheta) \stackrel{\text{def}}{=} \{e : count_{\vartheta}(e) \mid e \in \vartheta\}$$

   Here, in line 7, $\mathscr{F}_C$ denotes a counted set of the individuals in C, i.e., $\vartheta = [a, a, b]$, then $\mathscr{F}_C = \{a : 2, b : 1\}$. For Example 4.2, the results include:

$$\mathscr{F}_A = \{a : 3, b : 1, c : 1\},$$

$$\mathscr{F}_B = \{a : 3, b : 3, c : 3\},$$

$$\mathscr{F}_C = \{a : 3, b : 3, c : 3\}$$

   (line 7). Next, we seek out one "frequent" (or "faithful") interpretation. It is faithful because all individuals in each concept are reliable. We denote a "reliable" individual as an individual that appears in all concepts. We here use a threshold τ (line 8) to filter out the "reliable" individuals in each concept (lines 9 to 11). The threshold selected equals the size of the input set (κ) at line 10 ($\tau = |\kappa|$). So, the desired individuals hold a highly reliable threshold. At line 11, let $IFSI$ be "one interpretation" and $IFSI[C]$ be a list of individuals of C in $IFSI$. We collect the individual $inst$ into $IFSI$ for each concept. Regarding the example $\mathcal{P}_1$, $\tau = 3$, the selected result using the IFSI algorithm is as follows: $f_{\oplus}(\mathcal{P})$= $\langle A^{\mathcal{I}}{=}\{a\}, B^{\mathcal{I}}{=}\{a, b, c\}, C^{\mathcal{I}}{=}\{a, b, c\}\rangle$. Here we denote $\mathcal{I}^M = f_{\oplus}(\mathcal{P})$ as a final merged semantic result.

Algorithm 1 runs in a polynomial time on the size of the list of selected interpretation, given assign to an NP oracle in one step (line 6). Indeed, (i) computing the frequency of the individuals in each concept is in $O(n)$, and (ii) calculating the numbers of individuals is computed in $O(1)$ since the condition to filter the selected concepts is in $O(1)$ (line 10) and assigning the e into $IFSI$ is also in $O(1)$. This makes the complexity of Algorithm 1 in FP^P since it depends on the input selected interpretation.

[28] All interpretations contain these individuals.

Translating the merged semantic result to syntactic result

After collecting one "faithful" interpretation (resulting from the IFSI algorithm), we translate this into an ontology. To this end, we take advantage of the set of all axioms (concept patterns) α to seek the final ontology in which the merged semantic result is satisfied. We pick the merged syntactic result in α since it contains all possible axiom cases of ontology. A detailed explanation is that α contains the original axioms (in the $\mathcal{EL}$ normal form), which is also the first foundation for generating combinations and all interpretations. Therefore, the merged result will be axioms in α.

In detail, we represent the idea in the formal definitions. Now, we define a set of the axioms in which $\mathcal{I}^M$ is satisfied. It is also known as the merged syntactic result.

Definition 4.15 *Let α be a set of axioms (concept patterns), $f_\oplus$ be a full ontology merging operator, and $\mathcal{P}$ be a profile. A merged syntactic result, denoted by $\mathcal{T}^M$, is defined as follows:*

$$\mathcal{T}^M(\mathcal{P}) \stackrel{\text{def}}{=} \{\Phi \in \alpha \mid \mathcal{I}^M \models \Phi\}$$

where $\mathcal{I}^M = f_\oplus(\mathcal{P})$.

Intuitively, the syntactic result of merging depends entirely on the merged semantic result. In other words, the merged semantic result determines how the syntactic result is expressed. It is worth noting that we not only find axioms that satisfy most of the input sources but also discover new ones that fit the semantic outcome. Let us continue a running example to illustrate how to seek out the merged syntactic result.

Example 4.3 *Taking account of the running example (Example 4.2), the merged semantic result is $f_\oplus(\mathcal{P})$= $\langle A^\mathcal{I}$=$\{a\}$, $B^\mathcal{I}$=$\{a, b, c\}$, $C^\mathcal{I}$=$\{a, b, c\}\rangle$. Now, we apply Definition 4.15, then the result $\mathcal{T}^M$ is as follows:*

$$\mathcal{T}^M(\mathcal{P}) = \{A \sqsubseteq B, A \sqsubseteq C, B \sqsubseteq C, C \sqsubseteq B, A \sqsubseteq B \sqcap C, A \sqcap B \sqsubseteq C, A \sqcap C \sqsubseteq B\}.$$

In the case of expanding the selected distance, we select the close distance as $dist_{\max} = 1$. Then, the merged semantic result is $f_\oplus(\mathcal{P})$= $\langle A^\mathcal{I}$=$\{a\}$, $B^\mathcal{I}$=$\{a, b, c\}$, $C^\mathcal{I}$=$\{a, b\}\rangle$. Therefore, the merged syntactic result is

$$\mathcal{T}^M(\mathcal{P}) = \{A \sqsubseteq B, A \sqsubseteq C, C \sqsubseteq B, A \sqsubseteq B \sqcap C, A \sqcap B \sqsubseteq C, A \sqcap C \sqsubseteq B\}.$$

In $\mathcal{T}^M(\mathcal{P})$, since the axioms might be duplicate and redundant when representing knowledge implicitly, we apply normalization rules (see Section 2.3.4) to collect the axioms in the $\mathcal{EL}$ normal form, e.g., we have the axiom $A \sqsubseteq B \sqcap C$ as $\{A \sqsubseteq B, A \sqsubseteq C\}$. Here, we eliminate these axioms since both $A \sqsubseteq B$ and $A \sqsubseteq C$ are belonging to $\mathcal{T}^M(\mathcal{P})$. Here, we denote $\mathcal{N}$ as the complete rule algorithm of $\mathcal{EL}$, which applies normalization rules to obtain the axioms in normal form [BBL05]. Finally, a final merged syntactic outcome is the normalization of $\mathcal{T}^M(\mathcal{P})$ using $\mathcal{N}$.

Example 4.4 *Let us continue with an example 4.3. Now, we leverage $\mathcal{N}$ to normalize $\mathcal{T}^M(\mathcal{P})$, a final syntactic merged result is as follows:*

$$\mathcal{T}^M(\mathcal{P}) = \{A \sqsubseteq B, A \sqsubseteq C, C \sqsubseteq B, A \sqcap B \sqsubseteq C, A \sqcap C \sqsubseteq B\}.$$

The final syntactic merged result is reasonable, faithful, and consistent, since two sources state $C \sqsubseteq B$ ($\mathcal{O}_1$ and $\mathcal{O}_3$) and one source $\mathcal{O}_2$ says $B \sqsubseteq C$. This merging operator always covers the majority of the property.

Overall, the merged results are consistent, faithful, and reasonable. Moreover, the result is an acceptable and fair representation/representative of the input sources, works with the majority property, and will stay original (steady) when merging the same sources.

Proposition 4.1 *Let $\mathcal{P}$ be a profile and $\mathcal{O}_i \in \mathcal{P}$. If merging same ontologies, for all i such that $\mathcal{O}_i = \mathcal{O}_1$, then $\mathcal{T}^M(\mathcal{P}) = \mathcal{O}_1$.*

In the sequel, we will show the properties of the merging operator related to their rational behavior.

4.3 Postulates

In this section, we provide a characterization of ontology merging operators, i.e., we give a set of properties an operator has to satisfy in order to have rational behavior concerning ontology merging.

Let $\mathcal{P}$ be a profile, and let $\oplus_g$ be an ontology merging operator which assigns to each ontology set $\mathcal{P}$ a knowledge base $\oplus_g(\mathcal{P})$. Let $\mathcal{O}_1$, $\mathcal{O}_2 \in \mathcal{P}$. The ontology merging operator $\oplus_g$ satisfies the following postulates:

- **(P1)** $\oplus_g(\mathcal{P})$ is satisfiable;
- **(P2)** If $\mathcal{P}$ is satisfiable, then $\oplus_g(\mathcal{P}) = \bigcup \mathcal{P}$;
- **(P3)** If $\mathcal{P}_1 \equiv \mathcal{P}_2$, then $\oplus_g(\mathcal{P}_1) = \oplus_g(\mathcal{P}_2)$;
- **(P4)** If $\mathcal{O}_1 \cap \mathcal{O}_2$ is not satisfiable, then $\oplus_g(\mathcal{O}_1 \cup \mathcal{O}_2) \not\models \mathcal{O}_1$;
- **(P5)** $\oplus_g(\mathcal{P}_1) \cap \oplus_g(\mathcal{P}_2) \models \oplus_g(\mathcal{P}_1 \cup \mathcal{P}_2)$;
- **(P6)** If $\oplus_g(\mathcal{P}_1) \cap \oplus_g(\mathcal{P}_2)$ is satisfiable, then $\oplus_g(\mathcal{P}_1 \cup \mathcal{P}_2) \models \oplus_g(\mathcal{P}_1) \cap \oplus_g(\mathcal{P}_2)$.

These postulates are the basic properties an ontology merging operator has to satisfy since they give a rational behavior concerning the merging. The postulates' meaning is as follows: we always want to use the ontologies from a profile that is forced by (P1); all these ontologies from the profile are satisfiable. (P2) assures us that the result of the merging will be the disjunction of the profiles. (P3) states that the operator $\oplus_g$ simply obeys the syntax-irrelevancy principle. If two profiles are equivalent, then the resulting ontologies of each profile from the merging will also be logically equivalent. (P4) is a fairness postulate. It says that when merging the ontologies, the merging operator must not give any preference to one of them. (P5) expresses that if two profiles $\mathcal{P}_1$ and $\mathcal{P}_2$ agree on several alternatives, then these alternatives will be chosen if we join them. (P5) and (P6) mean that if two profiles agree on at least one alternative, then the result is those alternatives on which the two groups are in agreement. Note that the postulates (P5) and (P6) have been given by Revesz [Rev97] for weighted model fitting operators.

Proposition 4.2 *The model-based merging operator computed by the IFSI algorithm satisfies P1-P6.*

Note that we here provide the soundness of our framework with postulates and propositions. Regarding the completeness side, we implement an experimental evaluation showing the ability of our whole framework in practice. Therein, we present a qualitative analysis of the outputs and a discussion of how intuitive the results are for the test cases under consideration. We now present an experimental evaluation with our framework for practical sources in the next section.

4.4 Experimental Evaluation

In this section, we describe and interpret the experimental evaluation results of the approach that we have carried out. Our framework is implemented in the Python language.

Dataset	Type	Number of Concept	Number of Axiom	Number of is-a relation
SUMO	(1)	4,558	587,842	5,330
	(2)	3,432	2,138	2,138
Wikidata	(1)	69,188,843	2,941,036	2,941,036
	(2)	119,152	16,876	16,876
Babelnet	(1)	6,113,467	277,036,611	15,831,054
	(2)	119957	165,121	165,121

Table 4.3: The number of concepts, axioms, isa relations of three ontology sources.

We take account of the following three ontologies: SUMO[29], Wikidata[30], Babelnet[31], which are large open-domain ontologies. In our experiment, we take account of axioms of the form $A \sqsubseteq B$, one of the most common forms used to express knowledge in the ontologies [MAWNH15, BBM17]. It is also known as is-a or subsumption relation, which builds the hierarchy of an ontology. Here, since the concept names between ontologies are different, we have a mapping between concept names in the input sources to have a common concept name for merging. This step is performed before the merging process starts.

First of all, we collected the concepts from the three knowledge sources. The number of concepts gathered is as follows: $4,558$ concepts for SUMO, $69,188,843$ concepts for WikiData, and $6,113,467$ for Babelnet (including the total number of Babel synsets *(Babel synsets, i.e., sets of multilingual lexicalizations denoting a certain concept, are sets of synonyms words in Babelnet)*: $15,780,364$ and the total number of Babel senses *(i.e., sets of word senses expressed in different languages is called as Babel senses)*: $808,974,108$). The number of considered axioms (is-a relations) of the original ontologies is as follows: There are $5,330$ axioms for SUMO, $2,941,036$ for Wikidata, and $15,831,054$ for Babelnet. The application input is then the three open-domain ontologies. The outcome is the merged knowledge that agrees with the input ontologies.

As presented above, we mapped the concept names to each other in ontologies before merging the ontologies. Here, we provide some statistics about the ontologies before and after the mapping step in Table 4.3 (where (1) is the number of the original ontology, (2) is the number after the mapping process.). Here, our approach focuses crucially on the same concepts from the ontology sources. Therefore, we use the mapping process between the three sources to seek correspondence. We create new name synchronizations of concepts based on those mappings. The mapping process will be presented in the following subsection.

4.4.1 Mapping Process

The mapping process includes the following steps: In Step 1, for each concept name in SUMO, we extract the name from WordNet given in the annotation of the concept. Namely, we collect idWordNet in each concept (e.g., idWN: "*WN30-102569151*" in the *Fish* concept). For Step 2, we implement two methods to collect Wikidata ID:

- First, we use WordNet ID *(extracted at step 1)* to determine the synset of Babelnet. After that, we use BabelNet API[32] to retrieve the Wikidata ID (for each BabelSense in a BabelSynset); *(The*

[29] http://www.adampease.org/OP.
[30] https://www.wikidata.org.
[31] https://babelnet.org.
[32] available at https://babelnet.org/guide.

Level	SUMO	Wikidata	Babelnet
1	2,336	2,672	1,284
2	2,473	4,007	11,119
3	4,904	5,533	20,969
4	5,631	8,238	64,081
5	8,098	17,982	113,971
6	11,350	37,841	204,709
7	17,996	69,441	374,260
8	30,489	111,640	600,766
9	48,631	157,447	892,876
10	60,921	201,538	1,299,495

Table 4.4: The number of synchronized concepts for each level.

correspondences between Wikidata and Babelnet are available in the BabelNet dataset).

- The second one uses the SUMO concept name to seek its Wikidata ID through BabelSynset.

Here, the dataset is the collection of the (1) and the (2). In other words, we merge (1) and (2) into a file as the dataset. And then, we also remove the duplicate cases after collecting them. In Step 3, we check and collect the label/name of Wikidata ID through the Qwikidata library.[33] This work is to ensure that the concept names are the same.[34] Notably, a synchronization process of the concept names is conducted before merging. Specifically, we assign a new name at the end position of each mapping. Then, a list of synchronized concept names is collected, as in Figure 4.5. This process is the first part of the proposed framework (see Figure 4.1).

The synchronization of sources is essential because the concept names are different in practice. Moreover, this work concentrates on merging the ontologies with the same concept names. Therefore, the mapping process behaves as expected. The number of links between concept names of the three ontology sources is $156,152$ mappings. Once the mapping process is completed, we use the concepts of is-a relations with the new names.

4.4.2 Quantitative Results

We take account of the top 10 levels of the three ontologies to evaluate our merging method. The level starts at a root node for level 1 and expands the exploration of the next level when the depth increases $(n+1)$. The purpose of splitting the considered levels is to evaluate the computation time needed to generate the combinations/interpretations and determine the number of conflicts at each level. We determine the semantic conflicts based on explicit and implicit knowledge. First, we detect the common concept names at the testing level. Then, we filter out the semantic conflicts with the explicit knowledge *(i.e.,* $A12 \sqsubseteq A45$ *and* $A45 \sqsubseteq A12$ *where* $A12$ *and* $A45$ *are synchronized concept names)* and with the implicit knowledge *(i.e.,* $A24 \sqsubseteq A48, A48 \sqsubseteq A32$ *are known as* $A24 \sqsubseteq A32$ *and an axioms* $A32 \sqsubseteq A24$ *from another ontology)*. Moreover, we output the number of axioms, conflicts, and computed interpretations for each level. The number of is-a relations *(with synchronized names)* for each level is given in Table 4.4. The experimental results are provided in Table 4.6, where we show, for each level, the number of generated interpretations and the time needed to compute the result of

[33] https://qwikidata.readthedocs.io.

[34] The datasets and the implementation of the merging operator are available at https://github.com/ontologymerging/beliefmerging.

SUMO	Babelnet	Wikidata	New Name
...	...	...	...
Brandy	bn:03463167n	Q676745	A14
Brandy	bn:00874259n	Q301905	A15
Brandy	bn:0005720n	Q503252	A16
Brandy	bn:00041425n	Q693078	A17
Centimeter	bn:00017163n	Q174728	A18
Fish	bn:00004288n	Q2711102	A19
Fish	bn:02768902n	Q3840698	A20
Fish	bn:00052347n	Q168422	A21
Fish	bn:00015903n	Q625161	A22
Fish	bn:00014270n	Q640454	A23
Fish	bn:00023566n	Q1063438	A24
...	...	...	...

Table 4.5: Example of mappings and concept name synchronization.

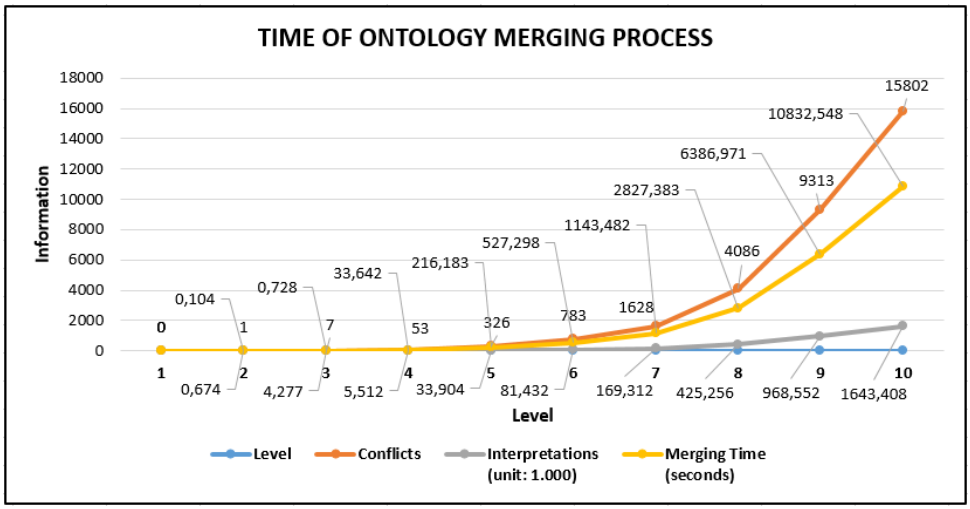


Figure 4.2: The computation time of the merging process (seconds).

merging. We evaluate the merging process from generating all possible interpretations to the result of the most plausible interpretation. We take account of different levels (where $n + 1$ level includes level n) because the extension of each level increases the number of axioms *(is-a relation)*, interpretations, and conflicts. It allows us to properly evaluate their impact on the time needed to compute the result of merging. Notably, we will first identify semantic conflicts and then apply our method to each of those semantic conflicts. If semantic conflicts found are separate (independent), we apply our framework in parallel (using multi-processing [35]).

Overall, from Table 4.6, we can see that our merging approach is of full potential and capable of solving semantic conflicts. First, we take account of the common concepts shared between the three ontology sources to seek conflicts. Namely, we focus on the axioms of the $is - a$ relation that represent ontology hierarchies in different ways. Next, we compute the interpretations needed to determine

[35] https://docs.python.org/3/library/multiprocessing.html

Level	Concepts	Axioms (is-a)	Conflicts	Interpretations	Merging Time (s)
1	29,748	S: 32 W: 10,772 B: 18,944	0	0	0
2	783,859	S: 66 W: 38,342 B: 745,451	1	104	0.674
3	1,717,363	S: 120 W: 549,710 B: 1,167,533	7	728	4.277
4	2,972,086	S: 251 W: 1,420,062 B: 1,551,773	53	5,512	33.642
5	3,968,698	S: 615 W: 1,682,041 B: 2,286,042	326	33,904	216.183
6	5,503,343	S: 1,452 W: 2,276,445 B: 3,225,446	783	81,432	527.298
7	8,601,261	S: 2,698 W: 3,872,494 B: 4,726,069	1,628	169,312	1143.482
8	12,907,446	S: 4,252 W: 6,439,584 B: 6,463,610	4,086	425,256	2,827.383
9	19,598,261	S: 5,656 W: 8,520,703 B: 11,071,902	9,313	968,552	6,386.971
10	28,189,097	S: 6,692 W: 9,786,145 B: 18,396,260	15,802	1,643,408	10,832.548

Table 4.6: The experimental results with SUMO (S), Wikidata (W) and BabelNet(B).

the result of merging. Note that concepts using our merging process must have synchronized names across the three sources. We have run an experiment in order to test our proposal based on the number of conflicts and the processing time. The conflicts start to appear in level 2 with $783,859$ concepts. Subsequently, the number of conflicts increases when increasing the number of levels. The processing time is acceptable and effective in practice. A piece of evidence is at level 10 with $15,802$ conflicts, and the processing time is around 180.54 minutes ($10,832.55$ seconds). Also, the merged result solves the problem of heterogeneous and ambiguous information well. We will discuss this qualitative result in the next subsection.

4.4.3 Qualitative Results

This section shows how our merging operators solve semantic conflicts between the three ontologies on the simple (yet instructive) illustrative case. In the example, we provide the input pieces of knowledge and the result of merging.

Example 4.5 *Let us consider the three concepts* $(Procedure, Technique, Process)$ *represented as in Figure 4.3. This example was presented in the introduction section.*

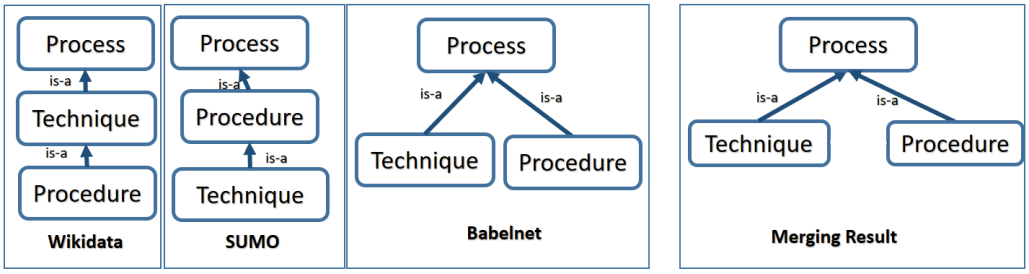


Figure 4.3: Construction and merged outcome of $Procedure$, $Technique$, and $Process$.

If we take account of the close distance as equal to 0. *We have the semantic merged outcome as follows:* $\mathcal{I}^M = f_{\oplus}(\mathcal{P}_{Ex5}) = \langle Procedure^{\mathcal{I}} = \{proced, tech\}, Technique^{\mathcal{I}} = \{tech, proced\}, Process^{\mathcal{I}} = \{tech, proceed, process\}\rangle$.

The result of the syntactic merging is $\mathcal{T}^M(\mathcal{P}_{Ex5}) = \{Procedure \sqsubseteq Process,\ Technique \sqsubseteq Process, Procedure \sqcap Technique \sqsubseteq Process, Procedure \sqcap Process \sqsubseteq Technique, Technique \sqcap Process \sqsubseteq Procedure\}$. *Example 5's merged result is reasonable and correct because they are syntactic of all axioms of sources. Moreover, they also pointed out the new relations among concepts (i.e.,* $Procedure \sqcap Process \sqsubseteq Technique$*). The semantics outcome is always satisfied by these found axioms.*

Example 4.6 *Let us consider the three concepts* $(Statement, Sentence, Proposition)$ *represented in Figure 4.4. We selected this example to illustrate that the merged syntactic result has a different structure w.r.t. the input sources.*

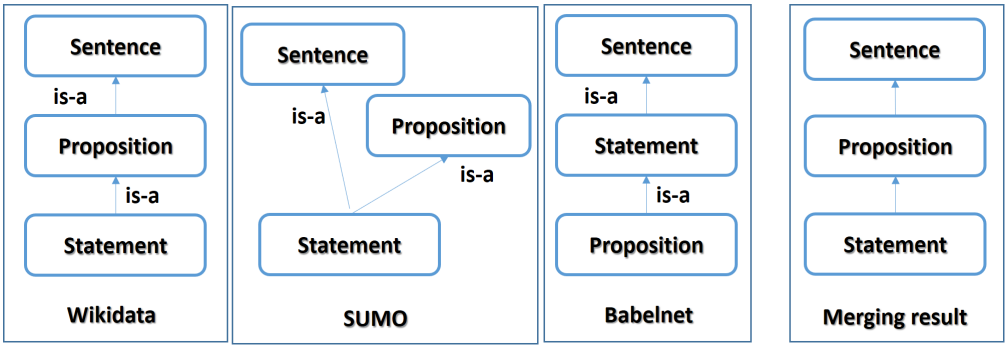


Figure 4.4: Structure and merged result of $Sentence$, $Statement$, and $Propositions$.

The semantic merged outcome is: $\mathcal{I}^M = f_{\oplus}(\mathcal{P}_{Ex6})$= $\langle Sentence^{\mathcal{I}} = \{se, st, pr\}, Proposition^{\mathcal{I}} = \{st, pr\}, Statement^{\mathcal{I}} = \{st\}\rangle$.

The result of the syntactic merging is $\mathcal{T}^M(\mathcal{P}_{Ex6}) = \{Proposition \sqsubseteq Sentence, Statement \sqsubseteq Sentence, Statement \sqsubseteq Proposition, Statement \sqcap Sentence \sqsubseteq Proposition, Statement \sqcap Proposition \sqsubseteq Sentence\}$. *This merged semantic and syntactic result is plausible and consistent.* $\mathcal{T}^M$ *is not only to contain the axioms of three sources but also to seek out other axioms.*

Example 4.7 *Let us consider the three concepts* $(Money, Cash, Currency)$ *represented as in Figure 4.5. We selected this example to demonstrate that the merged syntactic result has a quite different structure w.r.t. the input sources.*

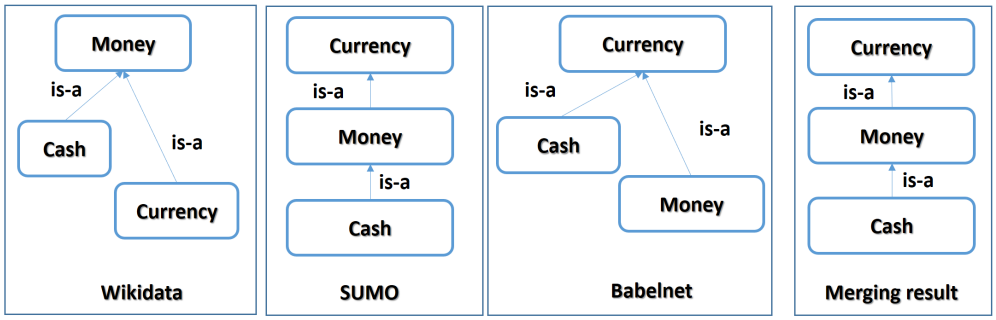


Figure 4.5: Hierarchy and merged outcome of $Money$, $Cash$, and $Currency$.

The semantic merged outcome is: $\mathcal{I}^M = f_{\oplus}(\mathcal{P}_{Ex7})$= $\langle Money^{\mathcal{I}}$= $\{mon, cas\}$, $Cash^{\mathcal{I}}$= $\{cas\}$, $Currency^{\mathcal{I}}$= $\{cur, cas, mon\}\rangle$.

The result of the syntactic merging is $\mathcal{T}^M(\mathcal{P}_{Ex7}) = \{Cash \sqsubseteq Money,\ Cash \sqsubseteq Currency,\ Money \sqsubseteq Currency,\ Money \sqcap Cash \sqsubseteq Currency,\ Currency \sqcap Cash \sqsubseteq Money\}$. *The merged result is tractable and consistent since some axioms correctly reflect the cases in practice (i.e.,* $Money \sqcap Cash \sqsubseteq Currency$*).*

4.5 Concluding Remarks

This work aims to develop an efficient framework and implementation for exploiting open-domain ontologies. When conflicts arise between knowledge from multiple open-domain ontology sources, we deal with merging problems. To this end, we formally defined a semantic-based merging operator. We then demonstrated our approach using real examples of open-domain ontologies. As a result, a framework for ontology merging and mapping between different sources has been developed and made available online. Finally, experimental results, as well as several practical examples, are provided.

Typically, most research focuses on determining methods to resolve logic conflicts when merging sources. However, our work addresses semantic conflicts. For instance, suppose we choose the "equivalence" relation for all merged outcomes for this problem. In that case, we will flatten the structure of the merged ontology. This issue is readily apparent in practice because multiple ontologies are constructed from diverse mental streams, e.g., the organizational structure of the companies is entirely different. Hence, if two knowledge base sources from different companies are merged, conflicts in terms of semantics are possible. Moreover, when we take into account the semantic problem, many types of semantic conflicts might arise. For example, one company states that a package includes a dozen bottles (with 12 items). In contrast, another enterprise says that a parcel has fifteen bottles (with 15 items). It is the same statement, but the number of bottles is different. Hence, if we merge two companies, what will happen? Therefore, it might be one of the future research directions to detect and solve the problems arising from semantic conflicts.

If this chapter focuses on resolving semantic conflicts, the following chapter will provide a strategy for resolving both sorts of conflicts (namely, logical and semantic problems).

Chapter 5

Region-Based Merging of Open-Domain Terminological Knowledge

In this chapter, we focus crucially on the description logic $\mathcal{EL}_\bot$. From the addition of the bottom concept ($\bot$) to $\mathcal{EL}$, logical conflicts will arise. Now, we consider an example to illustrate the merging problem. We assume that the first source states that every $Fruit$ is a $Vegetable$, while another source says that the concept $Fruit$ is disjoint with the concept $Vegetable$. Intuitively, these two viewpoints are conflicting if we merge them together.

To solve the problem of merging (i.e., semantic and logical conflicts), instead of simply merging ontologies in the existing language to tackle the merging problem, we will shift the input source into a more expressive framework to leverage inference for the merging. To do that, we moved the DL to the topological space. And then, we merge knowledge bases in this "intermediary" framework. Namely, we provide a two-way translation between DL $\mathcal{EL}$ into RCC-5 (see 5.2.2). Then, we express the knowledge bases into the qualitative constraint networks (see 5.2.3) [SPJ20, SLJ20, CKMS10]. We also take inspiration from several QCN merging operators of Condotta et al. [CKMS10] to propose our framework for merging $\mathcal{EL}_\bot$ ontologies. Therein, we propose an algorithm to select the merged result (Algorithm 2). Moreover, we mostly use distance to pick up merged constraints. Now let us examine the next section to determine why and what drives this strategy's implementation.

5.1 Context and Motivation

Open-domain knowledge Ontologies, frameworks for expressing terminological knowledge, are available and encode structured relations about the concepts and properties of a given domain. In this work, we focus on terminological knowledge about concepts that can be expressed using ontologies. The available ontologies (and KGs, as simple ontologies) are inevitably incomplete, where several rules and facts are missing. Several methods have been proposed for automated ontology (KG) completion [BCB+13, RR17, LBS19, KHKS20] that exploit statistical regularities in a given ontology to predict plausible missing rules or facts. Unfortunately, meaningful knowledge is difficult to predict, especially when only a few examples of facts or rules are available. Moreover, as most of the existing approaches are based on statistical regularities, the resulting predictions might conflict with each other. A repair-based mechanism is then required to maintain the consistency of the set of terminological statements, i.e., ensure that there are no conflicting (or contradictory) statements. In the same perspective, to widen the coverage of terminological knowledge to several domains and to deal with incompleteness and conflicting statements, one may combine knowledge from several sources. However, it turns out that merging open-domain knowledge bases is a particularly challenging task, as pointed out, for example, in

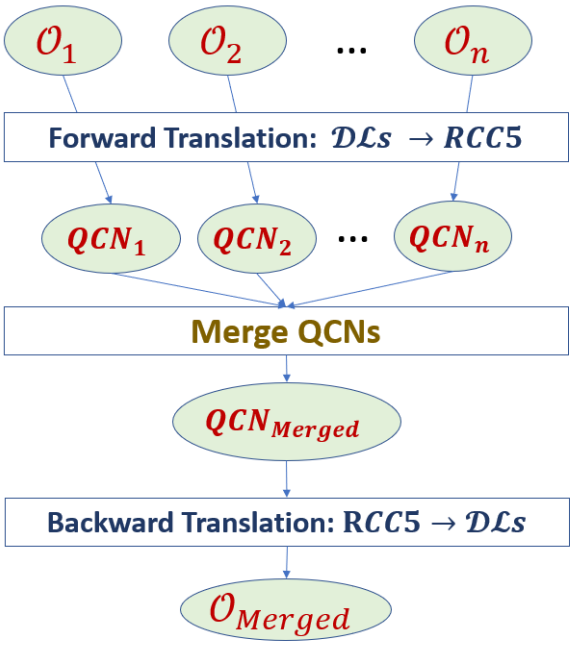


Figure 5.1: The idea of the contribution in Chapter 5.

[PTVS+16] reporting the different problems and difficulties encountered when merging Freebase with WikiData. Moreover, conflicting information may occur when the statements of several sources are simply gathered together. In this work, the primary goal of our approach is knowledge merging; hence, we again assume that all knowledge encoded by the different sources is already aligned and mapped to each other. Namely, we suppose that they share the same terminology. Hence, while ontology alignment (or matching) is the process of determining correspondences between terminologies of ontologies, ontology *merging* aims to combine two (or more) ontologies having the same terminology while handling conflicting statements.

Let us consider an example to illustrate the merging problem. Assume that a first source says that the concept $Paper$ is disjoint with the concept $Document$, while another source says that every $Paper$ is a $Document$. Obviously enough, these two statements are conflicting. Therefore, to be faithful to both sources while resolving conflicts, a sensible choice would be to assume that $paper$ and $Document$ are not disjoint concepts. However, every $Paper$ is not necessarily a $Document$, that is, the two concepts *partially overlap*. This kind of result is clearly consistent and can be seen as a good compromise between both sources. Finding a meaningful and relevant compromise between sources during the merging process is difficult to obtain. This is mainly due to the fact that ontology languages (description logics, for example) are not expressive enough to capture salient knowledge that might emerge during the merging process. The simple example pointed out above clearly shows some pieces of knowledge that should be taken into account during the merging process but that cannot be captured in the ontology language. The problem of ontology (or DL) merging is close to the problem of belief merging in a propositional setting [BBPW19, BBLR14, KH16, WWJQ12, PKM08]. For instance, Benferhat *et al.* ([BBPW19]) studied merging assertional bases in the $DL\text{-}Lite$ fragment. They have determined the minimal subsets of assertions to resolve conflicts based on the inconsistency minimization principle. Bouraoui et al. ([BKMV20b]) proposed a model-based merging operator for merging $\mathcal{EL}$ ontologies which solve semantic conflicts that arise during the merging process.

In this work, to handle the merging problem, we take inspiration from conceptual spaces, which are geometric representation frameworks in which the objects are represented as points in a topological space, and concepts are modelled as regions [HM97, Ard00, DEGM22, AB20]. Motivated by the fact that conceptual knowledge in an ontology can be to some extent modelled as geometric objects and constraints on metric spaces [BCCAS20], this work proposes a method for ontology merging that takes advantage of *qualitative spatial reasoning* [CCR93, CBGG97] to find out a relevant compromise between sources while resolving conflicts. Qualitative spatial reasoning [RN01, WQLY18] is a suitable paradigm for efficiently reasoning about spatial entities and their relationships, where knowledge is represented as a so-called *qualitative constraint network (QCN)*. Spatial information [HBFK10, KG05] is usually represented in terms of basic or non-basic relations in a qualitative calculus, where reasoning tasks are then formulated as solving a set of qualitative constraints [RCC92, CBGG97, BGWH11, SLJ20]. In particular, the Region Connection Calculus (RCC) is a well-studied formalism for qualitative topological representation and reasoning, including its subsets *RCC-5* [SL13] and *RCC-8* [RCC92, SK12]. Two significant advantages of the RCC framework are its ability to reason efficiently about the relationships between spatial entities and its ability to deal with conflicts in qualitative constraint merging, as shown in [CKMS10, TBL09]. In short, the representation of region constraints into QCNs allows for more expressivity than when using DL rules (or constraints). In particular, QCNs are expressive enough to allow for disjunctions in the constraints.

Several QCN merging operators have been introduced in the literature (e.g., [CKMS09, CKMS10, TBL09, JW12]. Roughly speaking, these operators compute a distance between QCN scenarios and the input QCNs. Then the scenarios with a minimal distance are selected as the best candidates for the merged result. Taking inspiration from these works, in this work, we take advantage of the RCC-5 formalism and propose a method for merging open-domain terminological knowledge (simply called ontologies) using QCNs. We first show how to translate such knowledge into qualitative spaces while preserving its semantics and properties. Second, we propose a merging operator that produces a single and consistent region space representing a compromise between sources. Lastly, we show how to express the region space in the input ontology language while maintaining all relevant information. The figure 5.1 graphically summarizes the main idea of this chapter.

5.2 Foundations

Our method is based on two complementary frameworks for merging open-domain terminological knowledge: we rely on a lightweight Description Logic (DL) framework to encode knowledge and use RCC-5 and qualitative constraints for performing the merging. This section briefly recalls the technical background required on RCC-5 and DL $\mathcal{EL}_\bot$[36].

5.2.1 Description Logic $\mathcal{EL}_\bot$

As presented in Section 2.3.4, $\mathcal{EL}_\bot$ or $\mathcal{EL}^\neg$ is a extended version of $\mathcal{EL}$. This work will use $\mathcal{EL}_\bot$, which is a logic language to express the knowledge base. We use this setting since this version can appear inconsistency with adding $\bot$ concept, i.e., one source $Tiger \sqsubseteq Cat$ while another source tell that $Tiger \sqcap Cat \sqsubseteq \bot$. In this work, we consider $\mathcal{EL}_\bot$ concept expressions [Kri20] which are built according to the following grammar:

$$C ::= \top \mid \bot \mid N_C \mid C \sqcap C \mid \exists r.C \text{ where } r \in N_R.$$

[36] An extended fragment of the family $\mathcal{EL}$.

Syntax	Semantics
$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
r	$r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
a	$a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
$\top$	$\Delta^{\mathcal{I}}$
$\bot$	$\emptyset$
$\exists r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}} s.t. (x,y) \in r^{\mathcal{I}}, y \in C^{\mathcal{I}}\}$

Table 5.1: Syntax and semantics of $\mathcal{EL}_{\bot}$.

A summary of the syntax and semantics of $\mathcal{EL}_{\bot}$ is shown in Table 5.1. Before translating the input ontologies into a region-based representation, we assume that each of them is in the *strict normal form*, i.e., if its $TBox$ $\mathcal{T}$ only consists of inclusions of the fundamental form:

$$A \sqsubseteq B, A \sqcap B \sqsubseteq \bot, A \sqsubseteq \exists r.B, \exists r.A \sqsubseteq B,$$

where $A, B \in N_C$. Such an assumption is made without loss of generality since for each ontology $\mathcal{O}$, one can compute an ontology $\mathcal{O}'$ in the strict normal form in polynomial time [BBL05]. For this work, we define a *fulfilling interpretation* that will be used in the next section (see Section 5.4).

Definition 5.1 *Let $\mathcal{O}$ be an ontology and let $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ be a model of $\mathcal{O}$. We say $\mathcal{I}$ is **fulfilling** if $A^{\mathcal{I}} \neq \emptyset$ for every $A \in N_C$.*

5.2.2 Region Connection Calculus

The study of topology, one of the most basic aspects of space, is well-documented in the mathematical literature. The RCC formalism allows one to represent and reason about the relationships between spatial entities [RCC92]. The RCC abstractly represents regions that are a non-empty regular closed subset of some topological space. Moreover, RCC is based on a single primitive relation between spatial regions. Among the fragments of the RCC theory, RCC-5 and RCC-8 are the most popular. RCC-8 includes eight basic relations, such as $\mathcal{B}^8 = \{DC$ (disconnected), EC (externally connected), PO (partially overlapping), EQ (Equals), TPP (Tangential proper part), $TPPi$ (Tangential proper part inverse), $NTPP$ (Non-tangential proper part), $NTPPi$ (Non-tangential proper part inverse)$\}$. Therein, the RCC-5 relations can be extended to the RCC-8. It is shown in Figure 5.2. Here, the RCC-5 fragment is expressive enough to explain set-theoretic relations between regions [Ben94, PT97]. In RCC-5, regions can simply be interpreted as non-empty subsets of a given set, and the focus is given on a set $\mathcal{B} = \{DR, PO, EQ, PP, PPi\}$ of five binary relations between regions called *basic relations*. The set $\mathcal{B}$ forms a jointly exhaustive and pairwise disjoint set of relations, that is, each pair of regions satisfies exactly one relation from $\mathcal{B}$: the relation DR (resp. PO, EQ, PP) holds between two regions whenever the two regions are disjoint (resp. when they partially overlap, are equal when the first is a strict subset of the second), and PPi is the converse of PP.

Moreover, for checking the correctness of all logical inferences, we present the composition table [BTB+20, SL13, LY03] of RCC-5 shown in Table 5.2.

5.2.3 Qualitative Constraint Networks

Based on $\mathcal{B}$, more complex pieces of information about the relative positions of a set of regions can be represented by means of qualitative constraint networks (QCNs). Here, we refer to a variable's name as

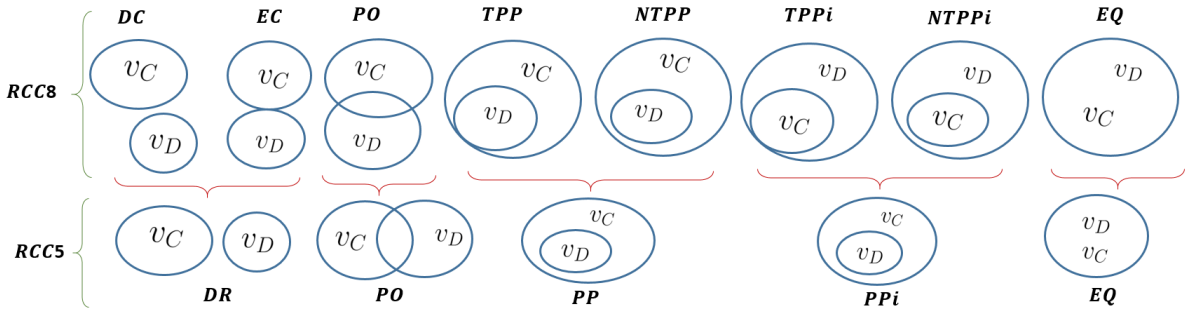


Figure 5.2: Relations of RCC-8 and RCC-5.

Relations	EQ	DR	PO	PP	PPi
EQ	EQ	DR	PO	PP	PPi
DR	DR	$\mathcal{B}$	DR, PO, PP	DR, PO, PP	DR
PO	PO	DR, PO, PPi	$\mathcal{B}$	PO, PP	DR, PO, PPi
PP	PP	DR	DR, PO, PP	PP	$\mathcal{B}$
PPi	PPi	DR, PO, PPi	PO, PPi	EQ, PO, PP, PPi	PPi

Table 5.2: The composition table of RCC-5.

V_C, where C is a known region (or concept) name in advance (i.e., V_{France}). Now, we formally define the QCN as follows:

Definition 5.2 (QCN) *A qualitative constraint network is a pair $\mathcal{N} = \langle V, \Psi \rangle$, where*

- *$V = \{v_C, v_D, \ldots\}$ is a non-empty finite set of variables corresponding to a set of region variables representing the spatial entities;*
- *Ψ is a set of binary constraints between these entities. Each constraint $\Psi_{CD} \in \Psi$ is a mapping from $V \times V$ to $2^{\mathcal{B}}$ such that $\Psi_{CC} = \{id\}$ and $\Psi_{CD} = (\Psi_{DC})^{-1}$ for each $v_C, v_D \in V$, and is simply denoted by $\Psi_{CD} = v_C \; \varphi \; v_D$, where $\varphi \subseteq 2^{\mathcal{B}}$; and Ψ_{CD} is said to be a singleton constraint whenever φ is a singleton.*

To illustrate the QCN, we show an in Figure 5.3 with $V = \{v_C, v_D, v_E\}$ and $\Psi = \{\Psi_{CD} = \{PP\}, \Psi_{CE} = \{DR\}, \Psi_{DE} = \{PO\}\}$.

Definition 5.3 (Interpretation of QCN) *An interpretation of a QCN $\mathcal{N}$ is defined as $\mathcal{S} = (\mathcal{D}^{\mathcal{S}}, \cdot^{\mathcal{S}})$, where*

- *$\mathcal{D}^{\mathcal{S}}$ is a non-empty set (the domain of the regions);*
- *$\cdot^{\mathcal{S}}$ is an interpretation function which maps each variable v_C to a non-empty subset $v_C^{\mathcal{S}}$ of $\mathcal{D}^{\mathcal{S}}$.*

Table 5.3 precises how singleton constraints from Ψ are interpreted in RCC-5, i.e., an interpretation $\mathcal{S}$ of $\mathcal{N}$ satisfies a singleton constraint Ψ_{CD}, denoted by $\mathcal{S} \models \mathcal{N}$, if the relation between $v_C^{\mathcal{S}}$ and $v_D^{\mathcal{S}}$ according to the table is satisfied (e.g., $\mathcal{S} \models v_C\{PP\}v_D\}$ whenever $v_C^{\mathcal{S}} \subset v_D^{\mathcal{S}}$). The satisfaction relation is extended to any (non-singleton) constraint from $2^{\mathcal{B}}$ as follows: for each $\varphi \in 2^{\mathcal{B}}$, $\mathcal{S} \models v_C \; \varphi \; v_D$ iff $\mathcal{S} \models v_C \; \{\varphi_i\} \; v_D$ for some $\varphi_i \in \varphi$ (e.g., $\mathcal{S} \models v_C\{PP, EQ\}v_D$ iff $\mathcal{S} \models v_C\{PP\}v_D$ or $v_C\{EQ\}v_D\}$).

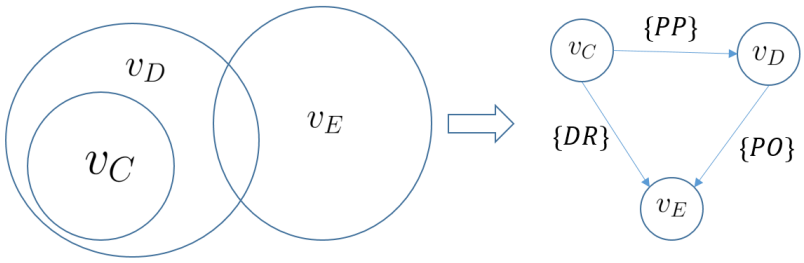


Figure 5.3: An example of the QCN.

Name (Symbol)	Syntax	Semantics
Proper Part of (PP)	$v_C\{PP\}v_D$	$v_C^{\mathcal{S}} \subset v_D^{\mathcal{S}}$
Inverse PP of (PPi)	$v_C\{PPi\}v_D$	$v_D^{\mathcal{S}} \subset v_C^{\mathcal{S}}$
Equals (EQ)	$v_C\{EQ\}v_D$	$v_C^{\mathcal{S}} = v_D^{\mathcal{S}}$
Disjoint From (DR)	$v_C\{DR\}v_D$	$v_C^{\mathcal{S}} \cap v_D^{\mathcal{S}} = \emptyset$
Partially Overlaps (PO)	$v_C\{PO\ \}v_D$	$v_C^{\mathcal{S}} \cap v_D^{\mathcal{S}} \neq \emptyset$ $v_C^{\mathcal{S}} \nsubseteq v_D^{\mathcal{S}}, v_D^{\mathcal{S}} \nsubseteq v_C^{\mathcal{S}}$

Table 5.3: Syntax and semantics of *RCC-5*, $v_C, v_D \in V$.

Definition 5.4 ([SLJ20]) *Let $\mathcal{N} = \langle V, \Psi \rangle$ be a QCN. We then have:*

- *An interpretation $\mathcal{S}$ of a QCN $\mathcal{N} = \langle V, \Psi \rangle$ is said to be a solution of $\mathcal{N}$, denoted by $\mathcal{S} \models \mathcal{N}$, if and only if $\mathcal{S} \models \Psi_{CD}$ for each $\Psi_{CD} \in \Psi$;*
- *A QCN is consistent if and only if it admits a solution;*
- *A sub-network of $\mathcal{N}$ is a QCN $\mathcal{N}' = (V, \Psi')$ such that $\Psi' \subseteq \Psi$;*
- *A quasi-atomic QCN $\langle V, \Psi \rangle$ is a QCN where for each $v_C, v_D \in V$, there is a unique constraint $\Psi_{CD} \in \Psi$, and where Ψ_{CD} is either a singleton or $\Psi_{CD} \in \{\{PP, EQ\}, \{PPi, EQ\}\}$;*
- *A scenario of a QCN is a quasi-atomic sub-network of $\mathcal{N}$. Noteworthy, a QCN is consistent if it admits a consistent scenario;*
- *$\mathcal{N}$ is atomic if and only if $\forall v_C, v_D \in V, \Psi_{CD} = \{b\}$ with $b \in \mathcal{B}$;*
- *$\mathcal{N}$ is trivially inconsistent, denoted by $\emptyset \in \mathcal{N}$, if and only if $\exists v_C, v_D$ such that $\Psi_{CD} = \emptyset$;*
- *$\mathcal{N}$ is the empty QCN on V, denoted by $\bot^V$, if and only if $\Psi_{CD} = \emptyset$ for all $v_C, v_D \in V$ such that $v_C \neq v_D$;*
- *The size of $\mathcal{N}$, denoted by $|\mathcal{N}|$, is $|\mathcal{N}| = |\{\Psi_{CD} \mid v_C, v_D \in V \wedge v_C \neq v_D\}|$;*
- *A QCN $\mathcal{N}'$ is equivalent to $\mathcal{N}$ if and only if it admits the same set of solutions with $\mathcal{N}$.*

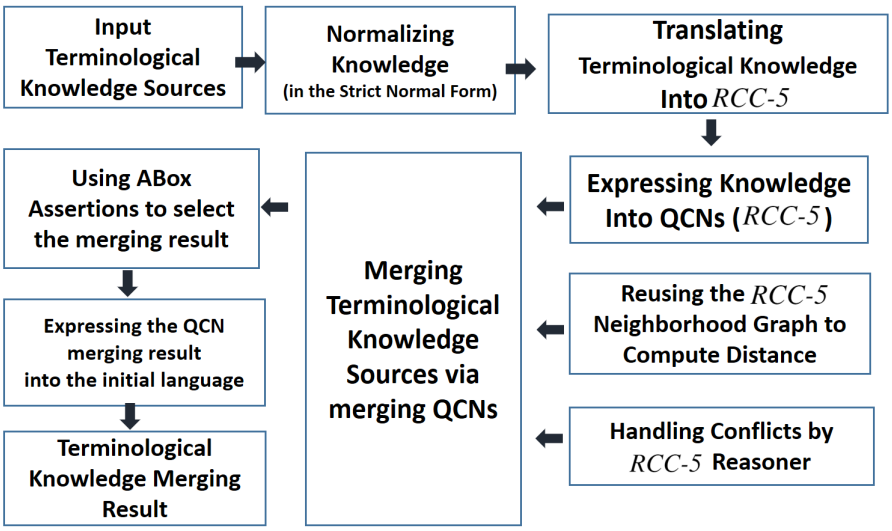


Figure 5.4: Framework of ontology merging via QCN merging.

5.3 Merging Framework Description

This short section summarizes our method for merging open-domain terminological knowledge (i.e., ontologies) using qualitative constraint networks (QCNs). As highlighted in the introduction, QCNs are expressive enough to capture some relevant information that might emerge during the merging process, which in turn allows one to select a consistent compromise between sources when expressing the merging result.

The first task is to find a "faithful translation" from an ontology to a QCN that preserves the semantics and maintains the initial knowledge encoded in the ontology. Then, it is assumed that every input ontology is translated into a QCN, so we are given a multiset of QCNs, also called *profile*. The second task is then to define a merging operator that associates the QCN profile into a single QCN, representing the QCN profile in a global and consistent way. As constraints of the merged QCN are *sets* of basic relations, this QCN cannot be translated back into the target ontology language in the general case. So we select one of its "best" consistent scenarios which, in contrast, can be expressed as an ontology. Our framework is presented in Figure 5.4.

More precisely, our approach involves the following main steps:

1. The translation of terminological knowledge sources (i.e., the input ontologies given in the strict normal form) into QCNs (Section 5.4). In this step, we present a translation function that ensures the faithfulness of the translation from an ontology into a QCN;

2. The definition of a QCN merging operator (Section 5.5). Exploiting the notion of "distance" between basic relations and constraints, this step associates the input QCN profile (the translated ontologies) with a single merged, consistent QCN;

3. The selection of a "best" consistent scenario of the merged QCN as a candidate of the merged result (Section 5.6). This selection process takes advantage of the notion of distance between scenarios and some plausible instantiation of the input ontologies (i.e., some interpretations);

4. The translation of the selected consistent scenario back into an ontology, i.e., the underlying language of the input sources (Section 5.7).

5.4 Translating Terminological Knowledge into QCNs

In this section, we present a translation function from any ontology to a QCN. More precisely, we (1) map atomic concepts names into QCN variables and axioms into constraints, and (2) show that the translation is faithful to the $TBox$ of the initial ontology.

Definition 5.5 (Forward translation $\tau_{\triangleright}$) *A **forward translation** is a function $\tau_{\triangleright} : N_C \longrightarrow V$ s.t. $\tau_{\triangleright}(C) \stackrel{\text{def}}{=} v_C$. $\tau_{\triangleright}$ is extended to map axioms in the (strict) normal form (see the end of Section 5.2.1) into constraints as follows:*

- $\tau_{\triangleright}(C \sqsubseteq D) \stackrel{\text{def}}{=} \tau_{\triangleright}(C)\{PP, EQ\}\tau_{\triangleright}(D)$, *and*
- $\tau_{\triangleright}(C \sqcap D \sqsubseteq \bot) \stackrel{\text{def}}{=} \tau_{\triangleright}(C)\{DR\}\tau_{\triangleright}(D)$.

Moreover, $\tau_{\triangleright}$ is extended to translate ontologies in the (strict) normal form into a set of constraints in the expected way:

$$\tau_{\triangleright}(\mathcal{O}_{\mathcal{EL}_{\bot}}) \stackrel{\text{def}}{=} \{\tau_{\triangleright}(\Phi) \mid \Phi \in \mathcal{O}_{\mathcal{EL}_{\bot}}\}.$$

To show that the translation is faithful, we provide a "semantic" mapping from $\mathcal{O}$ to $\tau_{\triangleright}(\mathcal{O}_{\mathcal{EL}_{\bot}})$, and conversely. Let us first show how models of $\mathcal{O}_{\mathcal{EL}_{\bot}}$ correspond to solutions of $\tau_{\triangleright}(\mathcal{O}_{\mathcal{EL}_{\bot}})$:

Definition 5.6 (Flattening of an interpretation) *Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be a fulfilling interpretation. With $\mathcal{S}_{\mathcal{I}} \stackrel{\text{def}}{=} (\Delta^{\mathcal{I}}, \cdot^{\mathcal{S}_{\mathcal{I}}})$ we denote the **flattening of** $\mathcal{I}$, where $\cdot^{\mathcal{S}_{\mathcal{I}}} : V \longrightarrow 2^{\Delta^{\mathcal{I}}}$ is such that $(v_C)^{\mathcal{S}_{\mathcal{I}}} = C^{\mathcal{I}}$.*

Notably, non-fulfilling interpretations are not considered (are irrelevant) in the work because the corresponding translation regions cannot be empty (are non-empty subsets) in RCC-5.

Theorem 5.1 *Let $\mathcal{O}_{\mathcal{EL}_{\bot}}$ be an ontology, and let $\mathcal{I}$ be a fulfilling interpretation of $\mathcal{O}_{\mathcal{EL}_{\bot}}$ such that $\mathcal{I} \models \mathcal{O}_{\mathcal{EL}_{\bot}}$. Then $\mathcal{S}_{\mathcal{I}} \models \tau_{\triangleright}(\mathcal{O}_{\mathcal{EL}_{\bot}})$.*

The other way around, let us show how solutions of $\tau_{\triangleright}(\mathcal{O}_{\mathcal{EL}_{\bot}})$ correspond to interpretations satisfying all axioms from $\mathcal{O}_{\mathcal{EL}_{\bot}}$.

Definition 5.7 (Inflation of a solution) *Let $\mathcal{S} = (\mathcal{D}^{\mathcal{S}}, \cdot^{\mathcal{S}})$ be a semantic solution to a QCN $\mathcal{N}$ over V. With $\mathcal{I}_{\mathcal{S}} \stackrel{\text{def}}{=} (\Delta^{\mathcal{I}_{\mathcal{S}}}, \cdot^{\mathcal{I}_{\mathcal{S}}})$, where $\Delta^{\mathcal{I}_{\mathcal{S}}} = \mathcal{D}^{\mathcal{S}}$ and, for every $A \in N_C$, $A^{\mathcal{I}_{\mathcal{S}}} = (v_A)^{\mathcal{S}}$, we call $\mathcal{I}_{\mathcal{S}}$ an **inflation of** $\mathcal{S}$.*

Intuitively, the inflation of $\mathcal{S}$ corresponds to an interpretation blown up from $\mathcal{S}$ by interpreting atomic concept names in the same way their corresponding variable names are "populated" by the solution. Notice that there are as many possible inflations of $\mathcal{S}$ as there are ways of interpreting N_R and N_I over $\Delta^{\mathcal{I}_{\mathcal{S}}}$. An immediate consequence of Definition 5.7 is that every inflation of a solution $\mathcal{S}$ is fulfilling.

Theorem 5.2 *Let $\mathcal{O}_{\mathcal{EL}_{\bot}}$ be an ontology and let $\mathcal{S}$ be a solution of $\tau_{\triangleright}(\mathcal{O}_{\mathcal{EL}_{\bot}})$. Then there is an inflation $\mathcal{I}_{\mathcal{S}}$ of $\mathcal{S}$ s.t. $\mathcal{I}_{\mathcal{S}} \models \Phi$ for each axiom Φ of $\mathcal{O}_{\mathcal{EL}_{\bot}}$.*

Theorems 5.1 and 5.2 establish that our translation is faithful, i.e., that the set of all fulfilling models of an ontology $\mathcal{O}_{\mathcal{EL}_{\bot}}$ are captured precisely in its translated QCN $\tau_{\triangleright}(\mathcal{O}_{\mathcal{EL}_{\bot}})$.

5.5 QCN Merging

We reduce the merging of a profile of ontologies $\mathcal{P} = \langle \mathcal{O}^1_{\mathcal{EL}_\bot}, \ldots, \mathcal{O}^n_{\mathcal{EL}_\bot} \rangle$ to the merging of a profile of QCNs $\mathcal{N} = \langle \mathcal{N}^1, \ldots, \mathcal{N}^n \rangle$, where for each $i \in \{1, \ldots, n\}$, $\mathcal{N}^i = (V, \Psi^i) = \tau_\triangleright(\mathcal{O}^i)$, based on the faithful translation given the previous section. Inspired by works on syntactical QCN merging [CKMS10], this QCN merging process is summarized as follows. We associate with the profile $\mathcal{N}$ a single merged and consistent QCN $\mathcal{N} = (V, \Psi)$ representing $\mathcal{N}$ in a "global" way. This is performed in a constraint-wise fashion: for each pair of variables $v_C, v_D \in V$, we associate each basic relation $b \in \mathcal{B}$ with a value representing its *distance* to the profile of constraints $\mathscr{C}_{CD} = \langle \Psi^1_{CD}, \ldots, \Psi^n_{CD} \rangle$. This distance is the key tool to form the constraint Ψ_{CD} of the merged QCN $\mathcal{N}$. Intuitively, each constraint Ψ_{CD} corresponds to the set of basic relations with the lowest distances to the profile $\mathscr{C}_{CD}$, while ensuring that the resulting QCN is consistent.

Example 5.1 *Consider the profile of ontologies* $\mathcal{P} = \langle \mathcal{O}^1, \mathcal{O}^2, \mathcal{O}^3, \mathcal{O}^4 \rangle$ *that encodes the following knowledge about the four concepts of* $Paper$, $Text$, $Document$ *and* $Book$, *respectively denoted by* P, T, D *and* B[37].

- $\mathcal{O}^1 = \langle \mathcal{T}^1 = \{P \sqsubseteq T, T \sqcap D \sqsubseteq \bot, P \sqsubseteq B, P \sqcap D \sqsubseteq \bot, B \sqcap D \sqsubseteq \bot\}, \mathcal{A}^1 = \{P(p_1), T(p_1), T(t_1), D(d_1), B(p_1), B(b_1)\} \rangle$,
- $\mathcal{O}^2 = \langle \mathcal{T}^2 = \{P \sqsubseteq T, T \sqsubseteq B, D \sqsubseteq B, D \sqsubseteq P\}, \mathcal{A}^2 = \{P(p_2), P(d_2), T(p_2), T(t_2), D(d_2), B(p_2), B(b_2)\} \rangle$,
- $\mathcal{O}^3 = \langle \mathcal{T}^3 = \{B \sqsubseteq D, D \sqsubseteq P, P \sqsubseteq T, D \sqsubseteq T\}, \mathcal{A}^3 = \{P(p_3), T(t_3), T(p_3), D(d_3), B(b_3)\} \rangle$,
- $\mathcal{O}^4 = \langle \mathcal{T}^4 = \{D \sqcap P \sqsubseteq \bot, P \sqsubseteq T, B \sqsubseteq D, T \sqcap B \sqsubseteq \bot, T \sqcap D \sqsubseteq \bot\}, \mathcal{A}^4 = \{P(p_4), T(t_4), T(p_4), D(b_4), D(d_4), B(b_4)\} \rangle$.

Using the forward translation $\tau_\triangleright$ *presented in the previous section, one associates with* $\mathcal{P}$ *a profile of QCNs* $\langle \tau_\triangleright(\mathcal{O}^i) \rangle = \langle \mathcal{N}^i \rangle$ *(*$i \in \{1, \ldots, 4\}$*). The four QCNs are depicted in Figure 5.5 (to alleviate the figures, we do not represent the* Ψ_{CD} *when* $\Psi_{CD} = \mathcal{B}$*, i.e., when the QCN does not provide any information between the relationship between* v_C *and* v_D*).*

Although we do not require it in the general case, note that in this example each ontology $\mathcal{O}^i$ is consistent, i.e., the $TBox$ of each input ontology does not contain conflicting information. As a direct consequence of Theorem 5.1, each QCN is consistent. However, simply combining these QCNs can easily lead to conflicts. For instance, there is no basic relation shared in the constraints Ψ^1_{BD} and Ψ^3_{BD}, since in $\mathcal{N}^1$ we have that $Book\ \{DR\}\ Document$ whereas in $\mathcal{N}^3$ we have that $Book\ \{PP, EQ\}\ Document$. This calls for our QCN merging procedure.

5.5.1 Computing a Distance between a Basic Relation and a Profile of Constraints

We start by considering a distance between basic relations. Firstly introduced in [Fre92] in the context of temporal reasoning, the notion of *conceptual neighborhood* (CN) between relations was later adapted to QCN merging in [CKS08, SPJ20] to define such a distance. Intuitively, two basic relations $b, b' \in \mathcal{B}$ are CNs if a continuous transformation of two regions which satisfy the basic relation $b \in B$ leads them to directly satisfy the basic relation b' without satisfying any other basic relation from $\mathcal{B}$. For instance, PP

[37] An implementation to illustrate our method is also made available at the following link: https://github.com/ontologymerging/MergingOntologyWithQCN.

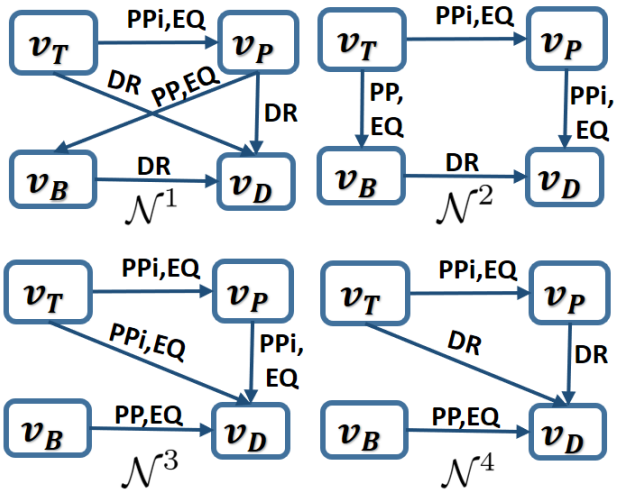


Figure 5.5: QCN profile of Example 5.1.

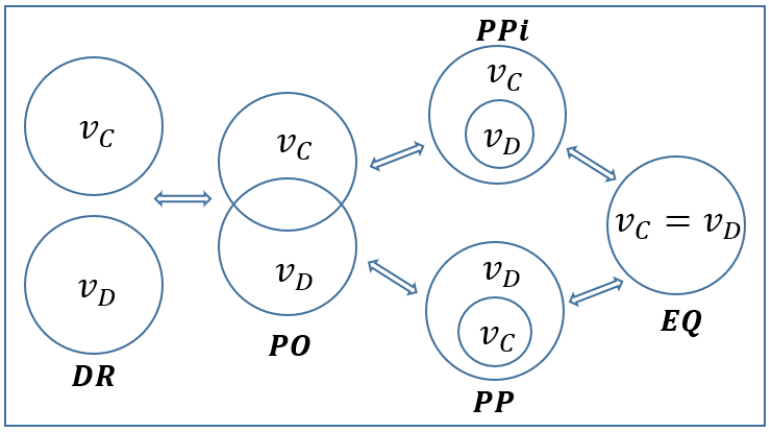


Figure 5.6: A conceptual neighborhood graph of *RCC-5*.

and EQ are CNs since "shrinking" a first region C initially equal to another region D directly makes it a proper part of D. This results in the neighborhood relation

$$\{(DR, PO), (PO, DR), (PO, PP), (PP, PO), (PO, PPi),$$

$$(PPi, PO), (PP, EQ), (EQ, PP), (PPi, EQ), (EQ, PPi)\}$$

It visually presents in Figure 5.6. This neighborhood relation induces a *neighborhood graph* $\mathcal{G}$ whose vertices are the elements of $\mathcal{B}$, and where there is an edge between two basic relations $b, b' \in \mathcal{B}$ whenever b and b' are CNs according the neighborhood relation. The distance $d(b, b')$ between two basic relations b and b' is defined as the length of the shortest path between b and b' in the neighborhood graph. For instance, $d(DR, PO) = 1$ since DR and PO are CNs, and $d(DR, EQ) = 3$ since DR and PO (resp. (PO, PP) and (PP, EQ)) are CNs, but DR and EQ are not. This distance is extended to a distance between a basic relation $b \in \mathcal{B}$ and a constraint $\varphi \in 2^{\mathcal{B}}$, defined as

$$d(b, \varphi) = \min_{b' \in \varphi} d(b, b')$$

Lastly, given two variables $v_C, v_D \in V$, the distance between each $b \in \mathcal{B}$ and the profile of constraints $\mathscr{E}_{CD} = \langle \Psi^1_{CD}, \ldots, \Psi^n_{CD} \rangle$ is defined by

$$d(b, \mathscr{E}_{CD}) = \sum_{i \in \{1,\ldots,n\}} d(b, \Psi^i_{CD})$$

Example 5.1 (continued) *Let us focus on $Text$ (T) and $Document$ (D). We have*

$$\mathscr{E}_{TD} = \langle \Psi^1_{TD}, \Psi^2_{TD}, \Psi^3_{TD}, \Psi^4_{TD} \rangle = \langle \{DR\}, \mathcal{B}, \{PPi, EQ\}, \{DR\} \rangle$$

For the distance between the basic relation PP and $\mathscr{E}_{TD}$, we have that $d(PP, \mathscr{E}_{TD}) = d(b, \Psi^1_{TD}) + d(b, \Psi^2_{TD}) + d(b, \Psi^3_{TD}) + d(b, \Psi^4_{TD})$, where:

$$\begin{aligned}
&d(PP, \Psi^1_{TD}) = d(PP, \{DR\}) = d(PP, DR) = 2,\\
&d(PP, \Psi^2_{TD}) = d(PP, \mathcal{B}) = d(PP, PP) = 0,\\
&d(PP, \Psi^3_{TD}) = d(PP, \{PPi, EQ\}) = d(PP, EQ) = 1,\\
&d(PP, \Psi^4_{TD}) = d(PP, \{DR\}) = d(PP, DR) = 2.
\end{aligned}$$

We get that $d(PP, \mathscr{E}_{TD}) = 5$. The distances between each basic relation from $\mathcal{B}$ and the profile of constraints $\mathscr{E}_{CD}$ for each pair of variables $v_C, v_D \in V$ is summarized in Table 5.4.

$\mathscr{E}$ / $\mathcal{B}$	$\mathscr{E}_{TP}$	$\mathscr{E}_{TB}$	$\mathscr{E}_{TD}$	$\mathscr{E}_{PB}$	$\mathscr{E}_{PD}$	$\mathscr{E}_{BD}$
DR	8	2	2	2	4	6
PO	4	2	3	1	4	4
PP	4	2	5	0	6	3
PPi	0	3	4	1	4	4
EQ	0	3	6	0	6	3

Table 5.4: Distances between relations from $\mathcal{B}$ and the profile of constraints $\mathscr{E}_{CD}$, for $v_C, v_D \in V$.

5.5.2 Using the Distance to Build a Merged Consistent QCN

We now describe our procedure which associates a profile of QCNs with a merged, consistent QCN. This takes advantage of the distance between basic relations and a profile of constraints $\mathscr{E}_{CD}$. Let us first formally define two intermediate functions $relax_{CD}$ and val_{CD} which are used in our procedure. Given a total preorder[38] $\preceq$ over a finite set E, let us denote by $\min(E, \preceq)$ the set of minimal elements of E w.r.t. $\preceq$, i.e.,

$$\min(E, \preceq) = \{e \in E \mid \forall e' \in E, e \preceq e'\}$$

Each pair of variables v_C, v_D is associated with a total preorder $\preceq_{CD}$ over the basic relations from $\mathcal{B}$ defined for all $b, b' \in \mathcal{B}$ as $b \preceq_{CD} b'$ iff $d(b, \mathscr{E}_{CD}) \leq d(b', \mathscr{E}_{CD})$. Then the function $relax_{CD}$ is a mapping $relax_{CD} : 2^{\mathcal{B}} \mapsto 2^{\mathcal{B}}$ defined for each $\varphi \in 2^{\mathcal{B}}$ as

$$relax_{CD}(\varphi) = \varphi \cup \min(\mathcal{B} \setminus \varphi, \preceq_{CD})$$

It corresponds to the *relaxation* of a constraint φ w.r.t.. the total preordering $\preceq_{CD}$. Noteworthy, $relax_{CD}(\emptyset)$ corresponds to the set of basic relations with a minimal distance to the profile of constraints $\mathscr{E}_{CD}$. The function val_{CD} is a mapping $val_{CD} : 2^{\mathcal{B}} \mapsto \mathbb{N}$ defined for each $\varphi \in 2^{\mathcal{B}}$ as

$$val_{CD}(\varphi) = \max_{b \in \varphi} d(b, \mathscr{E}_{CD})$$

[38] A total preorder over a set E is a total, symmetric and transitive relation.

Algorithm 2: Computing a merged QCN

input: A profile of QCNs $\mathcal{N} = \langle \mathcal{N}_1, \ldots, \mathcal{N}_n \rangle$
output: A merged, consistent QCN $\mathcal{N}$

```
1  begin
       // Initialization of the output QCN N
2      Ψ ⟵ {Ψ_CD | v_C, v_D ∈ V}
3      foreach (v_C, v_D) ∈ V × V do
4          Ψ_CD ⟵ relax_CD(∅)
5      N ⟵ (V, Ψ)
6      while N is not consistent do
           // One relaxes some constraints of N
7          S ⟵ {Ψ_CD | Ψ_CD ∈ Ψ, Ψ_CD ≠ B}
8          S ⟵ arg max{val_CD(Ψ_CD) | Ψ_CD ∈ S)}
9          foreach Ψ_CD ∈ S do
10             Ψ_CD ⟵ relax_CD(Ψ_CD)
11         N ⟵ (V, Ψ)
12     return N
```

For instance, according to Table 5.4 and focusing on $Book$ and $Document$ (cf. $\mathscr{E}_{BD}$), we get that $PP, EQ \preceq_{BD} PO, PPi \preceq_{BD} DR$, that $relax_{BD}(\emptyset) = \{PP, EQ\}$, that $relax_{BD}(\{PP, EQ\}) = \{PP, EQ, PO, PPi\}$, and that $val_{BD}(\{PP, EQ\}) = 3$.

We are ready to introduce our main procedure, whose outline is given in Algorithm 2 that defines an initial QCN $\mathcal{N}$ by setting each one of its constraints Ψ_{CD} to the set of basic relations from $\mathcal{B}$ having a distance to the profile of constraints $\mathscr{E}_{CD}$ that is minimal (lines 2 to 5). If this QCN is consistent, then it is returned as the merged QCN (line 12). If not, some of the constraints of $\mathcal{N}$ are relaxed in line 7, in the sense that some basic relations are added to these constraints. Such a set of constraints S is selected as follows. In line 7, S is first restricted to those constraints from $\mathcal{N}$ which *can* be relaxed, i.e., those constraints not equal to $\mathcal{B}$. Among those candidate constraints, one selects in line 8 the constraints Ψ_{CD} having a highest value $val_{CD}(\Psi_{CD})$. Indeed, we do not want to relax first the constraints consisting of basic relations which are "close" to the input profile, but rather would one prioritize the relaxation of more "controversial" constraints, i.e., those with a high value according to val_{CD}. For instance, let us look back at Table 5.4. It can be seen that $d(PPi, \mathscr{E}_{TP}) = d(EQ, \mathscr{E}_{TP}) = 0$, and thus $val_{TP}(\{PPi, EQ\}) = 0$; this low value reflects the consensus between sources on the fact that one of the basic relations PPi, EQ holds between $Text$ and $Paper$, and indeed it can be verified that the axiom $P \sqsubseteq T$ is consistent with each input $TBox$. On the contrary, one has that $val_{PD}(\{DR, PO, PPi\}) = 4$; this higher value reflects a disagreement between the input sources about the relationship between the concepts of $Paper$ and $Document$. And in the general case, whenever possible and in order to restore the consistency of the merged QCN, it is a sensible choice to keep unchanged those constraints unanimely accepted by the input sources, and rather weaken first the most disputed constraints. This "relaxation" process is repeated iteratively until the resulting QCN is consistent which, obviously enough, is guaranteed after a finite number of iterations.

Example 5.1 (continued) *Initially, the merged QCN $\mathcal{N} = \langle V, \Psi \rangle$ is defined by the following set of constraints, which correspond the basic relations highlighted in Table 5.4 (this QCN is also depicted in*

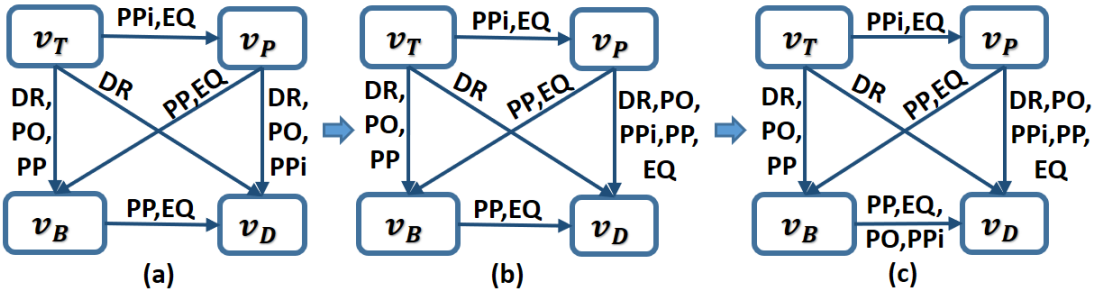


Figure 5.7: The QCNs iteratively generated by our algorithm.

Figure 5.7(a)):

$$\Psi_{TP} = \{PPi, EQ\} \qquad \Psi_{TB} = \{DR, PO, PP\}$$
$$\Psi_{TD} = \{DR\} \qquad \Psi_{PB} = \{PP, EQ\}$$
$$\Psi_{PD} = \{DR, PO, PPi\} \qquad \Psi_{BD} = \{PP, EQ\}$$

This QCN is inconsistent. One can see that the constraints Ψ_{PB} and Ψ_{BD} imply by transitivity that the relation between P and D must be PP or EQ, yet $\Psi_{PD} \cap \{PP, EQ\} = \emptyset$. Then the constraint Ψ_{PD} is selected (cf. line 8 in the algorithm) as the only candidate for relaxation at this point, since $val_{PD}(\Psi_{PD}) = 4$, which is the highest value among all constraints. And since $relax_{PD}(\Psi_{PD}) = \mathcal{B}$, one updates Ψ_{PD} to $\mathcal{B}$ which results in the QCN depicted in Figure 5.7(b). This QCN is, again, inconsistent (in this case, explaining its inconsistency is more complex as it involves dependencies between all four variables. We omit the details for space reasons). Then the constraint Ψ_{BD} is selected for relaxation ($val_{BD}(\Psi_{BD}) = 3$) and one updates Ψ_{BD} to $\{PP, EQ, PO, PPi\}$. The resulting QCN (cf. Figure 5.7(c)) is consistent and returned by the procedure. To sum up, Figure 5.7(a) and Figure 5.7(b) are inconsistent merged QCNs, Figure 5.7(c) corresponds to the final consistent merged QCN.

Algorithm 2 runs in a time that is polynomial on the size of the input QCN profile, given access to an NP oracle in one step (line 6). Indeed, (i) checking the consistency of an RCC-5 QCN can be performed by taking advantage of a standard SAT solver [CNS16], (ii) the number of possible relaxations of a given constraint is bounded by a constant (the number of RCC-5 basic relations), and as a consequence, the number of iterations performed in the loop starting from line 6 is in $O(|V \times V|)$; and (iii) the functions $relax_{CD}$ and val_{CD} are computed in $O(1)$ since the distance d is computed in $O(1)$, again because the number of RCC-5 basic relations is bounded by a constant. This makes the complexity of Algorithm 2 in FP^{NP}, that is reminiscent of the complexity of the inference problem for propositional belief merging, i.e., P^{NP}, for a vast majority of the existing operators [KM04].

Notably, the QCN merging step we propose in our framework can be seen as an adaptation of [CKMS10] that allows one to work with a single QCN as a merged result (the output of Algorithm 2). Nevertheless, the work of Condotta et al. [CKMS10] does not provide such a procedure.

5.6 Selecting a Representative Scenario of the Merged QCN

Once we have obtained a merged, consistent QCN, our goal is to express it in our initial (target) ontology language. However, not every constraint Ψ_{CD} from the QCN (i.e., a subset of $\mathcal{B}$) can easily be translated as a set of axioms, since non-singleton constraints express some disjunctive information between two concepts/regions. Moreover, since the merged QCN is consistent, one can remark it necessarily admits at least one consistent scenario, and since a scenario involves singleton constraints (as well as the two

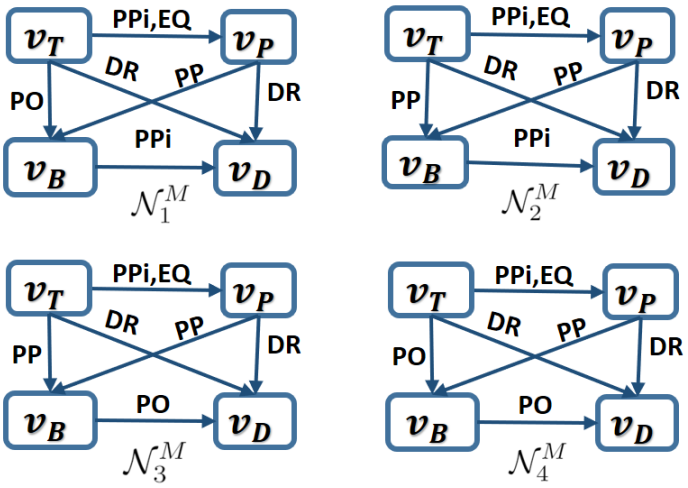


Figure 5.8: The consistent scenarios of the merged QCN.

constraints $\{PP, EQ\}$, $\{PPi, EQ\}$), it can easily be translated into a single ontology, as will be shown in the next section. Then our aim is to (i) focus on all consistent scenarios of the merged QCN, and (ii) select one representative scenario. This can be done by exploiting information provided by the input $ABoxes$ as we intend to show in the rest of this section.

In our running example, the merged QCN admits four consistent scenarios which are depicted in Figure 5.8. Let us first discuss why these four scenarios seem to be reasonable candidates to the input ontologies / QCNs provided by the sources. First, note that all input ontologies state that $Paper \sqsubseteq Text$ ($v_T\{PPi, EQ\}v_P$ in the QCN profile). And it can be seen that this consensus is reflected in the four candidate scenarios, which entail that information. Second, while the two sources $\mathcal{O}_1$ and $\mathcal{O}_4$ state that $Text$ and $Document$ are disjoint concepts ($v_T\{DR\}v_D$ in the corresponding input QCNs), only one source says that $Document \sqsubseteq Text$ ($v_T\{PP, EQ\}v_D$). In this case, it make sense to follow the point of view of the majority of the sources. And accordingly, in all four scenarios we have that $v_T\{DR\}v_D$, thus $Text$ and $Document$ are disjoint concepts. Third, the source $\mathcal{O}_1$ states that $Paper \sqsubseteq Book$ and all other sources have no information on these concepts. It is sensible to keep this information in the merged result, and accordingly all four scenarios entail that information. More, one sees that $v_P\{PP\}v_B$ holds in all scenarios, i.e., that $Paper$ is a *strict* part of $Book$, or stated otherwise, that both concepts cannot be equal while keeping the relationships between the remaining concepts consistent. This emergent property is also an interesting feature of the merging process. Last, the reason why there are four, equally reasonable, candidate scenarios is that some strong disagreements hold on the relationships between the concepts $Text$ and $Book$ on the one hand, and the concepts $Book$ and $Document$ on the other hand. Accordingly, the only differences between the four scenarios hold on the constraints between these two pairs of concepts ($v_T\{PO\}v_B$ / $v_T\{PP\}v_B$, and $v_B\{PPi\}v_D$ / $v_B\{PO\}v_D$).

What remains to be done is to select one of these four scenarios. For this purpose, one takes advantage of the $ABoxes$ from the input ontologies and see how these $ABoxes$ relate to each scenario. To be as faithful as possible to what each input source says, instead of simply considering each input $ABox$ as such, one considers the "closure" of it according to its corresponding $TBox$. For instance, if a given source states that $Paper \sqsubseteq Text$ in its $TBox$ and that $Paper(p)$ in its $ABox$, then it makes sense to also consider that $Text(p)$ also holds in that source's implicit knowledge. Formally, let $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ be

an ontology. The *deductive closure of* $\mathcal{A}$ w.r.t. $\mathcal{T}$, denoted by $\mathcal{C}l_{\mathcal{T}}(\mathcal{A})$, is defined [BCM+07, BBT15] as

$$\mathcal{C}l_{\mathcal{T}}(\mathcal{A}) \stackrel{\text{def}}{=} \{B(a) \mid B \in N_C, a \in N_I,\ \mathcal{O} \models B(a)\} \cup \{r(a,b) \mid \mathcal{O} \models r(a,b)\}$$

Accordingly, $\langle \mathcal{T}, \mathcal{A} \rangle$ is logically equivalent to $\langle \mathcal{T}, \mathcal{C}l_{\mathcal{T}}(\mathcal{A}) \rangle$.

To select the representative scenario, we define a distance between a scenario and the set of all input (closed) $ABoxes$, and then use this distance to choose the scenario. The representative scenario is the one having a minimal distance. So given a scenario, the idea is to count the number of individuals in each input $ABox$ which raise a conflict w.r.t. the constraints of that scenario. This can naturally be done for any scenario constraint $v_C\ \varphi\ v_D$ where $\varphi \neq \{PO\}$. For instance, if $Paper(p)$ and $Document(p)$ hold in the $ABox$ of a given source, and $v_P\{DR\}v_D$ holds in the scenario under consideration, then according to that $ABox$ p is an individual that raises a conflict with that scenario. Another example is if $Paper(p)$ holds but not $Text(p)$, then p is not a member of the concept $Text$ in the $ABox$ (recall that $ABoxes$ are closed w.r.t. their $TBox$); in that case p raises a conflict with the constraints $v_P\ \varphi\ v_T$ when $\varphi \in \{\{PP\}, \{EQ\}, \{PP, EQ\}\}$. More formally, given an $ABox$ $\mathcal{A}$, a scenario constraint $v_C\ \varphi\ v_D$ where $\varphi \neq \{PO\}$, and an individual p, we say that p *raises a conflict with* φ *w.r.t* $\mathcal{A}$ when:

$$\begin{array}{ll} C(p) \in \mathcal{A}, D(p) \notin \mathcal{A} & \text{when } \varphi \subseteq \{PP, EQ\}, \\ D(p) \in \mathcal{A}, C(p) \notin \mathcal{A} & \text{when } \varphi \subseteq \{PPi, EQ\}, \\ C(p), D(p) \in \mathcal{A} & \text{when } \varphi = \{DR\}. \end{array}$$

And the number of conflicts raised by an ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ w.r.t. a a scenario constraint $v_C\ \varphi\ v_D$ where $\varphi \neq \{PO\}$, is defined as

$$nbConf(\mathcal{O}, \varphi) = |\{p \in N_I \mid p \text{ raises a conflict with } \varphi \text{ w.r.t. } \mathcal{C}l_{\mathcal{T}}(\mathcal{A})\}|$$

The case of PO is more complex. Indeed, it can be easily seen that no individual can raise a conflict with $v_C\{PO\}v_D$. This is because all basic relations $\mathcal{B} \setminus \{PO\}$ express explicit dependencies between concepts / regions, whereas PO is a complementary relation that (explicitly) expresses a notion of *independency* between concepts. For instance, the concepts $Smoker$ and $Researcher$ can naturally be thought of as independent concepts, in the sense that one can easily find in a real-world context individuals that are members of either both concepts, only one of them, and none of them (note that this should not be confused with the case of DR, which expresses an explicit dependency between concepts, e.g., the concepts Dog and Cat.) So to evaluate the number of "conflicts" raised by an $ABox$ w.r.t. a constraint $v_C\{PO\}v_D$, we propose to count how "unbalanced" the number of conflicts are w.r.t. the remaining forms of constraints. Formally, focusing on the scenario constraint between two variables v_C and v_D,

$$nbConf(\mathcal{O}, \varphi) = \max_{\varphi' \neq \{PO\}} nbConf(\mathcal{O}, \varphi')) - \min_{\varphi' \neq \{PO\}} nbConf(\mathcal{O}, \varphi'))$$

For instance, when $nbConf(\mathcal{O}, \{PP, EQ\}) = nbConf(\mathcal{O}, \{PPi, EQ\}) = nbConf(\mathcal{O}, \{DR\})$, then $nbConf(\mathcal{O}, \{PO\}) = 0$: since individuals can be found equally (i) in both underlying concepts, and (ii) in one concept but not the other, $\mathcal{O}$ raises no conflict w.r.t. $\{PO\}$.

We have now a way to select the representative scenario from a given set of candidates. The distance between a scenario $\mathcal{N}^M$ and the input profile of ontologies $\mathcal{P} = \langle \mathcal{O}^1, \ldots, \mathcal{O}^n \rangle$ is simply defined as the overall number of conflicts raised by all input $ABoxes$ w.r.t. all constraints of $\mathcal{N}^M$, i.e.,

$$d(\mathcal{N}^M, \mathcal{P}) = \sum_{i \in \{1,\ldots,n\}, \varphi \in \mathcal{N}^M} nbConf(\mathcal{O}^i, \varphi)$$

Given a set of candidate scenarios, the representative scenario is then the one having a minimal distance.

Example 5.1 (continued) *Let us go back to our running example. We have that* $\mathcal{C}l_{\mathcal{T}^3}(\mathcal{A}^3) = \{P(p_3), P(b_3), P(d_3), T(t_3), T(d_3), T(b_3), T(p_3), D(d_3), D(b_3), B(b_3)\}$. *So focusing on* $Text$ *and* $Book$ *(i.e., on the scenario constraints between the variables* v_T *and* v_B*), we have that*

$$nbConf(\mathcal{O}^3, \{PP\}) = |\{t_3, d_3, p_3\}| = 3$$

and one can easily verify that

$$nbConf(\mathcal{O}^3, \{PO\}) = |\{t_3, d_3, p_3\}| = 3 - 0 = 3$$

Summing up all conflicts, we get that

$$d(\mathcal{N}_1^M, \mathcal{P}) = 20, d(\mathcal{N}_2^M, \mathcal{P}) = 18, d(\mathcal{N}_3^M, \mathcal{P}) = 22, d(\mathcal{N}_4^M, \mathcal{P}) = 24$$

Hence, the scenario $\mathcal{N}_2^M$ *is selected as a representative scenario of the merged QCN.*

5.7 Translating the Representative Scenario into a Terminological Knowledge

The last step of our framework is to translate back the resulting selected representative scenario into an ontology. The translation is defined as follows.

Definition 5.8 (Backward translation $\tau_\triangleleft$**)** *A* ***backward translation*** *is a function* $\tau_\triangleleft : V \longrightarrow \mathcal{C}$ *s.t.* $\tau_\triangleleft(v_C) \stackrel{\text{def}}{=} C$. $\tau_\triangleleft$ *is extended to map constraints into an ontology as follows, where* A', C', *and* D' *are new concept names and* a, b, *and* c *new individual names:*[39]

- $\tau_\triangleleft(v_C\{EQ\}v_D) \stackrel{\text{def}}{=} \langle\{C \equiv D\}, \emptyset\rangle$;
- $\tau_\triangleleft(v_C\{DR\}v_D) \stackrel{\text{def}}{=} \langle\{C \sqcap D \sqsubseteq \bot\}, \emptyset\rangle$;
- $\tau_\triangleleft(v_C\{PO\}v_D) \stackrel{\text{def}}{=} \langle\{A' \sqsubseteq C \sqcap D, C' \sqsubseteq C, C' \sqcap D \sqsubseteq \bot, D' \sqsubseteq D, D' \sqcap C \sqsubseteq \bot\}, \{A'(a), C(c), C(a), D(d), D(a), C'(c), D'(d)\}\rangle$;
- $\tau_\triangleleft(v_C\{PP, EQ\}v_D) \stackrel{\text{def}}{=} \langle\{C \sqsubseteq D\}, \emptyset\rangle$;
- $\tau_\triangleleft(v_C\{PPi, EQ\}v_D) \stackrel{\text{def}}{=} \langle\{D \sqsubseteq C\}, \emptyset\rangle$;
- $\tau_\triangleleft(v_C\{PP\}v_D) \stackrel{\text{def}}{=} \langle\{C \sqsubseteq D, D' \sqsubseteq D, C \sqcap D' \sqsubseteq \bot\}, \{D'(d), C(c), D(d), D(c)\}\rangle$, *and*
- $\tau_\triangleleft(v_C\{PPi\}v_D)) \stackrel{\text{def}}{=} \langle\{D \sqsubseteq C, C' \sqsubseteq C, D \sqcap C' \sqsubseteq \bot\}, \{C'(c), D(d), C(d), D(c)\}\rangle$.

Moreover, $\tau_\triangleleft$ *is extended to translate a set of constraints into an ontology in the (strict) normal form in the expected way:* $\tau_\triangleleft(\mathcal{N}) \stackrel{\text{def}}{=} \langle\mathcal{T}, \mathcal{A}\rangle$, *where*

$$\mathcal{T} \stackrel{\text{def}}{=} \bigcup_{\tau_\triangleleft(\Psi)=\langle\mathcal{T}', \mathcal{A}'\rangle, \Psi \in \mathcal{N}} \mathcal{T}'$$

and

$$\mathcal{A} \stackrel{\text{def}}{=} \bigcup_{\tau_\triangleleft(\Psi)=\langle\mathcal{T}', \mathcal{A}'\rangle, \Psi \in \mathcal{N}} \mathcal{A}'$$

[39] Notice that the constraint $\{PO\}$ cannot be translated into a set of GCIs only, whence the use of $ABox$ assertions in the translation.

Accordingly, our back translation is faithful. Using again the notions of inflation and flattening (cf. Definitions 5.6 and 5.7), we show that the set of solutions of a scenario $\mathcal{N}$ are captured precisely in its translated ontology $\tau_{\triangleleft}(\mathcal{N})$.

Theorem 5.3 *Let $\mathcal{N}$ be a scenario and $\mathcal{S}$ be solution of $\mathcal{N}$. Then there is an inflation $\mathcal{I}_{\mathcal{S}}$ of $\mathcal{S}$ s.t. $\mathcal{I}_{\mathcal{S}}$ is a model of $\tau_{\triangleleft}(\mathcal{N})$.*

Theorem 5.4 *Let $\mathcal{N}$ be a scenario, let $\mathcal{I}$ be a fulfilling interpretation of $\tau_{\triangleleft}(\mathcal{N})$ such that $\mathcal{I}$ is a model of $\tau_{\triangleleft}(\mathcal{N})$, and let $\mathcal{S}_{\mathcal{I}}$ be the flattening of $\mathcal{I}$. Then $\mathcal{S}_{\mathcal{I}} \models \mathcal{N}$.*

Example 5.1 (continued) *Let us translate the selected scenario $\mathcal{N}_2^M$ into an ontology. From Definition 5.8, we get that:* $\tau_{\triangleleft}(\mathcal{N}_2^M) = \langle \{(\Psi_{TP})\ P \sqsubseteq T, (\Psi_{TD})\ T \sqcap D \sqsubseteq \bot, (\Psi_{PD})\ P \sqcap D \sqsubseteq \bot, (\Psi_{TB})\ T \sqsubseteq B, SubBo1 \sqsubseteq B, SubBo1 \sqcap T \sqsubseteq \bot, (\Psi_{PB})\ P \sqsubseteq B, SubBo2 \sqsubseteq B, SubBo2 \sqcap P \sqsubseteq \bot, (\Psi_{BD})\ D \sqsubseteq B, SubBo3 \sqsubseteq B, SubBo3 \sqcap D \sqsubseteq \bot \}$,
$\{T(t_1), SubBo1(s_1), B(t_1), B(s_1), P(p_1), SubBo2(s_2), B(p_1), B(s_2), D(d_1), SubBo3(s_3), B(d_1), B(s_3)\}\rangle$.

5.8 Concluding Remarks

In this work, we introduced a terminological knowledge merging procedure based on qualitative reasoning over region spaces by providing a two-way translation between terminological knowledge and RCC-5-based qualitative constraint networks. Accordingly, even if the input sources are inconsistent when simply combined together, our approach returns a consistent result. It also remains as close as possible to the input sources, i.e., by preserving as much information as possible.

We have assumed that all input ontologies were provided in a strict normal form. Notably, the process of translating an ontology into another equivalent ontology in strict normal form can be done in polynomial time, but this process also allows one to obtain a full classification of all initial atomic concepts, for instance, by taking advantage of axioms involving constraints between complex concepts of the type $\exists r.C$ [BBL05]. For the scope of our work, we have focused on the merging of all atomic concepts from the input ontologies since taking account of more complex concepts may lead to considering constraints of higher arity in the translated qualitative constraint networks (e.g., for each role r in the signature, the assertion $\exists r.A \sqsubseteq \exists r.B$ holds when $A \sqsubseteq B$ holds, which would correspond to a constraint of arity four in the translation), and our merging procedure does not allow for these types of constraints. These kinds of considerations will be investigated in future work.

Moreover, another interesting direction for merging ontologies is an approach for representing regions and points in vector spaces [BS18, BGBS22] (combining natural language processing). Furthermore, the next work will also include the empirical evaluation of our method. In particular, we will assess the proposed merging operator and offer comparison results in the next chapter. Finally, using actual ontologies, we will provide a framework for measuring the result of merging in terms of computation time and operator quality.

Chapter 6

Tree Edit Distance Based Ontology Merging Evaluation Framework

Several proposals and implementations of merging operators have been developed in recent decades. To evaluate the quality of merging operators, we take inspiration from Tree Edit Distance (TED) [ZSS92] to create noisy ontologies and measure the distance between these ontologies. Furthermore, we will implement a framework to provide the significant information to compare the merging operators. Namely, we also evaluate the merging operators from two perspectives, including quantitative and qualitative results. Therein, we propose an algorithm to create noisy ontologies and a hybrid profile for applying the merging operators.

In addition, we carried out an empirical evaluation using the two operators suggested in Chapters 4 and 5 with nine different practical ontologies. Before we present the details of this approach, we shall motivate this work in the next section.

6.1 Context and Motivation

In the context of ontologies and structured knowledge, researchers and practitioners have to contend with the complex difficulties of multi-form fresh knowledge emerging from multiple sources. Because these knowledge sources are given by different agents (i.e., ontologies), merging many diverse sources will result in inconsistencies and conflicts. Furthermore, to broaden the scope of ontologies to several domains and deal with incompleteness and contradictory claims, one might merge knowledge from other sources. From that situation, the aspect of knowledge merging is one of the exciting issues in the last years, especially the view of ontology merging is a very interesting, invigorating, and challenging topic. In general, numerous merging operators have been proposed and developed in the last few years. The postulates are actually used to evaluate the behaviors of operators. However, to the best of our knowledge, there is no evaluation framework to assess and compare the quality of different merging operators. Their comparison and assessment have received little attention. Within the eyeshot of evaluating ontology merging methodologies, this work is motivated by two primary reasons:

1. *Evaluating the merging models for inputs from multiple fields (open-domain) is a difficult challenge*. Namely, while the open-domain concept names are same, their meanings differ, i.e., $Rock$ $is\text{-}a$ $Stone$ while $Rock$ $is\text{-}a$ $Music$ (in the music domain). Naturally, when merging these sources, there will be no meaningful relationship between $Stone$ and $Music$. Intuitively, neighboring concepts can relate to each other and might have the similar semantic kinds, i.e., $HeavyMetal$ $is\text{-}a$ $Music$, then it can be $HeavyMetal$ $is\text{-}a$ $Rock$. From this view, a simple idea is to merge

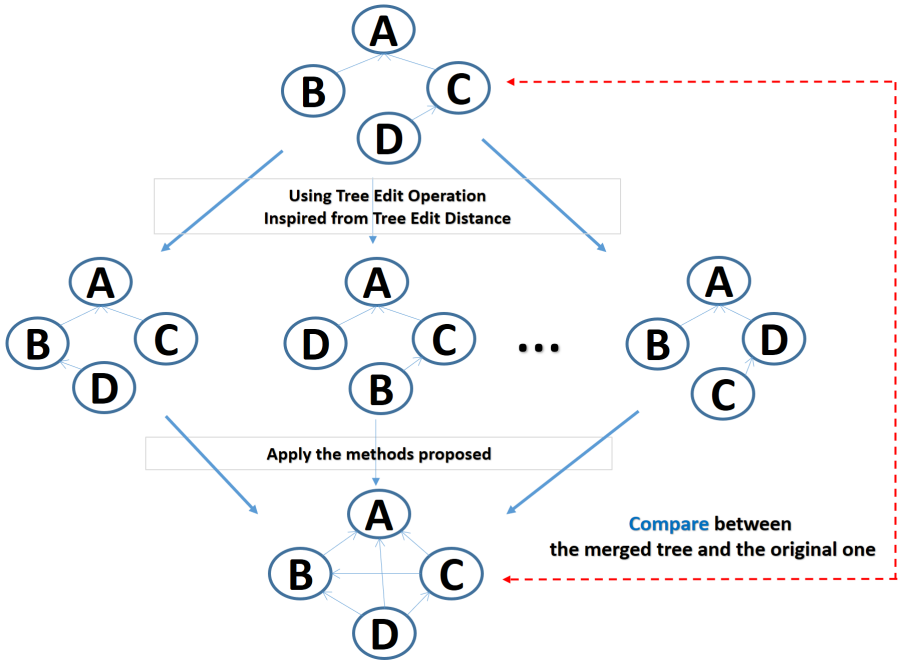


Figure 6.1: General idea of the ontology merging operator evaluation framework.

multiple close reproductions of a source. Such edited copies (close replicas) are interpreted as sources of local noise.

2. *Another challenge is to evaluate the merging models that require the same terminologies (signatures).* In fact, ontologies are naturally constructed based on the differing views of their builder. Therefore, even if they are the same field, the concept names are also different, i.e., $Vacation$ and $Holiday$. At these points, *the finding of ontology sources for the model evaluation is not an effortless task*

To this end, using noisy sources will be an exciting solution for measuring the efficacy of ontology merging frameworks. From the perspective of the noisy source, we take inspiration from the research branch on Graph Edit Distance [XHJJ19, Ser21] and Tree Edit Distance (TED)[40] [ZSS92, SPA17, ZS89] to propose a merging operator evaluation framework using a distance between attributed graphs (or trees). This distance is defined as the minimum amount of edit operations *(deletion, insertion, and substitution of nodes and edges)* needed to transform a graph (or tree) into another. Derived from the use of edit operations to compute the distances, we realize that the operations might *re-construct* the structured knowledge. Hence, we might use these modification functions to generate noisy ontologies (NoiOn(s)). At the same time, we take advantage of the TED to track the results during the merging process. The merged result that has the closest distance to the original tree will be a promising result.

Taking account of the foregoing, we propose an ontology merging evaluation framework to assess the quality of the ontology merging operators. The idea is to start with an ontology (gold standard), then build some NoiOn(s) (a merging profile), apply the merging operators to them, and compare the results to the gold standard. Particularly, we build the NoiOn(s) generation with "local" perturbation to make sense because neighboring concepts can be related to each other. Hence, we implement the modification of the substructure with a node's parent, child, and siblings to obtain the NoiOn. To this end, we provide

[40] http://tree-edit-distance.dbresearch.uni-salzburg.at

an algorithm. Then, we use the existing merging procedures to obtain the merged ontologies. Finally, we compare the merged and original results to evaluate the operators. The ability of noise reduction after merging will be a key to assessing the operators. Note that the ontology is represented as a tree (graph). The generation of NoiOn(s) and the merging of ontologies are two independent processes.

The figure 6.1 graphically summarizes the main idea of this contribution.

6.2 Background

Our approach is based on two foundations: (1) we rely on a lightweight Description Logic (DL) framework to encode terminological Boxes of ontologies, (2) we use edit operations and TED for creating NoiOn(s). This section briefly recalls the technical background required on Edit Operation and Tree Edit Distance. For the (1), we have detailed it in Section 2.3.3. Here, we use description logic $\mathcal{EL}_\bot$ for expressing the knowledge.

Edit Operation and Tree Edit Distance We here provide several formal definitions related to the TED to use in the sequels. A rooted tree, denoted by $\mathscr{T}$, is a connected graph with nodes $V(\mathscr{T})$ and edges $E(\mathscr{T}) \subseteq V(\mathscr{T}) \times V(\mathscr{T})$. The root of $\mathscr{T}$ is denoted by $\Re(\mathscr{T})$. Here, we write $\mathscr{T}$ to represent the set of nodes of $\mathscr{T}$ (replacing $\mathscr{T}$ by $V(\mathscr{T})$). For two nodes $v_1, v_2 \in \mathscr{T}$, a parent of v_1, denoted by $p(v_1)$, is the closest ancestor of A. The children of v_1, denote $c(v_1)$, is the set of closest nodes to v_1 among all descendants of v_1. A node v_1 is a leaf if it has no children. Moreover, we simply write $\mathscr{T} - v_1$ to denote the sub-graph of $\mathscr{T}$ obtained by removing a node v_1. We denote ϑ as a finite alphabet and $lb_\mathscr{T} : \mathscr{T} \to \vartheta$ as a labelling function. The labels of node in this work will be the concept names in an ontology $\mathcal{O}$. Hence, a finite alphabet ϑ is corresponding to a set of the concept names of N_C in $\mathcal{O}$.

The tree edit distance [ZSS92, SPA17, ZS89] between two trees $\mathscr{T}_1$ and $\mathscr{T}_2$ is defined as the minimum cost of edit operations to transform a tree to another. In other words, the TED is defined the minimun number of nodes that need to be replaced, removed, or inserted in $\mathscr{T}_1$ to obtain $\mathscr{T}_2$. We call these tasks *edit operations*.

In order to implement edit operations, we provide some formal definitions as follows:

Definition 6.1 *We say ϵ is a blank symbol and $\vartheta_\epsilon = \vartheta \cup \{\epsilon\}$ to represent the edit operations. We represent each of the edit operations $(v_1 \to v_2)$ by **a pair** denoted $(v_1, v_2) \in \vartheta_\epsilon \times \vartheta_\epsilon \setminus \{(\epsilon, \epsilon)\}$.*

Next, we define edit operations.

Definition 6.2 (Edit operations) *Let $\mathscr{T}$ be a tree, let $v_1, v_2 \in \mathscr{T}$, and let (v_1, v_2) be a pair of the edit operation. Edit operations on $\mathscr{T}$ include:*

- *Replacement:if $v_1 \neq \epsilon$ and $v_2 \neq \epsilon$, (v_1, v_2) is to replace v_1 with v_2 in $\mathscr{T}$. $Rep(v_1, v_2, \mathscr{T})$ is a function of the replacement operation;*
- *Deletion: if $v_1 \neq \epsilon$ and $v_2 = \epsilon$, (v_1, ϵ) means that we remove v_1 from $\mathscr{T}$ and keep $c(v_1)$ as children of $p(v_1)$. $Del(v_1, \epsilon, \mathscr{T})$ is a function of the deletion operation; and*
- *Insertion: if $v_1 = \epsilon$ and $v_2 \neq \epsilon$, (ϵ, v_2) means that we add v_1 into $\mathscr{T}$ such that v_1 is a child of v_2. $Ins(\epsilon, v_1, v_2, \mathscr{T})$ is a function of the insertion operation.*

Definition 6.3 *We denote $dist$ a metric cost function on edit operations defined on pairs of labels, i.e., $dist : \vartheta \times \vartheta \setminus \{(\epsilon, \epsilon)\} \to \mathbb{R}^+$. We define the cost of an edit operation by setting $dist(v_1 \to v_2) = dist(v_1, v_2)$. We simply write $dist(v_1, v_2)$ to represent $dist(lb(v_1), lb_2(v_2))$, where lb_1 and lb_2 are labelling functions on two trees $\mathscr{T}_1$ and $\mathscr{T}_2$.*

We will assume that $dist(v_1, v_2)$ is *a distance metric*.For any $v_1, v_2, v_3 \in \Psi_\epsilon$, the following conditions are satisfied:

1. $dist(v_1, v_2) \geq 0$;
2. $dist(v_1, v_1) = 0$;
3. $dist(v_1, v_2) = dist(v_2, v_1)$; and
4. $dist(v_1, v_3) \leq dist(v_1, v_2) + dist(v_2, v_3)$.

Normally, the edit operation cost of each operation is 1, i.e., $dist(v_1, v_2) = 1$ or $dist(v_1, \epsilon) = 1$. However, we may also re-define this cost (see [BBHS08] for more details). Notice that we will use the cost of each operation equals to 1 for this whole work.

Moreover, an edit distance mapping between $\mathscr{T}_1$ and $\mathscr{T}_2$ is a description of how a sequence of edit operations transforms from $\mathscr{T}_1$ into $\mathscr{T}_2$. Now, we present a mapping between two trees for computing the distance between them. Formally, we define the an edit mapping as follows

Definition 6.4 (Edit Mapping [PA20]) *A mapping $M_{\mathscr{T}_2}^{\mathscr{T}_1} \subseteq V(\mathscr{T}_1) \times V(\mathscr{T}_2)$ is an edit mapping from $\mathscr{T}_1$ to $\mathscr{T}_2$ if for any two node pairs $(v_1, s_1), (v_2, s_2) \in M_{\mathscr{T}_2}^{\mathscr{T}_1}$ the following conditions hold:*

1. ***One-to one:*** *$v_1 = v_2$ if and only if $s_1 = s_2$;*
2. ***Ancestor:*** *v_1 is an ancestor of v_2 if and only if s_1 is an ancestor of s_2; and*
3. ***Sibling:*** *v_1 is to the left of v_2 if and only if s_1 is to the left of s_2.*

A pair $(v_1, v_2) \in M_{\mathscr{T}_2}^{\mathscr{T}_1}$ is called a node alignment between $v_1 \in \mathscr{T}_1$ and $v_2 \in \mathscr{T}_2$. The set of all mappings between $\mathscr{T}_1$ and $\mathscr{T}_2$ is denoted by $\mathcal{M}(\mathscr{T}_1, \mathscr{T}_2)$. The cost of computing a mapping between the two trees is defined as:

Definition 6.5 *Let $\mathscr{T}_1$ and $\mathscr{T}_2$ be two trees, $M_{\mathscr{T}_2}^{\mathscr{T}_1}$ be a mapping between them, and $(v_1, v_2) \in M_{\mathscr{T}_2}^{\mathscr{T}_1}$. The cost of a mapping between $\mathscr{T}_1$ and $\mathscr{T}_2$ is defined as follows:*

$$dist(M_{\mathscr{T}_2}^{\mathscr{T}_1}) = \sum_{(v_1,v_2)\in M_{\mathscr{T}_2}^{\mathscr{T}_1}} dist(v_1, v_2) + \sum_{v_1 \in \mathscr{T}_1} dist(v_1, \epsilon) + \sum_{v_2 \in \mathscr{T}_2} dist(\epsilon, v_2)$$

Intuitively, the cost of mapping is a total of the cost of all edit operations with a mapping $M_{\mathscr{T}_2}^{\mathscr{T}_1}$ to transform $\mathscr{T}_1$ into $\mathscr{T}_2$. Now, we present how to compute the edit tree distance between two trees referring to [Tai79, ZS89] as follows:

Definition 6.6 (Tree Edit Distance) *Let $\mathscr{T}_1$ and $\mathscr{T}_2$ be two trees. The edit distance between $\mathscr{T}_1$ and $\mathscr{T}_2$ is defined as:*

$$dist(\mathscr{T}_1, \mathscr{T}_2) = \min\{dist(M_{\mathscr{T}_2}^{\mathscr{T}_1}) \mid M_{\mathscr{T}_2}^{\mathscr{T}_1} \in \mathcal{M}(\mathscr{T}_1, \mathscr{T}_2)\}$$

Note that we focus here on the original version of the TED with the a time complexity $\mathcal{O}(m^2 \times n^2)$ and a space complexity of $\mathcal{O}(m \times n)$, where m and n are the number of nodes in $\mathscr{T}_1$ and $\mathscr{T}_2$ [ZS89]. For this topic, there are many researches to improve the complexity, i.e., Pawlik and Agusten [PA16] proposed a new, memory efficient algorithm for the tree edit distance, AP-TED (All Path Tree Edit Distance). Their algorithms run at least as fast as RTED [PA11, PA14] without trading in memory efficiency. They have improved the worst-case time complexity to cubic time. Furthermore, there are many practical applications using this distance [Aku10, dCRGdSL04].

6.3 Ontology Merging Evaluation Framework

We propose a framework for generating and merging NoiOn(s). This framework's main purpose is to assess existing ontology merging operators. The mission is to establish a profile with many different NoiOn(s) from a single ontology source, then merge them. The input of the framework is an ontology, and the output is a merged noisy ontology in which the original axioms hold. Our framework includes seven steps S_i.

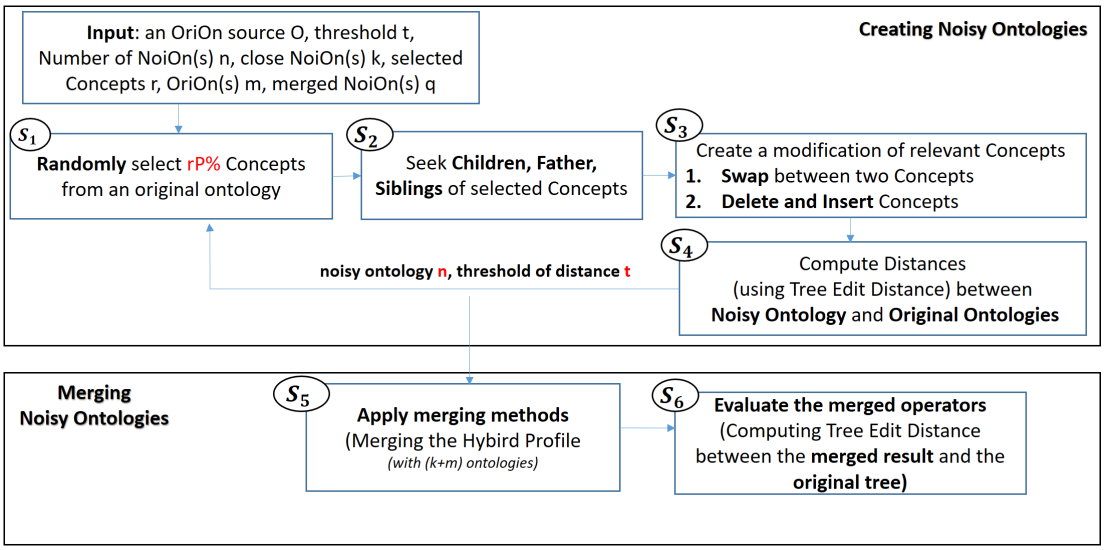


Figure 6.2: Framework of merging noisy ontologies.

S_1. We choose an input ontology and extract the pieces of knowledge. Here, we randomly collect the concepts' number $(rP\%)$ at step 1. These selected concepts will be the signatures to build NoiOn(s);

S_2. We filter out the relatives *(including Children, Fathers, Siblings)* of those concepts. This process is called collecting concept neighborhoods. Typically, assertions that are close to each other can be associated rather than searching for claims that are more apart. e.g., the assertion of the "Dog" concept is close to the statement of the "Wolf" concept because they are the same terrestrial animals that eat meat. Otherwise, the possibility of a relationship between "Dogs" and "Octopus" is not high since the two concepts can belong to two different species (groups). This is called local semantics, where the close relations can be meaningful. Therefore, we focus crucially on the surrounding neighbourhoods of concepts. Note that we *manage the local semantics* if they are *adjacent*. This is the preparation step towards establishing *a local perturbation*;

S_3. After collecting the concepts and their neighborhood, we use the edit operations (Def. 6.2) to create NoiOn(s). Note that we do not generate NoiOn(s) from all nodes but only concentrate on the $(rP\%)$ random nodes with their relatives. In particular, we re-structure the hierarchy of ontology by two methods:

- $[DI]$ Delete a node (or a concept) and insert it into an arbitrary position in the ontology hierarchy;

– $[Sw]$ Swap between two nodes (concepts) in the ontology hierarchy.

Regarding the $[Sw]$, we collect concept pairs *(i.e., Node-Children (NC), Node-Father (NF), Father-Children (FC)* from the set of selected concepts at S_2. Then, the pairs of NC, FC, and NF will be swapped to make noises for ontology. For the $[DI]$, we gather the pairs as (N_1, N_2) *with NC, FC, NF, and Node-Sibling (NS).* Then, edit operations include deleting N_1 and inserting N_1 as a child of the N_2;

S_4. We here select n-top NoiOn(s) close to the original one (based on the TED) to create a profile for merging. To this end, we compute the distances between an OriOn and NoiOn(s). Since the concept selection process (at S_1, S_2) and the edit operation of pairs (at S_3) perform in a random picking procedure, several NoiOn(s) will hence be distant (far) from the OriOn in terms of distance. Therefore, the distance computation step aim is to filter out the ontologies close to the input one. At the same time, we use a threshold t to detect the close NoiOn(s) and n to limit the number of NoiOn(s);

S_5. We take advantage of the existing ontology merging method for the framework. Here, if we only merge the NoiOn(s), the merging result can lost certain knowledge of the OriOn. Hence, we add m input ontologies into the profile to ensure that all original axioms are preserved. Then, a profile includes m OriOn(s) and k NoiOn(s). For the merging process, we use two merging methods of Bouraoui et al. (referring to [BKMV20b] and [BKM+22]);

S_6. After the merged result is obtained, we compute the distance between the merged result and the OriOn to measure the quality of merging operators.

In the sequel, we explicitly present how to collect NoiOn(s) based on the edit operations to create a profile for merging.

6.4 Building Ontology Profile

To begin, we denote $\mathscr{T}$ an ontology tree that corresponds to an ontology's taxonomic structure. Then, for simplicity of presentation, we refer to a noisy tree, a noisy ontology, and a noisy taxonomic structure of an ontology as the same thing. Here, the input of this process is an ontology, and the output is a set of NoiOn(s). Before collecting NoiOn(s), we define $\mathscr{T} \stackrel{\text{def}}{=} \mathbb{H}(\mathcal{O})$, where $\mathcal{O}$ is an ontology and $\mathbb{H} = \{\Phi \mid \Phi \in \mathcal{O}\}$ is a function to structure a hierarchical tree.

Example 6.1 *Let us consider a part of a "conference" ontology to illustrate our procedure, shown in Figure 6.3. Now, we denote $\mathscr{T}_{cf}$ a tree of "conference" ontology with 20 concepts as in Figure 6.3.*

To facilitate presentation in the latter, three first letters of each word in each concept name will be that concept name, i.e., $ComMem$ is known as $Committee\text{-}member$, $PaiApp$ is $Paid\text{-}applicant$, Rev is $Reviewer$, similar others. A set of axioms of an ontology tree is as follows: $\mathscr{T}_{cf} = \{Cha \sqsubseteq ComMem$, $RegAut \sqsubseteq ConCon$, $InvSpe \sqsubseteq ConCon$, $ConCon \sqsubseteq Per$, others $\}$.

Collection of Noisy Ontologies

We now present how to collect NoiOn(s) from an input ontology. The idea is to select some concept pairs and edit them. Formally, we represent how to collect the NoiOn(s) by Algorithm 3.

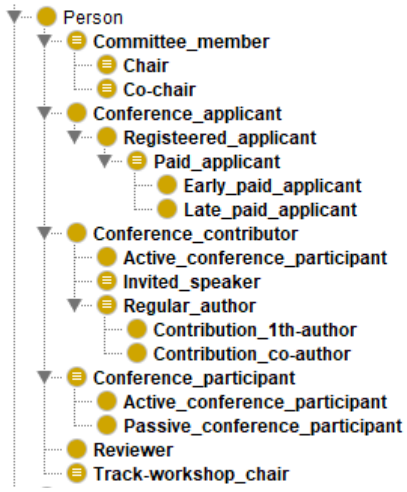


Figure 6.3: A part of a practical "conference" example.

We call Algorithm 3 as $\mathbb{G}(rP, n, t, \mathcal{O})$. The algorithm works as follows: we randomly collect concepts ($rP\%$) from OriOn(s) (line 4). e.g., $rP = 5$ and $n = 80$, then we have $5\% \times 80\ concepts = 4\ concepts$. We here define

$$\alpha = \Re_C(\mathcal{O}, rP) \stackrel{\text{def}}{=} \{C \in \sigma \mid \sigma \subseteq N_C, |\sigma| < nr\}$$

where

$$nr = \frac{rP \times |N_C|}{100}$$

Next, we collect the relative nodes in α (Children, Father, Sibling) at line 5. Given $A \in \alpha$, the concepts related to A denote as A's neighbors. We define

$$\boxplus_{Neighbours}(\mathcal{O}, \alpha) = \{g(\mathcal{O}, A) \mid A \in \alpha\}$$

where $g = \{NC, NF, FC, NS\}$. Therein,

$$NC(\mathcal{O}, A) = \{(A, B) \mid B \in \mathcal{O}, B \text{ is a Child of } A\},$$

$$NF(\mathcal{O}, A) = \{(B, A) \mid B \in \mathcal{O}, B \text{ is a Father of } A\},$$

$$FC(\mathcal{O}, A) = \{(B, A) \mid B \in \mathcal{O}, B \text{ is a close ancestor of } A\},$$

$$NS(\mathcal{O}, A) = \{(A, B) \mid B \in \mathcal{O}, B \text{ is a sibling of } A\}.$$

Example 6.2 *Continuing to Example 6.1, we assume that* $rP\% = 30\%$. *The concept number collected is* $30\% \times 20 = 6$. *Let us suppose we have* $\alpha = \{EarPaiApp, Rev, RegAut, RegApp, TraCha, CoCha\}$. *Then, taking account o* $RegAut \in \alpha$, *we have* $\boxplus_{Neighbours} = \{(RegAut, Con1thAu), (RegAut, ConCoAut), (ConCon, RegAut), (Per, RegAut), (RegAut, InvSpe), (RegAut, ActConPar)\}$.

From each node pair in $\boxplus_{Neighbours}$, we select randomly one method to modify the ontology, including the Deletion-Insertion operation ($[DI]$) (using $Del(v_C, \epsilon, \mathscr{T}^N)$ and $Ins(\epsilon, v_C, v_D, \mathscr{T}')$ (from line 9 to 10)) and the swap operation ($[Sw]$) (using replacement operation $Rep(v_C, v_D, \mathscr{T}^N)$ (at line 12)). Here, we define $\Re_M()$ is *a binary random function* (return True or False) to select either $[DI]$ or

Algorithm 3: Generating Noisy Ontologies.

input: rP: A Random Percentage of Concepts, n: Number of Noisy Trees, t: Threshold, $\mathcal{O}$: An OriOn
output: $\mathcal{P}$: A Set of Noisy Ontologies

1 **begin**
2 $\quad\mathscr{T} \longleftarrow \mathbb{H}(\mathcal{O}), \mathcal{P} \longleftarrow \emptyset$
3 $\quad$**while** $|\mathcal{P}| \leq n$ **do**
$\quad\quad$// Randomly collect $r\%$ concepts from ontologies
4 $\quad\quad\alpha \longleftarrow \Re_C(\mathcal{O}, rP)$
$\quad\quad$// Collect the neighbourhood concepts of the concepts in α
5 $\quad\quad V \longleftarrow \boxplus_{Neighbours}(\mathcal{O}, \alpha)$
$\quad\quad$// Extract the hierarchical structure of ontology
6 $\quad\quad\mathscr{T}^N \longleftarrow \mathbb{H}(\mathcal{O})$
7 $\quad\quad$**foreach** $(v_C, v_D) \in V \times V$ **do**
$\quad\quad\quad$// Randomly select one edit operation for a pair
$\quad\quad\quad$// True: use $[DI]$ method, False: use $[Sw]$ method
8 $\quad\quad\quad$**if** $\Re_M()\ is\ True$ **then**
9 $\quad\quad\quad\quad\mathscr{T}' \longleftarrow Del(v_C, \epsilon, \mathscr{T}^N)$
10 $\quad\quad\quad\quad\mathscr{T}^N \longleftarrow Ins(\epsilon, v_C, v_D, \mathscr{T}')$
11 $\quad\quad\quad$**else**
12 $\quad\quad\quad\quad\mathscr{T}^N \longleftarrow Rep(v_C, v_D, \mathscr{T}^N)$
13 $\quad\quad$**if** $0 < dist(\mathscr{T}^N, \mathscr{T}) \leq t$ *and* $\mathscr{T}^N \notin \mathcal{P}$ **then**
14 $\quad\quad\quad\mathcal{P} \longleftarrow \mathcal{P} \cup \mathscr{T}^N$
15 $\quad$**return** $\mathcal{P}$

$[Sw]$. We denote $\mathcal{P}$ as a set of NoiOn(s) or a profile. After the NoiOn obtains, we compute the distance between the NoiOn $\mathscr{T}^N$ and the OriOn $\mathscr{T}$ to measure their closeness. A NoiOn is collected into $\mathcal{P}$ ($\mathcal{P} \longleftarrow \mathcal{P} \cup \mathscr{T}^N$) if the distance is less than or equal to a threshold t ($dist(\mathscr{T}^N, \mathscr{T}) \leq t$) at line 13 and 14. We choose the acceptable threshold based on the number of concepts. The goal of threshold is to find a NoiOn close to the original one.

Finally, the number of NoiOn(s) depends on the parameter n at line 3 with the condition $|\mathcal{P}| \leq n$. This "$while$" loop works until the number of NoiOn(s) is fully collected. Note that the collection process of NoiOn(s) from line 3 to 14 runs in parallel to improve the computation time. Formally, a set of NoiOn(s) is defined as $\mathcal{P} \stackrel{\text{def}}{=} \mathbb{G}(rP, n, t, \mathcal{O})$. Here, Algorithm 3 runs in a time that is polynomial on the number n of NoiOn(s), given access to an NP oracle in one step (line 3). Indeed, (i) the size of $\mathcal{P}$ depends on threshold t (line 13), (ii), the function Del, Ins, Rep, and $\Re$ are computed in $O(1)$. (iii) the iteration's number from line 7 is in $O(|V \times V|)$. The algorithm complexity is in FP^{NP}.

Collecting a merging profile

After collecting the n NoiOn(s), we use the TED to filter the k NoiOn(s) that are closest to the OriOn. To this end, we say $\preceq_{\mathcal{P}}$ is a pre-order[41] on $\mathcal{P}$ such that

$$\forall \mathscr{T}_1^N, \mathscr{T}_2^N \in \mathcal{P}:\ \mathscr{T}_1^N \preceq_{\mathcal{P}} \mathscr{T}_2^N \text{iff } dist(\mathscr{T}_1^N, \mathscr{T}) \leq dist(\mathscr{T}_2^N, \mathscr{T})$$

[41] A preorder is a reflexive and transitive relation.

where $\mathscr{T}$ is a hierarchical structure of the OriOn $\mathcal{O}$. Note that, the noisy trees in $\mathcal{P}$ are always distinct since the condition $\mathscr{T}^N \notin \mathcal{P}$ at line 13 (Algorithm 3) is implemented.

Definition 6.7 *Let $\mathscr{T}$ be an ontology tree and $\preceq_{\mathcal{P}} = \{\mathscr{T}_1^N, \ldots, \mathscr{T}_n^N\}$ be a pre-ordered set of all noisy ontology trees. The set of "top-k" NoiOn trees close to the OriOn is defined as follows:*

$$\mathcal{P}_{Top}^k \stackrel{\text{def}}{=} \{\mathscr{T}_i^N \in \preceq_{\mathcal{P}} | \ 1 \leq i \leq k < n\}$$

Here, $\mathcal{P}_{Top}^k$ is a profile with k NoiOn(s). e.g., a profile with three noisy trees is $\mathcal{P}_{Top}^3$. As explained in S_5, because noisy trees are generated using randomization, in the worst-case scenario, an axiom of the original tree is not existing in all noisy trees. Therefore, it leads to no source stating that constraint. As a result, they may lose some of the axioms in OriOn. This is why we add some original trees to the profile to ensure that the merged result covers the OriOn. Formally, an extended profile is defined as:

Definition 6.8 *Let $\mathcal{P}_{Top}^k$ be a set of "k" NoiOn trees and $\mathscr{T}$ be an OriOn tree. A "hybrid" profile extended by adding "m" original trees is defined as follows:*

$$\mathcal{P}^{k,m} \stackrel{\text{def}}{=} \mathcal{P}_{Top}^k \cup \{\mathscr{T}_i = \mathscr{T} \mid 1 \leq i \leq m\}$$

Example 6.3 *Given $\mathcal{P}_{Top}^7 = \{\mathscr{T}_i^N\}$ with $i \in [1, 7]$. we apply Definition 6.8 with 3 OriOn(s). Then, we obtain $\mathcal{P}^{7,3} = \mathcal{P}_{Top}^7 \cup \{\mathscr{T}_1, \mathscr{T}_2, \mathscr{T}_3\}$ = $\{\mathscr{T}_1^N, \mathscr{T}_2^N, \mathscr{T}_3^N, \mathscr{T}_4^N, \mathscr{T}_5^N, \mathscr{T}_6^N, \mathscr{T}_7^N, \mathscr{T}_1, \mathscr{T}_2, \mathscr{T}_3\}$.*

Note that we denote $\mathcal{P}^{k,m} = \{\mathscr{T}_i\}$ corresponding to $\mathcal{P}^{k,m} = \{\mathcal{O}_i\}$, where $\mathscr{T}_i$ are hierarchical structures of $\mathcal{O}_i$. Now we apply the merging methods to the profile created.

6.5 Ontology Merging Operators

In this section, we evaluate two merging operators proposed in this thesis in order to merge NoiOn(s). Here, we denote $\mathbb{M}_{MBM}$ *as an ontology merging function with the "model-based merging" approach* [BKMV20b] (see Chapter 4) and denote $\mathbb{M}_{QM}$ *as an ontology merging function using the "QCN Merging" approach* [BKM+22] (see Chapter 5).

Let us denote $\mathcal{O}^M$ as the NoiOn merged result (using $\mathbb{M}_{MBM}$ and $\mathbb{M}_{QM}$). Note that the merging profiles are always different since the NoiOn(s) generated are different. Moreover, let us denote $dist(\mathcal{O}^M, \mathcal{O})$ as $dist(\mathscr{T}^M, \mathscr{T})$, where $\mathscr{T}^M = \mathbb{H}(\mathcal{O}^M)$. An merged result using the existing merging operators is defined as follows:

Definition 6.9 *Let $\mathcal{P}^{k,m}$ be a "hybrid" profile of NoiOn(s). A ontology merged result is defined as: $\mathcal{O}^M \stackrel{\text{def}}{=} y(\mathcal{P}^{k,m})$ where $y = \{\mathbb{M}_{MBM}, \mathbb{M}_{QM}\}$.*

Example 6.4 *We continue Example 6.3. Let us assume $q = 3$, $y = \mathbb{M}_{QM}$, $m = 3$, $k = 7$. We added original trees ($m = 3$) to maintain the integrity of merging (as explained above). We collect $\mathcal{W} = \{\mathcal{O}_1^M, \mathcal{O}_2^M, \mathcal{O}_3^M\}$, where $\mathcal{O}_i^M = \mathbb{M}_{MBM}(\mathcal{P}_i^{7,3})$ with $i \in [1, 3]$. Then, we compute the distance between each $\mathcal{O}_i^M$ and an OriOn $\mathcal{O}$. Assuming that we have $dist(\mathcal{O}_1^M, \mathcal{O}) = 32$, $dist(\mathcal{O}_2^M, \mathcal{O}) = 27$, $dist(\mathcal{O}_3^M, \mathcal{O}) = 38$. The final merged result will be $\mathcal{O}_2^M$ since $dist(\mathcal{O}_2^M, \mathcal{O}) = 27$ is a shortest distance. It also is known that $\mathcal{O}_2^M$ is closest to $\mathcal{O}$.*

6.6 Experimental Evaluation

In this section, we describe our implementation and interpret the experimental evaluation results.

Ontology Name	Nbr of Concepts of	Nbr of Relations is-a Relations	Nbr of Axioms
conference	57	55	397
cmt-2	30	38	318
ekaw	74	85	341
sigkdd	50	49	193
swo	86	144	349
cree-hydro	80	183	6,252
pto2	1,541	1,747	16,148
human	3,304	5,423	30,364
mouse	2,744	4,493	11,043

Table 6.1: Information of practical ontology sources.

6.6.1 Description of Implementation

We implement our framework[42] with both merging methods. We evaluate our framework with nine practical ontologies, namely *conference, cmt-2, ekaw, sigkdd, swo, Cree-hydro, pto, human, and mouse*. Table 6.1 shows the number of concepts and axioms. Here, we select these sources because we investigate how the framework reacts as the number of concepts and axioms increases. For hardware, we run the implementation with a computer with the following configuration: Intel core i7-7700 3.6Ghz (8 CPUs), 32Gb of memory, and Windows 10 Pro-64-bit. Otherwise, we run the experiment on a server (cluster) for large sources, such as $pto2$, $Human$, and $Mouse$. Recall that the process of merging is independent with the process of generating NoiOn(s).

Generating Noisy Ontologies In our experiment, we take account of axioms of the form $A \sqsubseteq B$, one of the most common forms used to express knowledge in the ontologies [MAWNH15, BBM17]. It is also known as is-a or subsumption relation, which builds the hierarchy of an ontology. If concepts do not have a father, the "Thing" concept will be assigned as their father. Based on this relation, we modify the structure of ontologies to obtain the NoiOn(s). In detail, we implement our whole framework on pure python (python 3.8). Regarding ontology, we take advantage of an Owlready2 library[43] to extract all the knowledge of the OriOn. We take account of the ontology hierarchy as a tree structure. We here use the python library named "networkx"[44] to optimize the paths (transitive relations) in the tree structure. The transitive axioms will be represented explicitly by this library. Note that the pre-processing step carries out to normalize the OriOn's axioms into the $\mathcal{EL}$ normal form [BBL05], which both methods require. Moreover, the "zss" python library[45] is also used to compute the distance between two trees (by the simple_distance() function). We implement the multi-processing procedure to improve the running time for generating noisy ontologies.

Implementing Merging Operations instead of taking account of all axioms into merging process, we filter out the set of the same axioms and the set of distinguishable axioms. We assume the same axioms as agreements statements and the different axioms as disagreement statements. The disagreement

[42] https://github.com/ontologymerging/NoisyOntologyMerging
[43] https://owlready2.readthedocs.io/en/v0.36/
[44] https://networkx.org/
[45] https://pythonhosted.org/zss/

statements will be taken into the merging procedures. This process reduces the number of considered axioms and improves the running time.

1. For the merging method of the "semantic" approach [BKMV20b], since the process of generating all possible interpretations is huge; therefore, a timeout variable is implemented to seek out the acceptable result. Otherwise, if the number of disagreement statements is large *(i.e., greater than 80 axioms for the human ontology)*, we cluster the axioms independently in order to collect local sub-trees, which we subsequently merge. A local sub-tree corresponds to a sub-hierarchy within a general ontology structure. If a sub-tree remains enormous (i.e., more than 40 axioms per sub-tree), we will disregard (ignore) it and return to the beginning in search of a new profile. This step resolves the enumeration knowledge explosion problem. Intuitively, it makes sense to ignore profiles for which a subtree contains an excessive amount of disagreement, as these profiles would be extremely distant from the OriOn.

2. We are interested in the disjoint relation for the merging method of the "QCN" approach [BKM$^+$22] along with the subsumption relation. ($A \sqcap B \sqsubseteq \bot$). We use a "PyRCC8" python library[46] to check the consistency of the QCN. As we know, if the number of concepts (regions) increases, the enumeration of all possible QCNs will also be an explosion and huge. Therefore, we solve the problem of generating all possible QCNs by the P-MAXSAT [CNS16]. First, we translate the merged QCN into the CNF format using the FCTE encoding. Next, based on this encoding, we build a CNF file underlying a DMACS format. Then, we use a MAXSAT solver[47] (using RC2) to enumerate the possible consistent QCNs. After collecting the solver's consistent solutions, we translate the CNF models back into the QCNs. Leveraging the power of the MAXSAT solver is a suitable choice to improve the processing time in the enumeration process because acceptable consistent solutions are always able to be found (with tolerable confidence) that do not need to enumerate all possible cases.

6.6.2 Quantitative Results

We evaluate the computation time of the two main parts for each dataset, including the time of generating NoiOn(s) denoted as TG and the time of merging them denoted as TM. Therein, we test our proposal with several parameters, including

1. The number of noisy trees generated n;
2. The number of noisy trees selected into a profile k;
3. NoiOn(s) combined with m OriOn(s);
4. The percentage of nodes to generate the NoiOn(s) rP;
5. The threshold t of selecting close NoiOn(s); and
6. The number merged ontologies q to seek out the final merged ontology.

These parameters and the computation time are shown in Table 6.2. The results are based on an average of ten evaluations (a ten-fold evaluation) of the running time. We investigate our framework on two merging methods, including $\mathbb{M}_{MBM}$ [BKMV20b], and $\mathbb{M}_{QM}$ [BKM$^+$22] (see Section 6.5).

[46] https://pypi.org/project/PyRCC8/
[47] https://pysathq.github.io/

Ontology Name	n	k,m	rP	t	TG	TM	
						$\mathbb{M}_{MBM}$	$\mathbb{M}_{QM}$
conference	400	20,5	15%	40	281.96	77.21	61.47
	500	25,5	17%	45	301.75	92.57	70.58
	600	30,5	19%	50	322.58	130.58	83.81
	700	35,10	21%	55	342.62	223.58	101.58
	800	40,10	23%	60	380.17	370.5	135.86
cmt-2	400	20,5	15%	40	129.86	58.68	42.54
	500	25,5	17%	45	141.47	78.56	56.57
	600	30,5	19%	50	169.58	125.85	71.19
	700	35,10	21%	55	190.11	201.56	83.92
	800	40,10	23%	60	219.78	305.11	101.58
ekaw	300	20,5	15%	40	212.47	91.49	85.27
	400	25,5	17%	45	237.58	143.57	101.58
	500	30,5	19%	50	261.68	212.36	126.58
	600	35,10	21%	55	290.41	329.10	153.51
	700	40,10	23%	60	309.17	475.79	190.92
sigkdd	300	15,5	15%	50	244.39	74.52	59.46
	400	20,5	17%	55	277.52	93.58	58.95
	500	25,5	19%	60	301.98	128.75	77.54
	600	30,10	21%	65	344.87	230.77	85.16
	700	40,10	23%	70	391.95	358.78	109.68
swo	200	15,5	10%	50	301.26	439.67	171.23
	300	20,5	12%	55	331.72	659.85	195.58
	400	25,5	14%	65	367.20	965.75	235.75
	500	30,10	16%	70	398.47	1,155.58	268.19
	600	40,10	17%	75	426.54	1,401.57	301.87
cree-hydro	200	15,5	10%	80	255.62	682.26	235.43
	200	20,5	12%	85	285.64	978.67	312.58
	300	25,5	14%	95	305.79	1,657.86	429.67
	400	30,10	16%	105	342.56	2,038.56	558.92
	500	40,10	17%	120	376.85	2,687.69	701.12
pto	135	10,5	3%	150	637.18	3,162.98	1,586.68
	150	15,5	5%	155	887.56	6,257.89	2,789.65
	170	20,10	7%	160	1,058.23	14,578.12	4,782.22
	190	25,10	9%	190	1,237.33	18,256.62	7,078.66
	210	30,10	11%	230	1,576.84	21,687.55	9,357.23
human	80	10,5	3%	400	1,765.21	7,158.19	3,758.52
	90	15,5	5%	600	2,368.71	13,785.55	5,598.56
	110	20,10	7%	800	4,172.65	23,928.23	9,218.87
	120	25,10	9%	1000	7,125.87	32,078.97	13,585.79
	150	30,10	11%	1300	9,075.87	42,231,78	17,178.56
mouse	80	10,5	3%	400	1,177.16	6,024.52	2,981.21
	90	15,5	5%	600	2207.89	10,786.83	4,257.89
	110	20,10	7%	800	3,920.81	17,859.77	7.587.58
	120	25,10	9%	1000	5,120.78	26,879.68	11.585.85
	150	30,10	11%	1000	7,783.45	37,781.12	14,792.34

Table 6.2: Parameters and computation time.

From the result obtained, the merging time (TM) of $\mathbb{M}_{QM}$ is faster than $\mathbb{M}_{MBM}$ (see Table 6.2). A reason is that the effective existing support tools of $\mathbb{M}_{QM}$ improve the computation time. Regarding the large ontology, such as $human$ and $mouse$, $\mathbb{M}_{QM}$'s merging time is acceptable and effective in practice with about 50-70 minutes, i.e., $3,758.22s$ for $human$ ontology. At the same time, $\mathbb{M}_{MBM}$ can spend about two hours on around 100 concepts ($3\% \times 3,304 = 99.12$). i.e., $7,158.19s$ for human ontology. In general, the merging time of $\mathbb{M}_{MBM}$ increases rapidly in proportion to the number of

ontology (subsumption) axioms, while the number of concepts influences the merging time of $\mathbb{M}_{QM}$. Otherwise, regarding TG, the computation time is also suitable in practice with around 20 minutes for a large ontology (i.e., $1,765.21$ for human ontology). Although the number of noisy ontologies collected is high (400 NoiOn(s)), the TG is still around 6-7 minutes. (i.e., it is $377.60s$ for the "conference" ontology). The threshold t directly impacts TG because NoiOn(s) collected depend on this threshold. Moreover, TG also depends on the number of CPUs since we implement this procedure with multi-processing. The calculation time is generally appropriate in practice through merging local sub-trees. Intuitively, the $\mathbb{M}_{QM}$ is sufficient and more efficient to solve the problems in practice than the $\mathbb{M}_{MBM}$ for the computation time aspect.

In a discussion of Table 6.2, we present an experiment to evaluate the running time of two operators. Here, we have tested many parameter sets with different types of ontology. When taking into account the noise ontology generation time, it is always consistent in practice, i.e., it takes about 5-10 minutes. Most of these generation steps will not take too long since we have already implemented the generation process in parallel. Moreover, the t parameter has been selected appropriately for each knowledge type (note that we have done the selection parameter t after many experimental runs). For large ontologies, the processing time is 2.5 hours (i.e., $human$). Intuitively, the generation time of noisy ontology is acceptable and suitable for real-life cases for a large number of concepts. Regarding TM, for the large ontology, the time of $\mathbb{M}_{MBM}$ is very large, while the time of $\mathbb{M}_{QM}$ is faster than $\mathbb{M}_{MBM}$. When taking account of a large number of concepts, the time of the large $\mathbb{M}_{MBM}$ is almost twice that of the $\mathbb{M}_{QM}$.

This shows that both respond well in real-time for a small dataset, but if the number of conflicting statements increases, the time of $\mathbb{M}_{MBM}$ will go uphill fast. The $\mathbb{M}_{QM}$'s time growth rate is moderate and agrees well with most ontologies' practice.

6.6.3 Qualitative Results

First, we provide a distance measurement table (see Table 6.3). It includes the distance between the OriOn and the NoiOn(s) (column 3-5) denoted by $[ON]$ and the distance between the OriOn and merged results denoted by $[OM]$ (column 6-7). Here, the TED is a measurable tool of an operator's noise-canceling ability. From these distances, we can compare and evaluate the operator's quality. For instance, these distances indicate that the NoiOn(s) generated are not distant to the original tree, i.e., the Human source's average distance is 187.57, which is small compared to $5,423$ subsumption relations. Otherwise, we also show how the merging operators work with noisy ontologies. In general, the $\mathbb{M}_{MBM}$ seems to be better than $\mathbb{M}_{QM}$ *(compare between column 6 and 7 in Table 6.3)*. However, both $\mathbb{M}_{MBM}$ and $\mathbb{M}_{QM}$ are effective because the $[OM]$ is always smaller than the average distance of $[ON]$. The TEDs have shown that the merge operator reduces the noise of the sources. Intuitively, the difference in the quality of the two operators is not much since the variance of distance between them is small (i.e., regarding $conference$ ontology, 4.46 of $\mathbb{M}_{MBM}$ and 5.82 $\mathbb{M}_{QM}$; the difference is $5.82 - 4.46 = 1.36$). Recall that, the distance of each edit operation is 1, for $human$ ontology, we have 127.68 of $\mathbb{M}_{MBM}$ (or 134.52 of $\mathbb{M}_{QM}$) is corresponding to 127.68 (or 134.52) modifications. Hence, if we compare 127.68 modifications with $30,364$ of input axioms, these changes are quite minimal (implying that the merged ontology is close to the OriOn).

In the following example, we provide the input sub-tree and the result of merging to illustrate that the merging outcome fully covers the original one.

Example 6.5 *Let us consider a sub-tree case of the "human" conference. We here have a hierarchy of* $ExtremityPart$ *in Figure 6.4(a). The merging result is shown in Figure 6.4(b). For this case, we generate* 100 *noisy ontologies with* $rP = 2\%$, $t = 80$, $q = 30$, $k = 10$, *and* $m = 5$. *Several new*

Ontology Names	Times	Distance ($[ON]$) (OriOn and NoiOn)			Distance ($[OM]$) (OriOn and Merged Result)	
		Avg	Min	Max	$\mathbb{M}_{MBM}$	$\mathbb{M}_{QM}$
conference	50	12.70	3	27	4.46	5.82
cmt-2	50	11.48	2	25	4.12	5.37
ekaw	50	12.78	2	28	6.78	7.86
sigkdd	50	15.85	3	32	5.83	7.03
swo	40	21.78	6	33	10.18	12.79
cree-hydro	40	33.86	9	67	14.25	16.76
pto	20	43.62	14	107	22.78	25.40
human	20	187.57	62	376	127.68	134.52
mouse	20	143.70	42	307	86.47	91.88

Table 6.3: Measurement of the distance (TED) to evaluate merging operators.

constraints using the $\mathbb{M}_{MBM}$ are as follows: $Toe \sqsubseteq Foot$, $Forearm \sqsubseteq Arm$, $Finger \sqsubseteq Hand$, $Hand \sqsubseteq Arm$, $Foot \sqsubseteq Leg$, others as shown in the structure in Figure 6.4(b).

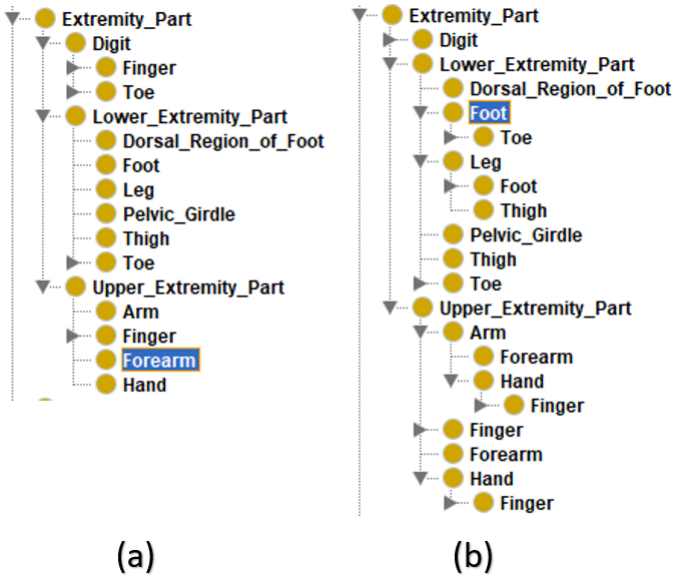


Figure 6.4: Structure of $ExtremityPart$ in the human ontology.

In this case, the merging result is plausible and acceptable. Since we add $m = 5$ OriOn(s), the main structure holds.

Example 6.6 *Let us consider Example 6.1. We here have a hierarchy of $Conference_Contribution$ in Figure 6.5(a). The merging result is shown in Figure 6.5(b) and 6.5(c). For this case, we generate 500 noisy ontologies with $rP = 10\%$, $t = 30$, $q = 100$, $k = 10$, and $m = 5$. Several new constraints using the $\mathbb{M}_{QM}$ are as follows: $InvSpe \sqsubseteq RegAut$, $ActConPar \sqsubseteq RegAut$, $InvSpe \sqsubseteq ConPar$, $ConCon \sqcap PassConPar \sqsubseteq \bot$, $InvSpe \sqcap PassConPar \sqsubseteq \bot$, $RegAut \sqcap PassConPar \sqsubseteq \bot$ shown in Figure 6.5(b).*

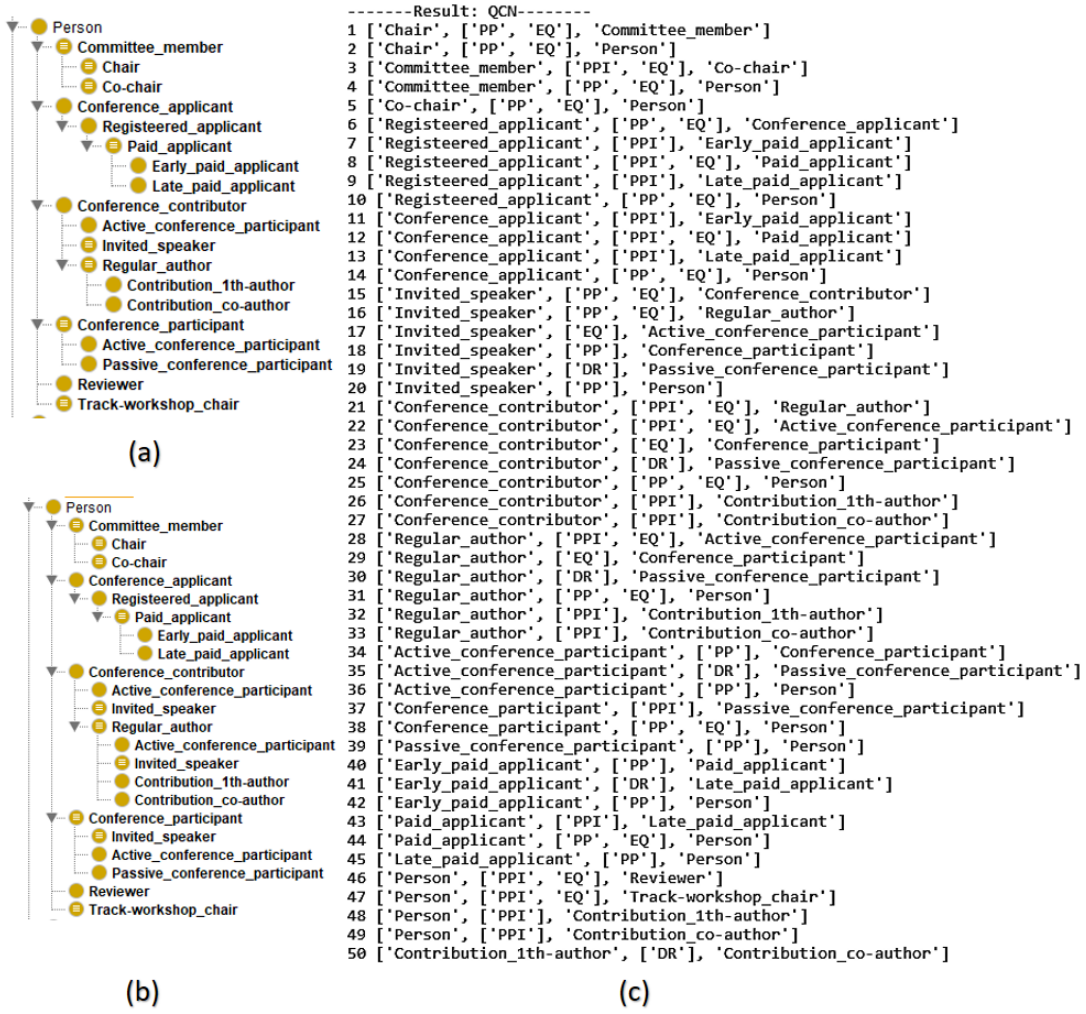


Figure 6.5: Structure of $Conference_Contribution$ in the conference ontology.

There are some disjoint relations in this cases, including $Con1ThAut \sqcap ConCoAuth \sqsubseteq \bot$, $EarPaiApp \sqcap LatPaiApp \sqsubseteq \bot$, $ActConPar \sqcap PasConPar \sqsubseteq \bot$. *The* $\mathbb{M}_{QM}$ *hence is reasonable to merge this ontology. In addition, holding the original axioms of ontology, this method represents all explicit axioms (shown all transitive axioms) in Figure 6.5(c).*

6.7 Concluding Remarks

In this chapter, we introduce a framework to assess ontology merging operators. Therein, we also provide an algorithm to create noisy ontologies by modifying the structure of a source. Moreover, this framework evaluates ontology merging models on the computation time and the distance measurement. Finally, an experimental result with the practical ontologies is shown and discussed. Intuitively, the majority of merged outcomes are acceptable and sensible since they are constantly cross-checked against the original (using TED). Furthermore, the process of generating the NoiOn(s) is separate from the ontology merging process. As a result, the evaluation of the models is unaffected (impacted) by any other external factors. This framework's advantage is that it provides several pieces of evidence for comparing operators and can be adjusted to accommodate various knowledge bases. Its limitation, however, should be noted

that the generation of ontologies is a process of local random selection and perturbations without taking into account the semantics and meanings of those shuffling concepts. Therefore, we can enhance the process of building noisy ontologies by taking account of the semantics of concept names rather than randomly selecting them in the future direction. In addition, we might experiment with other methods for computing the distance between two (or more) trees in order to improve accuracy and reduce calculation time [AJ21, ABG08, Wan22].

Chapter 7

Conclusion and Discussion

7.1 Summary

This work has contributed to the AI scientific community on several merging methodologies for structured knowledge bases. Our contributions addressed the issues caused by the merging of various knowledge sources. Once the problems are solved, we obtain a significant, centralized, and complete source of knowledge to support building applications for small and medium-sized enterprises and companies. Many AI scientists have recently pointed out the ontology completion challenges of finding common knowledge representing a subject. Naturally, multiple sources of knowledge are more comprehensive than a single source, i.e., an idiom says, "*Together we can change the world.*" Indeed, we can combine these many sources of knowledge and bring them into the "brain" of a computer. In that case, we obtain a world of knowledge to support many missions, i.e., chatbots, data mining, prediction, and others. However, each source will have different perspectives and interactions with the worldview. Hence, some pieces of knowledge will contradict each other when we put them into "a common box". Recall that we assumed that all knowledge bases always have reasonable beliefs. Therefore, it will not have any prioritization between sources. For this thesis, we proposed two merging operators to solve the arising problem when merging knowledge bases.

- In the first work, we focused on addressing different and even opposing opinions to reduce the flattening of structured knowledge (or to avoid leveling the ontology hierarchy). We proposed a merging solution inspired by the belief merging for propositional logic [KP02]. Here, we solved the semantic conflicts. Since the solution approach was to enumerate all of the worlds of interpretation and picked up the representative closest to the statements of the input sources, it could appear both positive and negative. The advantage is that we undoubtedly found a consistent, acceptable solution that could reflect common knowledge of sources and sometimes even found new rules from the listed world of interpretations. Otherwise, the disadvantage is that we consumed a ton of "resources and finance" to list all the solutions. In other words, this method took a great deal of time when faced with large amounts of knowledge where there are many conflicting opinions. However, we solved this problem by clustering dissent and processing each cluster to reduce the cost of enumeration. In particular, we enumerated interpretations and handled conflicts with each sub-structure in parallel. Nevertheless, it was still costly if there were many neighborhoods in the cluster under consideration. We discuss some future work for this problem in the next section.

- For the second task, we put attention to the logical inconsistencies (i.e., logical conflicts). Here, we did not directly propose the ontology merging operator to solve this kind of contradiction. Instead, we suggested an ontology merging solution in a different environment. Indeed, we merged the

knowledge into a region space that is more expressive than the ontology. Namely, we provided a faithful 2-way translation between description logic and regional space, i.e., a translation between $\mathcal{EL}_\bot$ and RCC-5. And then, we merged the knowledge in this space where the constraints were encoded in the QCNs. For the enumeration problem, at the last step, our merging method still needed to enumerate all the scenarios of the QCNs and to select a scenario closest to the originals. Fortunately, we took advantage of a study by Condotta et al., [CNS16] using MAXSAT to enumerate all the consistent scenarios of the QCN. Hence, we reduced the cost of automatically generating scenarios. Furthermore, this merging method was efficient and reliable because of the inference power of RCC-5, and a relaxation algorithm has been proposed. Generally, the merged results were consistent and faithful. However, some points for improvement had been pointed out in the study (cf. section 5.8), which were related to some special cases of existential rules, i.e., $\exists r.A \sqsubseteq \exists r.B$. This issue was a rare case in practice. Therefore, we had to disregard it and put it to work in the future.

In addition to the two operators suggested to handle real-world challenges associated with structured knowledge bases, we question how efficient our operators are. Which of the following operators is better? Which operator is most appropriate for the problems in practice? To answer these questions, we realized that we can evaluate operators by comparing the postulates reflecting the behaviors of the operators. However, no research addresses how to evaluate the quality of operators in practice (to the best of our knowledge). Therefore, we proposed a solution to evaluate the two above-mentioned operators. We measured the operator's ability to reduce noise in the knowledge merging process to investigate the operator's efficiency. In general, the advantage of our approach is that it does not endeavor to seek out knowledge sources with the same terminologies. Here, the important point is that we used distance and time to assess the performance of different operators. Because of this, it seems easy to pick the operators suitable for the specific problems. Although we took account of noise generation with local perturbations to guarantee the preservation of local semantic features, the random selection of concepts could lead to a lack of coverage and generalization. We also implemented the method to evaluate the merging operator with nine practical ontologies. Moreover, many flexible parameters had also been provided to adapt to different branches of representing knowledge. This measure had given an overview of operator evaluation.

In summary, from our experiments, the proposed operators are efficient and suitable for real-world problems. Moreover, merged results are always consistent, faithful, and can cover the sources of knowledge. In addition to the immediate benefits of this thesis' contribution, we will discuss the open questions for each research and provide the following directions for further advancement in favor of the thesis in the next section.

7.2 Open Questions and Future Directions

For any research, we will also look for extensions or new perspectives for the next steps. Hence, we will discuss the direction from general to specific based on what this thesis has to offer.

Some following questions appear from a general point of view:

1. In some special cases, the family of description logic $\mathcal{EL}$ will not be able to represent all statements of social life, i.e., Vietnam's population policy: whether it is a girl or a boy, no more than two children are allowed. Hence, how will we take account of merging more complex statements?

2. When considering ontology merging, some ones in the web semantic community can immediately

think of finding correspondences between two or more sources [48] [OVHP22]. For this thesis, we took advantage of mapping before the merging process is conducted. Here, merging and mapping are separate processes. Hence, some questions consist of: since the merged result will be directly affected by the mapping process, whether can we improve the mapping based on the idea of belief merging? Or, how is there a way to match and merge at the same time?

3. In the era of Industry 4.0, machine learning and big data techniques have been utilized in many practical applications. Hence, considering these techniques combined with a symbolic approach will be also one of the exciting combinations. Therefore, how can we merge structured knowledge with machine learning?

To solve these questions, we discuss some possible research directions in the future as follows:

(1) For the first question, because our merging approach pays attention to the description logics that fit the practical cases, i.e., description logic $\mathcal{EL}$ and its extension $\mathcal{EL}_\bot$, therefore, we can extend the investigation to other description logics, such as $\mathcal{ALC}, \mathcal{SROIQ}, \mathcal{SHIQ}, \mathcal{SHOIN}$, et al. to explore more complex knowledge sources. Moreover, when selecting more expressive logic languages, their complexity will increase progressively. Hence, we can choose languages that are close to the needs of businesses and companies for comprehensive/wide applications.

(2) Concerning the second question, for example, deciding on an issue, such as quitting a job, will not be easy. Here, we can seek consultation from many people. Indeed, we can use belief merging theory to decide which concepts are mapped. In other words, we take advantage of the knowledge of many sources to decide on the mapping. Namely, we can conduct: (i) leveraging belief merging to seek out the correspondences between concepts of sources and (ii) applying machine learning to predict pertinent knowledge about those concepts. Moreover, with the significant development of computer hardware and parallel technology, taking advantage of the parallel perspective to solve the merging and matching aspect will also be worthy of attention.

(3) Regarding the third question, the combination of the symbolic fashion branch and the machine learning branch has been an exciting and challenging direction for many scientists in recent years. One simple reason is that the black box (of machine learning) sometimes makes interpreting the obtained result more challenging than in symbolic algorithms. Leveraging machine learning can perform the prediction of potential concepts for the merging procedure or the semantic prediction of instances. Another interesting problem is the classification of the concept groups. In particular, merging operators will be suitable and faithful in various situations, but they can lose in some unique and specific cases. Therefore, if ideas are automatically grouped, we should choose the operator that best fits those groupings. Furthermore, this thesis might also contribute toward solving the problem of few-shot learning for merging structured knowledge bases and ontologies in future orientations. Another interesting approach is to represent conflicts into representation vectors and use machine learning to predict inconsistencies in ontological sources when merging..

Otherwise, for aiming to propose the contributions (Chapter 4, Chapter 5, Chapter 6) in this thesis, several discussions to improve are as follows:

- For the first contribution, as discussed in conclusion, the cost of listing all the interpretations still needs to be improved. In a quick outline, we have conducted many supplementary process to improve this step, such as enumerating with sub-hierarchy (locally), ignoring redundant, incomplete interpretations, checking the far-from-the-source interpretation group in parallel, and splitting concepts into several subgroups to enumerate. Here, we can investigate some restrictions/properties of knowledge to limit the number of interpretations.

[48] Production ontologies can be merged via terminological and structural matching, domain knowledge, and interactive debugging.

- With regard to the second work, the general idea is to borrow another environment to merge and then translate the final result back to the original environment. Therefore, we can consider more expressive environments to investigate its merging capabilities. At first glance, it appears that we would be applying old ideas to a new environment. However, it is useful to acknowledge that each type of framework will have different strengths and weaknesses. Moreover, each kind of knowledge will need a different effective solution. From this idea, we can build a common framework to carry the knowledge into a suitable framework/environment/setting for the merging problem. On the other hand, from the introduction of the second contribution, we were inspired by Bouraoui et al.'s modeling [BCCAS20, BS18] of conceptual spaces into vector spaces. We can take account of geometric objects and contraints on metric spaces. So we can leverage machine learning [HL20] for this merging problem;
- Regarding the last contribution, noisy ontologies will accurately represent the outcome of merging if we consider each concept's semantics for the perturbation step. We may also develop an easy-to-use interface. Here, it will support updating the parameters and evaluating the operators for the case of merging each subtree following the user's choices rather than at random;

In general, the problem of merging knowledge bases is always a very critical challenge in real life. Therefore, the development of merging methods leveraging advanced techniques is one of the significant topics, i.e., knowledge graph [SPS22, SPST21], machine learning (graph neural network [LBS19], convolution neural network [FF19]), natural language processing (BERT[49], Word2vec[50], Sentence2Vec[51]), and others, nowadays to improve output quality. Indeed, combining symbolic and machine learning is evolving the development trend to find effective and explainable methods. Therefore, this thesis will not be able to isolate the general development direction of humanity from the subsequent efforts.

[49] https://github.com/google-research/bert
[50] https://pypi.org/project/gensim/
[51] https://github.com/stanleyfok/sentence2vec

Appendix A

First Appendix

A.1 Proofs of Chapter 4

Proposition 4.1. Let $\mathcal{P}$ be a profile and $\mathcal{O}_i \in \mathcal{P}$. If merging same ontologies for all $i, O_i = \mathcal{O}_1$, then $\mathcal{T}^M(\mathcal{P}) = \mathcal{O}_1$.

Proof A.1 *Given $\mathcal{P} = \{\mathcal{O}_i\}$, $\forall i, \mathcal{O}_i = \mathcal{O}_1$, and $\mathcal{O}_1 = \{\Phi_k\}$. Since we selected the $\mathcal{I}$(s) close to input ontologies and the input ontologies are same, the minimize distance will be selected, i.e., $dist_g(\mathcal{I}, \mathcal{P}) = 0$ (for $g = \{\sum, \max\}$) or $dist_g(\mathcal{I}, \mathcal{P}) = (0, \ldots, 0)$ (for $g = Gmax$) in such a way that $\forall \Phi_k \in \mathcal{O}_1$, $\mathcal{I} \models \Phi_k$. So, $\mathcal{I} \models \mathcal{O}_1$. By $\forall \mathcal{O}_1 \in \mathcal{P}$, hence $\mathcal{I} \in Mod(\oplus_g(\mathcal{P}))$.*

Here, we have $\mathcal{I}^M = IFSI(Mod(\oplus_g(\mathcal{P})))$ (cf. Definition 4.14) iff $\mathcal{I}^M = IFSI(Mod(\oplus_g(\{\mathcal{O}_i\})))$. Then, by $\mathcal{O}_i = \mathcal{O}_1$, so $\mathcal{I}^M \models \mathcal{O}_1$. Therefore, we have that $\mathcal{I}^M \models \Phi_k$ iff $\mathcal{I}^M \models \mathcal{O}_1$ iff $\mathcal{I}^M \models \mathcal{O}_i$ ($\forall i, \mathcal{O}_i = \mathcal{O}_1$) iff $\mathcal{I}^M \models \mathcal{P}$.

In addition, based on $\mathcal{I}^M$, we collect the merged axioms (c.f. Definition 4.15). Here, since input ontologies $\mathcal{O}_1$ are same, so $\mathcal{T}^M(\mathcal{P})$ contains $\mathcal{O}_1$'s all axioms. Thus, we have $\mathcal{T}^M(\mathcal{P}) = \mathcal{O}_1$.

Proposition 4.2. The model-based merging operator computed by the IFSI algorithm satisfies P1-P6.

Proof A.2 *Let's consider an assignment referring to Definition 4.12 [KP02] that maps each ontology set $\mathcal{P}$ to a total pre-order $\leq_{\mathcal{P}}$ and defined an ontology merging operator $\oplus_g$ by putting $Mod(\oplus_g(\mathcal{P})) = min(\mathcal{W}, \leq^g_{\mathcal{P}})$. We show that $\oplus_g$ satisfies P1-P6 (referring to proofs of postulates' belief merging [KP98]).*

- ***(P1)*** *if $\mathcal{I} \in Mod(\mathcal{P})$ and $\mathcal{J} \in Mod(\mathcal{P})$, then $dist_g(\mathcal{I}, \mathcal{P}) = dist_g(\mathcal{J}, \mathcal{P}) = 0$ (for $g \in \{\max, \Sigma\}$). Regarding $g = GMax$, $\mathscr{L}^{\mathcal{I}}_{\mathcal{P}} = (0, \ldots, 0)$ and $\mathscr{L}^{\mathcal{J}}_{\mathcal{P}} = (0, \ldots, 0)$. Hence $\mathcal{I} \simeq^g_{\mathcal{P}} \mathcal{J}$.*
- ***(P2)*** *If $\mathcal{I} \in Mod(\mathcal{P})$ and $\mathcal{J} \notin Mod(\mathcal{P})$, then $dist_g(\mathcal{I}, \mathcal{P}) = 0$ and $dist_g(\mathcal{J}, \mathcal{P}) > 0$ (for $g \in \{\max, \Sigma\}$). Regarding $g = GMax$, $\mathscr{L}^{\mathcal{I}}_{\mathcal{P}} = (0, \ldots, 0)$ and $\mathscr{L}^{\mathcal{J}}_{\mathcal{P}} \neq (0, \ldots, 0)$. Hence $\mathcal{I} <^g_{\mathcal{P}} \mathcal{J}$.*
- ***(P3)*** *If $\mathcal{P}_1 \equiv \mathcal{P}_2$, then obvious that $\leq^g_{\mathcal{P}_1} = \leq^g_{\mathcal{P}_2}$.*
- ***(P4)*** *Suppose $\mathcal{O}_1 \cap \mathcal{O}_2 \models \bot$, so $dist(\mathcal{I}, \mathcal{O}_1) > 0$ or $dist(\mathcal{I}, \mathcal{O}_2) > 0$. Hence, if $\exists \mathcal{I} \in Mod(\mathcal{O}_1)$, $\forall \mathcal{J} \in Mod(\mathcal{O}_2)$, and $dist(\mathcal{I}, \mathcal{O}_1) \leq dist(\mathcal{J}, \mathcal{O}_2)$, then $dist(\mathcal{I}, \mathcal{O}_1) = k > 0$ and $k \leq dist(\mathcal{J}, \mathcal{O}_2)$. It also means that $k = min\{dist_g(\mathcal{X}, \mathcal{O}_1 \cup \mathcal{O}_2) : \mathcal{X} \in \mathcal{W}\}$ thus $\mathcal{I} \in Mod(\oplus_g(\mathcal{O}_1 \cup \mathcal{O}_2))$ and $\mathcal{J} \in Mod(\oplus_g(\mathcal{O}_1 \cup \mathcal{O}_2))$, hence $\oplus_g(\mathcal{O}_1 \cup \mathcal{O}_2) \not\models \mathcal{O}_1$*

(P5) *If $\mathcal{I} \in min(\mathcal{W}, \leq^g_{\mathcal{P}_1})$ and $\mathcal{I} \in min(\mathcal{W}, \leq^g_{\mathcal{P}_2})$, then $\forall \mathcal{J}$, $dist_g(\mathcal{I}, \mathcal{P}_1) \leq dist_g(\mathcal{J}, \mathcal{P}_1)$ and $dist_g(\mathcal{I}, \mathcal{P}_2) \leq dist_g(\mathcal{J}, \mathcal{P}_2)$. So, $\forall \mathcal{J}$ $dist_g(\mathcal{I}, \mathcal{P}_1) + dist_g(\mathcal{I}, \mathcal{P}_2) \leq dist_g(\mathcal{J}, \mathcal{P}_1) + dist_g(\mathcal{J}, \mathcal{P}_2)$. It is easy to see that for any $\mathcal{Z}$, $\mathcal{O}_1$, $\mathcal{O}_2$, we have $dist_g(\mathcal{Z}, \mathcal{O}_1 \cup \mathcal{O}_2) = dist_g(\mathcal{Z}, \mathcal{O}_1) + dist_g(\mathcal{Z}, \cup\mathcal{O}_2)$. Then, $dist_g(\mathcal{I}, \mathcal{P}_1 \cup \mathcal{P}_2) \leq dist_g(\mathcal{J}, \mathcal{P}_1 \cup \mathcal{P}_2)$. Hence, $\mathcal{I} \in min(\mathcal{W}, \leq^g_{\mathcal{P}_1 \cup \mathcal{P}_2})$. (for $g \in \{\max, \Sigma\}$) Regarding $g = GMax$, $\forall \mathcal{J} \in \mathcal{W}$, we have $\mathscr{L}^{\mathcal{I}}_{\mathcal{P}_1} \leq_{GMax} \mathscr{L}^{\mathcal{J}}_{\mathcal{P}_1}$ and $\mathscr{L}^{\mathcal{I}}_{\mathcal{P}_2} \leq_{GMax} \mathscr{L}^{\mathcal{J}}_{\mathcal{P}_2}$. By Lemma 3.1 [KP98], we have $\forall \mathcal{J}, \mathscr{L}^{\mathcal{I}}_{\mathcal{P}_1 \cup \mathcal{P}_2} \leq_{GMax} \mathscr{L}^{\mathcal{J}}_{\mathcal{P}_1 \cup \mathcal{P}_2}$. Hence, $\mathcal{I} \in min(\mathcal{W}, \leq^{GMax}_{\mathcal{P}_1 \cup \mathcal{P}_2})$. So $\mathcal{I} \models \oplus_{GMax}(\mathcal{P}_1 \cup \mathcal{P}_2)$.*

(P6) *We assume that $\oplus_g(\mathcal{P}_1) \cap \oplus_g(\mathcal{P}_2)$ is satisfiable. We show that $\oplus_g(\mathcal{P}_1 \cup \mathcal{P}_2) \models \oplus_g(\mathcal{P}_1) \cap \oplus_g(\mathcal{P}_2)$. Take $\mathcal{I} \models \oplus_g(\mathcal{P}_1 \cap \mathcal{P}_2)$, so $\forall \mathcal{J}$ $\mathcal{I} \leq^g_{\mathcal{P}_1 \cap \mathcal{P}_2} \mathcal{J}$. Suppose a contradiction that $\mathcal{I} \not\models \oplus_g(\mathcal{P}_1) \cap \oplus_g(\mathcal{P}_2)$. So $\mathcal{I} \not\models \oplus_g(\mathcal{P}_1)$ or $\mathcal{I} \not\models \oplus_g(\mathcal{P}_2)$. Suppose that $\mathcal{I} \not\models \oplus_g(\mathcal{P}_1)$ (symmetric). As $\oplus_g(\mathcal{P}_1) \cap \oplus_g(\mathcal{P}_2)$ is satisfiable, $\exists \mathcal{J} \models \oplus_g(\mathcal{P}_1) \cap \oplus_g(\mathcal{P}_2)$. So $\mathcal{J} \leq^g_{\mathcal{P}_1} \mathcal{I}$ and $\mathcal{J} \leq^g_{\mathcal{P}_2} \mathcal{I}$, hence $\mathcal{J} \leq^g_{\mathcal{P}_1 \cup \mathcal{P}_2} \mathcal{I}$ and then $\mathcal{I} \not\models \oplus_g(\mathcal{P}_1 \cup \mathcal{P}_2)$. Contradiction (for $g \in \{\max, \Sigma\}$) (1). Moreover, for $g = GMax$, if $min(\mathcal{W}, \leq^{GMax}_{\mathcal{P}_1}) \cap min(\mathcal{W}, \leq^{GMax}_{\mathcal{P}_2}) \neq \emptyset$ and $\mathcal{I} \notin min(\mathcal{W}, \leq^{GMax}_{\mathcal{P}_1})$. Now, let $\mathcal{J} \in min(\mathcal{W}, \leq^{GMax}_{\mathcal{P}_1}) \cap min(\mathcal{W}, \leq^{GMax}_{\mathcal{P}_2})$, then we have $\mathscr{L}^{\mathcal{J}}_{\mathcal{P}_1} <_{GMax} \mathscr{L}^{\mathcal{I}}_{\mathcal{P}_1}$ and $\mathscr{L}^{\mathcal{J}}_{\mathcal{P}_2} \leq_{GMax} \mathscr{L}^{\mathcal{I}}_{\mathcal{P}_2}$. By Lemma 3.2 [KP98], we have $\mathscr{L}^{\mathcal{J}}_{\mathcal{P}_1 \cup \mathcal{P}_2} <_{GMax} \mathscr{L}^{\mathcal{I}}_{\mathcal{P}_1 \cup \mathcal{P}_2}$. Then, $\mathcal{I} \notin min(\mathcal{W}, \leq^{GMax}_{\mathcal{P}_1 \cup \mathcal{P}_2})$ (2). Hence, from (1)+(2), if $\oplus_g(\mathcal{P}_1) \cap \oplus_g(\mathcal{P}_2)$ is satisfiable, then $\oplus_g(\mathcal{P}_1 \cup \mathcal{P}_2) \models \oplus_g(\mathcal{P}_1) \cap \oplus_g(\mathcal{P}_2)$.*

A.2 Proofs of Chapter 5

Theorem 5.1. Let $\mathcal{O}_{\mathcal{EL}_\bot}$ be an ontology, and let $\mathcal{I}$ be a fulfilling interpretation of $\mathcal{O}_{\mathcal{EL}_\bot}$ such that $\mathcal{I} \models \mathcal{O}_{\mathcal{EL}_\bot}$. Then $\mathcal{S}_\mathcal{I} \models \tau_\triangleright(\mathcal{O}_{\mathcal{EL}_\bot})$.

Proof A.3 *Let $\mathcal{O}_{\mathcal{EL}_\bot} = \langle \mathcal{A}, \mathcal{T} \rangle$ be an ontology and $\mathcal{I}$ be a fulfilling interpretation of $\mathcal{O}_{\mathcal{EL}_\bot}$ such that $\mathcal{I} \models \mathcal{O}_{\mathcal{EL}_\bot}$. Let us denote by $\tau_\triangleright(\mathcal{O}_{\mathcal{EL}_\bot}) = \langle V, \Psi \rangle$ the forward translation of $\mathcal{O}_{\mathcal{EL}_\bot}$. We need to show that $\mathcal{S}_\mathcal{I} \models \tau_\triangleright(\mathcal{O}_{\mathcal{EL}_\bot})$. Since $\mathcal{I} \models \mathcal{O}_{\mathcal{EL}_\bot}$, we know that for each axiom $\Phi \in \mathcal{T}$, we have that $\mathcal{I} \models \Phi$. On the other hand, to show that $\mathcal{S}_\mathcal{I} \models \tau_\triangleright(\mathcal{O}_{\mathcal{EL}_\bot})$, it is enough to show that $\mathcal{S}_\mathcal{I} \models \varphi$ for each constraint $\varphi \in \Psi$. Hence, we need to show that $\mathcal{S}_\mathcal{I} \models \tau_\triangleright(\Phi)$ for each axiom $\Phi \in \mathcal{T}$. Let $\Phi \in \mathcal{T}$. We fall into one of the following cases:*

- *Φ is of the form $C \sqsubseteq D$. Since $\mathcal{I} \models \Phi$, we have that $C^\mathcal{I} \subseteq D^\mathcal{I}$ (cf. Table 2.5), or stated equivalently, that $v_C^{\mathcal{S}_\mathcal{I}} \subseteq v_D^{\mathcal{S}_\mathcal{I}}$ by definition of $\mathcal{S}_\mathcal{I}$ (cf. Definition 5.6). So $v_C^{\mathcal{S}_\mathcal{I}} \subset v_D^{\mathcal{S}_\mathcal{I}}$ or $v_C^{\mathcal{S}_\mathcal{I}} = v_D^{\mathcal{S}_\mathcal{I}}$. Thus from Table 5.3, we get that $\mathcal{S}_\mathcal{I} \models v_C\{PP\}v_D$ or $\mathcal{S}_\mathcal{I} \models v_C\{EQ\}v_D$, which can equivalently be written as $\mathcal{S}_\mathcal{I} \models v_C\{PP, EQ\}v_D$. Yet we know that $\tau_\triangleright(C \sqsubseteq D) = v_C\{PP, EQ\}v_D$ (cf. Definition 5.5), so we have that $\mathcal{S}_\mathcal{I} \models \tau_\triangleright(C \sqsubseteq D)$. Hence, $\mathcal{S}_\mathcal{I} \models \tau_\triangleright(\Phi)$.*
- *Φ is of the form $C \sqcap D \sqsubseteq \bot$. Since $\mathcal{I} \models \Phi$, we have that $C^\mathcal{I} \cap D^\mathcal{I} \subseteq \emptyset$ (cf. Table 2.5), or stated equivalently, that $v_C^{\mathcal{S}_\mathcal{I}} \cap v_D^{\mathcal{S}_\mathcal{I}} \subseteq \emptyset$ by definition of $\mathcal{S}_\mathcal{I}$ (cf. Definition 5.6), i.e., $v_C^{\mathcal{S}_\mathcal{I}} \cap v_D^{\mathcal{S}_\mathcal{I}} = \emptyset$. Thus from Table 5.3, we get that $\mathcal{S}_\mathcal{I} \models v_C\{DR\}v_D$. Yet we know that $\tau_\triangleright(C \sqcap D \sqsubseteq \bot) = v_C\{DR\}v_D$ (cf. Definition 5.5), so we have that $\mathcal{S}_\mathcal{I} \models \tau_\triangleright(C \sqcap D \sqsubseteq \bot)$. Hence, $\mathcal{S}_\mathcal{I} \models \tau_\triangleright(\Phi)$.*

This concludes the proof.

Theorem 5.2. Let $\mathcal{O}_{\mathcal{EL}_\bot}$ be an ontology and let $\mathcal{S}$ be a solution of $\tau_\triangleright(\mathcal{O}_{\mathcal{EL}_\bot})$. Then there is an inflation $\mathcal{I}_\mathcal{S}$ of $\mathcal{S}$ s.t. $\mathcal{I}_\mathcal{S} \models \Phi$ for each axiom Φ of $\mathcal{O}_{\mathcal{EL}_\bot}$.

Proof A.4 *Let $\mathcal{O}_{\mathcal{EL}_\bot} = \langle \mathcal{A}, \mathcal{T} \rangle$ be an ontology. Let us denote by $\tau_\triangleright(\mathcal{O}_{\mathcal{EL}_\bot}) = \langle V, \Psi \rangle$ the forward translation of $\mathcal{O}_{\mathcal{EL}_\bot}$. Let $\mathcal{S} = (\mathcal{D}^\mathcal{S}, \cdot^\mathcal{S})$ be a solution of $\tau_\triangleright(\mathcal{O}_{\mathcal{EL}_\bot})$, e be any element of $\mathcal{D}^\mathcal{S}$, and*

$\mathcal{I}_\mathcal{S} \stackrel{\text{def}}{=} (\Delta^{\mathcal{I}_\mathcal{S}}, \cdot^{\mathcal{I}_\mathcal{S}})$ be the inflation of $\mathcal{S}$ defined for each $a \in N_I$ as $\cdot^\mathcal{S}(a) = e$ and for each $a, b \in N_I$. Since $\mathcal{I}_\mathcal{S}$ is an inflation of $\mathcal{S}$, we know that $\Delta^{\mathcal{I}_\mathcal{S}} = \mathcal{D}^\mathcal{S}$ and for every $A \in N_C$, $A^{\mathcal{I}_\mathcal{S}} = (v_A)^\mathcal{S}$ (cf. Definition 5.7).

Let Φ be any axiom of $\mathcal{T}$ and let us show that $\mathcal{I}_\mathcal{S} \models \Phi$. We fall into one of the following cases:

- *Φ is of the form $C \sqsubseteq D$. Since $\mathcal{S}$ is a solution of $\tau_\triangleright(\mathcal{O}_{\mathcal{EL}_\bot})$, we have that $\mathcal{S} \models \tau_\triangleright(C \sqsubseteq D)$. Yet we know that $\tau_\triangleright(C \sqsubseteq D) = v_C\{PP, EQ\}v_D$ (cf. Definition 5.5), so we have that $\mathcal{S} \models v_C\{PP, EQ\}v_D$, which can equivalently be written as $\mathcal{S} \models v_C\{PP\}v_D$ or $\mathcal{S} \models v_C\{EQ\}v_D$. Thus from Table 5.3, we get that $v_C^\mathcal{S} \subset v_D^\mathcal{S}$ or $v_C^\mathcal{S} = v_D^\mathcal{S}$, i.e., $v_C^\mathcal{S} \subseteq v_D^\mathcal{S}$. So by definition of $\mathcal{I}_\mathcal{S}$, we get that $C^{\mathcal{I}_\mathcal{S}} \subseteq D^{\mathcal{I}_\mathcal{S}}$. From Table 2.5, this means that $\mathcal{I}_\mathcal{S} \models \Phi$.*
- *Φ is of the form $C \sqcap D \sqsubseteq \bot$. Since $\mathcal{S}$ is a solution of $\tau_\triangleright(\mathcal{O}_{\mathcal{EL}_\bot})$, we have that $\mathcal{S} \models \tau_\triangleright(C \sqcap D \sqsubseteq \bot)$. Yet we know that $\tau_\triangleright(C \sqcap D \sqsubseteq \bot) = v_C\{DR\}v_D$ (cf. Definition 5.5), so we have that $\mathcal{S}_\mathcal{I} \models v_C\{DR\}v_D$. Thus from Table 5.3, we get that $v_C^{\mathcal{S}_\mathcal{I}} \cap v_D^{\mathcal{S}_\mathcal{I}} \subseteq \emptyset$, i.e., $v_C^{\mathcal{S}_\mathcal{I}} \cap v_D^{\mathcal{S}_\mathcal{I}} = \emptyset$. So by definition of $\mathcal{I}_\mathcal{S}$, we get that $C^\mathcal{I} \cap D^\mathcal{I} \subseteq \emptyset$. From Table 2.5, this means that $\mathcal{I}_\mathcal{S} \models \Phi$.*

This concludes the proof.

Theorem 5.3. Let $\mathcal{N}$ be a scenario and $\mathcal{S}$ be solution of $\mathcal{N}$. Then there is an inflation $\mathcal{I}_\mathcal{S}$ of $\mathcal{S}$ s.t. $\mathcal{I}_\mathcal{S}$ is a model of $\tau_\triangleleft(\mathcal{N})$.

Proof A.5 *Let $\mathcal{N} = \langle V, \Psi\rangle$ be a scenario. Let us denote by $\tau_\triangleleft(\mathcal{N}) = \langle \mathcal{A}, \mathcal{T}\rangle$ the backward translation of $\mathcal{N}$. Let $\mathcal{S} = (\mathcal{D}^\mathcal{S}, \cdot^\mathcal{S})$ be a solution of $\mathcal{N}$. Note that the set of concept names N_C of $\tau_\triangleleft(\mathcal{N})$ is formed of two parts: the concept names C which are directly associated with a variable v_C from V, and the new concept names introduced in the backward translation (cf. Definition 5.8). For instance, some new concept names A', C' and D' appear in the translation of a constraint $v_C\{PO\}v_D$. Let us denote by N_C^* the subset of N_C formed by these new concept names. In addition, note that the set of individual names N_I of $\tau_\triangleleft(\mathcal{N})$ is formed only of individual names artificially introduced in the translation; for instance, the individual names a, c and d appear in the translation of a constraint $v_C\{PO\}v_D$, and all these individual names form the set N_I. Now, let $\mathcal{I}_\mathcal{S} = (\Delta^{\mathcal{I}_\mathcal{S}}, \cdot^{\mathcal{I}_\mathcal{S}})$ be an interpretation defined as $\Delta^{\mathcal{I}_\mathcal{S}} = \mathcal{D}^\mathcal{S}$, and for every $A \in N_C \setminus N_C^*$ as $A^{\mathcal{I}_\mathcal{S}} = (v_A)^\mathcal{S}$. Note that at this point, $\mathcal{I}_\mathcal{S}$ is only partially defined, so one needs to complete its definition. Indeed, one needs to define $(A')^{\mathcal{I}_\mathcal{S}}$ for each $A' \in N_C^*$ and to define $\cdot^{\mathcal{I}_\mathcal{S}}(a)$ and for each $a \in N_I$. For this purpose, let us consider only the forms of constraints that actually introduce new concept names and new individual names, i.e., the elements of N_C^* and N_I. These constraints are of the form $v_C\ \{PO\}\ v_D$ and of the form $v_C\ \{PP\}\ v_D$ (the constraints of the form $v_C\ \{PPi\}\ v_D$ can be dealt with similarly as to the case of the constraints of the form $v_C\ \{PP\}\ v_D$, since $v_C\ \{PPi\}\ v_D$ is equivalent to $v_D\ \{PP\}\ v_C$.) So let φ be a constraint of the form $v_C\ \{PO\}\ v_D$ or the form $v_C\ \{PP\}\ v_D$. We consider the two cases separately:*

- *φ is a constraint of the form $v_C\ \{PO\}\ v_D$. Note that since the concept names C and D are members of $N_C \setminus N_C^*$, $C^{\mathcal{I}_\mathcal{S}}$ and $D^{\mathcal{I}_\mathcal{S}}$ are already defined as $C^{\mathcal{I}_\mathcal{S}} = v_C^\mathcal{S}$ and $D^{\mathcal{I}_\mathcal{S}} = v_D^\mathcal{S}$, where $v_C^\mathcal{S}$ and $v_D^\mathcal{S}$ are non-empty subsets of $\Delta^{\mathcal{I}_\mathcal{S}}$. Additionally, since $\mathcal{S} \models v_C\ \{PO\}\ v_D$, according to Table 5.3 we have that $v_C^\mathcal{S} \cap v_D^\mathcal{S} \neq \emptyset$, $v_C^\mathcal{S} \not\subseteq v_D^\mathcal{S}$ and $v_D^\mathcal{S} \not\subseteq v_C^\mathcal{S}$. Now, let A', C' and D' be the concept names newly introduced in the translation of $v_C\ \{PO\}\ v_D$ according to Definition 5.8. We have that $A', C', D' \in N_C^*$. Let us define $(A')^{\mathcal{I}_\mathcal{S}}$, $(C')^{\mathcal{I}_\mathcal{S}}$, and $(D')^{\mathcal{I}_\mathcal{S}}$ as follows:*
  - *let $(A')^{\mathcal{I}_\mathcal{S}} = v_{A'}^\mathcal{S}$, where $v_{A'}^\mathcal{S}$ is any non-empty subset of $\Delta^{\mathcal{I}_\mathcal{S}}$ such that $v_{A'}^\mathcal{S} \subseteq v_C^\mathcal{S} \cap v_D^\mathcal{S}$ (this assignment is realizable since $v_C^\mathcal{S} \cap v_D^\mathcal{S} \neq \emptyset$);*
  - *let $(C')^{\mathcal{I}_\mathcal{S}} = (v_{C'})^\mathcal{S}$, where $v_{C'}^\mathcal{S}$ is any non-empty subset of $\Delta^{\mathcal{I}_\mathcal{S}}$ such that $v_{C'}^\mathcal{S} \subseteq v_C^\mathcal{S}$ and $v_{C'}^\mathcal{S} \cap v_D^\mathcal{S} = \emptyset$ (this assignment is realizable since $v_C^\mathcal{S} \not\subseteq v_D^\mathcal{S}$);*

- let $(D')^{\mathcal{I}_\mathcal{S}} = v^\mathcal{S}_{D'}$, where $v^\mathcal{S}_{D'}$ is any non-empty subset of $\Delta^{\mathcal{I}_\mathcal{S}}$ such that $v^\mathcal{S}_{D'} \subseteq v^\mathcal{S}_D$ and $v^\mathcal{S}_{D'} \cap v^\mathcal{S}_C = \emptyset$ (this assignment is realizable since $v^\mathcal{S}_D \not\subseteq v^\mathcal{S}_C$).

Similarly, let a, c, d be the individual names newly introduced in that translation. We have that a, c, $d \in N_I$. Let us define $\cdot^{\mathcal{I}_\mathcal{S}}(a)$ (respectively, $\cdot^{\mathcal{I}_\mathcal{S}}(c)$, $\cdot^{\mathcal{I}_\mathcal{S}}(d)$) as any element of $v^\mathcal{S}_{A'}$ (respectively, of $v^\mathcal{S}_{C'}$, of $v^\mathcal{S}_{D'}$).

- *φ is a constraint of the form $v_C \ \{PP\} \ v_D$. Note that since the concept names C and D are members of $N_C \setminus N^*_C$, $C^{\mathcal{I}_\mathcal{S}}$ and $D^{\mathcal{I}_\mathcal{S}}$ are already defined as $C^{\mathcal{I}_\mathcal{S}} = v^\mathcal{S}_C$ and $C^{\mathcal{I}_\mathcal{S}} = v^\mathcal{S}_D$, where $v^\mathcal{S}_C$ and $v^\mathcal{S}_D$ are non-empty subsets of $\Delta^{\mathcal{I}_\mathcal{S}}$. Additionally, since $\mathcal{S} \models v_C \ \{PP\} \ v_D$, according to Table 5.3 we have that $v^\mathcal{S}_C \subset v^\mathcal{S}_D$. Now, let D' be the concept name newly introduced in the translation of $v_C \ \{PP\} \ v_D$ according to Definition 5.8. We have that $D' \in N^*_C$ and c, $d \in N_I$. Let us define $(D')^{\mathcal{I}_\mathcal{S}}$ as $(D')^{\mathcal{I}_\mathcal{S}} = v^\mathcal{S}_{D'}$, where $v^\mathcal{S}_{D'}$ is any non-empty subset of $\Delta^{\mathcal{I}_\mathcal{S}}$ such that $v^\mathcal{S}_{D'} \subseteq v^\mathcal{S}_D$ and $v^\mathcal{S}_{D'} \cap v^\mathcal{S}_C = \emptyset$ (this assignment is realizable since $v^\mathcal{S}_C \subset v^\mathcal{S}_D$). Similarly, let c, d be the individual names newly introduced in that translation. We have that c, $d \in N_I$. Let us define $\cdot^{\mathcal{I}_\mathcal{S}}(c)$ (respectively, $\cdot^{\mathcal{I}_\mathcal{S}}(d)$) as any element of $v^\mathcal{S}_C$ (respectively, of $v^\mathcal{S}_{D'}$).*

At this point, $\mathcal{I}_\mathcal{S}$ is completely defined: we have defined $A^{\mathcal{I}_\mathcal{S}}$ for each $A \in N_C$, and we have defined $\cdot^\mathcal{S}(a)$ for each $a \in N_I$. First, it can easily be verified that $\mathcal{I}_\mathcal{S}$ is an inflation of $\mathcal{S}$, according to Definition 5.7. Indeed, we have that $\Delta^{\mathcal{I}_\mathcal{S}} = \mathcal{D}^\mathcal{S}$, and for every $A \in N_C$ that $A^{\mathcal{I}_\mathcal{S}} = (v_A)^\mathcal{S}$. We now intend to show that $\mathcal{I}_\mathcal{S}$ is a model of $\tau_\lhd(\mathcal{N})$. For this purpose, let us consider each contraint $v_C \ \varphi \ v_D \in \Psi$, and show that $\mathcal{I}_\mathcal{S}$ is such that $\mathcal{I}_\mathcal{S} \models \tau_\lhd(v_C \ \varphi \ v_D)$. We fall into one of the following cases:

- *φ is a constraint of the form $v_C \ \{EQ\} \ v_D$. Since $\mathcal{S}$ is a solution of $\mathcal{N}$, we have that $\mathcal{S} \models v_C \ \{EQ\} \ v_D$. Thus $v^\mathcal{S}_C = v^\mathcal{S}_D$ (cf. Table 5.3). So by definition of $\mathcal{I}_\mathcal{S}$, we get that $C^{\mathcal{I}_\mathcal{S}} = D^{\mathcal{I}_\mathcal{S}}$. Thus, from Table 2.5, we get that $\mathcal{I}_\mathcal{S} \models C \equiv D$. Yet, from Definition 5.8, we know that $\tau_\lhd(v_C \ \{EQ\} \ v_D) = \langle\{C \equiv D\}, \emptyset\rangle$. Hence, $\mathcal{I}_\mathcal{S} \models \tau_\lhd(v_C \ \{EQ\} \ v_D)$.*

- *φ is a constraint of the form $v_C \ \{DR\} \ v_D$. Since $\mathcal{S}$ is a solution of $\mathcal{N}$, we have that $\mathcal{S} \models v_C \ \{DR\} \ v_D$. Thus $v^\mathcal{S}_C \cap v^\mathcal{S}_D = \emptyset$ (cf. Table 5.3). So by Definition of $\mathcal{I}_\mathcal{S}$, we get that $C^{\mathcal{I}_\mathcal{S}} \cap D^{\mathcal{I}_\mathcal{S}} = \emptyset$, i.e, $C^{\mathcal{I}_\mathcal{S}} \cap D^{\mathcal{I}_\mathcal{S}} \subseteq \emptyset$. Thus, from Table 2.5, we get that $\mathcal{I}_\mathcal{S} \models C \sqcap D \sqsubseteq \bot$. Yet, from Definition 5.8, we know that $\tau_\lhd(v_C \ \{DR\} \ v_D) = \langle\{C \sqcap D \sqsubseteq \bot\}, \emptyset\rangle$. Hence, $\mathcal{I}_\mathcal{S} \models \tau_\lhd(v_C \ \{DR\} \ v_D)$.*

- *φ is a constraint of the form $v_C \ \{PO\} \ v_D$. Since $\mathcal{S}$ is a solution of $\mathcal{N}$, we have that $\mathcal{S} \models v_C \ \{PO\} \ v_D$. And according to the way we previously defined $(A')^{\mathcal{I}_\mathcal{S}}$, $(C')^{\mathcal{I}_\mathcal{S}}$, and $(D')^{\mathcal{I}_\mathcal{S}}$, we have that $v^\mathcal{S}_{A'} \subseteq v^\mathcal{S}_C \cap v^\mathcal{S}_D$, $v^\mathcal{S}_{C'} \subseteq v^\mathcal{S}_C$, $v^\mathcal{S}_{C'} \cap v^\mathcal{S}_D = \emptyset$, $v^\mathcal{S}_{D'} \subseteq v^\mathcal{S}_D$, and $v^\mathcal{S}_{D'} \cap v^\mathcal{S}_C = \emptyset$. Hence, from Table 2.5, we get that:*

  - $\mathcal{I}_\mathcal{S} \models A' \sqsubseteq C \sqcap D$ (since $v^\mathcal{S}_{A'} \subseteq v^\mathcal{S}_C \cap v^\mathcal{S}_D$);
  - $\mathcal{I}_\mathcal{S} \models C' \sqsubseteq C$ (since $v^\mathcal{S}_{C'} \subseteq v^\mathcal{S}_C$);
  - $\mathcal{I}_\mathcal{S} \models C' \sqcap D \sqsubseteq \bot$ (since $v^\mathcal{S}_{C'} \cap v^\mathcal{S}_D = \emptyset$);
  - $\mathcal{I}_\mathcal{S} \models D' \sqsubseteq D$ (since $v^\mathcal{S}_{D'} \subseteq v^\mathcal{S}_D$);
  - $\mathcal{I}_\mathcal{S} \models D' \sqcap C \sqsubseteq \bot$ (since $v^\mathcal{S}_{D'} \cap v^\mathcal{S}_C = \emptyset$).

  In addition, according to the way we previously defined $\cdot^{\mathcal{I}_\mathcal{S}}(a)$, $\cdot^{\mathcal{I}_\mathcal{S}}(c)$, $\cdot^{\mathcal{I}_\mathcal{S}}(d)$, we know that $\cdot^{\mathcal{I}_\mathcal{S}}(a) \in v^\mathcal{S}_{A'}$, $\cdot^{\mathcal{I}_\mathcal{S}}(c) \in v^\mathcal{S}_{C'}$, and $\cdot^{\mathcal{I}_\mathcal{S}}(d) \in v^\mathcal{S}_{D'}$. Hence, we get that:

  - $\mathcal{I}_\mathcal{S} \models A'(a)$ (since $\cdot^{\mathcal{I}_\mathcal{S}}(a) \in v^\mathcal{S}_{A'}$);
  - $\mathcal{I}_\mathcal{S} \models C(c)$ (since $\cdot^{\mathcal{I}_\mathcal{S}}(c) \in v^\mathcal{S}_{C'}$ and $v^\mathcal{S}_{C'} \subseteq v^\mathcal{S}_C$);

- $\mathcal{I}_\mathcal{S} \models C(a)$ *(since* $\cdot^{\mathcal{I}_\mathcal{S}}(a) \in v^\mathcal{S}_{A'}$ *and* $v^\mathcal{S}_{A'} \subseteq v^\mathcal{S}_C$*);*
- $\mathcal{I}_\mathcal{S} \models D(d)$ *(since* $\cdot^{\mathcal{I}_\mathcal{S}}(d) \in v^\mathcal{S}_{D'}$ *and* $v^\mathcal{S}_{D'} \subseteq v^\mathcal{S}_D$*);*
- $\mathcal{I}_\mathcal{S} \models D(a)$ *(since* $\cdot^{\mathcal{I}_\mathcal{S}}(a) \in v^\mathcal{S}_{A'}$ *and* $v^\mathcal{S}_{A'} \subseteq v^\mathcal{S}_D$*);*
- $\mathcal{I}_\mathcal{S} \models C'(c)$ *(since* $\cdot^{\mathcal{I}_\mathcal{S}}(c) \in v^\mathcal{S}_{C'}$*);*
- $\mathcal{I}_\mathcal{S} \models D'(d)$ *(since* $\cdot^{\mathcal{I}_\mathcal{S}}(d) \in v^\mathcal{S}_{D'}$*).*

Overall, we got that $\mathcal{I}_\mathcal{S} \models \langle\{A' \sqsubseteq C \sqcap D, C' \sqsubseteq C, C' \sqcap D \sqsubseteq \bot, D' \sqsubseteq D, D' \sqcap C \sqsubseteq \bot\}, \{A'(a), C(c), C(a), D(d), D(a), C'(c), D'(d)\}\rangle$*. Hence,* $\mathcal{I}_\mathcal{S} \models \tau_\triangleleft(v_C\ \{PO\}\ v_D)$*.*

- φ *is a constraint of the form* $v_C\ \{PP, EQ\}\ v_D$*. Since* $\mathcal{S}$ *is a solution of* $\mathcal{N}$*, we have that* $\mathcal{S} \models v_C\ \{PP, EQ\}\ v_D$*. Thus* $v^\mathcal{S}_C \subseteq v^\mathcal{S}_D$ *(cf. Table 5.3). So by definition of* $\mathcal{I}_\mathcal{S}$*, we get that* $C^{\mathcal{I}_\mathcal{S}} \subseteq D^{\mathcal{I}_\mathcal{S}}$*. Thus, from Table 2.5, we get that* $\mathcal{I}_\mathcal{S} \models C \sqsubseteq D$*. Yet, from Definition 5.8, we know that* $\tau_\triangleleft(v_C\ \{PP, EQ\}\ v_D) = \langle\{C \sqsubseteq D\}, \emptyset\rangle$*. Hence,* $\mathcal{I}_\mathcal{S} \models \tau_\triangleleft(v_C\ \{PP, EQ\}\ v_D)$*.*

- φ *is a constraint of the form* $v_C\ \{PPi, EQ\}\ v_D$*. The proof that* $\mathcal{I}_\mathcal{S} \models \tau_\triangleleft(v_C\ \{PPi, EQ\}\ v_D)$ *can be reduced equivalently to the proof that* $\mathcal{I}_\mathcal{S} \models \tau_\triangleleft(v_D\ \{PP, EQ\}\ v_C)$ *similarly to the previous case, since* $v_C\ \{PPi, EQ\}\ v_D$ *is equivalent to* $v_D\ \{PP, EQ\}\ v_C$*.*

- φ *is a constraint of the form* $v_C\ \{PP\}\ v_D$*. Since* $\mathcal{S}$ *is a solution of* $\mathcal{N}$*, we have that* $\mathcal{S} \models v_C\ \{PP\}\ v_D$*. Thus* $v^\mathcal{S}_C \subset v^\mathcal{S}_D$ *(cf. Table 5.3). And according to the way we previously defined* $(D')^{\mathcal{I}_\mathcal{S}}$*, we have that* $v^\mathcal{S}_{D'} \subseteq v^\mathcal{S}_D$ *and* $v^\mathcal{S}_{D'} \cap v^\mathcal{S}_C = \emptyset$*. Hence, from Table 2.5, we get that:*

  - $\mathcal{I}_\mathcal{S} \models C \sqsubseteq D$ *(since* $v^\mathcal{S}_C \subset v^\mathcal{S}_D$*);*
  - $\mathcal{I}_\mathcal{S} \models D' \sqsubseteq D$ *(since* $v^\mathcal{S}_{D'} \subseteq v^\mathcal{S}_D$*);*
  - $\mathcal{I}_\mathcal{S} \models C \sqcap D' \sqsubseteq \bot$ *(since* $v^\mathcal{S}_{D'} \cap v^\mathcal{S}_C = \emptyset$*).*

  In addition, according to the way we previously defined $\cdot^{\mathcal{I}_\mathcal{S}}(c)$ *and* $\cdot^{\mathcal{I}_\mathcal{S}}(d)$*, we know that* $\cdot^{\mathcal{I}_\mathcal{S}}(c) \in v^\mathcal{S}_C$*, and* $\cdot^{\mathcal{I}_\mathcal{S}}(d) \in v^\mathcal{S}_{D'}$*. Hence, we get that:*

  - $\mathcal{I}_\mathcal{S} \models D'(d)$ *(since* $\cdot^{\mathcal{I}_\mathcal{S}}(d) \in v^\mathcal{S}_{D'}$*);*
  - $\mathcal{I}_\mathcal{S} \models C(c)$ *(since* $\cdot^{\mathcal{I}_\mathcal{S}}(c) \in v^\mathcal{S}_C$*);*
  - $\mathcal{I}_\mathcal{S} \models D(d)$ *(since* $\cdot^{\mathcal{I}_\mathcal{S}}(d) \in v^\mathcal{S}_{D'}$ *and* $v^\mathcal{S}_{D'} \subseteq v^\mathcal{S}_D$*);*
  - $\mathcal{I}_\mathcal{S} \models D(c)$ *(since* $\cdot^{\mathcal{I}_\mathcal{S}}(c) \in v^\mathcal{S}_C$ *and* $v^\mathcal{S}_C \subset v^\mathcal{S}_D$*).*

  Overall, we got that $\mathcal{I}_\mathcal{S} \models \langle\{C \sqsubseteq D, D' \sqsubseteq D, C \sqcap D' \sqsubseteq \bot\}, \{D'(d), C(c), D(d), D(c)\}\rangle$*. Hence,* $\mathcal{I}_\mathcal{S} \models \tau_\triangleleft(v_C\ \{PP\}\ v_D)$*.*

- φ *is a constraint of the form* $v_C\ \{PPi\}\ v_D$*. The proof that* $\mathcal{I}_\mathcal{S} \models \tau_\triangleleft(v_C\ \{PPi\}\ v_D)$ *can be reduced equivalently to the proof that* $\mathcal{I}_\mathcal{S} \models \tau_\triangleleft(v_D\ \{PP\}\ v_C)$ *similarly to the previous case, since* $v_C\ \{PPi\}\ v_D$ *is equivalent to* $v_D\ \{PP\}\ v_C$*.*

We have proved that for each contraint $v_C\ \varphi\ v_D \in \Psi$*, we have that* $\mathcal{I}_\mathcal{S} \models \tau_\triangleleft(v_C\ \varphi\ v_D)$*. Therefore,* $\mathcal{I}_\mathcal{S} \models \tau_\triangleleft(\mathcal{N})$*. This concludes the proof.*

Theorem 5.4. Let $\mathcal{N}$ be a scenario, let $\mathcal{I}$ be a fulfilling interpretation of $\tau_\triangleleft(\mathcal{N})$ such that $\mathcal{I}$ is a model of $\tau_\triangleleft(\mathcal{N})$, and let $\mathcal{S}_\mathcal{I}$ be the flattening of $\mathcal{I}$. Then $\mathcal{S}_\mathcal{I} \models \mathcal{N}$..

Proof A.6 *Let $\mathcal{N} = \langle V, \Psi \rangle$ be a scenario and let $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ be a fulfilling interpretation of $\tau_{\triangleleft}(\mathcal{N})$ such that $\mathcal{I} \models \tau_{\triangleleft}(\mathcal{N})$. Let us denote by $\tau_{\triangleleft}(\mathcal{N}) = \langle \mathcal{A}, \mathcal{T} \rangle$ the backward translation of $\mathcal{N}$. Note that the set of concept names N_C of $\tau_{\triangleleft}(\mathcal{N})$ is formed of two parts: (1) the concept names C which are directly associated with a variable v_C from V and (2) the new concept names introduced in the backward translation (cf. Definition 5.8). For instance, some new concepts A', C', and D' appear in the translation of a constraint $v_C \ \{PO\} \ v_D$. Let us denote by N_C^* the subset of N_C formed by these new concept names. In addition, the set of N_I of $\tau_{\triangleleft}(\mathcal{N})$ is formed only of individual names artificially introduced in the translation; for instance, the individual a, c, and d appear in the translation of a constraint $v_C \ \{PO\} \ v_D$, and all these individual names of N_I.*

From Definition 5.6, we have $\mathcal{S}_{\mathcal{I}} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{S}_{\mathcal{I}}} \rangle$ that is a solution of $\mathcal{N}$, for every $(v_A)^{\mathcal{S}_{\mathcal{I}}} = A^{\mathcal{I}}$ where $A \in N_C \setminus N_C^*$ and $v_A \in V$ and $\cdot^{\mathcal{S}_{\mathcal{I}}}(a)$ for each $a \in N_I$. It can easily be verified that $\mathcal{S}_{\mathcal{I}}$ is the flattening of $\mathcal{I}$ (cf. Definition 5.6). Now we need to show that $\mathcal{S}_{\mathcal{I}} \models \mathcal{N}$. For this purpose, let us consider each constraint $v_C \ \varphi \ v_D$, and show that $\mathcal{I} \models \tau_{\triangleleft}(v_C \ \varphi \ v_D)$ then $\mathcal{S}_{\mathcal{I}} \models v_C \ \varphi \ v_D$. We fall into one of the following cases:

- *φ is a constraint of the form $v_C \ \{EQ\} \ v_D$. Since $\mathcal{I} \models \tau_{\triangleleft}(\mathcal{N})$, we have that $\mathcal{I} \models \tau_{\triangleleft}\{v_C \ \{EQ\} \ v_D\}$. From Definition 5.8, we get that $\tau_{\triangleleft}(v_C \ \{EQ\} \ v_D) = \langle C \equiv D, \emptyset \rangle$. Thus, we have $\mathcal{I} \models C \equiv D$, it is equivalent to $C^{\mathcal{I}} \equiv D^{\mathcal{I}}$ (cf. Table 2.5). So by definition of $\mathcal{S}_{\mathcal{I}}$, we get that $(v_C)^{\mathcal{S}_{\mathcal{I}}} = (v_D)^{\mathcal{S}_{\mathcal{I}}}$. Thus, from Table 5.3, we get that $\mathcal{S}_{\mathcal{I}} \models v_C \ \{EQ\} \ v_D$.*

- *φ is a constraint of the form $v_C \ \{DR\} \ v_D$. Since $\mathcal{I} \models \tau_{\triangleleft}(\mathcal{N})$, we have that $\mathcal{I} \models \tau_{\triangleleft}\{v_C \ \{DR\} \ v_D\}$. From Definition 5.8, we get that $\tau_{\triangleleft}(v_C \ \{DR\} \ v_D) = \langle C \sqcap D \sqsubseteq \bot, \emptyset \rangle$. Thus we have $\mathcal{I} \models C \sqcap D \sqsubseteq \bot$, it is equivalent to $C^{\mathcal{I}} \cap D^{\mathcal{I}} \subseteq \emptyset$ (cf. Table 2.5). So by definition of $\mathcal{S}_{\mathcal{I}}$, we get that $(v_C)^{\mathcal{S}_{\mathcal{I}}} \cap (v_D)^{\mathcal{S}_{\mathcal{I}}} = \emptyset$. Thus, from Table 5.3, we get that $\mathcal{S}_{\mathcal{I}} \models v_C \ \{DR\} \ v_D$.*

- *φ is a constraint of the form $v_C \ \{PO\} \ v_D$. Since $\mathcal{I} \models \tau_{\triangleleft}(\mathcal{N})$, we have that $\mathcal{I} \models \tau_{\triangleleft}\{v_C \ \{PO\} \ v_D\}$. From Definition 5.8, we get that $\tau_{\triangleleft}(v_C \ \{PO\} \ v_D) = \langle \{A' \sqsubseteq C \sqcap D, C' \sqsubseteq C, C' \sqcap D \sqsubseteq \bot, D' \sqsubseteq D, D' \sqcap C \sqsubseteq \bot\}, \{A'(a), C'(c), D'(d), C(c), C(a), D(d), D(a)\} \rangle$. Thus, we have $\mathcal{I} \models \{A' \sqsubseteq C \sqcap D, C' \sqsubseteq C, C' \sqcap D \sqsubseteq \bot, D' \sqsubseteq D, D' \sqcap C \sqsubseteq \bot\}$ and $\mathcal{I} \models \{A'(a), C'(c), D'(d), C(c), C(a), D(d), D(a)\}$. From Table 2.5, we get that $(A')^{\mathcal{I}} \subseteq C^{\mathcal{I}} \cap D^{\mathcal{I}}$, $(C')^{\mathcal{I}} \subseteq C^{\mathcal{I}}, (C')^{\mathcal{I}} \cap D^{\mathcal{I}} \subseteq \emptyset$, and $(D')^{\mathcal{I}} \subseteq D^{\mathcal{I}}, (D')^{\mathcal{I}} \cap C^{\mathcal{I}} \subseteq \emptyset$. Now, we need to consider the following cases:*

  - *Since $\mathcal{I} \models \mathcal{A}'(a)$, we get that $a^{\mathcal{I}} \in (A')^{\mathcal{I}}$. Moreover, $(A')^{\mathcal{I}} \subseteq C^{\mathcal{I}} \cap D^{\mathcal{I}}$. Thus $a^{\mathcal{I}} \in C^{\mathcal{I}} \cap D^{\mathcal{I}}$. Then we get that $C^{\mathcal{I}} \cap D^{\mathcal{I}} \neq \emptyset$.*
  - *Since $\mathcal{I} \models \{C'(c), C(c)\}$, we get that $c^{\mathcal{I}} \in (C')^{\mathcal{I}}$ and $c^{\mathcal{I}} \in C^{\mathcal{I}}$. Moreover, $(C')^{\mathcal{I}} \subseteq C^{\mathcal{I}}, (C')^{\mathcal{I}} \cap D^{\mathcal{I}} \subseteq \emptyset$. Thus $c^{\mathcal{I}} \in C^{\mathcal{I}}$ but $c^{\mathcal{I}} \notin D^{\mathcal{I}}$. Then, we get that $C^{\mathcal{I}} \not\subseteq D^{\mathcal{I}}$.*
  - *Since $\mathcal{I} \models \{D'(d), D(d)\}$, we get that $d^{\mathcal{I}} \in (D')^{\mathcal{I}}$ and $d^{\mathcal{I}} \in D^{\mathcal{I}}$. Moreover, $(D')^{\mathcal{I}} \subseteq D^{\mathcal{I}}, (D')^{\mathcal{I}} \cap C^{\mathcal{I}} \subseteq \emptyset$. Thus $d^{\mathcal{I}} \in D^{\mathcal{I}}$ but $d^{\mathcal{I}} \notin C^{\mathcal{I}}$. Then, we get that $D^{\mathcal{I}} \not\subseteq C^{\mathcal{I}}$.*

  From the above cases and by definition of $\mathcal{S}_{\mathcal{I}}$, we get that $(v_C)^{\mathcal{S}_{\mathcal{I}}} \cap (v_D)^{\mathcal{S}_{\mathcal{I}}} \neq \emptyset$, $(v_C)^{\mathcal{S}_{\mathcal{I}}} \not\subseteq (v_D)^{\mathcal{S}_{\mathcal{I}}}$, and $(v_D)^{\mathcal{S}_{\mathcal{I}}} \not\subseteq (v_C)^{\mathcal{S}_{\mathcal{I}}}$. In other words, we also have $\mathcal{S}_{\mathcal{I}} \models \{v_{A'} \subseteq v_C \cap v_D, v_C \not\subseteq v_D, v_D \not\subseteq v_C\}$. Thus, from Table 5.3, we have $\mathcal{S}_{\mathcal{I}} \models v_C \ \{PO\} \ v_D$.

- *φ is a constraint of the form $v_C \ \{PP, EQ\} \ v_D$. Since $\mathcal{I} \models \tau_{\triangleleft}(\mathcal{N})$, we have that $\mathcal{I} \models \tau_{\triangleleft}\{v_C \ \{PP, EQ\} \ v_D\}$. From Definition 5.8, we get that $\tau_{\triangleleft}(v_C \ \{PP, EQ\} \ v_D) = \langle C \sqsubseteq D, \emptyset \rangle$. Thus, we have $\mathcal{I} \models C \sqsubseteq D$, it is equivalent to $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ (cf. Table 2.5). So by definition of $\mathcal{S}_{\mathcal{I}}$, we get that $(v_C)^{\mathcal{S}_{\mathcal{I}}} \subseteq (v_D)^{\mathcal{S}_{\mathcal{I}}}$. Moreover, we also have that $(v_C)^{\mathcal{S}_{\mathcal{I}}} \subset (v_D)^{\mathcal{S}_{\mathcal{I}}}$ or $(v_C)^{\mathcal{S}_{\mathcal{I}}} = (v_D)^{\mathcal{S}_{\mathcal{I}}}$. Thus, from Table 5.3, we get that $\mathcal{S}_{\mathcal{I}} \models \{v_C \ \{PP\} \ v_D$ or $v_C \ \{EQ\} \ v_D\}$. It is also equivalent to $\mathcal{S}_{\mathcal{I}} \models v_C \ \{PP, EQ\} \ v_D$.*

- *φ is a constraint of the form $v_C\ \{PPi, EQ\}\ v_D$. The proof that $\mathcal{S}_\mathcal{I} \models v_C\ \{PPi, EQ\}\ v_D$ can be reduced equivalently to the proof that $\mathcal{S}_\mathcal{I} \models v_D\ \{PP, EQ\}\ v_C$ similarly to the previous case, since $v_C\ \{PP, EQ\}\ v_D$ is equivalent to $v_D\ \{PPi, EQ\}\ v_C$.*

- *φ is a constraint of the form $v_C\{PP\}v_D$. Since $\mathcal{I} \models \tau_\triangleleft(\mathcal{N})$, we have that $\mathcal{I} \models \tau_\triangleleft\{v_C\ \{PP\}\ v_D\}$. From Definition 5.8, we get that $\tau_\triangleleft(v_C\{PP\}v_D) = \langle\{C \sqsubseteq D, D' \sqsubseteq D, C \sqcap D' \sqsubseteq \bot\}, \{D'(d), C(c), D(d), D(c)\}\rangle$. Thus, we have $\mathcal{I} \models \{C \sqsubseteq D, D' \sqsubseteq D, C \sqcap D' \sqsubseteq \bot\}$ and $\mathcal{I} \models \{D'(d), C(c), D(d), D(c)\}$. From Table 2.5, we get that $C^\mathcal{I} \subseteq D^\mathcal{I}, (D')^\mathcal{I} \subseteq D^\mathcal{I}, C^\mathcal{I} \cap (D')^\mathcal{I} \subseteq \emptyset$. Moreover, since $\mathcal{I} \models \{D'(d), D(d)\}$, we have $d^\mathcal{I} \in (D')^\mathcal{I}$, $d^\mathcal{I} \in D^\mathcal{I}$. Then, $d^\mathcal{I} \notin C^\mathcal{I}$ (by $C^\mathcal{I} \cap (D')^\mathcal{I} \subseteq \emptyset$). Hence, $C^\mathcal{I} \not\subset D^\mathcal{I}$. Now, by definition of $\mathcal{S}_\mathcal{I}$, we get that $(v_C)^{\mathcal{S}_\mathcal{I}} \subset (v_D)^{\mathcal{S}_\mathcal{I}}$. In other words, we also have $\mathcal{S}_\mathcal{I} \models v_C \subset v_D$ (shown in the above part). Thus, from Table 5.3, we have $\mathcal{S}_\mathcal{I} \models v_C\ \{PP\}\ v_D$.*

- *φ is a constraint of the form $v_C\ \{PPi\}\ v_D$. The proof that $\mathcal{S}_\mathcal{I} \models v_C\ \{PPi\}\ v_D$ can be reduced equivalently to the proof that $\mathcal{S}_\mathcal{I} \models v_D\{PP\}v_C$ similarly to the previous case, since $v_C\ \{PP\}\ v_D$ is equivalent to $v_D\ \{PPi\}\ v_C$.*

This concludes the proof.

Appendix B

Second Appendix

B.1 Application and Experiments of Chapter 4

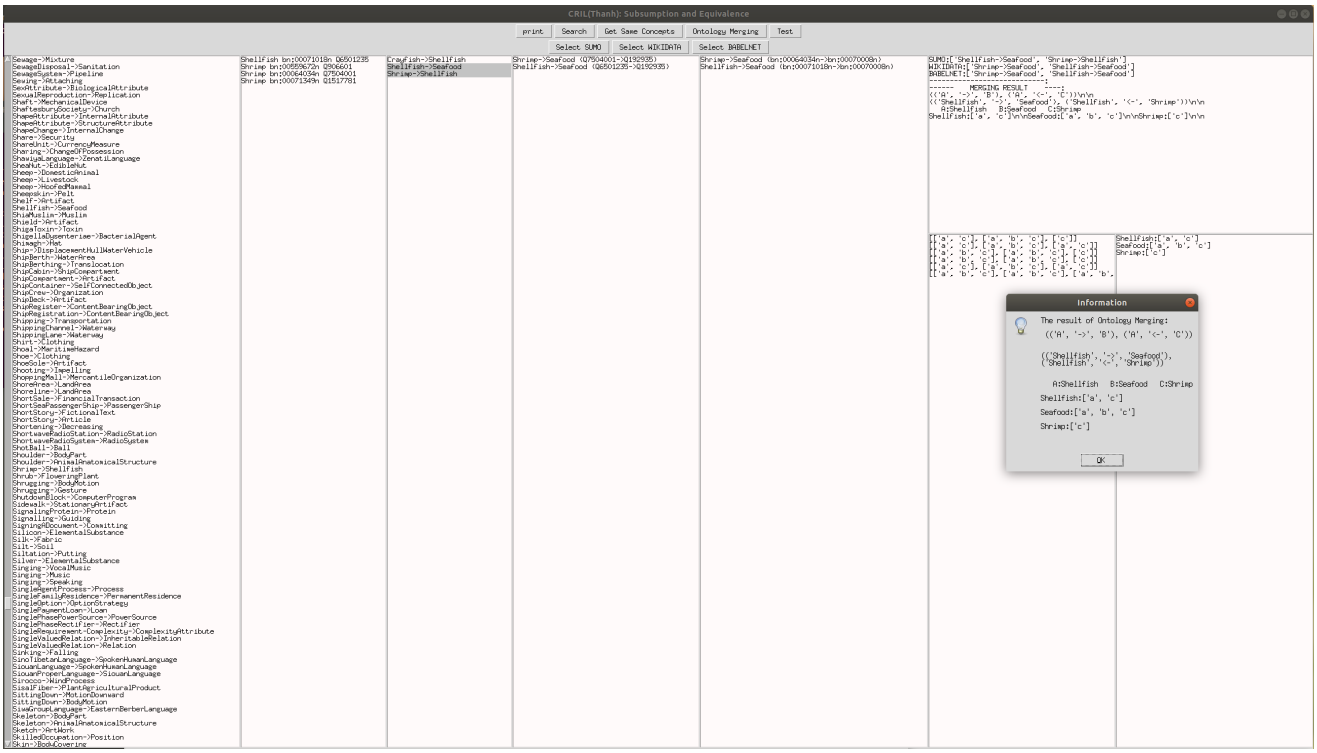


Figure B.1: Interface of knowledge base merging application.

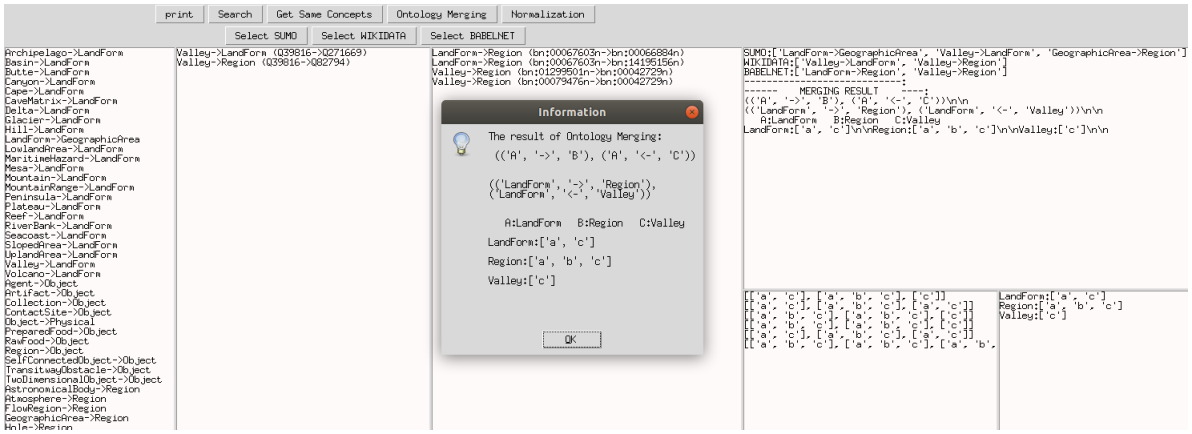


Figure B.2: Interface of merged result notification.

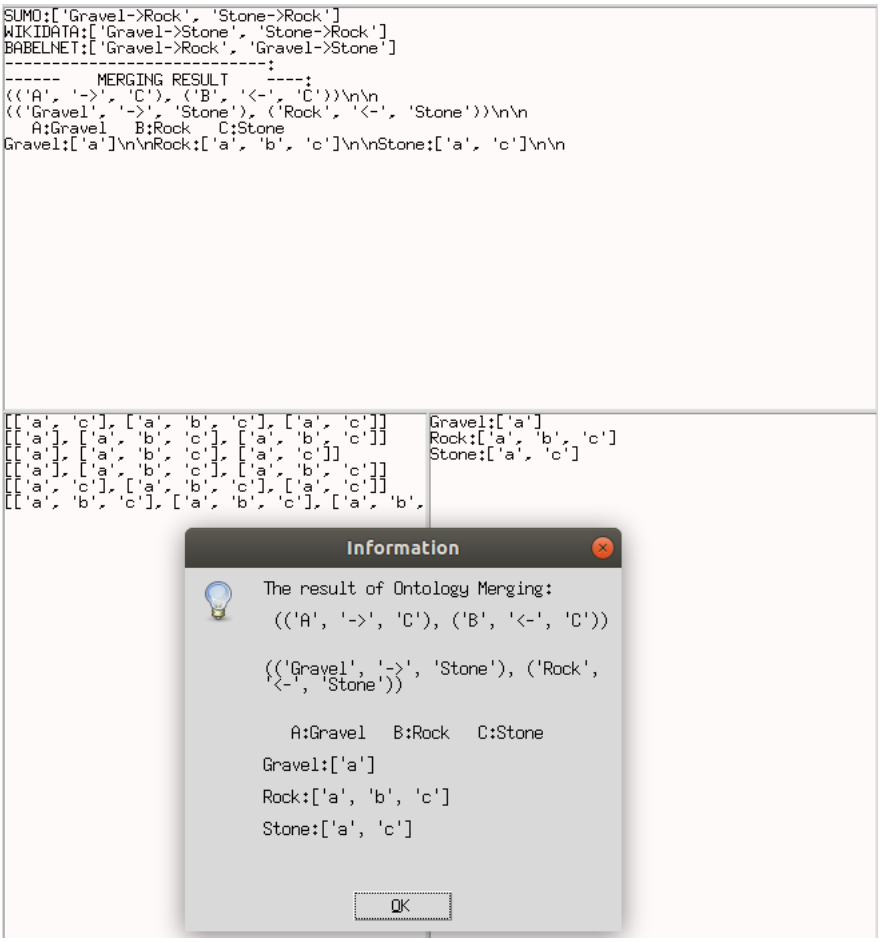


Figure B.3: Merged outcome of the case of *Gravel*, *Rock*, *Stone*

B.2 Implementation of Chapter 5

```
==================================
| The input Sources (Ontologies):
| 1 [['Paper', 'is-a', 'Text'], ['Text', 'dj', 'Document'], ['Paper', 'is-a', 'Book'], ['Pap
er', 'dj', 'Document'], ['Book', 'dj', 'Document']]
| 2 [['Paper', 'is-a', 'Text'], ['Text', 'is-a', 'Book'], ['Document', 'is-a', 'Book'], ['Do
cument', 'is-a', 'Paper']]
| 3 [['Paper', 'is-a', 'Text'], ['Book', 'is-a', 'Document'], ['Document', 'is-a', 'Paper'],
 ['Document', 'is-a', 'Text']]
| 4 [['Document', 'dj', 'Paper'], ['Paper', 'is-a', 'Text'], ['Book', 'is-a', 'Document'], [
'Text', 'dj', 'Book'], ['Text', 'dj', 'Document']]
|------------------
| The input Sources (QCNs):|
| 1 [['Text', ['PPI', 'EQ'], 'Paper'], ['Text', ['DR'], 'Document'], ['Paper', ['PP', 'EQ'],
 'Book'], ['Paper', ['DR'], 'Document'], ['Book', ['DR'], 'Document']]
| 2 [['Text', ['PPI', 'EQ'], 'Paper'], ['Text', ['PP', 'EQ'], 'Book'], ['Book', ['PPI', 'EQ'
], 'Document'], ['Paper', ['PPI', 'EQ'], 'Document']]
| 3 [['Text', ['PPI', 'EQ'], 'Paper'], ['Book', ['PP', 'EQ'], 'Document'], ['Paper', ['PPI',
 'EQ'], 'Document'], ['Text', ['PPI', 'EQ'], 'Document']]
| 4 [['Paper', ['DR'], 'Document'], ['Text', ['PPI', 'EQ'], 'Paper'], ['Book', ['PP', 'EQ'],
 'Document'], ['Text', ['DR'], 'Book'], ['Text', ['DR'], 'Document']]
```

Figure B.4: Implementation of input practical ontology sources.

```
==============--Result--===============
| The final merged QCN result:
| 1 ['Text', ['PPI', 'EQ'], 'Paper']
| 2 ['Text', ['PP'], 'Book']
| 3 ['Text', ['DR'], 'Document']
| 4 ['Paper', ['PP'], 'Book']
| 5 ['Paper', ['DR'], 'Document']
| 6 ['Book', ['PPI'], 'Document']
-------Merged Ontology---------
| 1 ['Paper', 'is-a', 'Text']
| 2 ['Text', 'is-a', 'Book']
| 3 ['Sub1', 'is-a', 'Book']
| 4 ['Sub1', 'dj', 'Text']
| 5 ['Text', 'dj', 'Document']
| 6 ['Paper', 'is-a', 'Book']
| 7 ['Sub2', 'is-a', 'Book']
| 8 ['Sub2', 'dj', 'Paper']
| 9 ['Paper', 'dj', 'Document']
| 10 ['Document', 'is-a', 'Book']
| 11 ['Sub3', 'is-a', 'Book']
| 12 ['Sub3', 'dj', 'Document']

Note: is-a: subsumption, dj: disjoint With
```

Figure B.5: Merged result of the case of $Text$, $Book$, $Document$, and $Paper$.

B.3 Implementation of Chapter 6

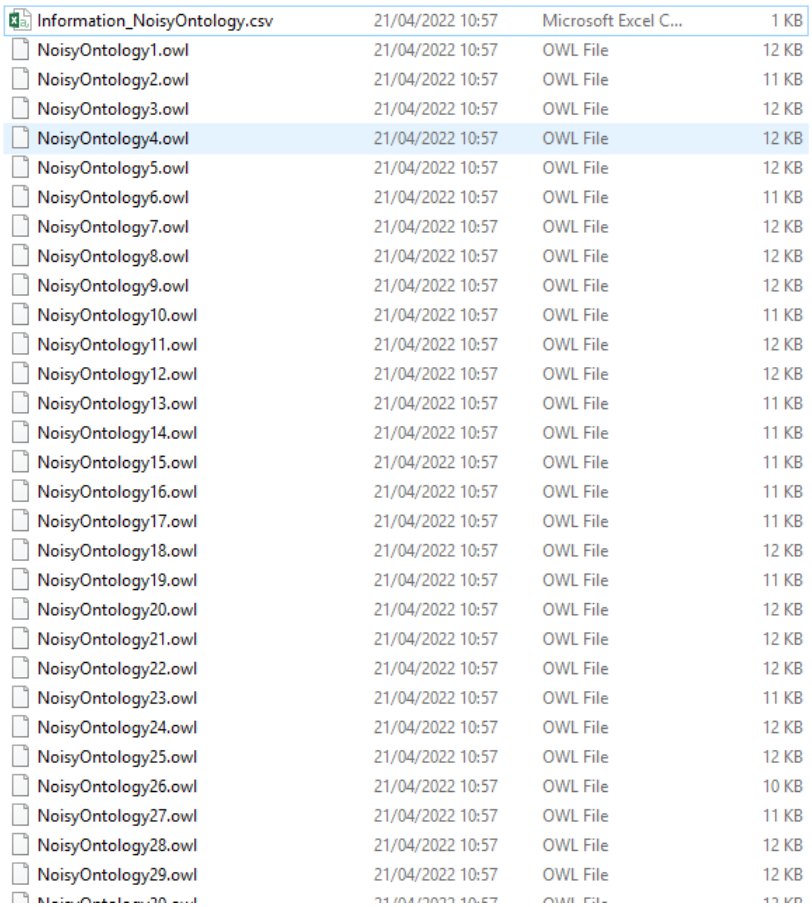


Figure B.6: Result of generating noisy ontologies.

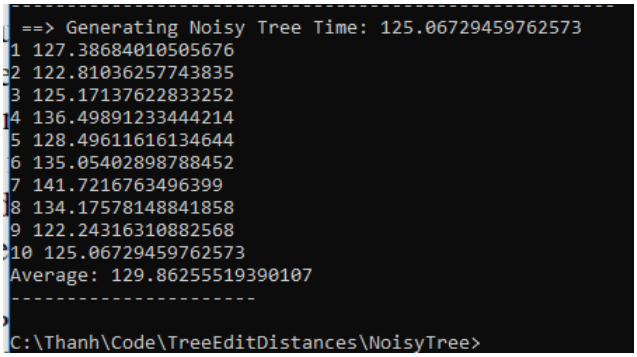


Figure B.7: Computation time of generating noisy ontologies of the *cmt* ontology.

Bibliography

[AABH18] Marc AIGUIER, Jamal ATIF, Isabelle BLOCH, and Céline HUDELOT. « Belief revision, minimal change and relaxation: A general framework based on satisfaction systems, and applications to description logics ». *Journal of Artificial Intelligence*, pages 160–180, 2018. 3.1.1

[AB20] Chris ALVIN and Adam BYERLY. « Analysis of Region-Based Integral Calculus Problems ». In *Proceedings of the Thirty-Third International Flairs Conference (FLAIRS'20)*, 2020. 5.1

[ABB+00] Michael ASHBURNER, Catherine A BALL, Judith A BLAKE, David BOTSTEIN, Heather BUTLER, J Michael CHERRY, Allan P DAVIS, Kara DOLINSKI, Selina S DWIGHT, Janan T EPPIG, and OTHERS. « Gene Ontology: Tool for The Unification of Biology ». *Journal of Nature Genetics*, 25(1):25–29, 2000. 2.3.4

[ABC99] Marcelo ARENAS, Leopoldo BERTOSSI, and Jan CHOMICKI. « Consistent Query Answers in Inconsistent Databases ». In *Proceedings of the Eighteenth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS'99)*, Page 68–79, 1999. 3.5.2

[ABG08] Nikolaus AUGSTEN, Michael BÖHLEN, and Johann GAMPER. « The PQ-Gram Distance between Ordered Labeled Trees ». *Journal of ACM Transactions on Database Systems (TODS)*, pages 1–36, 2008. 6.7

[ABM15] Fatima ARDJANI, Djelloul BOUCHIHA, and Mimoun MALKI. « Ontology-Alignment Techniques: Survey and Analysis ». *International Journal of Modern Education and Computer Science*, 7:67–78, 11 2015. 1.1

[AC04] Leila AMGOUD and Claudette CAYROL. « Inferring from Inconsistency in Preference-Based Argumentation Frameworks ». *Journal of Automated Reasoning*, 29:125–169, 2004. 3.5.1

[AFH05] Grigoris ANTONIOU, Enrico FRANCONI, and Frank van HARMELEN. « *Introduction to Semantic Web Ontology Languages* », pages 1–21. Springer Berlin Heidelberg, 2005. 2.4.2

[AGM85] Carlos ALCHOURRÓN, Peter GÄRDENFORS, and David MAKINSON. « On the Logic of Theory Change: Partial Meet Contraction and Revision Functions ». *Journal of Symbolic Logic*, pages 510–530, 06 1985. 3.1, 3.1, 3.2, 3.2.1, 3.3

[AJ21] Shyan AKMAL and Ce JIN. « Faster Algorithms for Bounded Tree Edit Distance ». In *Proceedings of the 48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*, pages 12:1–12:15, 2021. 6.7

[Aku10] Tatsuya AKUTSU. « Tree Edit Distance Problems: Algorithms and Applications to Bio-Informatics ». *Journal Of Transactions on Information and Systems*, 93(2):208–218, 2010. 6.2

[Ala09] Ahmed Khalifa ALASOUD. « *A Multi-Matching Technique for Combining Similarity Measures in Ontology Integration* ». PhD thesis, Concordia University, 2009. 3.4

[Ard00] Peter ARDENFORS. *Conceptual Spaces: The Geometry of Thought*. MIT Press, US, 2000. 5.1

[Arm73] David Malet ARMSTRONG. *Belief, Truth and Knowledge*. Cambridge University Press, 1973. 3.1.1

[Arr51] Kenneth Joseph ARROW. *Social Choice and Individual Values*, volume 12. New York: Wiley, 1951. 3.3.1, 3.4

[Bar77] Jon BARWISE. An introduction to first-order logic. In *Studies in Logic and the Foundations of Mathematics*, volume 90, pages 5–46. Elsevier, 1977. 2.2.1

[BB20] Meghyn BIENVENU and Camille BOURGAUX. « Querying and Repairing Inconsistent Prioritized Knowledge Bases: Complexity Analysis and Links with Abstract Argumentation ». In *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning (KR'20)*, pages 141–151, 2020. 3.5.1

[BBC+16] Salem BENFERHAT, Zied BOURAOUI, Huma CHADHRY, Mohd SHAFRY BIN MOHD RAHIM, Karim TABIA, and Abdelmoutia TELLI. « How to Reason from Inconsistent Ontologies with Prioritized Assertional Bases? ». In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI'16)*, 2016. 3.5.1

[BBC21] Sihem BELABBES, Salem BENFERHAT, and Jan CHOMICKI. « Handling Inconsistency in Partially preordered Ontologies: the Elect Method ». *Journal of Logic and Computation*, 31(5):1356–1388, 2021. 3.36

[BBHS08] Marc BERNARD, Laurent BOYER, Amaury HABRARD, and Marc SEBBAN. « Learning Probabilistic Models of Tree Edit Distance ». *Journal of Pattern Recognition*, pages 2611–2629, 08 2008. 6.2

[BBL05] Franz BAADER, Sebastian BRANDT, and Carsten LUTZ. « Pushing the $\mathcal{EL}$ Envelope ». In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI'05)*, Page 364–369, 2005. 1.3, 2.3.1, 2.3.3, 2.3.4, 2.3.4, 2.3.4, 2.2, 2.3, 2.1, 2.19, 2.3.4, 2.4.1, 2.4.2, 4.1, 4.2.2, 12, 5.2.1, 5.8, 6.6.1

[BBL10] Franz BAADER, Sebastian BRANDT, and Carsten LUTZ. « Pushing the $\mathcal{EL}$ Envelope Further ». *Proceedings of the OWL: Experiences and Directions Workshop (OWLED'10)*, 11 2010. 2.3.4

[BBL13] Salem BENFERHAT, Zied BOURAOUI, and Zied LOUKIL. « Min-based Fusion of Possibilistic DL-Lite Knowledge Bases ». In *Proceedings of the IEEE WIC ACM International Conference on Web Intelligence (WI'13)*, volume 1, pages 23–28. IEEE, 2013. 1.1, 2.4.1

[BBLR14] Salem BENFERHAT, Zied BOURAOUI, Sylvain LAGRUE, and Julien ROSSIT. « Min-based Assertional Merging Approach for Prioritized DL-Lite Knowledge Bases ». In

Proceedings of the 13th International Conference on Scalable Uncertainty Management (SUM'14), pages 8–21, 2014. 3.4, 5.1

[BBM17] Fernando BOBILLO, Carlos BOBED, and Eduardo MENA. « On the Generalization of the Discovery of Subsumption Relationships to the Fuzzy Case ». In *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'17)*, pages 1–6, 2017. 4.4, 6.6.1

[BBPW19] Salem BENFERHAT, Zied BOURAOUI, Odile PAPINI, and Eric WÜRBEL. « Assertional Removed Sets Merging of $DL\text{-}Lite$ Knowledge Bases ». In *Proceedings of the 13th International Conference of Scalable Uncertainty Management (SUM'19)*, volume 11940, pages 207–220, 2019. 1.2, 3.4, 5.1

[BBT15] Salem BENFERHAT, Zied BOURAOUI, and Karim TABIA. « How to Select One Preferred Assertional-Based Repair from Inconsistent and Prioritized DL-Lite Knowledge Bases? ». In *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI'15)*, Page 1450–1456, 2015. 3, 3.4, 5.6

[BCB+13] Islam BELTAGY, Cuong CHAU, Gemma BOLEDA, Dan GARRETTE, Katrin ERK, and Raymond MOONEY. « Montague Meets Markov: Deep Semantics with Probabilistic Logical Form ». In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics (*SEM'13)*, pages 11–21, 2013. 5.1

[BCCAS20] Zied BOURAOUI, José CAMACHO-COLLADOS, Luis Espinosa ANKE, and S. SCHOCKAERT. « Modelling Semantic Categories using Conceptual Neighborhood ». In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'20)*, pages 7448–7455, 2020. 2, 5.1, 7.2

[BCM+07] Franz BAADER, Diego CALVANESE, Deborah MCGUINNESS, Daniele NARDI, and Peter PATEL-SCHNEIDER. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 01 2007. 2.2.1, 2.2.2, 2.3, 2.3.1, 2.3.1, 2.3.2, 2.3.4, 5.6

[BDP93] Salem BENFERHAT, Didier DUBOIS, and Henri PRADE. « Argumentative Inference in Uncertain and Inconsistent Knowledge Bases ». In *Proceedings of the Uncertainty in Artificial Intelligence (UAI'93)*, pages 411–419, 1993. 3.5.1

[BDP97] Salem BENFERHAT, Didier DUBOIS, and Henri PRADE. « Syntactic Combination of Uncertain Information: A Possibilistic Approach ». In *Proceedings of the European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU'97)*, pages 30–42, 1997. 3.5.1

[BDP99] Salem BENFERHAT, Didier DUBOIS, and Henri PRADE. « An Overview of Inconsistency-Tolerant Inferences in Prioritized Knowledge Bases ». *Journal of Fuzzy sets, logics and reasoning about knowledge*, pages 395–417, 1999. 3.3.2

[BEF+06] Jos de BRUIJN, Marc EHRIG, Cristina FEIER, Francisco MARTÍNS-RECUERDA, Francois SCHARFFE, and Moritz WEITEN. « Ontology Mediation, Merging, and Aligning ». *Journal of Semantic Web Technologies*, pages 95 – 113, 07 2006. 3.4

[Ben94] Brandon BENNETT. « Spatial Reasoning with Propositional Logics ». In *Proceedings of the Principles of Knowledge Representation and Reasoning (KR'94)*, pages 51–62, 1994. 5.2.2

[BGBS22] Zied BOURAOUI, Víctor GUTIÉRREZ-BASULTO, and Steven SCHOCKAERT. « Integrating Ontologies and Vector Space Embeddings Using Conceptual Spaces ». In *International Research School in Artificial Intelligence in Bergen (AIB'22)*, pages 3:1–3:30, 2022. 5.8

[BGWH11] Mehul BHATT, Hans GUESGEN, Stefan WÖLFL, and Shyamanta HAZARIKA. « Qualitative Spatial and Temporal Reasoning: Emerging Applications, Trends, and Directions ». *Journal of Spatial Cognition & Computation*, 11:1–14, 02 2011. 5.1

[BH91] Franz BAADER and Bernhard HOLLUNDER. « A Terminological Knowledge Representation System with Complete Inference Algorithms ». In *Proceedings of the International Workshop on Processing Declarative Knowledge (PDK'91)*, pages 67–86, 1991. 2.3.2

[BH95] Philippe BESNARD and Anthony HUNTER. « Quasi-classical Logic: Non-trivializable Classical Reasoning from Incosistent Information ». In *Proceedings of the European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU'95)*, pages 44–51, 1995. 3.5.1

[BHLS17] Franz BAADER, Ian HORROCKS, Carsten LUTZ, and Uli SATTLER. « *Reasoning in the $\mathcal{EL}$ Family of Description Logics* », Page 140–167. An Introduction to Description Logic, 2017. 2.3

[Bie12] Meghyn BIENVENU. « On the Complexity of Consistent Query Answering in the Presence of Simple Ontologies ». In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI'12)*, Page 705–711, 2012. 3.5.2

[Bie19] Meghyn BIENVENU. « Inconsistency Handling in Ontology-Mediated Query Answering: A Progress Report ». In *Proceedings of the 32nd International Workshop on Description Logics (DL'19)*, 2019. 3.5.2

[BK03] Salem BENFERHAT and Souhila KACI. « Fusion of Possibilistic Knowledge Bases from A Postulate Point of View ». *International Journal of Approximate Reasoning*, 33(3):255–285, 2003. 2.4.2, 3.4

[BK14] Jasmin Christian BLANCHETTE and Daniel KÜHLWEIN. A Survey of Axiom Selection as a Machine Learning Problem. In *Infinity, Computability, and Metamathematics*. Tributes, College Publications, 2014.

[BKM91] Chitta BARAL, Sarit KRAUS, and Jack MINKER. « Combining Multiple Knowledge Bases ». *Journal of IEEE transactions on knowledge and data engineering*, 3(2):208–220, 1991. 3.3.2, 3.3.2

[BKM+22] Zied BOURAOUI, Sébastien KONIECZNY, Thanh MA, Nicolas SCHWIND, and Ivan VARZINCZAK. « Region-Based Merging of Open-Domain Terminological Knowledge ». In *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning (KR'22)*, 2022. 2, 1.3, 6.3, 6.5, 2, 6.6.2

[BKMS91] Chitta BARAL, Sarit KRAUS, Jack MINKER, and VS SUBRAHMANIAN. « Combining Knowledge Bases Consisting of First-order Theories ». In *Proceedings of the International Symposium on Methodologies for Intelligent Systems (ISMIS'91)*, pages 92–101, 1991. 3.3.2

[BKMV20a] Zied BOURAOUI, Sebastien KONIECZNY, Truong-Thanh MA, and Ivan . VARZINCZAK. « An Application for Merging Open-Domain Ontologies ». In *Journées d'Intelligence Artificielle Fondamentale, JIAF'20*, pages 79–88, 7 2020. 1

[BKMV20b] Zied BOURAOUI, Sébastien KONIECZNY, Thanh MA, and Ivan VARZINCZAK. « Model-based Merging of Open-Domain Ontologies ». In *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence (ICTAI'20)*, pages 29–34, 2020. 1, 1.3, 5.1, 6.3, 6.5, 1, 6.6.2

[BKMV22] Zied BOURAOUI, Sébastien KONIECZNY, Thanh MA, and Ivan VARZINCZAK. « Tree Edit Distance Based Ontology Merging Evaluation Framework ». In *Proceedings of the 15th International Conference on Knowledge Science, Engineering and Management (KSEM' 2022)*, 2022. 3

[BKR02] Abraham BOOKSTEIN, Vladimir A KULYUKIN, and Timo RAITA. « Generalized Hamming Distance ». *Information Retrieval Journal*, 5(4):353–375, 2002. 3.1.2

[BL04] Ronald BRACHMAN and Hector LEVESQUE. *Knowledge Representation and Reasoning*. Morgan Kaufmann Publishers Inc., 2004. 2.1

[BLS05] Franz BAADER, Carsten LUTZ, and Boontawee SUNTISRIVARAPORN. « Is Tractable Reasoning in Extensions of the Description Logic $\mathcal{EL}$ Useful in Practice? ». In *Proceedings of the Methods for Modalities Workshop (M4M'05)*, 2005. 2.3.4

[BLSW02] Franz BAADER, Carsten. LUTZ, Holger STURM, and Frank WOLTER. « Fusions of Description Logics and Abstract Description Systems ». *Journal of Artificial Intelligence Research*, Page 1–58, 2002. 1.2, 2.3, 3.4

[BM91] Robert S. BOYER and J. Strother MOORE. « *MJRTY-A Fast Majority Vote Algorithm* », pages 105–117. Journal of Automated Reasoning, 1991. 3.13, 4.1, 4.2

[BM93] Sasa BUVAC and Ian A MASON. « Propositional Logic of Context ». In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'93)*, pages 412–419, 1993. 2.1, 2.2.1

[BOPP20] Camille BOURGAUX, Ana OZAKI, Rafael PENALOZA, and Livia PREDOIU. « Provenance for the Description Logic ELHr ». In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI'20)*, pages 1862–1869, 7 2020. 2.3.4

[BPS07] Franz BAADER, Rafael PEÑALOZA, and Boontawee SUNTISRIVARAPORN. « Pinpointing in the Description Logic $\mathcal{EL}$ ». In *Proceedings of the 30th German Conference on Artificial Intelligence (KI'2007)*, 2007. 2.3.4

[BQL07] David BELL, Guilin QI, and Weiru LIU. « Approaches to Inconsistency Handling in Description-logic Based Ontologies ». In *Proceedings of the OTM Confederated International Conferences (OTM'07)*, pages 1303–1311. Springer, 2007. 2.12, 2.13

[BS92] Belaid BENHAMOU and Lakhdar SAIS. « Theoretical Study of Symmetries in Propositional Calculus and Applications ». In *Proceedings of the International Conference on Automated Deduction (CADE'92)*, pages 281–294. Springer, 1992. 2.2

[BS18] Zied BOURAOUI and S. SCHOCKAERT. « Learning Conceptual Space Representations of Interrelated Concepts ». In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'18)*, pages 1760–1766, 2018. 5.8, 7.2

[BTB+20] George BARYANNIS, Ilias TACHMAZIDIS, Sotirios BATSAKIS, Grigoris ANTONIOU, Mario ALVIANO, and Emmanuel PAPADAKIS. « A Generalised Approach for Encoding and Reasoning with Qualitative Theories in Answer Set Programming ». *Journal of Theory and Practice of Logic Programming*, pages 687–702, 07 2020. 5.2.2

[BVHH+04] Sean BECHHOFER, Frank VAN HARMELEN, Jim HENDLER, Ian HORROCKS, Deborah L MCGUINNESS, Peter F PATEL-SCHNEIDER, Lynn Andrea STEIN, and OTHERS. « OWL Web Ontology Language Reference ». *Journal of W3C recommendation*, pages 1–53, 2004. 2.4.1, 2.4.2, 2.4.2

[BZ77] Richard E BELLMAN and Lotfi Asker ZADEH. Local and Fuzzy Logics. In *Modern Uses of Multiple-Valued Logic*, pages 103–165. Springer, 1977. 2.4.1

[Car86] Thomas P CARPENTER. « Conceptual Knowledge as a Foundation for Procedural Knowledge ». *Journal of Conceptual and Procedural Knowledge: The Case of Mathematics*, pages 113–132, 1986. 2

[CBGG97] Anthony COHN, Brandon BENNETT, John GOODAY, and Mark GOTTS. « Qualitative Spatial Representation and Reasoning with the Region Connection Calculus ». *GeoInformatica*, pages 275–316, 1997. 1.3, 5.1

[CCR93] Zhang CUI, Anthony G COHN, and David A RANDELL. « Qualitative and Topological Relationships in Spatial Databases ». In *Proceedings of the International Symposium on Spatial Databases (ISSD'93)*, pages 296–315. Springer, 1993. 5.1

[CdLKM16] Thomas CARIDROIT, Tiago de LIMA, Sébastien KONIECZNY, and Pierre MARQUIS. « On Distances Between KD45n Kripke Models and Their Use for Belief Revision ». In *Proceedings of the Twenty Second European Conference on Artifical Intelligence (ECAI'16)*, pages 1053–1061, 2016. 3.2.2

[CF90] Roldano CATTONI and Enrico FRANCONI. « Walking through the semantics of frame-based description languages: A case study ». In *Proceedings of the Fifth International Symposium on Methodologies for Intelligent Systems (SMIS'90)*, 1990. 2.3.1

[CFF+98] Vinay K. CHAUDHRI, Adam FARQUHAR, Richard FIKES, Peter D. KARP, and James RICE. « OKBC: A Programmatic Foundation for Knowledge Base Interoperability ». In *Proceedings of the Annual Conference on Innovative Applications of Artificial Intelligence (IAAI-98)*, 1998. 2.4.1, 3.4

[CH85] Ashok K CHANDRA and David HAREL. « Horn Clause Queries and Generalizations ». *Journal of Logic Programming*, 2(1):1–15, 1985. 2.3.4

[Che80a] Brian F. CHELLAS. *Modal Logic - An Introduction*. Cambridge University Press, 1980. 2.2.1

[Che80b] Brian F CHELLAS. *Modal logic: an introduction*. Cambridge university press, 1980.

[CKMS09] Jean-Francois CONDOTTA, Souhila KACI, Pierre MARQUIS, and Nicolas SCHWIND. « Merging Qualitative Constraint Networks in a Piecewise Fashion ». In *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence (ICTAI'09)*, pages 605–608, November 2009. 2, 5.1

[CKMS10] Jean-Francois CONDOTTA, Souhila KACI, Pierre MARQUIS, and Nicolas SCHWIND. « A Syntactical Approach to Qualitative Constraint Networks Merging ». In *Proceedings of the International Conference on Logic Programming and Automated Reasoning (LPAR'10)*, pages 233–247. Springer, 2010. 2, 5, 5.1, 5.5, 12

[CKP18] Nadia CREIGNOU, Raïda KTARI, and Odile PAPINI. « Belief Update within Propositional Fragments ». *Journal of Artificial Intelligence Research*, pages 807–834, 2018. 3.1

[CKS08] Jean-Francois CONDOTTA, Souhila KACI, and Nicolas SCHWIND. « A Framework for Merging Qualitative Constraints Networks ». In *Proceedings of the International Florida Artificial Intelligence Research Society Conference (FLAIRS'08)*, pages 586–591, 2008. 5.5.1

[CMM01] Sylvie COSTE-MARQUIS and Pierre MARQUIS. « Knowledge Compilation for Closed World Reasoning and Circumscription ». *Journal of Logic and Computation*, 11(4):579–607, 2001. 2.2

[CNS16] Jean-François CONDOTTA, Issam NOUAOURI, and Michael SIOUTIS. « A SAT Approach for Maximizing Satisfiability in Qualitative Spatial and Temporal Constraint Networks ». In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR'16)*, pages 342–442, 2016. 12, 2, 7.1

[CP03] Peter CLARK and Bruce W. PORTER. « KM – The Knowledge Machine 2.0: Users Manual ». In *Computer Science*, 2003. 2.4.1

[CPP12] José Luis CHACÓN and Ramón PINO PÉREZ. « Exploring the Rationality of Some Syntactic Merging Operators ». In *Proceedings of the Ibero-American Conference on Artificial Intelligence (IBERAMIA'12)*, pages 21–30, 2012. 3.1

[CPPW14] Nadia CREIGNOU, Odile PAPINI, Reinhard PICHLER, and Stefan WOLTRAN. « Belief Revision within Fragments of Propositional Logic ». *Journal of Computer and System Sciences*, 80(2):427–449, 2014. 3.1, 3.1.2

[CPRW16] Nadia CREIGNOU, Odile PAPINI, Stefan RÜMMELE, and Stefan WOLTRAN. « Belief Merging within Fragments of Propositional Logic ». *Journal of ACM Transactions on Computational Logic (TOCL)*, pages 1–28, 2016. 2.2, 3.1

[CPSV06] Sik CHUN, Jeff PAN, Derek SLEEMAN, and Wamberto VASCONCELOS. « Ontology Inconsistency Handling: Ranking and Rewriting Axioms ». *Technical report, University of Aberdeen*, 2006. 3.5.2

[CR16] Enrico G. CALDAROLA and Antonio RINALDI. « An Approach to Ontology Integration for Ontology Reuse ». In *Proceedings of the International Conference on Information Reuse & Integration for Data Science (IRI'16)*, pages 384–393, 07 2016. 1.2

[CS16] Paula CHOCRON and Marco SCHORLEMMER. « Attuning Ontology Alignments to Semantically Heterogeneous Multi-Agent Interactions ». In *Proceedings of the Twenty Second European Conference on Artifical Intelligence (ECAI'16)*, Page 871–879, 2016. 3.4

[CSAH06] Ronald CORNET, Stefan SCHLOBACH, and Ameen ABU-HANNA. « An Approach for Debugging Description Logic-based Ontologies ». *International Journal of Human-Computer Studies*, 2006. 3.33, 3.34

[CSH06] Namyoun CHOI, Il-Yeol SONG, and Hyoil HAN. « A Survey on Ontology Mapping ». *SIGMOD Record*, 35:34–41, 09 2006. 3.4

[Cur10] Olivier CURÉ. « *Merging Expressive Spatial Ontologies Using Formal Concept Analysis with Uncertainty Considerations* », volume 256, pages 189–209. Methods for Handling Imperfect Spatial Information, 10 2010. 3.4

[CwCZ19] Fei CHANG, Guo wei CHEN, and Songmao ZHANG. « FCAMap-KG Results for OAEI 2019 ». In *Proceedings of the International Semantic Web Conference (OM@ISWC'19)*, Page 138–145, 2019. 3.4

[CWWM06] Huajun CHEN, Zhaohui WU, Heng WANG, and Yuxin MAO. « RDF/RDFS-based Relational Database Integration ». In *Proceedings of the International Conference on Data Engineering (ICDE'06)*, pages 94–94, 2006. 2.3, 2.4.2

[CWY19] Yu-Hua CHEN, Bryan WANG, and Yi-Hsuan YANG. « Demonstration of Performancenet: A Convolutional Neural Network Model for Score-to-Audio Music Generation ». *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI'19)*, 2019. 1.1

[CZ97] Alexander V. CHAGROV and Michael ZAKHARYASCHEV. *Modal Logic*, volume 35 of *Oxford logic guides*. Oxford University Press, 1997. 2.2.1

[D+06] Kevin DONNELLY and OTHERS. « SNOMED-CT: The Advanced Terminology and Coding System for eHealth ». *Journal of Studies in health technology and informatics*, 121:279, 2006. 2.3.4

[Dal88a] Mukesh DALAL. « Investigations into a Theory of Knowledge Base Revision: Preliminary Report ». In *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI'88)*, volume 2, pages 475–479. Citeseer, 1988. 3.1.2, 3.5

[Dal88b] Mukesh DALAL. « Updates in Propositional Databases ». In *Technical report, Rutgers University*, 1988. 3.3.1

[DBS77] Randall DAVIS, Bruce BUCHANAN, and Edward SHORTLIFFE. « Production Rules As a Representation for A Knowledge-based Consultation Program ». *Journal of Artificial Intelligence*, pages 15–45, 1977. 4

[dCRGdSL04] Davi de CASTRO REIS, Paulo Braz GOLGHER, Altigran Soares da SILVA, and Alberto HF LAENDER. « Automatic Web News Extraction Using Tree Edit Distance ». *Proceedings of the 13th International Conference on World Wide Web (WWW'04)*, 10:988672–988740, 2004. 6.2

[DEGM22] Igor DOUVEN, Shira ELQAYAM, Peter GÄRDENFORS, and Patricia MIRABILE. « Conceptual Spaces and the Strength of Similarity-based Arguments ». *Journal of Cognition*, 218:104951, 2022. 5.1

[DEGV01] Evgeny DANTSIN, Thomas EITER, Georg GOTTLOB, and Andrei VORONKOV. « Complexity and expressive power of logic programming ». *Journal of ACM Computing Surveys (CSUR)*, 33(3):374–425, 2001. 2.1

[DEKP20] Didier DUBOIS, Patricia EVERAERE, Sébastien KONIECZNY, and Odile PAPINI. Main Issues in Belief Revision, Belief Merging and Information Fusion. In *A Guided Tour of Artificial Intelligence Research*, pages 441–485. Springer, 2020. 2.2, 3, 3.3, 3.4

[DGL96] Giuseppe DE GIACOMO and Maurizio LENZERINI. « TBox and ABox Reasoning in Expressive Description Logics. ». *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR'96)*, 96(316-327):10, 1996. 2.3

[DKC05] Rose DIENG-KUNTZ and Olivier CORBY. « Conceptual Graphs for Semantic Web Applications ». In *International Conference on Conceptual Structures (ICCS'05)*, pages 19–50, 07 2005. 2.4.1

[DKP03] Didier DUBOIS, Sébastien KONIECZNY, and Henri PRADE. « Quasi-Possibilistic Logic and its Measures of Information and Conflict ». *Journal of Fundamenta Informaticae*, 57:101–125, 10 2003. 3.5.1

[DM02] Adnan DARWICHE and Pierre MARQUIS. « A Knowledge Compilation Map ». *Journal of Artificial Intelligence Research*, 17:229–264, 2002. 2.2

[DP94] Didier DUBOIS and Henri PRADE. « La Fusion d'Informations Imprécises ». *TS. Traitement du signal*, 11(6):447–458, 1994. 3.4

[DP97] Adnan DARWICHE and Judea PEARL. « On the Logic of Iterated Belief Revision ». *Journal of Artificial intelligence*, pages 1–29, 1997. 3.1

[DP98] Didier DUBOIS and Henri PRADE. Introduction: Revising, Updating and Combining Knowledge. In *Belief Change*, pages 1–15. Springer, 1998. 3.1.1

[DRH20] Jin DONG, Marc-Antoine RONDEAU, and William L HAMILTON. « Distilling Structured Knowledge for Text-Based Relational Reasoning ». In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP'20)*, pages 6782–6791, 2020. 1.1

[dSRABE18] Jomar da SILVA, Kate REVOREDO, Fernanda ARAUJO BAIÃO, and Jérôme EUZENAT. « Interactive Ontology Matching: Using Expert Feedback to Select Attribute Mappings ». In *Proceedings of the International Semantic Web Conference (ISWC'18)*, pages 25–36, October 2018. 4.1

[EKM05] Patricia EVERAERE, Sébastien KONIECZNY, and Pierre MARQUIS. « Quota and Gmin Merging Operators ». In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'05)*, Page 424, 2005. 3.3.3

[EKM07] Patricia EVERAERE, Sébastien KONIECZNY, and Pierre MARQUIS. « The Strategy-Proofness Landscape of Merging ». *Journal of Artificial Intelligence Research*, 28:49–105, 2007. 3.3.1, 4

[EKM10] Patricia EVERAERE, Sébastien KONIECZNY, and Pierre MARQUIS. « Disjunctive Merging: Quota and Gmin Merging Operators ». *Journal of Artificial Intelligence*, 174(12-13):824–849, 2010. 3.3.3

[EKM12] Patricia EVERAERE, Sébastien KONIECZNY, and Pierre MARQUIS. « Compositional Belief Merging ». In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR'12)*, 2012. 3.3.1, 4

[EKM15] Patricia EVERAERE, Sébastien KONIECZNY, and Pierre MARQUIS. « Belief Merging versus Judgment Aggregation ». In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS'15)*, Page 999–1007, Richland, SC, 2015. 3.3

[Euz17] Jérôme EUZENAT. « Interaction-based Ontology Alignment Repair with Expansion and Relaxation ». In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI'17)*, pages 185–191, 2017. 4.1

[FBB+09] Stefano FERILLI, Teresa Maria Altomare BASILE, Marenglen BIBA, Nicola DI MAURO, and Floriana ESPOSITO. « A General Similarity Framework for Horn Clause Logic ». *Journal of Fundamenta Informaticae*, 90(1-2):43–66, 2009. 2.3.4

[FBC13] Messaouda FAREH, Omar BOUSSAID, and Rachid CHALAL. « *Mapping System for Merging Ontologies* », pages 205–216. Springer International Publishing, 2013. 3.4

[FF19] Yang FENG and Lidan FAN. « Ontology Semantic Integration based on Convolutional Neural Network ». *Journal of Neural Computing and Applications*, pages 8253–8266, 2019. 7.2

[FH11] Eduardo FERMÉ and Sven Ove HANSSON. « AGM 25 Rears: Twenty-Five Years of Research in Belief Change ». *Journal of Philosophical Logic*, pages 295–331, 2011. 3

[FH18] Eduardo FERMÉ and Sven Ove HANSSON. *Belief change: introduction and overview*. Springer, 2018. 3.1

[FHP+06] Giorgos FLOURIS, Zhisheng HUANG, Jeff Z. PAN, Dimitris PLEXOUSAKIS, and Holger WACHE. « Inconsistencies, Negations and Changes in Ontologies ». In *Proceedings of the National Conference on Artificial intelligence (AAAI'06)*, Page 1295–1300, 2006. 2.12, 3.2.2

[FK85] Richard FIKES and Tom KEHLER. « The Role of Frame-Based Representation in Reasoning ». *Journal of Association for Computing Machinery*, Page 904–920, 1985. 3

[For96] Peter FORREST. « From Ontology to Topology in The Theory of Regions ». *Journal of The Monist*, 79(1):34–50, 1996. 2

[FP01] George FLOURIS and Dimitris PLEXOUSAKIS. « Belief Revision in Propositional Knowledge Bases ». In *Proceedings of the 8th Panhellenic Conference on Informatics*, pages 412–421, 2001. 3.1.1

[FPA06] Giorgos FLOURIS, Dimitris PLEXOUSAKIS, and Grigoris ANTONIOU. « On generalizing the AGM postulates ». In *Proceedings of the European Starting AI Researchers' Symposium (STAIRS'06)*, volume 142, pages 132–143, 2006. 3.9

[Fra96] Hubert FRANK. « A new axiom system of fuzzy logic ». *Journal of Fuzzy sets and systems*, 77(2):203–205, 1996. 2.4.1

[Fre92] Christian FREKSA. « Temporal Reasoning Based on Semi-Intervals ». *Journal of Artificial Intelligence*, 54(1):199–227, 1992. 5.5.1

[FS19] Eduardo FERMÉ and Guillermo R SIMARI. « Introduction to the Special Issue on Belief Revision, Argumentation, Ontologies, and Norms ». *Journal of Annals of Mathematics and Artificial Intelligence*, 87(3):185–186, 2019. 3.2

[FUV83] Ronald FAGIN, Jeffrey D. ULLMAN, and Moshe Y. VARDI. « On the Semantics of Updates in Databases ». In *Proceedings of the 2nd ACM SIGACT-SIGMOD Symposium on Principles of Database Systems (PODS'83)*, Page 352–365, 1983. 3.1, 3.4

[FVHH+01] Dieter FENSEL, Frank VAN HARMELEN, Ian HORROCKS, Deborah L MCGUINNESS, and Peter F PATEL-SCHNEIDER. « OIL: An ontology infrastructure for the semantic web ». *Journal of IEEE intelligent systems*, 16(2):38–45, 2001. 2.4.2

[Gär88] Peter GÄRDENFORS. *Knowledge in Flux: Modeling the Dynamics of Epistemic States.* The MIT press, 1988. 3.1, 3.1, 3.2, 3.2.1, 3.6, 3.4

[GH11] John GRANT and Anthony HUNTER. « Measuring the Good and the Bad in Inconsistent Information ». In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI'11)*, Page 2632–2637, 2011. 3.5.1, 3.5.1

[GH17] John GRANT and Anthony HUNTER. « Analysing Inconsistent Information Using Distance-Based Measures ». *International Journal of Approximate Reasoning*, Page 3–26, 2017. 3.5.1

[GHL+19] Rafael GONÇALVES, Matthew HORRIDGE, Rui LI, Yu LIU, Mark MUSEN, Csongor NYULAS, Evelyn OBAMOS, Dhananjay SHROUTY, and David TEMPLE. « Use of OWL and Semantic Web Technologies at Pinterest ». In *Proceedings of the International Semantic Web Conference (ISWC'19)*, pages 418–435, 10 2019. 1.1

[GHLS08] Birte GLIMM, Ian HORROCKS, Carsten LUTZ, and Ulrike SATTLER. « Conjunctive Query Answering for the Description Logic $\mathcal{SHIQ}$ ». *Journal of Artificial Intelligence Research*, Page 157–204, 2008. 2.3.1

[GHM+14] Birte GLIMM, Ian HORROCKS, Boris MOTIK, Giorgos STOILOS, and Zhe WANG. « HermiT: An OWL 2 Reasoner ». *Journal of Automated Reasoning*, 53(3):245–269, 2014. 2.4.2

[GHVD03] Benjamin N. GROSOF, Ian HORROCKS, Raphael VOLZ, and Stefan DECKER. « Description Logic Programs: Combining Logic Programs with Description Logic ». In *Proceedings of the 12th International Conference on World Wide Web (WWW'03)*, Page 48–57. Association for Computing Machinery, 2003. 2.4.2

[GL86] Michael P GEORGEFF and Amy L LANSKY. « Procedural Knowledge ». *Proceedings of the IEEE*, 74(10):1383–1398, 1986. 2

[GL00] Benjamin N GROSOF and Yannis LABROU. An approach to using XML and a rule-based content language with an agent communication language. In *Issues in Agent Communication*, pages 96–117. Springer, 2000. 2.4.2

[GM88] Peter GÄRDENFORS and David MAKINSON. « Revisions of Knowledge Systems using Epistemic Entrenchment ». In *Proceedings of the 2nd Conference on Theoretical Aspects of Reasoning about Knowledge (TARK'88)*, pages 83–95, 1988. 3.2.1

[GPC02] Asunción GÓMEZ-PÉREZ and Oscar CORCHO. « Ontology Languages for the Semantic Web ». *Journal of IEEE Intelligent systems*, 17(1):54–60, 2002. 2.3, 2.4.1, 2.4.2

[GPZ88] Georg GOTTLOB, Paolo PAOLINI, and Roberto ZICARI. « Properties and Update Semantics of Consistent Views ». *Journal of ACM Transactions on Database Systems (TODS)*, 13(4):486–524, 1988. 1.1

[Gro04] Bernard GROFMAN. « *Arrow's Impossibility Theorem* », pages 349–351. Springer US, 2004. 3.3.1

[GRP09] Dov GABBAY, Odinaldo RODRIGUES, and Gabriella PIGOZZI. « Connections between Belief Revision, Belief Merging and Social Choice ». *Journal of Logic and Computation*, pages 445–446, 2009. 1.1, 3.3, 3.4

[GRS16] Sarah GAGGL, Sebastian RUDOLPH, and Lukas SCHWEIZER. « Fixed-Domain Reasoning for Description Logics ». In *Proceedings of the Twenty-Second European Conference on Artificial Intelligence (ECAI'16)*, Page 819–827, 2016. 2.3.1, 2.9, 2.3.1, 4.2.2

[Gru93] Thomas R GRUBER. « A Translation Approach to Portable Ontology Specifications ». *Journal of Knowledge Acquisition*, pages 199–220, 1993. 2.4

[Gru95] Thomas R. GRUBER. « Toward Principles for the Design of Ontologies Used for Knowledge Sharing? ». *International Journal of Human-Computer Studies*, 43(5):907–928, 1995. 2.4.1

[Gru03] Michael GRUNNINGER. Enterprise Modelling. In *Handbook on Enterprise Architecture*, pages 515–541. Springer, 2003. 2.4.1

[GS13] S.A. GÓMEZ and Guillermo SIMARI. « Merging of Ontologies using Belief Revision and Defeasible Logic Programming ». *Journal of Inteligencia Artificial*, pages 16–28, 01 2013. 3.4

[GSG00] Aldo GANGEMI, Geri STEVE, and Fabrizio GIACOMELLI. « ONIONS: An Ontological Methodology for Taxonomic Knowledge Integration ». *Proceedings of the Twenty Second European Conference on Artifical Intelligence (ECAI'00)*, 04 2000. 3.4

[GU89] Giorgio GALLO and Giampaolo URBANI. « Algorithms for Testing the Satisfiability of Propositional Formulae ». *Journal of Logic Programming*, 7(1):45–61, 1989. 2.2

[GZK03] Martin GOGOLLA, Paul ZIEMANN, and Sabine KUSKE. « Towards an Integrated Graph Based Semantics for UML ». *Electronic Notes in Theoretical Computer Science*, 72(3):160–175, 2003. 2.4.1

[Han98] Sven Ove HANSSON. Revision of Belief Sets and Belief Bases. In *Belief change*, pages 17–75. Springer, 1998. 3.1.1

[HBC+21] Aidan HOGAN, Eva BLOMQVIST, Michael COCHEZ, Claudia D'AMATO, Gerard De MELO, Claudio GUTIERREZ, Sabrina KIRRANE, José Emilio Labra GAYO, Roberto NAVIGLI, Sebastian NEUMAIER, and et AL. « Knowledge Graphs ». *Journal of ACM Computing Surveys*, 54:1–37, 2021. 4.1

[HBFK10] Frederik HOGENBOOM, Bram BORGMAN, Flavius FRASINCAR, and Uzay KAYMAK. « Spatial Knowledge Representation on the Semantic Web ». In *IEEE Internet Computing*, pages 252 – 259, 10 2010. 5.1

[HDRDA11] Sajjad HUSSAIN, Jos DE ROO, Ali DANIYAL, and Syed Sibte Raza ABIDI. « Detecting and Resolving Inconsistencies in Ontologies using Contradiction Derivations ». In *Proceedings of the 2011 IEEE 35th Annual Computer Software and Applications Conference (SAC'11)*, pages 556–561. IEEE, 2011. 3.5, 3.5.2

[HFB+00] Ian HORROCKS, Dieter FENSEL, Jeen BROEKSTRA, Stefan DECKER, Michael ERDMANN, Carole GOBLE, Frank van HARMELEN, Michel KLEIN, Steffen STAAB, Rudi STUDER, and OTHERS. « The Ontology Inference Layer OIL », 2000. 2.4.2

[HH00] Jeff HEFLIN and James HENDLER. « Searching the Web with SHOE ». In *AAAI-2000 Workshop on AI for Web Search*, pages 35–40, 2000. 2.4.2

[HK05] Anthony HUNTER and Sébastien KONIECZNY. « *Approaches to Measuring Inconsistent Information* », pages 191–236. Springer Berlin Heidelberg, 2005. 3.5.1

[HK08] Anthony HUNTER and Sébastien KONIECZNY. « Measuring Inconsistency through Minimal Inconsistent Sets ». In *Proceedings of the Eleventh International Conference on Principles of Knowledge Representation and Reasoning (KR'08)*, pages 358–366, 2008. 2.2, 3, 3.32, 3.5.1

[HK10] Anthony HUNTER and Sébastien KONIECZNY. « On the Measure of Conflicts: Shapley Inconsistency Values ». *Journal of Artificial Intelligence*, 174(14):1007–1026, 2010. 3.5.1

[HKP03] Andreas HERZIG, Sébastien KONIECZNY, and Laurent PERRUSSEL. « On Iterated Revision in the AGM Framework ». In *Proceedings of the European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU'03)*, pages 477–488. Springer, 2003. 3.1.2, 3.2

[HKP+09] Pascal HITZLER, Markus KRÖTZSCH, Bijan PARSIA, Peter F PATEL-SCHNEIDER, and Sebastian RUDOLPH. « OWL 2 Web Ontology Language Primer ». *Journal of W3C recommendation*, Page 123, 2009. 2.3, 2.4.2

[HL08] Christoph HAASE and Carsten LUTZ. « Complexity of Subsumption in the $\mathcal{EL}$ Family of Description Logics: Acyclic and Cyclic TBoxes ». In *Proceedings of the Twenty Second European Conference on Artifical Intelligence (ECAI'08)*, volume 178, pages 25–29, 2008. 2.3.4

[HL20] Patrick HOHENECKER and Thomas LUKASIEWICZ. « Ontology Reasoning with Deep Neural Networks ». In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'20)*, pages 5060–5064, 7 2020. 7.2

[HM97] Volker HAARSLEV and Ralf MÖLLER. « SBox: A Qualitative Spatial Reasoner ». In *Proceedings of the 11th International Workshop on Qualitative Reasoning (QR'97)*, pages 105–113, 1997. 5.1

[HM01] Volker HAARSLEV and Ralf MÖLLER. « RACER System Description ». In *Proceedings of the International Joint Conference on Automated Reasoning (IJCAR'01)*, pages 701–705. Springer, 2001. 2.4.2

[Hoe09] Rinke HOEKSTRA. « *Ontology Representation: design patterns and ontologies that make sense* ». PhD thesis, University of Amsterdam, 2009. 2.1

[Hor05] Ian HORROCKS. « OWL: A Description Logic Based Ontology Language ». In *Proceedings of the International Conference on Principles and Practice of Constraint Programming (CP'05)*, pages 5–8. Springer, 2005. 2.3, 2.4.1, 2.4.2

[HP14] Sebastian J.I. HERZIG and Christiaan J.J. PAREDIS. « A Conceptual Basis for Inconsistency Management in Model-based Systems Engineering ». *Journal of Procedia CIRP*, 21:52–57, 2014. 3.5.1

[HPPS14] Andreas HERZIG, Pilar POZOS-PARRA, and François SCHWARZENTRUBER. « Belief Merging in Dynamic Logic of Propositional Assignments ». In *Proceedings of the International Symposium on Foundations of Information and Knowledge Systems (FoIKS'14)*, pages 381–398, 2014. 3.3.2

[HPW07a] Julien HUE, Odile PAPINI, and Eric WURBEL. « Syntactic propositional belief bases fusion with removed sets ». In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 66–77. Springer, 2007. 3.1

[HPW07b] Julien HUE, Odile PAPINI, and Eric WURBEL. « Syntactic Propositional Belief Bases Fusion with Removed Sets ». In *Proceedings of the European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU'07)*, pages 66–77, 2007. 3.3.2, 3.4, 3.4

[HPW09] Julien HUÉ, Odile PAPINI, and Eric WÜRBEL. « Merging belief bases represented by logic programs ». In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 371–382. Springer, 2009. 3.1

[HS01] Ian HORROCKS and Ulrike SATTLER. « Ontology Reasoning in the $\mathcal{SHOQ}$ (D) Description Logic ». In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'01)*, pages 199–204, 2001. 2.4.2

[Hun02] Anthony HUNTER. « Measuring Inconsistency in Knowledge via Quasi-Classical Models ». In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (ECAI'02)*, Page 68–73, 2002. 1.1, 3.5.1

[HW19] Adrian HARET and Stefan WOLTRAN. « Belief Revision Operators with Varying Attitudes Towards Initial Beliefs ». In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI'19)*, pages 1726–1733, 2019. 3, 3.1, 3.4, 3.4

[HWP08] Julien HUÉ, Eric WÜRBEL, and Odile PAPINI. « Removed Sets Fusion: Performing Off The Shelf. ». In *ECAI*, volume 178, pages 94–98, 2008. 3.1

[JBQ19] Qiu JI, Khaoula BOUTOUHAMI, and Guilin QI. « Resolving logical contradictions in description logic ontologies based on integer linear programming ». *Journal of IEEE Access*, 7:71500–71510, 2019. 3.5

[JL74] Neil D JONES and William T LAASER. « Complete problems for deterministic polynomial time ». In *Proceedings of the sixth annual ACM symposium on Theory of computing*, pages 40–46, 1974. 2.1

[JW12] Hué JULIEN and Matthias WESTPHAL. « Revising Qualitative Constraint Networks: Definition and Implementation ». *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence (ICTAI'12)*, 1:548–555, 2012. 5.1

[K+95] Jan KRAJICEK and OTHERS. *Bounded arithmetic, propositional logic and complexity theory*, volume 60. Cambridge University Press, 1995. 2.2.1

[KG05] Yarden KATZ and Bernardo GRAU. « Representing Qualitative Spatial Information in OWL-DL. ». In *CEUR Workshop Proceedings*, volume 188, 01 2005. 5.1

[KH16] Sri Krishna KUMAR and Jennifer A. HARDING. « Description Logic–Based Knowledge Merging for Concrete- and Fuzzy-Domain Ontologies ». *Journal of Engineering Manufacture*, 230:954 – 971, 2016. 5.1

[KHKS20] Bosung KIM, Taesuk HONG, Youngjoong KO, and Jungyun SEO. « Multi-Task Learning for Knowledge Graph Completion with Pre-trained Language Models ». In *Proceedings of 11th International Conference on Computational Logistics (ICCL'20)*, pages 1737–1743, 2020. 5.1

[KL07] Adila KRISNADHI and Carsten LUTZ. « Data Complexity in the $\mathcal{EL}$ Family of Description Logics ». In *Proceedings of the International Conference on Logic for Programming Artificial Intelligence and Reasoning (LPAR'07)*, pages 333–347, 2007. 2.3.4

[KLW95] Michael KIFER, Georg LAUSEN, and James WU. « Logical Foundations of Object-Oriented and Frame-Based Languages ». *Journal of Association for Computing Machinery*, Page 741–843, 1995. 2.4.1

[KM89] Hirofumi KATSUNO and Alberto O MENDELZON. « A unified view of propositional knowledge base updates ». In *Proceedings of the 11th international joint conference on Artificial intelligence-Volume 2*, pages 1413–1419, 1989. 3.1

[KM91a] Hirofumi KATSUNO and Alberto O MENDELZON. « Propositional Knowledge Base Revision and Minimal Change ». *Journal of Artificial Intelligence*, 52(3):263–294, 1991. 3, 3.1, 3.2.1, 3.7, 3.1, 3.3

[KM91b] Hirofumi KATSUNO and Alberto O MENDELZON. « Updating a Knowledge Base and Revising it ». In *Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference (KR91)*, Page 387. Morgan Kaufmann Pub, 1991. 3.1

[KM02] Sébastien KONIECZNY and Pierre MARQUIS. « Three-Valued Logics for Inconsistency Handling ». In *Journal of Logics in Artificial Intelligence*, pages 332–344. Springer Berlin Heidelberg, 2002. 3, 3.5.1

[KM04] Sébastien KONIECZNY and Pierre MARQUIS. « DA2 Merging Operators ». *Artificial Intelligence*, 157:49–79, 08 2004. 3, 3.1.2, 3.3, 3.3.3, 3.28, 3.3.3, 3.4, 4.2.3, 12

[KM05] Sébastien KONIECZNY and Pierre MARQUIS. « Reasoning under Inconsistency: The Forgotten Connective ». *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'01)*, 01 2005. 3.1, 3.5.1

[KMB08] Sébastien KONIECZNY, Pierre MARQUIS, and Philippe BESNARD. « Journal of Bipolarity in Billatice Logics ». *International Journal of Intelligent System*, 23:1046–1061, 10 2008. 3.5.1

[KMS18] Sudipta KAR, Suraj MAHARJAN, and Thamar SOLORIO. « Folksonomication: Predicting Tags for Movies from Plot Synopses using Emotion Flow Encoded Neural Network ». In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2879–2891, 2018. 1.1

[KMV15] Sebastien KONIECZNY, Pierre MARQUIS, and Srdjan VESIC. « On Supported Inference and Extension Selection in Abstract Argumentation Frameworks ». In *Proceedings of the thirteenth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'15)*, pages 49–59, 2015. 3.5.1

[Kni03] Kevin M KNIGHT. « Two information measures for inconsistent sets ». *Journal of Logic, Language and Information*, 12:227–248, 2003. 2.2

[KNPSQ20] Armita KHAJEH NASSIRI, Nathalie PERNELLE, Fatiha SAÏS, and Gianluca QUERCINI. « Generating Referring Expressions from RDF Knowledge Graphs for Data Linking ». In *Journal of International Semantic Web Conference (ISWC'20)*, pages 311–329. Springer, 2020. 4.1

[Kon00] Sébastien KONIECZNY. « On the Difference between Merging Knowledge Bases and Combining them ». In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR'00)*, 2000. 1.3, 3.1, 3.3, 3.3.2, 3.25, 3.26, 3.27, 3.3.2, 4.1, 4.2.3

[KP98] Sébastien KONIECZNY and Ramón Pino PÉREZ. « On the Logic of Merging ». In *Proceedings of the sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, pages 488–498, 1998. 3.1, 3.24, 3.1, 3.2, A.2

[KP99] Sébastien KONIECZNY and Ramón Pino PÉREZ. « Merging with Integrity Constraints ». In *Proceedings of the European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU'99)*, pages 233–244. Springer, 1999. 3.3, 3.12, 3.3, 3.4

[KP02] Sébastien KONIECZNY and Ramón Pino PÉREZ. « Merging Information under Constraints: A Logical Framework ». *Journal of Logic and computation*, 12(5):773–808, 2002. 1.1, 1.2, 1, 1.3, 2.2, 3.1, 3.1, 3.1.2, 3.1.2, 3.12, 3.3.1, 3.4, 3.5, 3.4, 3.4, 4, 4.1, 4.2.3, 27, 7.1, A.2

[KP17] Yevgeny KAZAKOV and Denis PONOMARYOV. « On the Complexity of Semantic Integration of OWL Ontologies ». In *Proceedings of the Description Logic Workshop (DL'17)*, 2017. 2.3.1

[KPP01] Sébastien KONIECZNY and Ramon PINO PÉREZ. « Some Operators for Iterated Revision ». In *Proceedings of the Sixth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'01)*, pages 498–509, 2001. 3.1

[KPP11] Sébastien KONIECZNY and Ramon PINO PÉREZ. « Logic Based Merging ». *Journal of Philosophical Logic*, 40(2):239–270, 2011. 1.1, 1.3, 3, 3.1.2, 3.4, 3.4, 4.2.3

[KPP13] Sébastien KONIECZNY and Ramón PINO PÉREZ. « Confluence Operators and Their Relationships with Revision, Update and Merging ». *Journal of Annals of Mathematics and Artificial Intelligence*, 69:73–101, 2013. 1.1, 3, 3.1, 3.2.2, 3.4

[KRH07] Markus KRÖTZSCH, Sebastian RUDOLPH, and Pascal HITZLER. « Complexity Boundaries for Horn Description Logics ». In *Proceedings of the International Joint Conference on Artificial Intelligence (AAAI'07)*, volume 7, pages 452–457, 2007. 2.3.1

[Kri20] Francesco KRIEGEL. « Most Specific Consequences in the Description Logic $\mathcal{EL}$ ». *Journal of Discrete Applied Mathematics*, pages 172–204, 2020. 1.3, 2.3.3, 2.3.4, 5.2.1

[KS03] Yannis KALFOGLOU and Marco SCHORLEMMER. « Ontology Mapping: The State of The Art ». *Journal of Semantic Interoperability and Integration*, 2, 01 2003. 3.4

[KS14] Sharifullah KHAN and Muhammad SAFYAN. « Semantic Matching in Hierarchical Ontologies ». *Journal of Computer and Information Sciences*, 26, 05 2014. 3.4

[KT18] Seiji KOIDE and Hideaki TAKEDA. « Semantic Web KM: A Knowledge Machine for Semantic Webs ». In *Journal of Computer Science*, 2018. 2.4.1

[KVS06] Konstantinos KOTIS, George A. VOUROS, and Konstantinos STERGIOU. « Towards Automatic Merging of Domain Ontologies: The HCONE-Merge Approach ». *Journal of Web Semantic*, Page 60–79, 2006. 3.4

[Lam19] Patrick LAMBRIX. « Completing and Debugging Ontologies: State of the Art and Challenges ». *Semantic Web Journal*, 2019. 3.5.2

[LBS19] Na LI, Zied BOURAOUI, and Steven SCHOCKAERT. « Ontology Completion Using Graph Convolutional Networks ». In *Proceedings of the International Semantic Web Conference (ISWC'19)*, pages 435–452, 2019. 5.1, 7.2

[Leg07] Catherine LEGG. « Ontologies on the Semantic Web ». *Journal of Annual Review of Information Science and Technology*, 41:407–451, 2007. 2.4

[Leh92] Fritz LEHMANN. « Semantic Networks ». *Computers & Mathematics with Applications*, 23(2):1–50, 1992. 2, 2.4.1

[LGM+17] Amir LAADHAR, Faïza GHOZZI, Imen MEGDICHE, Franck RAVAT, Olivier TESTE, and Faïez GARGOURI. « POMap Results for OAEI 2017 ». In *Proceedings of the International Semantic Web Conference (ISWC'17)*, pages pp. 1–7, 2017. 1.2, 4.1

[LLHX19] Christy Y. LI, Xiaodan LIANG, Zhiting HU, and Eric P. XING. « Knowledge-driven Encode, Retrieve, Paraphrase for Medical Image Report Generation ». In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'19)*, volume 33, pages 6666–6673, 2019. 1.1

[LLR+10] Domenico LEMBO, Maurizio LENZERINI, Riccardo ROSATI, Marco RUZZI, and Domenico Fabio SAVO. « Inconsistency-Tolerant Semantics for Description Logics ». In *Proceedings of the Fourth International Conference on Web Reasoning and Rule Systems (RR'10)*, pages 103–117, 09 2010. 3.35, 3.5.2

[LM99] Jinxin LIN and Alberto O. MENDELZON. « *Knowledge Base Merging by Majority* », pages 195–218. Springer, 1999. 1.2, 3.3.1, 3.4

[LM02] Jérôme LANG and Pierre MARQUIS. « Resolving Inconsistencies by Variable Forgetting ». In *Proceedings of the 8th International Conference on Principles of Knowledge Representation and Reasoning (KR'02)*, pages 239–250, Toulouse, France, 2002. 3.5.1

[LN92] Gerhard LAKEMEYER and Bernhard NEBEL. « Foundations of Knowledge Representation and Reasoning ». In *Lecture Notes in Computer Science*, pages 1–12, 01 1992. 2.1, 2.1

[LSRH97] Sean LUKE, Lee SPECTOR, David RAGER, and James HENDLER. « Ontology-based Web Agents ». In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS'97)*, pages 59–66, 1997. 2.4.2

[LT06] Patrick LAMBRIX and He TAN. « SAMBO-A System for Aligning and Merging Biomedical Ontologies ». *Journal of Web Semantics*, 4(3):196–206, sep 2006. 3.4

[LT08] Patrick LAMBRIX and He TAN. « *Ontology Alignment and Merging* », pages 133–149. Springer London, 2008. 3.4

[Luk20] Jan LUKASIEWICZ. « On the History of the Logic of Propositions ». *Journal of Polish Logic*, 1939:67–87, 1920. 2.2.1

[LY03] Sanjiang LI and Mingsheng YING. « Region Connection Calculus: Its models and composition table ». *Journal of Artificial Intelligence*, 145(1):121 – 146, 2003. 5.2.2

[LZLT07] Yi LI, Qian ZHONG, Juanzi LI, and Jie TANG. « Result of Ontology Alignment with RiMOM at OAEI'07 ». In *Proceedings of the 2nd International Conference on Ontology Matching (OM'07)*, OM'07, Page 227–235, 2007. 3.4

[MAWNH15] Dana MOVSHOVITZ-ATTIAS, Steven Euijong WHANG, Natalya NOY, and Alon HALEVY. « Discovering Subsumption Relationships for Web-Based Ontologies ». In *Proceedings of the International Workshop on the Web and Databases (WebDB'15)*, Page 62–69, 2015. 4.4, 6.6.1

[McB04] Brian MCBRIDE. The Resource Description Framework (RDF) and Its Vocabulary Description Language RDFS. In *Handbook on ontologies*, pages 51–65. Springer, 2004. 2.4.2

[Mei15] Christian MEILICKE. « MAMBA - Results for the OAEI ». In *Proceedings of the International Semantic Web Conference (OM@ISWC'15)*, 2015. 1.2

[MET11] Julian MENDEZ, Andreas ECKE, and Anni-Yasmin TURHAN. « Implementing completion-based inferences for the EL-family ». In *Proceedings of the 24th International Workshop on Description Logics (DL'11)*, volume 745, pages 334–344, 2011. 2.3.4, 2.19, 2.3.4

[MFSH03] Deborah L MCGUINNESS, Richard FIKES, Lynn Andrea STEIN, and James A HENDLER. « DAML-ONT: An Ontology Language for the Semantic Web. ». In *Spinning the Semantic Web*, pages 65–93, 2003. 2.3, 2.4.2

[MGC02] Thomas Andreas MEYER, Aditya K. GHOSE, and Samir CHOPRA. « Syntactic Representations of Semantic Merging Operations ». In *Proceedings of the Pacific Rim International Conference on Artificial Intelligence (PRICAI'22)*, 2002. 3.4

[MLJ11] Kedian MU, Weiru LIU, and Zhi JIN. « A general framework for measuring inconsistency through minimal inconsistent sets ». *Knowledge and Information Systems*, 27:85–114, 2011. 2.2

[MPP20] Pierre MARQUIS, Odile PAPINI, and Henri PRADE. *A Guided Tour of Artificial Intelligence Research: Volume I: Knowledge Representation, Reasoning and Learning*. Springer Nature, 2020. 2.1

[MSM+20] Joao MARQUES-SILVA, Carlos MENCÍA, and OTHERS. « Reasoning about Inconsistent Formulas ». In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence. (IJCAI'20)*, 2020. 3.30

[MVH+04] Deborah L MCGUINNESS, Frank VAN HARMELEN, and OTHERS. « OWL Web Ontology Language Overview ». *Journal of W3C recommendation*, Page 2004, 2004. 2.4.2, 2.4.2

[Nav18] Roberto NAVIGLI. « Natural Language Understanding: Instructions for (Present and Future) Use ». In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI'18)*, pages 5697–5702, 07 2018. 1.1

[Neb89] Bernhard NEBEL. « A Knowledge Level Analysis of Belief Revision ». In *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning (KR'89)*, pages 301–311, 1989. 3.2.2

[Neb90] Bernhard NEBEL. « Reasoning and Revision in Hybrid Representation Systems ». In *Lecture Notes in Computer Science*, 1990. 3.2.2

[Neb98] Bernhard NEBEL. How Hard Is It to Revise a Belief Base? In *Belief Change*, pages 77–145. Springer, 1998. 3

[NM00] Natalya F. NOY and Mark A. MUSEN. « PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment ». In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI/IAAI'00)*, 2000. 3.4

[NM03] Natalya NOY and Mark Alan MUSEN. « The PROMPT Suite: Interactive Tools for Ontology Merging and Mapping ». *International Journal of Human-Computer Studies*, Page 983–1024, 2003. 3.4

[NPP03] Abhaya C NAYAK, Maurice PAGNUCCO, and Pavlos PEPPAS. « Dynamic Belief Revision Operators ». *Journal of Artificial Intelligence*, 146(2):193–228, 2003. 3.1.2, 3.4

[OBYD21] Inès OSMAN, Sadok BEN YAHIA, and Gayo DIALLO. « Ontology Integration: Approaches and Challenging Issues ». *Journal of Information Fusion*, 71:38–63, 2021. 3.4

[OCK+22] Barlas OGUZ, Xilun CHEN, Vladimir KARPUKHIN, Stan PESHTERLIEV, Dmytro OKHONKO, Michael SCHLICHTKRULL, Sonal GUPTA, Yashar MEHDAD, and Scott YIH. « UniK-QA: Unified Representations of Structured and Unstructured Knowledge for Open-Domain Question Answering ». In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL'22)*, pages 1535–1546, 2022. 1.1

[OK18] Peter OCHIENG and Swaib KYANDA. « Large-Scale Ontology Matching: State-of-the-Art Analysis ». *Journal of ACM Computing Surveys*, 2018. 3.4

[OVHP22] Felix OCKER, Birgit VOGEL-HEUSER, and Christiaan JJ PAREDIS. « A Framework for Merging Ontologies in the Context of Smart Factories ». *Journal of Computers in Industry*, 135:103571, 2022. 2

[OYD21] Inès OSMAN, Sadok Ben YAHIA, and Gayo DIALLO. « Ontology Integration: Approaches and Challenging Issues ». *Journal of Information Fusion*, 71:38–63, 2021. 3.4

[PA11] Mateusz PAWLIK and Nikolaus AUGSTEN. « RTED: A Robust Algorithm for the Tree Edit Distance ». *Proceedings of the International Conference on Very Large Data Bases (VLDB'11)*, 2011. 6.2

[PA14] Mateusz PAWLIK and Nikolaus AUGSTEN. « A Memory-Efficient Tree Edit Distance Algorithm ». In *Proceedings of the International Conference on Database and Expert Systems Applications (DEXA'14)*, pages 196–210, 2014. 6.2

[PA16] Mateusz PAWLIK and Nikolaus AUGSTEN. « Tree Edit Distance: Robust and Memory-Efficient ». *Journal of Information Systems*, 56:157–173, 2016. 6.2

[PA20] Mateusz PAWLIK and Nikolaus AUGSTEN. « Minimal edit-based diffs for large trees ». In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management (CIKM'20)*, pages 1225–1234, 2020. 6.4

[Pap92] Odile PAPINI. « A Complete Revision Function in Propositional Calculus ». In *Proceedings of the 10th European Conference on Artificial Intelligence (ECAI'92)*, Page 339–343, 1992. 3.1, 3.2, 3.4

[Pap00] Odile PAPINI. « Knowledge-base revision ». *The Knowledge Engineering Review*, 15(4):339–370, 2000. 3.2.1

[Pap01] Odile PAPINI. Iterated revision operations stemming from the history of an agent's observations. In *Frontiers in belief revision*, pages 279–301. Springer, 2001. 3.2, 3.4

[PC19] M PRIYA and Aswani Kumar CHERUKURI. « An Approach to Merge Domain Ontologies using Granular Computing ». *Journal of Granular Computing*, 08 2019. 3.4

[PDYP05] Rong PAN, Zhongli DING, Yang YU, and Yun PENG. « A Bayesian Network Approach to Ontology Mapping ». In *Proceedings of the International Semantic Web Conference (ISWC'05)*, pages 563–577, 2005. 3.4

[PE11] Daniele PORELLO and Ulle ENDRISS. « Ontology Merging as Social Choice ». In *Proceedings of the International Workshop on Computational Logic in Multi-Agent Systems (CLIMA'11)*, pages 157–170, 2011. 3.4, 3.4

[PKM08] Everaere PATRICIA, Sébastien KONIECZNY, and Pierre MARQUIS. « Conflict-based Merging Ooperators ». In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR'08)*, pages 348–357, 2008. 5.1

[PKM14] Everaere PATRICIA, Sébastien KONIECZNY, and Pierre MARQUIS. « On Egalitarian Belief Merging ». In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'14)*, 07 2014. 4.2.3

[PM98] Liberatore PAOLO and Schaerf MARCO. « Arbitration (or how to merge knowledge bases) ». *Journal of IEEE Transactions on Knowledge and Data Engineering*, 1998. 3.13

[Pos74] Howard POSPESEL. « Introduction to Logic. Propositional Logic ». *Journal of Symbolic Logic*, 43(2):383–383, 1974. 2.2.1

[PP18] Madalena PAVÃO and Catia PESQUITA. « Complex matching for multiple ontologies: an exploratory study ». In *Proceedings of the International Semantic Web Conference (OM@ISWC'18)*, 2018. 1.1

[PSHH04] P PATEL-SCHNEIDER, Patrick HAYES, and I HORROCKS. « OWL Web Ontology Language Semantics and Abstract Syntax ». *Journal of W3C recommendation*, 01 2004. 2.4.2

[PT97] Jonsson PETER and Drakengren THOMAS. « A Complete Classification of Tractability in RCC-5 ». *Journal of Artificial Intelligence Research*, 6:211–221, 1997. 2, 1.3, 5.2.2

[PTC+17] Daniele PORELLO, Nicolas TROQUARD, Roberto CONFALONIERI, Pietro GALLIANI, Oliver KUTZ, and Rafael PEÑALOZA. « Repairing Socially Aggregated Ontologies Using Axiom Weakening ». In *Proceedings of the International Conference of Principles and Practice of Multi-Agent Systems (PRIMA'17)*, pages 441–449, 2017. 3.5.1

[PTVS+16] Thomas PELLISSIER TANON, Denny VRANDECIC, Sebastian SCHAFFERT, Thomas STEINER, and Lydia PINTSCHER. « From Freebase to Wikidata: The Great Migration ». In *Proceedings of the International World Wide Web Conference (WWW'2016)*, pages 1419–1428, 2016. 5.1

[QD09] Guilin QI and J.F. DU. « Model-based Revision Operators for Terminologies in Description Logics. ». In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'19)*, pages 891–897, 01 2009. 1.1, 3.1.2, 3.2.2

[QHH+08] Guilin QI, Peter HAASE, Zhisheng HUANG, Qiu JI, Jeff Z. PAN, and Johanna VÖLKER. « A Kernel Revision Operator for Terminologies – Algorithms and Evaluation ». In *International Semantic Web Conference (ISWC'08)*, Page 419–434. Springer-Verlag, 2008. 3.5.2

[QLB06] Guilin QI, Weiru LIU, and David A. BELL. « Knowledge Base Revision in Description Logics ». In *Journal of Logics in Artificial Intelligence*, pages 386–398, 2006. 3.1.1, 3.2.2, 3.10, 3.11, 3.2, 3.2.2

[RACA18] Carlos Ramón RANGEL, Junior ALTAMIRANDA, Mariela CERRADA, and Jose AGUILAR. « Procedure Based on Semantic Similarity for Merging Ontologies by Non-Redundant Knowledge Enrichment ». *International Journal of Knowledge Management*, Page 16–36, 2018. 1.2

[RCC92] David A. RANDELL, Zhan CUI, and Anthony G. COHN. « A Spatial Logic Based on Regions and Connection ». In *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning (KR'92)*, Page 165–176, 1992. 5.1, 5.2.2

[Rev97] Peter REVESZ. « On the Semantics of Arbitration ». *International Journal of Algebra and Computation*, 07, 05 1997. 1.2, 1, 3.3.1, 3.4, 4.1, 4.3

[RM70] Nicholas RESCHER and Ruth MANOR. « On Inference from Inconsistent Premisses ». *Journal of Theory and Decision*, 1:179–217, 1970. 3.5.1

[RN01] J. RENZ and B. NEBEL. « Efficient Methods for Qualitative Spatial Reasoning ». *Journal of Artificial Intelligence Research*, 15:289–318, Oct 2001. 5.1

[RNC+94] Alan L RECTOR, William Anthony NOWLAN, Galen CONSORTIUM, and OTHERS. « The GALEN Project ». *Journal of Computer Methods and Programs in Bio-medicine*, 45(1-2):75–78, 1994. 2.3.4

[Ros11] Riccardo ROSATI. « On the Complexity of Dealing with Inconsistency in Description Logic Ontologies ». In *Proceedings of the International Joint Conference on Artificial Intelligence (AAAI'11)*, 2011. 2.3.1

[RR11] Salvatore RAUNICH and Erhard RAHM. « ATOM: Automatic target-driven ontology merging ». In *Proceedings of the IEEE International Conference on Data Engineering (ICDE'11)*, pages 1276 – 1279, 05 2011. 1.2, 3.4

[RR17] Tim ROCKTÄSCHEL and Sebastian RIEDEL. « End-to-end Differentiable Proving ». In *Proceedings of the International Conference on Neural Information Processing Systems (NIPS'17)*, pages 3791–3803, 2017. 5.1

[RS17] Sebastian RUDOLPH and Lukas SCHWEIZER. « Not Too Big, Not Too Small... Complexities of Fixed-Domain Reasoning in First-Order and Description Logics ». In *Progress in Artificial Intelligence*, pages 695–708, Cham, 2017. 1.2, 2.3.1, 2.3.1, 4.2.2

[RSS19] Herilalaina RAKOTOARISON, Marc SCHOENAUER, and Michèle SEBAG. « Automated Machine Learning with Monte-Carlo Tree Search ». In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI'19)*, pages 3296–3303, 08 2019. 1.1

[RT22] Ryokan RI and Yoshimasa TSURUOKA. « Pretraining with Artificial Language: Studying Transferable Knowledge in Language Models ». In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL'22)*, pages 7302–7315, 2022. 1.1

[Rud11] Sebastian RUDOLPH. « *Foundations of Description Logics* », pages 76–136. Reasoning Web Semantic Technologies. Springer, 2011. 2.1

[Sal21] Yakoub SALHI. « Inconsistency Measurement for Improving Logical Formula Clustering ». In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence (IJCAI'21)*, pages 1891–1897, 2021. 3.29, 3.30, 3.31

[Sch10] Nicolas SCHWIND. « *Fusion de réseaux de contraintes qualitatives* ». PhD thesis, Artois University, 2010. 3.1

[Sch11] Eric SCHWITZGEBEL. Belief. In *The Routledge Companion to Epistemology*, pages 40–50. Routledge, 2011. 3.1.1

[SE13] Pavel SHVAIKO and Jérôme EUZENAT. « Ontology Matching: State of the Art and Future Challenges ». *Journal of IEEE Transactions on Knowledge and Data Engineering*, 25(1):158–176, 2013. 3.4, 3.4

[Ser21] Francesc SERRATOSA. « Redefining the Graph Edit Distance ». *Journal of SN Computer Science*, 2, 11 2021. 6.1

[SFL77] Paul SLOVIC, Baruch FISCHHOFF, and Sarah LICHTENSTEIN. « Behavioral Decision Theory ». *Annual Review of Psychology*, 28(1):1–39, 1977. 3.3.1

[SGH20] Luke T SLATER, Georgios V GKOUTOS, and Robert HOEHNDORF. « Towards Semantic Interoperability: Finding and Repairing Hidden Contradictions in Biomedical Ontologies ». *Journal of BMC Medical Informatics and Decision Making*, 20(10):1–13, 2020. 3.5

[SIK+19] Nicolas SCHWIND, Katsumi INOUE, Sébastien KONIECZNY, Jean-Marie LAGNIEZ, and Pierre MARQUIS. « What Has Been Said? Identifying the Change Formula in A Belief Revision Scenario ». In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'19)*, pages 1865–1871, 2019. 3.1, 3.2.2

[SK12] Michael SIOUTIS and Manolis KOUBARAKIS. « Consistency of Chordal RCC-8 Networks ». In *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence (ICTAI'12)*, volume 1, pages 436–443, 2012. 5.1

[SL13] Steven SCHOCKAERT and Sanjiang LI. « Combining RCC5 Relations with Betweenness Information ». In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13)*, pages 1083–1089, 2013. 5.1, 5.2.2

[SLJ20] Michael SIOUTIS, Zhiguo LONG, and Tomi JANHUNEN. « On Robustness in Qualitative Constraint Networks ». In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI'20)*, pages 1813–1819, 7 2020. 5, 5.1, 5.4

[SM01] Gerd STUMME and Alexander MAEDCHE. « FCA-MERGE: Bottom-Up Merging of Ontologies ». In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'01)*, 2001. 3.4

[Smu68] Raymond M. SMULLYAN. *First-order Logic*. Springer-Verlag, 1968. 1.2, 2.1, 2.2.1, 2.2.2, 2.4.1

[Sow99] John F SOWA. *Knowledge Representation: Logical, Philosophical and Computational Foundations*. Brooks/Cole Publishing Company, 1999. 1

[SPA17] Stefan SCHWARZ, Mateusz PAWLIK, and Nikolaus AUGSTEN. « A New Perspective on the Tree Edit Distance ». In *Proceedings of the International Conference on Similarity Search and Applications (SISAP'17)*, pages 156–170, 2017. 6.1, 6.2

[SPD06] Julien SEINTURIER, Odile PAPINI, and Pierre DRAP. « A reversible framework for propositional bases merging ». In *Proceedings of the Workshop on Non-Monotomic Reasoning (NRM'06)*, pages 409–496, 2006. 3.1

[SPG+07] Evren SIRIN, Bijan PARSIA, Bernardo Cuenca GRAU, Aditya KALYANPUR, and Yarden KATZ. « Pellet: A Practical OWL-DL Reasoner ». *Journal of Web Semantics*, 5(2):51–53, 2007. 2.4.1, 2.4.2

[SPJ20] Michael SIOUTIS, Anastasia PAPARRIZOU, and Tomi JANHUNEN. « On Neighbourhood Singleton-Style Consistencies for Qualitative Spatial and Temporal Reasoning ». *Journal of Information and Computation*, Page 104638, 2020. 5, 5.5.1

[SPS22] Lucas SIMONNE, Nathalie PERNELLE, and Fatiha SAÏS. « Counter Effect Rules Mining in Knowledge Graphs ». In *Proceedings of the 23rd International Conference of Knowledge Engineering and Knowledge Management (EKAW'22)*, pages 167–173, 2022. 7.2

[SPST21] Lucas SIMONNE, Nathalie PERNELLE, Fatiha SAÏS, and Rallou THOMOPOULOS. « Differential Causal Rules Mining in Knowledge Graphs ». In *Proceedings of the 11th on Knowledge Capture Conference (KCAP'21)*, pages 105–112, 2021. 7.2

[SPSW01] Michael Q STEARNS, Colin PRICE, Kent A SPACKMAN, and Amy Y WANG. « SNOMED Clinical Terms: Overview of the Development Process and Project Status ». In *Proceedings of the AMIA Symposium*, Page 662, 2001. 2.3.4

[SSS91] Manfred SCHMIDT-SCHAUSS and Gert SMOLKA. « Attributive Concept Descriptions with Complements ». *Journal of Artificial intelligence*, 48(1):1–26, 1991. 2.3.2

[Sun09] Boontawee SUNTISRIVARAPORN. *Polynomial-Time Reasoning Support for Design and Maintenance of Large-Scale Biomedical Ontologies*. Ph.D. thesis. Technical University of Dresden, Germany, 06 2009. 2.3.3

[SY21] Nakul SHARMA and Prasanth YALLA. « Visualizing UML's Sequence and Class Diagrams Using Graph-Based Clusters ». *IOP Series: Materials Science and Engineering*, Page 012010, 02 2021. 2.4.1

[Tai79] Kuo-Chung TAI. « The Tree-to-Tree Correction Problem ». *Journal of the ACM*, Page 422–433, 1979. 6.2

[TBL09] David THAU, Shawn BOWERS, and Bertram LUDÄSCHER. « Merging Taxonomies under RCC-5 Algebraic Articulations ». *Journal of Computing Science and Engineering*, 3:109–126, 06 2009. 1.3, 5.1

[TCG+18] Nicolas TROQUARD, Roberto CONFALONIERI, Pietro GALLIANI, Rafael PENALOZA, Daniele PORELLO, and Oliver KUTZ. « Repairing Ontologies via Axiom Weakening ». In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'18)*, 2018. 3.5.2

[TH06] Dmitry TSARKOV and Ian HORROCKS. « FaCT++ Description Logic Reasoner: System Description ». In *Proceedings of the International Joint Conference on Automated Reasoning (IJCAR'06)*, pages 292–297. Springer, 2006. 2.4.2

[Thi19] Matthias THIMM. « Inconsistency Measurement ». In *Proceedings of the International Conference on Scalable Uncertainty Management (SUM'19)*, pages 9–23. Springer, 2019. 3.32

[THT18] Élodie THIÉBLIN, Ollivier HAEMMERLÉ, and Cássia TROJAHN. « CANARD Complex Matching System ». In *Proceedings of the International Semantic Web Conference (OM@ISWC'18)*, Page 138–143, 2018. 1.2, 3.4, 4.1

[TML07] Paul D THOMAS, Huaiyu MI, and Suzanna LEWIS. « Ontology Annotation: Mapping Genomic Regions to Biological Function ». *Journal of Current opinion in chemical biology*, 11(1):4–11, 2007. 3.4

[TPPvdS05] Christophe TILLIER, Béatrice PESQUET-POPESCU, and Mihaela van der SCHAAR. « Improved Update Operators for Lifting-based Motion-Compensated Temporal Filtering ». *IEEE Signal Processing Letters*, 12(2):146–149, 2005. 1.1

[Ung68] Peter UNGER. « An Analysis of Factual Knowledge ». *Journal of Philosophy*, pages 157–170, 1968. 2

[Var18] Ivan VARZINCZAK. « A Note on a Description Logic of Concept and Role Typicality for Defeasible Reasoning Over Ontologies ». *Logica Universalis*, 12:1–29, 09 2018. 2.3.2

[VHLP08] Frank VAN HARMELEN, Vladimir LIFSCHITZ, and Bruce PORTER. *Handbook of Knowledge Representation*. Elsevier, 2008. 2.2.1, 2.4.2

[Wan22] Yue WANG. « Two metrics on rooted unordered trees with labels ». *Journal of Algorithms for Molecular Biology*, pages 1–17, 2022. 6.7

[Wel10] Katrin WELLER. Knowledge Representation in the Social Semantic Web. In *Knowledge Representation in the Social Semantic Web*. KG Saur, 2010. 2.4.2

[WLB12] Wilson WONG, Wei LIU, and Mohammed BENNAMOUN. « Ontology Learning from Text: A Look Back and into the Future ». *Journal of ACM Computing Surveys (CSUR)*, 44(4):1–36, 2012. 2.4.1

[WQLY18] Yandong WANG, Mengling QIAO, Hui LIU, and Xinyue YE. « Qualitative Spatial Reasoning on Topological Relations by Combining the Semantic Web and Constraint Satisfaction ». *Journal of Geo-spatial Information Science*, 21(2):80–92, 2018. 5.1

[WWJQ12] Zhe WANG, Kewen WANG, Yifan JIN, and Guilin QI. « Ontomerge: A System for Merging DL-Lite Ontologies ». *CEUR Workshop proceedings*, 969:16–27, 2012. 1.2, 3.4, 5.1

[WWT10] Zhe WANG, Kewen WANG, and Rodney TOPOR. « A New Approach to Knowledge Base Revision in DL-Lite ». In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'10)*, pages 369–374, 2010. 3.2.2

[XHJJ19] Chen XIAOYANG, Huo HONGWEI, Huan JUN, and Scott Vitter JEFFREY. « An Efficient Algorithm for Graph Edit Distance Computation ». *Journal of Knowledge-Based Systems*, 163:762–775, 2019. 6.1

[YJH+21] Wenhao YU, Meng JIANG, Zhiting HU, Qingyun WANG, Heng JI, and Nazneen RAJANI. « Knowledge-Enriched Natural Language Generation ». In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'21)*, pages 11–16, 2021. 1.1

[ZS89] Kaizhong ZHANG and Dennis SHASHA. « Simple Fast Algorithms for the Editing Distance Between Trees and Related Problems ». *SIAM journal on computing*, 18(6):1245–1262, 1989. 6.1, 6.2, 6.2, 6.2

[ZSS92] Kaizhong ZHANG, Rick STATMAN, and Dennis SHASHA. « On the Editing Distance between Unordered Labeled Trees ». *Information Processing Letters*, 42(3):133–139, 1992. 3, 1.3, 6, 6.1, 6.2

[ZSWY19] Chen ZHENG, Yu SUN, Shengxian WAN, and Dianhai YU. « RLTM: An Efficient Neural IR Framework for Long Documents ». In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI'19)*, pages 5457–5463, 2019. 1.1

[ZXLVDB14] Xiaowang ZHANG, Guohui XIAO, Zuoquan LIN, and Jan VAN DEN BUSSCHE. « Inconsistency-Tolerant Reasoning with OWL DL ». *International Journal of Approximate Reasoning*, pages 557–584, 2014. 1.1

[ZZ16] Mengyi ZHAO and Songmao ZHANG. « FCA-Map Results for OAEI 2016 ». In *Proceedings of the International Semantic Web Conference (OM@ISWC'16)*, pages 1–7, 2016. 1.2, 4.1

Résumé

Le sujet de la thèse est la fusion d'ontologies, une approche pour intégrer diverses sources d'ontologies en une seule qui gère les conflits émergents. Cette thèse s'inspire de la théorie de la fusion de croyances pour fusionner des ontologies afin de produire une base de connaissances cohérente et unifiée. Dans ce contexte, nous proposons trois contributions suivantes.

Dans la première contribution, nous formalisons une nouvelle modèle de fusion basé sur la sémantique. Il s'agit d'une stratégie de fusion qui se concentre principalement sur la gestion des conflits sémantiques. Un conflit sémantique, qui n'est pas nécessairement logique, est une connaissance représentée de nombreuses manières différentes ou opposées. Nous proposons également une caractérisation formelle du modèle et nous montrons l'efficacité de la méthode à travers une évaluation expérimentale.

La deuxième contribution propose une méthode permettant de fusionner les connaissances terminologiques d'un domaine ouvert. Elle se focalise sur RCC5 qui est un formalisme de représentation des régions dans un espace topologique. Ce formalisme permet également de raisonner sur leurs relations ensemblistes. De plus, nous proposons une traduction fidèle des connaissances terminologiques issues des sources conflictuelles dans les espaces régionaux. Ici, nous fusionnons les bases de connaissances dans ces espaces et nous traduisons le résultat vers la langue des sources d'entrée. Notre technique utilise l'expressivité et la flexibilité de RCC5 pour traiter des connaissances contradictoires.

La dernière contribution est un cadre d'évaluation des opérateurs de fusion d'ontologies. La stratégie part d'une ontologie originale pour créer des ontologies bruitées en tant qu'ensembles de données et les utiliser afin d'évaluer les opérateurs de fusion. Finalement, nous analysons l'efficacité des opérateurs de fusion en termes de temps de calcul et leur capacité à couvrir l'ontologie d'origine. Enfin, nous mettons en œuvre une expérience avec des ontologies pratiques pour évaluer les opérateurs de fusion.

Mots-clés: Fusion de Croyances, Fusion d'Ontologie, Logiques de Description, Domaine Ouvert, Conflits.

Abstract

The subject of the thesis is ontology merging, an approach for integrating various ontology sources into a unique one that handles emerging conflicts. This dissertation takes inspiration from the belief merging theory for merging ontologies to produce a consistent and unified knowledge base. We propose the three following contributions.

The first one is a semantic-based merging approach. In particular, a model-based merging strategy mainly focuses on handling semantic conflicts. A semantic conflict, which is not necessarily logical, is knowledge represented in many different or opposite ways. We also propose a formal model characterization and show the method's effectiveness with an experimental evaluation.

The second approach proposes a new framework to merge open-domain terminological knowledge. It leverages RCC5, a formalism for representing regions in a topological space and reasoning about their set-theoretic relationships. We propose a faithful translation of terminological knowledge from conflicting sources into region spaces. Here, we merge knowledge bases in this space and translate the outcome into the input sources' language. Our technique uses RCC5's expressivity and flexibility to deal with contradictory knowledge.

The last contribution is a framework for evaluating ontology merging operators. The primary strategy starts with an original ontology to create noisy ontologies as datasets and use them to assess the merging operators. Then, we analyze merging operators' computation time effectiveness and ability to cover the original ontology. Finally, we experimented with practical ontologies to evaluate the merging operators.

Keywords: Belief Merging, Ontology Merging, Description Logics, Open Domain, Conflicts.