UNIVERSITY OF CATANIA

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

DOCTOR OF PHILOSOPHY IN MATHEMATICS AND COMPUTER SCIENCE

Automated Reasoning via a Multi-sorted Fragment of Computable Set Theory with Applications to Semantic Web

Author:
Daniele Francesco Santamaria

Supervisors:
Prof. Domenico Cantone
Prof. Marianna Nicolosi Asmundo

XXXI CICLO
Settore Scientifico Disciplinare INF/01

Table of contents

List of Figures

List of Tables

Acknowledgements

The work proposed in this dissertation has been published as:

- Claudia Cantale, Domenico Cantone, Marianna Nicolosi-Asmundo and Daniele Francesco Santamaria, *Distant Reading Through Ontologies: The Case Study of Catania's Benedictines Monastery.* JLIS.it 8-3, pp. 203-219 (September 2017).

- Domenico Cantone, Marianna Nicolosi-Asmundo and Daniele Francesco Santamaria, *An optimized KE-tableau-based system for reasoning in the description logic* $\mathcal{DL}_{\mathbf{D}}^{4,\times}$. To App. in Proc. of RuleML+RR 2018, 2nd International Joint Conference on Rules and Reasoning 18-21 Sep 2018, Luxembourg.

- Domenico Cantone, Marianna Nicolosi-Asmundo and Daniele Francesco Santamaria, *A set-based reasoner for the description logic* $\mathcal{DL}_{\mathbf{D}}^{4,\times}$. To App. in Proc. of SETS 2018, 5 June 2018, Southampton, United Kingdom, CEUR Workshop Proceedings.

- Claudia Cantale, Domenico Cantone, Manuela Lupica Rinato, Marianna Nicolosi-Asmundo, Daniele Francesco Santamaria, *The Shape of a Benedictine Monastery: The SaintGall Ontology.* Proceedings of the third Joint Ontology Workshops (JOWO-17), Bolzano, Italy, 21-23 September 2017, CEUR Workshop Proceedings, ISSN 1613-0073, Vol. 2050.

- Domenico Cantone, Marianna Nicolosi-Asmundo and Daniele Francesco Santamaria, *A C++ reasoner for the description logic* $\mathcal{DL}_{\mathbf{D}}^{4,\times}$. Proceedings of CILC 2017, 26-28 September 2017, Naples, Italy, CEUR Workshop Proceedings, ISSN 1613-0073, Vol. 1949, pp. 276-280.

- Domenico Cantone, Marianna Nicolosi-Asmundo and Daniele Francesco Santamaria, *A set-theoretic approach to ABox reasoning services.* In Costantini S., Franconi E., Van Woensel W., Kontchakov R., Sadri F., Roman D. (eds) Rules and Reasoning. RuleML+RR 2017. Lecture Notes in Computer Science, vol 10364, pp. 87-102. Springer, Cham (2017).

- Domenico Cantone, Andrea Fornaia, Marianna Nicolosi-Asmundo, Daniele Francesco Santamaria and Emiliano Tramontana, *An OWL framework for rule-based recognition of places in Italian non-structured text.* Proc. of the 2nd International Workshop on Knowledge Discovery on the Web, KDWEB 2016, Cagliari, Italy, September 8-10, 2016, CEUR Workshop Proceedings, ISSN 1613-0073, Vol. 1748, Section I, Paper 8.

- Domenico Cantone, Marianna Nicolosi-Asmundo, Daniele Francesco Santamaria, *Conjunctive Query Answering via Fragment of Set Theory.* Proc. of the 17th Italian Conference on Theoretical Computer Science, ICTCS 2016, Lecce, Italy, September 7-9, 2016, CEUR Workshop Proceedings, ISSN 1613-0073, Vol. 1720, pp. 23-35.

- Domenico Cantone, Marianna Nicolosi-Asmundo, Daniele Francesco Santamaria and Francesca Trapani, *OntoCeramic: an OWL ontology for ceramics classification.* Proceedings of the 30th Italian Conference on Computational Logic, CILC 2015, Genova, Italy, July 1-3, 2015, CEUR Workshop Proceedings, ISSN 1613-0073, Vol. 1459, pp. 122-127.

- Domenico Cantone, Cristiano Longo, Marianna Nicolosi-Asmundo and Daniele Francesco Santamaria, *Web ontology representation and reasoning via fragments of set theory.* In Cate B., Mileo A. (eds) Web Reasoning and Rule Systems. RR 2015. Lecture Notes in Computer Science, vol 9209. Springer, Cham (2015).

- Claudia Cantale and Daniele Francesco Santamaria, *I Benedettini: un modello di rappresentazione delle informazioni per un modello di riuso dei beni architettonici.* Quinto Convegno Annuale dell' Associazione per l'Informatica Umanistica e la Cultura Digitale, AIUCD 2016, Venice, Italy, 7-9 September, 2016.

Abstract

Semantic Web is a vision of the web in which machine-readable data enables software agents to manipulate and query information on behalf of human agents. To achieve such goal, machines are provided with appropriate languages and tools. Investigating new technologies which can extend the power of knowledge representation and reasoning systems is the main task of this research work started by observing the lack of some desirable characteristics concerning the expressiveness of semantic web languages and their integration with some features of rule-based languages, arisen with the study of some application domains.

Specifically in this dissertation, we consider some results of *Computable Set Theory*, a research field rich of several interesting theoretical results, in particular for what concerns multi-sorted and multi-level syllogistic fragments, in order to provide a novel powerful knowledge representation and reasoning framework particularly devoted to the context of Semantic Web.

For this purpose, we use a syllogistic fragment of computable set theory called $\mathsf{4LQS^R}$, admitting variables of four sorts and a restricted form of quantification over variables of the first three sorts, to represent and reason on expressive decidable *description logics* (DLs) which can be used to represent ontological knowledge via Semantic Web technologies. We show that the fragment $\mathsf{4LQS^R}$ allows one to represent an expressive DL called $\mathcal{DL}(\mathbf{D})\langle\mathsf{4LQS^R}\rangle$ ($\mathcal{DL}_{\mathbf{D}}^{4}$ for short) and its extension $\mathcal{DL}^{\times}(\mathbf{D})\langle\mathsf{4LQS^R}\rangle$ ($\mathcal{DL}_{\mathbf{D}}^{4,\times}$ for short). The DL $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ admits, among other features, Boolean operations on concepts and roles, data types, and product of concepts. Moreover, $\mathsf{4LQS^R}$ provides a unique formalism which combines the features of the DL $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ with a rule language admitting full negation and subject to no safety condition. We show that $\mathsf{4LQS^R}$ can be used to represent a novel Web Ontology Language (OWL) 2 profile, and hence can be used as reasoning framework for a large family of ontologies. Then, we study the most widespread reasoning tasks concerning both $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-TBoxes and $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-ABoxes. In particular, we consider the *Consistency Problem* of a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-knowledge base (KB), the *Conjunctive Query Answering* (*CQA*) for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$, which provides simple mechanism allowing users and applications to interact with data stored in the KB, and the *Higher-Order Conjunctive Query Answering* (*HOCQA*) for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$, a generalization of the *CQA* problem admitting three sorts of variables, namely, individuals, concepts, and roles. The decidability of the above mentioned problems are proved by reducing them to analogous problems in the context of the set-theoretic language $\mathsf{4LQS^R}$ by means of a suitable mapping function from axioms and assertions of $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ to formulae of $\mathsf{4LQS^R}$.

Then, we provide a correct and terminating algorithm for the *HOCQA* problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ based on the KE-tableau system, a refutation system inspired to the Smullyan's semantic tableaux, giving also computational complexity results. The algorithm also serves for the consistency problem of $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KBs and for other reasoning tasks which the *HOCQA* problem can be instantiated to.

We also introduce an implementation in C++ of the algorithm to provide an effective reasoner for the DL $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ which admits ontology serialized in the OWL/XML format.

Investigating the efficiency of the resulting reasoner and its improvements is another goal of this dissertation. We study an optimization of the considered algorithm based on a variant of the KE-tableau which provides a generalization of the elimination rule incorporating a rule for treating universally quantified formulae. We also show that the implementation of the resulting system turns out to be more efficient than the previous system and, incidentally, than the implementation of the First-Order (FO) KE-tableau.

The benchmarking process is based on a huge amount of KBs of different sizes and kinds, constructed *ad hoc* just for the purpose of comparing the three systems, and on some real-world ontologies.

Developing semantic web ontologies which are mainly applied to address problems concerning the field of *Human and Social Sciences* is the main application domain of this dissertation.

Specifically, we introduce three ontologies. The first one called *OntoLocEstimation* has been conceived for recognizing location names in non-structured texts and to handle data and reason about them, even in presence of name ambiguities. *OntoLocEstimation* can be represented in the DL $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ and can be adopted for different application domains since it has been conceived as general purpose ontology.

The second ontology, called *ArchivioMuseoFabbrica* and representable in $\mathcal{DL}_{\mathbf{D}}^{4,\times}$, has been introduced to study the history of the renovation of the Benedictine Monastery of San Nicolò L'Arena in Catania by the architect Giancarlo De Carlo, the areas of the building and the people involved, and the social events occurred during such process of renovations which lasted 28 years (1977–2005). The ontology describes the Benedictine Monastery in its contemporary conformation as university campus, whereas its original design and meaning is framed by the last presented ontology called *SaintGall*.

The ontology *SaintGall* is intended to study the shape of the ideal Benedictine monastery and its peculiarities which identify and distinguish it from other religious buildings and which are analysed in the Saint Gall map, one of the most ancient descriptions of a Benedictine monastery. Therefore, the ontology can be considered a first step towards the definition of the architectonic type of monastery.

1 Introduction

1.1 Automated reasoning for Web Knowledge

Automated Reasoning concerns the capability of automatically concluding new facts (inferences) from what is already known by a predetermined set of assertions and by a formal description of the environment engrossing the elements involved in those statements. The term "automated reasoning" is commonly associated with "deductive reasoning" as meant in mathematics and logic and has been conceived to prove theorems of a certain calculus in an efficient manner. Although its original goal, the field is nowadays richly and deeply investigated to solve a huge variety of research problems in several contexts. Indeed, automated reasoning is essential for many web systems and applications, especially when huge quantities of data need to be collected, integrated, and automatically and independently processed. The outcomes of these tasks are relevant in several data exchange-centric contexts such as electronic commerce and financial, social web interaction, tracking and analytic, human and social sciences, and many other applications in which data cannot be processed and queried in a human interactive manner or in which relationships among data are very intertwined.

One of the main issue of automated reasoning tools when dealing with a specific problem is to identify the right deduction calculi which fits properly with the domain of interest. This task turns out to be unfeasible without the intervention of a computer scientist with logic backgrounds. For these reasons, scholars and researchers have been focus their interest to *Semantic Web*, a general-purpose framework of automated reasoning on the web based on linked data, which is recently emerging at a fast pace.

The main standard languages of the semantic web are the *Resource Description Framework* (RDF) [136], a variation on the entity-relationship schema language, and the *Web Ontology Language* (OWL) [131], an ontological language stemming from *Description Logics* (DLs) [3], which reached version 2.1 at the time of this writing. The *World Wide Web Consortium*[1] (W3C) identifies several OWL 2 profiles, the most widespread of which is OWL 2 DL [133], a profile conjoining high expressiveness and decidability. An OWL 2 profile is identified by the expressive power of the language underpinning it. For the sake of efficiency of reasoning and to suit different application scenarios, profiles are usually defined by restricting the expressive power of OWL 2 DL. For instance, OWL 2 RL [132] restricts the way in which constructs are used in order to define ontologies for rule-based reasoning systems, while still providing desirable

[1] https://www.w3c.org

computational guarantees.

Hence, the scientific community identified the DLs family as the principal deduction calculi for the semantic web. DLs in the semantic web ambit permit to successfully answer challenging questions from a variety of fields but it also presents several drawbacks. For example, DLs don't allow one to directly mix concepts and roles. This issue has been partially addressed by the introduction of the *Semantic Web Rule Language* (SWRL) [129], an extension of OWL to Horn-like rules, which however not only provides different syntax and semantics, thus splitting the formalism of the representation system, but it is also not robustly decidable, i.e, any tiny extension of a decidable fragment of SWRL can easily result in an undecidable combination [105]. Another major drawback of integrating SWRL with OWL is that, although the DL component is very expressive, the rule component is not, since it is restricted to *Datalog* [105]. Another major issue concerns the capability of DLs to deal with *meta-modeling*, i.e., to provide statements about statements. In facts, even if the integration of meta-modeling features in description logics have been studied (see [93]), their integration affects the semantics and no working implementation has been standardized. In addition, DLs do not in general provide reasoning with propositions qualified in terms of time and modality which are crucial in many web applications especially in reactive systems, i.e., systems that continuously interact with the environment and whose basic functionality is to maintain a certain behaviour of the environment [62].

One of the most important features of a knowledge representation system consists of the capability of users to interact with the (inferred) information provided by the knowledge base. The problem of querying DL knowledge bases has been mainly addressed by introducing the *Conjunctive Query Answering* problem, which admits a restricted form of first-order queries [64] using the logical conjunction operator. DL conjunctive queries are mainly limited by admitting query variables ranging over the set of individual of knowledge base, whereas higher order queries involving concept and role variables are defined on top of the DL formalism (i.e., exploiting the inducted knowledge graph).

Finally, to the best of our knowledge, implementations of the decision procedures for DLs suitable for parallel computing systems are unknown.

1.2 Related works

DLs as been introduced as reasoning framework for the semantic web in [7] and in [6] because they are conceived with the focus on tractable reasoning, they well fit for practical modeling purposes, and they enjoy interesting computational properties such as decidability. DLs have proved to be useful in a wide variety of applications such as federated databases and cooperative information systems [11] and in the configuration of complex telecommunications systems [9]. In particular, the DLs $\mathcal{SHIF}$, $\mathcal{SHOIN}$, $\mathcal{SHOIQ}$, and $\mathcal{SROIQ}$ [3, 69] have been adopted for knowledge web reasoning over the years from the foundation of semantic web.

The satisfiability problem for knowledge bases of these DLs can be solved using a sound and complete algorithm based on the tableau calculus [113, 35]. In order to solve inefficiency introduced by the existential quantification and by cardinality restrictions, many optimisation techniques such as the hyper-resolution calculus [96] are applied to these tableau-based algorithms. The hyper-tableau calculus permits a systematic branch saturation approach in which the proof procedure does not have to start with a new tableau from scratch once the resources are exhausted on the current tableau during iterative deepening, in a similar way to the resolution calculus [106].

In addition to the problem of satisfiability, one of the most investigated problems in Dls and inherited form database theory [122] is the *CQA* problem. Techniques to solve the *CQA* problem for DLs are mainly based on a sophisticated reduction to the satisfiability problem [107, 14, 73]. In these approaches a rewriting of the KB and of the query is required in order to verify whether the query is entailed by the knowledge base. The problem of defining queries, which admit variables ranging over the set of concepts and roles defined in the knowledge base, falls outside the definition of the CQA problem. The former problem is addressed by SPARQL [130], a SQL-style query language, which however operates outside the DL formalism, i.e, on the graph induced by the knowledge base.

Finally, the task of pushing the expressive limits of DLs while keeping decidability and implementability features by extending the DL formalism with rules has been deeply investigated. The main results which can be found in [85, 37, 59, 79, 90] lead to an extension of OWL to rules called SWRL language. The approach used in SWRL consists of introducing *DL rules* as an expressive new rule language which combines DLs with a limited form of first-order rules. DL-safe rules simply restrict the applicability of rules to a finite set of named individuals in order to retain decidability.

The idea of applying computable set theory in the context of semantic web knowledge has been presented in a previous work of our [108]. In facts, to best of our knowledge, this dissertation first adopts computable set theory as a means of representing and reasoning on semantic web knowledge even though computable set theory received an in-deep theoretical understanding.

Set theory is a branch of mathematical logic based on *sets*. Sets are basic elements for representing and reasoning on application domains in various fields of computer science, logic, and mathematics. In particular, *Computable Set Theory* is a research field started in the late seventies with the purpose of studying the decidability of the satisfiability problem for fragments of set theory.

Within the research field of computable set theory, a wealth of decidability results has been collected over the years [52, 20, 28, 111, 29]. The most efficient decision procedures designed in this area have been implemented within the set-based proof verifier Ætnanova/Referee [111] and constitute its inferential core.

Most of the decidability results and applications in computable set theory concerns one-sorted multi-level syllogistic, namely collections of formulae admitting variables of one sort only, which range over the von Neumann universe of sets. Only a few stratified syllogistics endowed with variables of multiple sorts have been investigated, despite the fact that in many fields of computer science and mathematics one often has to deal with multi-sorted languages. For instance, in description logics one has to consider entities of different types such as *individual elements*, *concepts*, namely sets of individuals, and *roles*, namely binary relations over elements.

In [51] an efficient decision procedure was presented for the satisfiability of the *Two-Level Syllogistic* language 2LS. 2LS admits variables of two sorts and propositional connectives together with the basic set-theoretic operators $\cup, \cap, \setminus$ and the predicate symbols $=, \in$, and $\subseteq$. Then, in [17], it was shown that the extension of 2LS with the singleton operator and the Cartesian product operator is decidable. Tarski's and Presburger's arithmetics extended with sets have been analysed in [19]. Subsequently, in [18], a three-sorted language 3LSSPU (Three-Level Syllogistic with Singleton, Powerset and general Union) has been proved decidable. In [25], it was shown that the language $\mathsf{3LQS}^{\mathsf{R}}$ (*Three-Level Quantified Syllogistic* with restricted quantifiers), admitting variables of three sorts and a restricted form of quantification, has a decidable satisfiability problem.

In the last decade, two-sorted multi-level fragments of set theory allowing one to express constructs related to *multi-valued maps* have been studied (see [22, 23, 21]) and applied in the realm of knowledge representation. In [24], for instance, an expressive description logic, called $\mathcal{DL}\langle\mathsf{MLSS}^{\times}_{2,m}\rangle$, has been introduced and the consistency problem for $\mathcal{DL}\langle\mathsf{MLSS}^{\times}_{2,m}\rangle$-knowledge bases has been proved to be **NP**-complete. $\mathcal{DL}\langle\mathsf{MLSS}^{\times}_{2,m}\rangle$ has been extended with additional description logic constructs and SWRL rules in [23], proving that the decision problem for the resulting description logic, called $\mathcal{DL}\langle\forall^{\pi}_{\mathbf{0,2}}\rangle$, is still **NP**-complete under suitable conditions. In [21], $\mathcal{DL}\langle\forall^{\pi}_{\mathbf{0,2}}\rangle$ has been extended with some *meta-modeling* features. In particular, it has been shown how to relax the distinction between concepts and individuals, still preserving the **NP**-completeness of the decision problem.

However, none of the above-mentioned description logics provides functionalities to deal with data types, a simple form of concrete domains that are relevant in real-world applications. In [55] an extension of the description logics fragment $\mathcal{ALC}$ has been represented via a very rudimentary axiomatic set theory Ω consisting of only four axioms characterizing binary union, set difference, inclusion, and the power-set.

Sets have also several applications in logic programming [43]. Numerous decidability results in this field can be found in [44, 46, 13, 45]. Finally, extension of programming language with a restricted form of intensional sets has been studied in [32, 33].

1.3 Automated reasoning via a fragment of set-theory

In this dissertation we investigate a novel framework for automated reasoning, in particular for the semantic web context, based on *Elementary Set Theory* which moves in the direction of providing a general and efficient deduction system allowing the desired features missing from DLs.

In [26], the authors have introduced the decidable four-level stratified fragment of set theory $\mathsf{4LQS^R}$, involving variables of four sorts, pair terms, and a restricted form of quantification over variables of the first three sorts. The satisfiability problem for formulae of $\mathsf{4LQS^R}$ has been proved decidable by showing that it enjoys a small model property. In addition, it has been also shown that the modal logic **K45** can be formalized in one of such collections, thus redetermining the **NP**-completeness of its decision problem [81].

The high expressiveness and potentiality of $\mathsf{4LQS^R}$ motivates the adoption of the fragment of computable set theory as a knowledge representation and reasoning framework for the semantic web, which we address in this dissertation. We carry the work of [108] on by formalizing the representation and reasoning system based on the set-theoretical fragment $\mathsf{4LQS^R}$, by addressing the main reasoning problems which are solved by suitable algorithms presented in this dissertation and whose complexity and efficiency is also analysed. In addition, we develop in C++ a novel reasoner [30] based on the above mentioned algorithms.

Specifically, we first identify an expressive description logic called $\mathcal{DL}(\mathbf{D})\langle\mathsf{4LQS^R}\rangle$ (more simply referred to as $\mathcal{DL}^4_{\mathbf{D}}$), which can be represented in $\mathsf{4LQS^R}$. An extension of $\mathcal{DL}(\mathbf{D})\langle\mathsf{4LQS^R}\rangle$ will be discussed next.

The DL $\mathcal{DL}^4_{\mathbf{D}}$ supports data types and admits concept constructs such as full negation, union and intersection of concepts, concept domain and range, Boolean operations on concepts, existential quantification and minimum cardinality restriction on the left-hand-side of inclusion axioms, and universal quantification and maximum cardinality restriction on the right-hand-side of inclusion axioms. The DL $\mathcal{DL}_{\mathbf{D}}^{4}$ also supports role constructs such as role chains on the left-hand-side of inclusion axioms, union, intersection, and complement of roles, as well as properties on roles such as transitivity, symmetry, reflexivity, and irreflexivity. We prove that the consistency problem for $\mathcal{DL}_{\mathbf{D}}^{4}$-knowledge bases (KBs) is decidable via a reduction to the satisfiability problem for formulae of $\mathsf{4LQS^R}$ through an appropriate translation process.

We also show that the consistency problem for $\mathcal{DL}_{\mathbf{D}}^{4}$-KBs involving only suitably constrained $\mathcal{DL}_{\mathbf{D}}^{4}$-formulae is **NP**-complete. Such constraints are not particularly restrictive: in fact, it turns out that the constrained logic allows one to represent real world ontologies such as *Ontoceramic*, designed for ancient ceramic cataloguing ([27, 108]), *Ontoluoghi*, which is used for recognizing location names in non-structured texts, and *ArchivioMuseoFabbrica*, an ontology concerning the history of the renovation of the Benedictine Monastery of San Nicolò l'Arena in Catania. These ontologies are some of the outcomes of investigating on the practical aspects of the semantic web in the human and social sciences field.

Some considerations on the expressiveness of $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ are in order. The logic $\mathcal{DL}_{\mathbf{D}}^{4}$ does not extend $\mathcal{SROIQ}(\mathbf{D})$, the description logic upon which the standard OWL 2 DL profile is based, as $\mathcal{DL}_{\mathbf{D}}^{4}$ admits existential (resp., universal) quantification only on the left-hand (resp., right-hand) side of inclusion axioms. However, $\mathcal{DL}_{\mathbf{D}}^{4}$ allows one to express chain axioms not supported by $\mathcal{SROIQ}(\mathbf{D})$, as they can involve roles that are not subject to any regularity restriction. Moreover, Boolean combinations of roles are admitted even on the right-hand-side of chain axioms. The latter fact is particularly relevant to the problem of expressing rules in OWL. In particular, expressiveness of SWRL is framed in $\mathsf{4LQS^R}$.

Then, we exploit the fragment $\mathsf{4LQS^R}$ to represent the description logic $\mathcal{DL}^{\times}(\mathbf{D})\langle\mathsf{4LQS^R}\rangle$ ($\mathcal{DL}_{\mathbf{D}}^{4,\times}$, for short) which extends the logic $\mathcal{DL}_{\mathbf{D}}^{4}$ with Boolean operations on concrete roles and with the product of concepts. We prove that, as in the case of $\mathcal{DL}_{\mathbf{D}}^{4}$, under suitable not very restrictive constraints, the consistency problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KBs is **NP**-complete.

Next, we address the problem of *Conjunctive Query Answering* (*CQA*) for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ (and hence for $\mathcal{DL}_{\mathbf{D}}^{4}$), one of the most simple and natural ways to interact with digital knowledge. The result is obtained by formalizing $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KBs and $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive queries in terms of formulae of the set-theoretic fragment $\mathsf{4LQS^R}$, through a decision procedure for the satisfiability problem for $\mathsf{4LQS^R}$. We further define a KE-tableau-based procedure for the same problem, more suitable for implementation purposes, and we analyse its computational complexity. The procedure also serves as a decision procedure for the consistency problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KBs. The choice of the KE-tableau-system instead of the traditional Smullyan's semantic tableaux is motivated by the fact that KE-tableau-based systems [35] construct tableaux whose distinct branches define mutually exclusive situations, thus preventing the proliferation of redundant branches, typical of semantic tableaux. The KE-tableau-system turns out to be a crucial feature since it allows one to define deterministic decision procedures.

Then, we investigate and define the *Higher-Order Conjunctive Query Answering* (*HOCQA*) problem, a generalization of the *CQA* problem which admits variables of three sorts and can be instantiated to the most widespread reasoning tasks for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-ABox, such as *instance checking* and *concept retrieval*. Decidability of the *HOCQA* problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ is established via a reduction to the satisfiability problem for the set-theoretic fragment $\mathsf{4LQS^R}$. We also solve the *HOCQA* problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ by generalizing the KE-tableau-based procedure introduced for the *CQA* problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$.

Furthermore, we illustrate an implementation of the procedures for the consistency problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KBs and for the *HOCQA* problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$. Specifically, we describe a C++ reasoner for the $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-TBox and $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-ABox reasoning tasks that supports $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KBs serialized in the OWL/XML format and admitting SWRL rules. Concerning $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-TBox reasoning tasks, the reasoner checks the consistency of $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KBs represented in terms of the set-theoretic fragment $\mathsf{4LQS^R}$, whereas, concerning $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-ABox reasoning tasks, the reasoner computes the $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-higher-order answer set of a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-higher-order conjunctive query with respect to a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KB.

Finally, we also modify the KE-tableau-based procedure mentioned above by eliminating the preprocessing phase introduced for universally quantified formulae and replacing the standard KE-elimination rule with a novel elimination rule, called E^γ-Rule, incorporating the standard rule for treating universally quantified formulae (γ-rule). The resulting system turns out to be more efficient than the KE-system and the First-Order (FO) KE-system in [36] as shown by suitable benchmarking tests executed on the C++ implementations of the three mentioned systems. The main reason for such a speed-up relies on the fact that the novel E^γ-Rule does not need to store the instances of universally quantified formulae on the KE-tableau.

1.4 Automated reasoning in Digital Humanities

Digital Humanities concerns the critical study of how computer science potentially affects humanities scholarship and scholarly communication. Basically, it investigates the use of automatic computation to solve humanities research questions and how it can enhance disciplines such as *Art History*, *Classical Studies*, *History*, *Literature*, *Music* and many others. Even though, computer science is commonly associated with humanities by a mere act of digital storing and of brute computation, it not only allows scholars to approach old problems with new means but also to ask new questions that could not have been asked with the traditional means of humanistic enquiry [100]. Since the advantages of computer science in humanities cannot be exhaustively addressed here and falls outside the scope of this work, the reader is referred to [110] for a good overview. In particular, the advantages of ontologies in cultural and social sciences are addressed in [125], whereas in architectural design in [50, 66, 60, 116]. In [41], the authors proposed an ontology which provides definitions and a formal structure for describing the implicit and explicit concepts and relationships used in cultural heritage documentation. An ontology for linguistic and cognitive engineering called *DOLCE* has been introduced in [54]. A set of vocabulary terms that can be used to describe digital resources as well as physical resources has been presented in [126]. An ontology for the digital development and publication of archival heritage of the Italian archival system *Sistema Archivistico Nazionale (SAN)* of the *Istituto centrale per gli Archivi (ICAR)* of *Ministero dei beni e delle attività culturali e del turismo (MIBACT)* is available at [74]. Our approach to the problem of ontological classification of pottery can be found in [27].

In this dissertation, we introduce several OWL ontologies in order to address some problems concerning topics in the field of human and social sciences, which is the main application domain of the investigation efforts of this work.

The first ontology, called Ontoluoghi and expressible in $\mathcal{DL}_{\mathbf{D}}^{4,\times}$, has been introduced for recognizing location names in non-structured texts and for handling such information and reasoning on it, even in presence of name ambiguities. The ontology serves as middle-ware for techniques introduced for recognizing location names in non-structured texts based on grammar rules devised for the Italian language. The problem of name recognition has been usually approached mainly on top of the English language, thus leading to a difficult generalization of the technique which produces unsatisfactory results when applied to Italian places mentioned in Italian texts.

The second one, called ArchivioMuseoFabbrica, is an OWL ontology expressible in $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ concerning the history of the renovation of the Benedictine monastery of San Nicolò l'Arena in Catania by the architect Giancarlo De Carlo [38, 89, 15]. Specifically, we consider for the purpose of modeling the ontology, a wide subset of public and private documents collected from 1977 to 2006 during the process of restoration and adaptation of the monastery to a campus for the university of Catania. The task of modeling and populating the ontology has been carried out from the analysis of documents stored in the *Archivio del Museo della Fabbrica*, in the new archive of professor Giuseppe Giarrizzo, in the private collection of Antonino Leonardi, and from the conceptual map of the locations of the monastery. To the best of our knowledge, this is the first effort of ontological modeling the renovation and refurbishment process of a (historical) building.

The third ontology is an OWL 2 ontology representing the Saint Gall plan, one of the most ancient documents arrived intact to us, which describes the ideal model of a Benedictine monastic complex which inspired the design of many European monasteries. Such work can be considered as an enrichment of ArchivioMuseoFabbrica. The ontology SaintGall modeling the structural, functional, and architectural specification of an ideal Benedictine monastery allows one to analyse the monastery of San Nicolò L'Arena in Catania as it was originally intended before the renovation and re-adaptation as university campus in the Seventies. In order to provide support to the analysis and to the actions concerning the monastery architectonic type, it was deemed appropriate to model the ontology of an ideal monastery such as the one described in the Saint Gall map. The ontology can be potentially extended/specialized, on a case-by-case basis, by providing the specification of the considered monastery. Indeed, the ontology introduced in this dissertation can be considered as a general support tool for all the refurbishment, recovery, and change-in-use actions concerning a Benedictine monastery, or more generally, the the architectural type of Monastery.

Since the ontology makes a wide usage of OWL 2 existential and universal quantification restrictions, it cannot be expressed in $\mathcal{DL}_{\mathbf{D}}^{4,\times}$. The task of modifying the set-theoretic fragment underpinning $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ so as to include a restricted version of the operator of relational composition will be addressed in future works.

The first ontology has been developed in 2016 by the joint effort with Andrea Fornaia, at that time a Ph.D. student in *Mathematics and Computer Science* at the *University of Catania*, whereas the latter two ontologies have been developed thanks to a three-year collaboration with Claudia Cantale which was at the time of cooperation a Ph.D. student in *Social Theory, Communication and Media* at the *University of Catania*. A meaningful contribution to the latter ontology was given by Dr. Maria Rossella Stufano Melone, postdoctoral research fellow at *DICATECh - Department of Civil, Environmental, Building Engineering, and Chemistry* of the *Politecnico di Bari*.

1.5 Organization of the dissertation

The dissertation is organized as follows.

In Chapter 2 we give the main notions useful to understand the results presented in the dissertation. Specifically, we introduce the description logic underpinning the language OWL 2, called $\mathcal{SROIQ}(\mathbf{D})$, and we describe the main features of OWL 2 DL and of other standard OWL 2 profiles. Then, we provide the syntax and semantics of the set-theoretic language $\mathsf{4LQS^R}$ and the effective class of $\mathsf{4LQS^R}$-formulae, called $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$, adopted for the purpose of this work and *de facto* involved in the set-theoretic representation of the DLs $\mathcal{DL}_{\mathbf{D}}^{4}$ and $\mathcal{DL}_{\mathbf{D}}^{4,\times}$.

In Chapter 3 we illustrate the description logic $\mathcal{DL}_{\mathbf{D}}^{4}$ and its extension $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ expressible in $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$. We prove that the decidability of the consistency problem for $\mathcal{DL}_{\mathbf{D}}^{4}$- and for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KBs can be reduced to the satisfiability problem for $\mathsf{4LQS^R}$-formulae.

In Chapter 4 we present the most widespread reasoning tasks for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ and $\mathcal{DL}_{\mathbf{D}}^{4}$, as well as the algorithms designed to solve them. Specifically, we introduce the *CQA* and the *HOCQA* problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ and we provide the algorithms for the *CQA* and the *HOCQA* problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ which also serve for the consistency problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KBs. We also provide the proofs of correctness and termination of the above mentioned procedures and give some complexity results. Finally, we introduce an efficient variant of the algorithm for the consistency problem of $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KBs and for the *HOCQA* problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$.

In Chapter 5 we describe the C++ implementation of the algorithms mentioned above, providing also some benchmark tests.

In Chapter 6 we show the ontologies that we use to address several problems concerning human and social sciences. Specifically, we introduce the ontology *Ontoluoghi* for the recognition of location names in non-structured texts, the ontology *ArchivioMuseoFabbrica* concerning the history of the renovation of the Benedictine Monastery of San Nicolò L'Arena in Catania, and the ontology *SaintGall* concerning the specifications of an ideal Benedictine monastery.

Finally, in Chapter 7 we draw our conclusions and give some hints towards future developments.

2 Preliminaries

In this chapter we will introduce notions and notations useful for the comprehension of the results of this dissertation. Specifically, in Section 2.1 we will introduce the description logic $\mathcal{SROIQ}(\mathbf{D})$ upon which the main ontological language for the semantic web OWL 2 is founded. The OWL 2 language and the standard OWL 2 profiles will be presented in Section 2.2.

Finally, in Section 2.3 we will recall the set-theoretical fragment $\mathsf{4LQS}^{\mathsf{R}}$ exploited in this dissertation to define a reasoning framework for the semantic web. In the same section We will also discuss the expressiveness of $\mathsf{4LQS}^{\mathsf{R}}$ and present $\mathsf{4LQS}^{R}_{\mathcal{DL}^{4,\times}_{\mathbf{D}}}$, the subclass of formulae of $\mathsf{4LQS}^{\mathsf{R}}$ *de facto* involved in the implementation of the reasoner.

2.1 Description Logics

Description logics (DLs) are a family of formalisms widely used in the field of *Knowledge Representation* to model application domains (known as terminological knowledge) and to reason on them [6]. DLs include light-weight logics with feasible inference problems as well as very expressive formalisms for which reasoning is undecidable. DL-knowledge bases (KBs) describe models that are based on individual elements (or, more simply, individuals), classes whose elements are individuals, and binary relationships among individuals. These three types of semantic entities are syntactically denoted by means of individual names, concept names, and role names. In addition, DLs provide operators for combining concept and role names into complex concept and role expressions. Details on DLs can be found in [6, 5].

One of the leading application domains for DLs is the *Semantic Web* [6]. In fact, the most recently developed semantic web language *Web Ontology Language* (OWL 2) is based on a very expressive DL admitting data types called $\mathcal{SROIQ}(\mathbf{D})$ [69].

Data types are a limited form of concrete domain particularly important in several applications such as OWL 2. The W3C lists most *XML Schema* data types as normative [135], including facets for restricting the range of built-in data types. For example, the *minExclusive* facet can be applied to *xsd:integer* to obtain a subset of integers larger than a particular value. Extensions of DLs with concrete domains and data types have received in-depth theoretical treatment [4, 61, 72, 87, 86]. Furthermore, data type groups [103] provide an architecture for integrating different data types, and the OWL-Eu approach [104] provides a way to restrict data types using expressions.

Data type reasoning can be performed using an external data type checking procedure. Informally, a data type checker verifies if the usage of a data type value conforms to the specifications of the data type it is instance of. Standard DL-tableau calculi can be extended to handle data types, by invoking the data type checker as an oracle. These results assume that data type reasoning can be performed using an external data type checking procedure. In particular, in [94] it has been shown that data type checking in OWL 2 is **NP**-hard in the general case, but may become trivial in many (hopefully typical) cases. The authors also argue that certain data types listed as normative in the current OWL 2 Working Draft may be unsuitable from both a modeling and an implementation perspective. They also present a modular data type checking algorithm that can support any data type for which it is possible to implement a small set of basic operations and discuss how to implement data type checkers for number and string data types.

2.1.1 The description logic $\mathcal{SROIQ}(\mathbf{D})$.

$\mathcal{SROIQ}(\mathbf{D})$ [69, 75] is the description logic underlying the most expressive OWL 2 profile, namely OWL 2 DL. $\mathcal{SROIQ}(\mathbf{D})$ is an extension of the DL $\mathcal{ALC}$ with transitive roles ($\mathcal{S}$), complex role inclusion axioms, reflexivity and and local reflexivity ($\exists R.Self$), irreflexivity, role disjointness ($\mathcal{R}$), nominals ($\mathcal{O}$), inverse properties ($\mathcal{I}$), qualified cardinality restrictions ($\mathcal{Q}$), and data types ($\mathbf{D}$). $\mathcal{SROIQ}(\mathbf{D})$ also admits asymmetric roles, negated role assertions, and the universal role U.

In order to introduce the syntax and the semantics of the DL $\mathcal{SROIQ}(\mathbf{D})$, it is convenient to introduce the definition of data type. Data types are defined according to [94] as follows.

Let $\mathbf{D} = (N_D, N_C, N_F, \cdot^{\mathbf{D}})$ be a *data type map*, where N_D is a finite set of data types, N_C is a map assigning a set of constants $N_C(d)$ to each data type $d \in N_D$, N_F is a map assigning a set of facets $N_F(d)$ to each $d \in N_D$, and $\cdot^{\mathbf{D}}$ is a map assigning

(i) a data type interpretation $d^{\mathbf{D}}$ to each data type $d \in N_D$,

(ii) a facet interpretation $f^{\mathbf{D}} \subseteq d^{\mathbf{D}}$ to each facet $f \in N_F(d)$, and

(iii) a data value $e_d^{\mathbf{D}} \in d^{\mathbf{D}}$ to every constant $e_d \in N_C(d)$.

We shall assume that the interpretations of the data types in N_D are non-empty pairwise disjoint sets.

A *facet expression* for a data type $d \in N_D$ is a formula ψ_d constructed from the elements of $N_F(d) \cup \{\top_d, \bot_d\}$ by applying a finite number of times the connectives $\neg$, $\wedge$, and $\vee$. The function $\cdot^{\mathbf{D}}$ is extended to facet expressions for $d \in N_D$ by putting, for $f, f_1, f_2 \in N_F(d)$,

- $\top_d^{\mathbf{D}} := d^{\mathbf{D}}$,
- $\bot_d^{\mathbf{D}} := \emptyset$,
- $(\neg f)^{\mathbf{D}} := d^{\mathbf{D}} \setminus f^{\mathbf{D}}$,
- $(f_1 \wedge f_2)^{\mathbf{D}} := f_1^{\mathbf{D}} \cap f_2^{\mathbf{D}}$,
- $(f_1 \vee f_2)^{\mathbf{D}} := f_1^{\mathbf{D}} \cup f_2^{\mathbf{D}}$.

A *data range dr* for $\mathbf{D}$ is either a data type $d \in N_D$, or a finite enumeration of data type constants $\{e_{d_1}, \ldots, e_{d_n}\}$, with $e_{d_i} \in N_C(d_i)$ and $d_i \in N_D$, or a facet expression ψ_d, for $d \in N_D$.

Let $\mathbf{R_A}$, $\mathbf{R_D}$, $\mathbf{C}$, $\mathbf{Ind}$ be denumerable pairwise disjoint sets of *abstract role names*, *concrete role names*, *concept names*, and *individual names*, respectively. The set of abstract roles is defined as $\mathbf{R_A} \cup \{R^- \mid R \in \mathbf{R_A}\} \cup \{U\}$, where U is the universal role and R^- is the inverse role of R.

A *role inclusion axiom* (RIA) is an expression of the form $w \sqsubseteq R$, where w is a finite string of roles not including the universal role U and R is an abstract role name distinct from the universal role U. An *abstract role hierarchy* R_a^H is a finite collection of RIAs. A strict partial order $\prec$ on a set A is an irreflexive and transitive relation on A. A strict partial order $\prec$ on $\mathbf{R_A} \cup \{R^- \mid R \in \mathbf{R_A}\}$ is called *a regular order* if $\prec$ satisfies, additionally, $S \prec R$ iff $S^- \prec R$, for all roles R and S.

A RIA $w \sqsubseteq R$ is $\prec$*-regular* if R is an abstract role name, and one of the following conditions holds:

1. $w = RR$,
2. $w = R^-$,
3. $w = S_1 \ldots S_n$ and $S_i \prec R$, for all $1 \le i \le n$,
4. $w = RS_1 \ldots S_n$ and $S_i \prec R$, for all $1 \le i \le n$,
5. $w = S_1 \ldots S_n R$ and $S_i \prec R$, for all $1 \le i \le n$,

where $S_1, \ldots, S_n$ are abstract role names.

An abstract role hierarchy R_a^H is regular if there exists a regular order $\prec$ such that each RIA in R_a^H is $\prec$-regular.

A *concrete role hierarchy* $\mathsf{R}_{\mathbf{D}}^H$ is a finite collection of concrete role inclusion axioms $T_i \sqsubseteq T_j$, where $T_i, T_j \in \mathbf{R_D}$.

A *role assertion* is an expression of one of the following types:

$$\mathsf{Ref}(R),\quad \mathsf{Irref}(R),\quad \mathsf{Sym}(R),\quad \mathsf{Asym}(R),\quad \mathsf{Tra}(R),\quad \mathsf{Dis}(R,S),$$

where $R, S \in \mathbf{R_A} \cup \{R^- \mid R \in \mathbf{R_A}\}$.

Given an abstract role hierarchy R_a^H and a set of role assertions R^A without transitivity or symmetry assertions ($\mathsf{Sym}(R)$ can be represented by a RIA of type $R^- \sqsubseteq R$ and $\mathsf{Tra}(R)$ by $R \circ R \sqsubseteq R$), the set of roles that are *simple* in $\mathsf{R}_a^H \cup \mathsf{R}^A$ is inductively defined as follows:

(i) a role name is simple if it does not occur on the right-hand-side of a RIA in R_a^H,

(ii) an inverse role R^- is simple if R is, and

(iii) if R occurs on the right hand of a RIA in R_a^H, then R is simple if, for each $w \sqsubseteq R \in \mathsf{R}_a^H$, $w = S$, for a simple role S.

A set of role assertions R^A is called simple if all roles R, S appearing in role assertions of the form $\mathsf{Irref}(R)$, $\mathsf{Asym}(R)$, or $\mathsf{Dis}(R,S)$ are simple in R^A.

Informally, the *ABox* is the assertional part of the knowledge base used to describe concrete situation by stating properties of individuals (just as the data in database). The *TBox* is the terminological part of the knowledge base, that is the vocabulary of the application domain. In addition to an *ABox* and a *TBox*, $\mathcal{SROIQ}(\mathbf{D})$ provides an *RBox* that includes all the statements concerning roles.

An $\mathcal{SROIQ}(\mathbf{D})$-*RBox* is a set $\mathsf{R} = \mathsf{R}_a^H \cup \mathsf{R}_\mathbf{D}^H \cup \mathsf{R}^A$ such that R_a^H is a regular abstract role hierarchy, $\mathsf{R}_\mathbf{D}^H$ is a concrete role hierarchy, and R^A is a finite simple set of role assertions. A formal definition of regular abstract role hierarchy can be found in [69].

Before introducing the formal definitions of *TBox* and of *ABox*, we define the set of $\mathcal{SROIQ}(\mathbf{D})$-concepts as the smallest set such that:

(a) every concept name and the constants $\top$, $\bot$ are concepts,

(b) if C, D are concepts, R is an abstract role (possibly inverse), S is a simple role (possibly inverse), T is a concrete role, dr is a data range for $\mathbf{D}$, a is an individual, and n is a non-negative integer, then

$$\begin{array}{ccccc} C \sqcap D, & C \sqcup D, & \neg C, & \{a\}, & \forall R.C, \exists R.C, \\ \exists S.Self, & \forall T.dr, & \exists T.dr, & \geq nS.C, & \leq nS.C \end{array}$$

are also concepts.

A *General Concept Inclusion* (GCI) axiom is an expression $C \sqsubseteq D$, where C and D are $\mathcal{SROIQ}(\mathbf{D})$-concepts. An $\mathcal{SROIQ}(\mathbf{D})$-*TBox* $\mathcal{T}$ is a finite set of CGIs.

An *individual assertion* is any expression of one of the following forms:

$$a : C, \quad (a, b) : R, \quad (a, e_d) : T, \quad (a, b) : \neg R, \quad (a, e_d) : \neg T, \quad a = b, \quad a \neq b,$$

where a, b are individuals, e_d is a constant in $N_C(d)$, R is a (possibly) inverse abstract role, P is a concrete role, and C is a concept. An $\mathcal{SROIQ}(\mathbf{D})$-*ABox* $\mathcal{A}$ is a finite set of individual assertions.

An $\mathcal{SROIQ}(\mathbf{D})$-KB is a triple $\mathcal{K} = (\mathcal{R}, \mathcal{T}, \mathcal{A})$ such that $\mathcal{R}$ is an $\mathcal{SROIQ}(\mathbf{D})$-*RBox*, $\mathcal{T}$ an $\mathcal{SROIQ}(\mathbf{D})$-*TBox*, and $\mathcal{A}$ an $\mathcal{SROIQ}(\mathbf{D})$-*ABox*. The semantics of $\mathcal{SROIQ}(\mathbf{D})$ is given by means of an interpretation $\mathbf{I} = (\Delta^{\mathbf{I}}, \Delta_{\mathbf{D}}, \cdot^{\mathbf{I}})$, where $\Delta^{\mathbf{I}}$ and $\Delta_{\mathbf{D}}$ are non-empty disjoint domains such that $d^{\mathbf{D}} \subseteq \Delta_{\mathbf{D}}$, for every $d \in N_D$, and $\cdot^{\mathbf{I}}$ is an interpretation function.

The interpretation of concepts and roles, axioms and assertions is defined in Table 1.

Name	Syntax	Semantics
concept	A	$A^{\mathbf{I}} \subseteq \Delta^{\mathbf{I}}$
abstract role	R	$R^{\mathbf{I}} \subseteq \Delta^{\mathbf{I}} \times \Delta^{\mathbf{I}}$
concrete role	T	$T^{\mathbf{I}} \subseteq \Delta^{\mathbf{I}} \times \Delta_{\mathbf{D}}$
individual	a	$a^{\mathbf{I}} \in \Delta^{\mathbf{I}}$
data type constant	e_d	$e_d^{\mathbf{D}} \in d^{\mathbf{D}}$
nominal	$\{a\}$	$\{a\}^{\mathbf{I}} = \{a^{\mathbf{I}}\}$
data type	d	$d^{\mathbf{D}} \subseteq \Delta_{\mathbf{D}}$
negated data type	$\neg d$	$\Delta_{\mathbf{D}} \setminus d^{\mathbf{D}}$
data range	$\{e_{d_1}, \ldots, e_{d_n}\}$	$\{e_{d_1}, \ldots, e_{d_n}\}^{\mathbf{D}} = \{e_{d_1}^{\mathbf{D}}\} \cup \ldots \cup \{e_{d_n}^{\mathbf{D}}\}$
data range	ψ_d	$\psi_d^{\mathbf{D}}$
data range	$\neg dr$	$\Delta_{\mathbf{D}} \setminus dr^{\mathbf{D}}$
top	$\top$	$\Delta^{\mathbf{I}}$
bottom	$\perp$	$\emptyset$
negation	$\neg C$	$(\neg C)^{\mathbf{I}} = \Delta^{\mathbf{I}} \setminus C$
conjunction	$C \sqcap D$	$(C \sqcap D)^{\mathbf{I}} = C^{\mathbf{I}} \cap D^{\mathbf{I}}$
disjunction	$C \sqcup D$	$(C \sqcup D)^{\mathbf{I}} = C^{\mathbf{I}} \cup D^{\mathbf{I}}$
universal restriction	$\forall R.C$	$(\forall R.C)^{\mathbf{I}} = \{x \in \Delta^{\mathbf{I}} : \forall y \in \Delta^{\mathbf{I}}. \langle x, y \rangle \in R^{\mathbf{I}} \rightarrow y \in C^{\mathbf{I}}\}$
existential restriction	$\exists R.C$	$(\exists R.C)^{\mathbf{I}} = \{x \in \Delta^{\mathbf{I}} : \exists y \in C^{\mathbf{I}}. \langle x, y \rangle \in R^{\mathbf{I}}\}$
self concept	$\exists R.Self$	$(\exists R.Self)^{\mathbf{I}} = \{x \in \Delta^{\mathbf{I}} : \langle x, x \rangle \in R^{\mathbf{I}}\}$

data type exists	$\exists P.dr$	$(\exists P.dr)^{\mathbf{I}} = \{x \in \Delta^{\mathbf{I}} : \exists y \in dr^{\mathbf{D}}.\langle x,y\rangle \in P^{\mathbf{I}}\}$
data type value	$\forall P.dr$	$(\forall P.dr)^{\mathbf{I}}=\{x \in \Delta^{\mathbf{I}} : \forall y \in \Delta_{\mathbf{D}}.\langle x,y\rangle \in P^{\mathbf{I}} \rightarrow y \in dr^{\mathbf{D}}\}$
qualified number restriction	$\leq_n R.C$	$(\leq_n R.C)^{\mathbf{I}} = \{x \in \Delta^{\mathbf{I}} : \|\{y \in C^{\mathbf{I}} : \langle x,y\rangle \in R^{\mathbf{I}}\}\| \leq n\}$
qualified number restriction	$\geq_n R.C$	$(\geq_n R.C)^{\mathbf{I}} = \{x \in \Delta^{\mathbf{I}} : \|\{y \in C^{\mathbf{I}} : \langle x,y\rangle \in R^{\mathbf{I}}\}\| \geq n\}$
qualified data type number restriction	$\leq_n P.dr$	$(\leq_n P.dr)^{\mathbf{I}} = \{x \in \Delta^{\mathbf{I}} : \|\{y \in dr^{\mathbf{D}} : \langle x,y\rangle \in P^{\mathbf{I}}\}\| \leq n\}$
qualified data type number restriction	$\geq_n P.dr$	$(\geq_n P.dr)^{\mathbf{I}} = \{x \in \Delta^{\mathbf{I}} : \|\{y \in dr^{\mathbf{D}} : \langle x,y\rangle \in P^{\mathbf{I}}\}\| \geq n\}$
nominals	$\{a_1,\ldots,a_n\}$	$\{a_1,\ldots,a_n\}^{\mathbf{I}} = \{a_1^{\mathbf{I}}\} \cup \ldots \cup \{a_n^{\mathbf{I}}\}$
universal role	U	$(U)^{\mathbf{I}} = \Delta^{\mathbf{I}} \times \Delta^{\mathbf{I}}$
inverse role	R^-	$(R^-)^{\mathbf{I}} = \{\langle y,x\rangle \mid \langle x,y\rangle \in R^{\mathbf{I}}\}$
concept subsumption	$C_1 \sqsubseteq C_2$	$\mathbf{I} \models_{\mathbf{D}} C_1 \sqsubseteq C_2 \iff C_1^{\mathbf{I}} \subseteq C_2^{\mathbf{I}}$
abstract role subsumption	$R_1 \sqsubseteq R_2$	$\mathbf{I} \models_{\mathbf{D}} R_1 \sqsubseteq R_2 \iff R_1^{\mathbf{I}} \subseteq R_2^{\mathbf{I}}$
role inclusion axiom	$S_1 \ldots S_n \sqsubseteq R$	$\mathbf{I} \models_{\mathbf{D}} S_1 \ldots S_n \sqsubseteq R \iff S_1^{\mathbf{I}} \circ \ldots \circ S_n^{\mathbf{I}} \subseteq R^{\mathbf{I}}$
concrete role subsumption	$T_1 \sqsubseteq T_2$	$\mathbf{I} \models_{\mathbf{D}} T_1 \sqsubseteq T_2 \iff T_1^{\mathbf{I}} \subseteq T_1^{\mathbf{I}}$
symmetric abstract role	$\mathsf{Sym}(R)$	$\mathbf{I} \models_{\mathbf{D}} \mathsf{Sym}(R) \iff (R^-)^{\mathbf{I}} \subseteq R^{\mathbf{I}}$
asymmetric abstract role	$\mathsf{Asym}(R)$	$\mathbf{I} \models_{\mathbf{D}} \mathsf{Asym}(R) \iff R^{\mathbf{I}} \cap (R^-)^{\mathbf{I}} = \emptyset$
transitive abstract role	$\mathsf{Tra}(R)$	$\mathbf{I} \models_{\mathbf{D}} \mathsf{Tra}(R) \iff R^{\mathbf{I}} \circ R^{\mathbf{I}} \subseteq R^{\mathbf{I}}$
disjoint abstract role	$\mathsf{Dis}(R_1,R_2)$	$\mathbf{I} \models_{\mathbf{D}} \mathsf{Dis}(R_1,R_2) \iff R_1^{\mathbf{I}} \cap R_2^{\mathbf{I}} = \emptyset$
disjoint concrete role	$\mathsf{Dis}(P_1,P_2)$	$\mathbf{I} \models_{\mathbf{D}} \mathsf{Dis}(P_1,P_2) \iff P_1^{\mathbf{I}} \cap P_2^{\mathbf{I}} = \emptyset$
reflexive abstract role	$\mathsf{Ref}(R)$	$\mathbf{I} \models_{\mathbf{D}} \mathsf{Ref}(R) \iff \{\langle x,x\rangle \mid x \in \Delta^{\mathbf{I}}\} \subseteq R^{\mathbf{I}}$
irreflexive abstract role	$\mathsf{Irref}(R)$	$\mathbf{I} \models_{\mathbf{D}} \mathsf{Irref}(R) \iff R^{\mathbf{I}} \cap \{\langle x,x\rangle \mid x \in \Delta^{\mathbf{I}}\} = \emptyset$
functional abstract role	$\mathsf{Fun}(R)$	$\mathbf{I} \models_{\mathbf{D}} \mathsf{Fun}(R) \iff (R^-)^{\mathbf{I}} \circ R^{\mathbf{I}} \subseteq \{\langle x,x\rangle \mid x \in \Delta^{\mathbf{I}}\}$
functional concrete role	$\mathsf{Fun}(P)$	$\mathbf{I} \models_{\mathbf{D}} \mathsf{Fun}(P) \iff \langle x,y\rangle \in P^{\mathbf{I}}$ and $\langle x,z\rangle \in P^{\mathbf{I}}$ imply $y = z$
concept assertion	$a : C_1$	$\mathbf{I} \models_{\mathbf{D}} a : C_1 \iff (a^{\mathbf{I}} \in C_1^{\mathbf{I}})$
agreement	$a = b$	$\mathbf{I} \models_{\mathbf{D}} a = b \iff a^{\mathbf{I}} = b^{\mathbf{I}}$
disagreement	$a \neq b$	$\mathbf{I} \models_{\mathbf{D}} a \neq b \iff \neg(a^{\mathbf{I}} = b^{\mathbf{I}})$

abstract role assertion	$(a, b) : R$	$\mathbf{I} \models_{\mathbf{D}} (a, b) : R \iff \langle a^{\mathbf{I}}, b^{\mathbf{I}} \rangle \in R^{\mathbf{I}}$
concrete role assertion	$(a, e_d) : P$	$\mathbf{I} \models_{\mathbf{D}} (a, e_d) : P \iff \langle a^{\mathbf{I}}, e_d^{\mathbf{D}} \rangle \in P^{\mathbf{I}}$
negated abstract role assertion	$(a, b) : \neg R$	$\mathbf{I} \models_{\mathbf{D}} (a, b) : \neg R \iff \neg(\langle a^{\mathbf{I}}, b^{\mathbf{I}} \rangle \in R^{\mathbf{I}})$
negated concrete role assertion	$(a, e_d) : \neg P$	$\mathbf{I} \models_{\mathbf{D}} (a, e_d) : \neg P \iff \neg(\langle a^{\mathbf{I}}, e_d^{\mathbf{D}} \rangle \in P^{\mathbf{I}})$

Table 1: Semantics of $\mathcal{SROIQ}(\mathbf{D})$.

Let $\mathcal{A}$, $\mathcal{R}$, $\mathcal{T}$ be, respectively, an $\mathcal{SROIQ}(\mathbf{D})$-*ABox*, an $\mathcal{SROIQ}(\mathbf{D})$-*RBox*, and an $\mathcal{SROIQ}(\mathbf{D})$-*TBox*. An interpretation $\mathbf{I} = (\Delta^{\mathbf{I}}, \Delta_{\mathbf{D}}, \cdot^{\mathbf{I}})$ is a **D**-model of $\mathcal{R}$ (resp., $\mathcal{T}$), and we write $\mathbf{I} \models_{\mathbf{D}} \mathcal{R}$ (resp., $\mathbf{I} \models_{\mathbf{D}} \mathcal{T}$), if $\mathbf{I}$ satisfies each axiom in $\mathcal{R}$ (resp., $\mathcal{T}$) according to the semantic rules in Table 1. Analogously, $\mathbf{I} = (\Delta^{\mathbf{I}}, \Delta_{\mathbf{D}}, \cdot^{\mathbf{I}})$ is a **D**-model of $\mathcal{A}$, and we write $\mathbf{I} \models_{\mathbf{D}} \mathcal{A}$, if $\mathbf{I}$ satisfies each assertion in $\mathcal{A}$, according to the semantic rules in Table 1.

An $\mathcal{SROIQ}(\mathbf{D})$-KB $\mathcal{K} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$ is consistent if there is an interpretation $\mathbf{I} = (\Delta^{\mathbf{I}}, \Delta_{\mathbf{D}}, \cdot^{\mathbf{I}})$ that is a **D**-model of $\mathcal{A}$, $\mathcal{T}$, and $\mathcal{R}$. Further details concerning the syntax and semantics of $\mathcal{SROIQ}(\mathbf{D})$ can be found in [69, 75].

Decidability of the consistency problem for $\mathcal{SROIQ}(\mathbf{D})$-KBs was proved in [69] by means of a tableau-based decision procedure and its computational complexity was shown to be **N2EXPTime**-complete in [75].

2.2 The Web Ontology Language OWL

Applications that need to automatically process the content of information, instead of just presenting information to humans, require an appropriate language with formally defined semantics. The goal of such language should be to simplify machine interoperability of information content by providing additional expressive power along with a formal syntax and semantics. This task turns out to be particularly critical in the case of web and distributed content. For this purpose, the *Word Wide Web Consortium* (W3C) identifies the OWL language, which at time of this writing is upgraded to the version 2.1, as the standard language for representing ontologies.

An ontology [99, 65] is a formal description of a domain of interest carried out by combining three basic syntactic categories: *entities*, *expressions*, and *axioms*, which constitute the logical part of ontologies, namely what ontologies can express and the type of inferences that can be drawn. Ontologies can also be combined together in order to describe more complex domains.

We briefly describe the main features of the standard OWL 2 DL profile. For the rest of this dissertation we will refer to OWL 2 DL as OWL 2, and to an OWL 2 DL ontology as ontology.

Entities represent the primitive terms of an ontology and are identified in a unique way. They are *individuals* (actors), *properties* (actions), and *classes* (sets of actors with common features). Properties are of two types: object properties and data type properties. Object properties relate pairs of individuals, whereas data type properties relate individuals with data type values, namely strings and numbers. Expressions, summarized in Table 2, are obtained by applying OWL 2 constructs to entities. For example, in the ontology SaintGall (illustrated in Section 6.3), the OWL 2 expression

*ObjectIntersectionOf(:Element ObjectSomeValuesFrom(
:uses ObjectUnionOf(:Fire :HotAir :HotWater)))*

defines the collection of individuals that can be classified as "heating elements".

OWL 2 Expression	Description
OWL 2 Property Expressions	
Object Property Expressions	
ObjectProperty	Simple object property
InverseObjectProperty	Inverse object property
Data Property Expressions	
Dataproperty	Simple data property
OWL 2 Class Expression	
Class	Simple class
Propositional Connectives and Enumeration of Individuals	
ObjectIntersectionOf	Intersection of classes
ObjectUnionOf	Union of classes
ObjectComplementOf	Complement of classes
ObjectOneOf	Enumeration of individuals
Object Property Restrictions	
ObjectSomeValuesFrom	Existential quantification
ObjectAllValuesFrom	Universial quantification
ObjectHasValue	Value restriction
ObjectHasSelf	Self restriction
Object Property Cardinality Restrictions	
ObjectMinCardinality	Minimum cardinality restric.
ObjectMaxCardinality	Maximum cardinality restic.

ObjectExactCardinality	Exactly cardinality restric.
data type Property Restrictions	
DataSomeValuesFrom	Data existential quantification
DataAllValuesFrom	Data universal quantificaion
DataHasValue	Data value restriction
data type Property Cardinality Restrictions	
DataMinCardinality	Data min. cardinality restric.
DataMaxCardinality	Data max. cardinality restric.
DataExactCardinality	Data exact cardinality restric.

Table 2: OWL 2 Expressions.

OWL 2 Axiom	Description
Object Property Axiom	
SubObjectPropertyOf	Axiom of object subproperty
EquivalentObjectProperties	Axiom of equivalent object property
DisjointObjectProperties	Axiom of disj. object property
InverseObjectProperties	Axiom of inverse object property
ObjectPropertyDomain	Axiom of object property domain
ObjectPropertyRange	Axiom of object property range
FunctionalObjectProperty	Axiom of functional object property
InverseFunctionalObjectProperty	Axiom of inv. func. object prop.
ReflexiveObjectProperty	Axiom of reflexive object prop.
IrreflexiveObjectProperty	Axiom of irreflexive object prop.
SymmetricObjectProperty	Axiom of symmetric object prop.
AsymmetricObjectProperty	Axiom of asymmetric object prop.
TransitiveObjectProperty	Axiom of transitive object prop.
Data Property Axioms	
SubDataPropertyOf	Axiom of data type subproperty
EquivalentDataProperties	Axiom of equivalent data type prop.
DisjointDataProperties	Axiom of disjoint data type prop.
DataPropertyDomain	Axiom of data type prop. domain
DataPropertyRange	Axiom of data type prop. range
FunctionalDataProperty	Axiom of functional data type prop.

Table 3: OWL 2 Axioms.

The first type of expressions admitted in OWL 2 is the *property expression*. Property expressions can be either *object property expressions*, which represent binary relationships between individuals, or *data type property expressions*, representing binary relationships between individuals and data type values. OWL 2 provides two types of object property expression, *ObjectProperty*, indicating the simplest form of object property, and *InverseObjectProperty*, the latter allowing one to define the inverse of a given object property.

OWL 2 Axiom	Description
Class Axiom	
SubClassOf	Axiom of subclass
EquivalentClasses	Axiom of equiv. classes
DisjointClasses	Axiom of disj. classes
DisjointUnion	Axiom of disj. union of classes
Assertions	
SameIndividual	Same individual axiom
DifferentIndividuals	Different individual axiom
ClassAssertion	Class instance axiom
ObjectPropertyAssertion	Object property axiom
NegativeObjectPropertyAssertion	Negative object property axiom
DataPropertyAssertion	data type property axiom
NegativeDataPropertyAssertion	Negative data type property
Key Axiom	
HasKey	Key axiom

Table 4: OWL 2 Axioms.

OWL 2 Axiom	Example	Description
ObjectPropertyAssertion	ObjectPropertyAssertion (:isDirectlyIncludedIn :Novice_Fountain :Novitiate)	The novice fountain is directly included in to the novitiate.
ClassAssertion	ClassAssertion (:Novice_Fountain :Fountain)	The novice fountain is a fountain.
SubClassOf	SubClassOf (:Chamber a:AbbotSittingRoom)	The abbot sitting room is subclass of chamber.

DisjointClasses	DisjointClasses (:AbbotHouse :BakeHouseBrewHouse)	Nothing can be both an abbot house and a bake/brew house.

Table 5: OWL 2 Axioms examples.

The second type of OWL 2 expression is the *class expression*. It represents sets of individuals sharing common characteristics. Such individuals are said to be instances of the respective class expression. Class expressions are constructed recursively by using classes, properties and class expressions.

The class expression *ObjectIntersectionOf* collects all the individuals that are instances of all the class expressions involved in the intersection. *ObjectUnionOf* allows one to aggregate all the individuals that are instances of at least one of the class expressions involved in the union. *ObjectComplementOf* allows one to aggregate all the individuals that are not instances of the class expression involved in the complement. Finally, *ObjectOneOf* permits to construct enumerations of individuals.

Class expressions can also be defined by applying restrictions on object property expressions. OWL 2 admits four types of class expressions defined by restrictions. The first one is *ObjectSomeValuesFrom*, formalizing the existential quantification and collecting those individuals that are connected by an object property expression to at least one instance of a given class expression. The second one, called *ObjectAllValuesFrom*, formalizes the universal quantification over an object property expression. It collects those individuals that are connected by an object property expression only to instances of a given class expression. The class expression *ObjectHasValue* consists of those individuals that are connected by an object property expression to a specific individual. Finally, *ObjectHasSelf* collects those individuals that are connected by an object property expression to themselves.

Another type of restriction admitted by OWL 2 concerns the cardinality of object property expressions. Such restrictions are called "cardinality restrictions". They can be either "qualified", when the individuals that are connected by the object property expression are also instances of a qualifying class expression, or "unqualified". Class expressions *ObjectMinCardinality*, *ObjectMaxCardinality*, and *ObjectExactCardinality* collect those individuals that are connected by an object property expression *at least*, *at most*, and *exactly* to a given number of instances of a specified class expression, respectively.

All of the restrictions described above can also involve data type values instead of individuals in an analogous way. They are represented in OWL 2 by the class expressions *DataSomeValuesFrom*, *DataAllValuesFrom*, *DataHas-*

Value, *DataHasSelf*, *DataMinCardinality*, *DataMaxCardinality*, and *DataExact-Cardinality*, respectively.

Statements asserting what is true in a domain are called axioms and they are the main components of OWL 2 ontologies. OWL 2 admits several types of axiom such as axioms about class expressions, about object or data property expressions, assertions, keys, data type definitions, and annotations. We focus on the first four types of axioms and refer the reader to the W3C documents for the latter two.

The first type of OWL 2 axiom concerns relationships between class expressions. *SubClassOf* allows one to state that, given two class expressions, each instance of the first class expression is also an instance of the second class expression, and thus to construct a hierarchy of classes.

EquivalentClasses allows one to state that some class expressions are equivalent to each other. *DisjointClasses* permits to state that some class expressions are pairwise disjoint, that is, they do not share any instance. Finally, *DisjointUnion* allows one to define a class expression as a disjoint union of several class expressions and thus to express covering constraints.

The second type of axioms admitted by OWL 2 can be used to characterize and establish relationships between object property expressions. The axiom *SubObjectPropertyOf* allows one to state that a object property expression is included in another object property expression. *EquivalentObjectProperties* permits to establish that several object property expressions are the same. *DisjointObjectProperties* allows one to state that several object property expressions are pairwise disjoint, that is, they do not share pairs of connected individuals. *InverseObjectProperties* is used to state that two object property expressions are the inverse of each other.

The property axiom *FunctionalObjectProperty* allows one to state that an object property expression is functional, namely that each individual can have at most one outgoing connection of the specified object property expression. *InverseFunctionalObjectProperty* states that an object property expression is inverse-functional, that is, each individual can have at most one incoming connection of the specified object property expression. *ReflexiveObjectProperty* allows one to state that an object property expression is reflexive, i.e., each individual is connected by the object property expression to itself, whereas *IrreflexiveObjectProperty* states that an object property expression is irreflexive, and hence no individual is connected to itself. *SymmetricObjectProperty* allows one to state that an object property expression is symmetric, that is, if an individual x is connected by a given object property expression `OPE` to an individual `y`, then `y` is also connected to `x` by the same property expression `OPE`. Conversely, *AsymmetricObjectProperty* states that the object property expression is asymmetric. *TransitiveObjectProperty* states that an object property expression is transitive, that is, if an individual x is connected by a given object property expression `OPE` to an individual `y`, and `y` is connected to an individual `z` by *OPE*, then `x` is also connected to `z` by `OPE`. Finally, *ObjectPropertyDomain* and *ObjectPropertyRange* are used, respectively, to restrict the first and the second individual related by an object property expression to be instances of the specified class expression.

Concerning data type properties axioms, OWL 2 provides analogous constructs, namely *SubDataPropertyOf*, *EquivalentDataProperties*, *DisjointDataProperties*, *DataPropertyDomain*, *DataPropertyRange*, and *FunctionalDataProperty*.

The last type of axiom, also called assertion or fact, concerns individuals. *SameIndividual* allows one to state that two individuals are the same entity, while *DifferentIndividuals* that they are distinct. The axiom *ClassAssertion* allows one to state that an individual is instance of a class. *ObjectPropertyAssertion* states that two individuals are connected by an object property expression, while *NegativeObjectPropertyAssertion* that they are not. Analogous axioms are introduced to connect individuals with data type values, namely *DataPropertyAssertion* and *NegativeDataPropertyAssertion*.

A key axiom *HasKey* states that each instance of a given class expression is uniquely identified by given object property expressions and/or by given data property expressions.

The mapping from OWL 2 constructs to DL [68] is illustrated in Table 6. OWL 2 constructs are described via the *OWL 2 Abstract Syntax* [134].

OWL constructs	DL Syntax
Object Properties (R)	
R	R
inv(R)	R^-
Data type Properties (P)	
P	P
Individuals (o)	
o	o
Data Values (d)	
d	d
Data ranges (D)	
D	D
oneOf($d_1 \ldots d_n$)	$\{d_1\} \sqcup \ldots \sqcup \{d_n\}$
Descriptions of Concepts	
A	A

owl:Thing	$\top$
owl:Nothing	$\bot$
intersectionOf($C_1 \ldots C_n$)	$C_1 \sqcap \ldots \sqcap C_n$
unionOf($C_1 \ldots C_n$)	$C_1 \sqcup \ldots \sqcup C_n$
complementOf(C)	$\neg C_1$
oneOf($o_1 \ldots o_n$)	$\{o_1\} \sqcup \ldots \sqcup \{o_n\}$
restriction(R someValuesFrom(C))	$\exists R.C$
restriction(R allValuesFrom(C))	$\forall R.C$
restriction(R hasValue(o))	$R : o$
restriction(R minCardinality(n))	$\geq_n R$
restriction(R maxCardinality(n))	$\leq_n R$
restriction(P someValuesFrom(D))	$\exists P.D$
restriction(P allValuesFrom(D))	$\forall P.D$
restriction(P hasValue(d))	$P : d$
restriction(P minCardinality(n))	$\geq_n P$
restriction(P maxCardinality(n))	$\leq_n P$
SubClassOf($C_1 \ldots C_n$)	$C_1 \sqsubseteq \ldots \sqsubseteq C_n$
EquivalentClasses($C_1 \ldots C_n$)	$C_1 \equiv \ldots \equiv C_n$
DisjointClasses($C_1 \ldots C_n$)	$C_i \sqcap C_j \subseteq \bot, i \neq j$
Class(C partial $C_1 \ldots C_n$)	$A \sqsubseteq C_1 \sqcap \ldots \sqcap C_n$
Class(C complete $C_1 \ldots C_n$)	$A \equiv C_1 \sqcap \ldots \sqcap C_n$
EnumeratedClass(C $o_1 \ldots o_n$)	$A \equiv \{o_1\} \sqcup \ldots \sqcup \{o_n\}$
Descriptions of Object Properties	
SubPropertyOf(R R_1)	$R \sqsubseteq R_1$
EquivalentProperties($R \ldots R_n$)	$R \equiv \ldots \equiv R_n$
ObjectPropety(R Transitive)	Tran(R)
ObjectPropety(R Functional)	$\top \sqsubseteq \leq_1 R$
ObjectPropety(R InverseFunctional)	$\top \sqsubseteq \leq_1 R^-$
ObjectPropety(R Symmetric)	$R \equiv R^-$
ObjectProperty(R InverseOf(R_1))	$R \equiv R_1^-$
ObjectProperty(R domain(C))	$\geq_1 R \sqsubseteq C$
ObjectProperty(R range(C))	$\top \sqsubseteq \forall R.C$
Descriptions of Data Properties	
SubPropertyOf(P P_1)	$P \sqcap P_1$
EquivalentProperties($P_1 \ldots P_n$)	$P_1 \equiv \ldots \equiv P_n$
Functional(P)	$\top \sqsubseteq \leq_1 P$
ObjectProperty(P domain(C))	$\geq_1 P \sqsubseteq C$
ObjectProperty(P range(C))	$\top \sqsubseteq \forall P.C$
Descriptions of Individuals	
Individual(o type(C))	$o \in C$
Individual(o value(R type(o_1))	$\langle o, o_1 \rangle \in R$

Individual(o value(P type(d))	$\langle o, d\rangle \in P$
SameIndividual($o_1 \dots o_n$)	$\{o_1\} \equiv \dots \equiv o_n$
DifferentIndividuals($o_1 \dots o_n$)	$\{o_i\} \sqsubseteq \neg o_j, i \neq j$

Table 6: Mapping from OWL constructs to DL.

2.2.1 OWL 2 Profiles

As stated above, an OWL 2 profile is a trimmed-down version of OWL 2 DL that trades some expressive power for the efficiency of reasoning. The standard OWL 2 profiles currently identified are OWL 2 QL, OWL 2 RL, and OWL 2 EL.

The OWL 2 QL profile is designed for efficient query answering, which can be performed in **LOGSPACE** with respect to the size of the ABox by using a suitable reasoning technique. Such an efficiency therefore requires a limitation of the profile expressive power.

Basically, OWL 2 QL admits pure existential quantification but it is very limited for what concerns the number of supported constructs. In fact, OWL 2 QL does not support

- existential quantification to a class expression or a data range (*ObjectSomeValuesFrom* and *DataSomeValuesFrom*) in the subclass position;
- self-restriction (*ObjectHasSelf*);
- valued existential quantification (*ObjectHasValue*, *DataHasValue*);
- enumeration of individuals and literals (*ObjectOneOf*, *DataOneOf*);
- universal quantification to a class expression or a data range (*ObjectAllValuesFrom*, *DataAllValuesFrom*);
- cardinality restrictions (*ObjectMaxCardinality*, *ObjectMinCardinality*, *ObjectExactCardinality*, *DataMaxCardinality*, *DataMinCardinality*, *DataExactCardinality*);
- disjunction (*ObjectUnionOf*, *DisjointUnion*, and *DataUnionOf*);
- property inclusions (*SubObjectPropertyOf*) involving property chains;
- functional and inverse-functional properties (*FunctionalObjectProperty*, *InverseFunctionalObjectProperty*, and *FunctionalDataProperty*);
- transitive properties (*TransitiveObjectProperty*);
- keys (*HasKey*), and

- individual equality assertions and negative property assertions.

The OWL 2 RL profile is designed for scalable reasoning without sacrificing too much the expressive power by defining a syntactic subset of OWL 2 which is compliant to be implemented with rule-based framework. OWL 2 RL supports all axioms of OWL 2 apart from disjoint unions of classes (*DisjointUnion*) and reflexive object property axioms (*ReflexiveObjectProperty*). It admits the following constructs in subclass expressions:

- enumeration of individuals (*ObjectOneOf*);
- intersection of class expressions (*ObjectIntersectionOf*);
- union of class expressions (*ObjectUnionOf*), and
- existential quantification (*ObjectSomeValuesFrom*, *DataSomeValuesFrom*, *ObjectHasValue*, *DataHasValue*).

It also admits the following constructs in superclass expressions:

- intersection of classes (*ObjectIntersectionOf*);
- negation (*ObjectComplementOf*);
- universal quantification to a class expression (*ObjectAllValuesFrom*);
- existential quantification to an individual (*ObjectHasValue*);
- at-most 0/1 cardinality restriction to a class expression (*ObjectMaxCardinality 0/1*);
- universal quantification to a data range (*DataAllValuesFrom*);
- existential quantification to a literal (*DataHasValue*), and
- at-most 0/1 cardinality restriction to a data range (*DataMaxCardinality 0/1*).

OWL EL is suitable for applications employing ontologies containing a very large number of classes and/or properties. Such profile does not admit:

- universal quantification to a class expression (*ObjectAllValuesFrom*) or a data range (*DataAllValuesFrom*);
- cardinality restrictions (*ObjectMaxCardinality*, *ObjectMinCardinality*, *ObjectExactCardinality*, *DataMaxCardinality*, *DataMinCardinality*, and *DataExactCardinality*);

- disjunction (*ObjectUnionOf*, *DisjointUnion*, and *DataUnionOf*);
- class negation (*ObjectComplementOf*);
- enumerations involving more than one individual (*ObjectOneOf* and *DataOneOf*);
- disjoint properties (*DisjointObjectProperties* and *DisjointDataProperties*);
- irreflexive object properties (*IrreflexiveObjectProperty*);
- inverse object properties (*InverseObjectProperties*);
- functional and inverse-functional object properties (*FunctionalObjectProperty* and *InverseFunctionalObjectProperty*);
- symmetric object properties (*SymmetricObjectProperty*), and
- asymmetric object properties (*AsymmetricObjectProperty*).

OWL EL also restricts the set of supported class expressions (*ObjectIntersectionOf*, *ObjectSomeValuesFrom*, *ObjectHasSelf*, *ObjectHasValue*, *DataSomeValuesFrom*, *DataHasValue*, and *ObjectOneOf*) to contain a single individual.

To sum up, OWL EL admits a very limited form of pure existential quantification at the price of excluding numerous OWL 2 constructs.

2.2.2 The Semantic Web Rule Language

The extension of ontologies with rules has become a fundamental requirement in order to increase the expressiveness and the reasoning power of OWL knowledge bases. In the last decade scholars had interest in studying how rules can be translated in DL axioms by extending OWL syntax and semantics in a coherent manner [95, 77].

Many of the limitations of OWL derive from the fact that, even if it provides several class expressions, it comes with few property constructors. Moreover, although OWL is provided with several sorts of conditionals, these are, however, very constrained. In fact, it is not possible to mix directly classes (concepts) and properties (roles) since OWL disallows for arbitrary patterns of variables and for fairly free mixing of expressions (e.g., property and class expressions). For instance, one can express in OWL the following axiom:

SubClassOf(:Person ObjectUnionOf(:Human :IntelligentComputer)),

but cannot mix classes and properties as in the following case:

SubClassOf(:parentOf ObjectUnionOf(:Human :IntelligentComputer)),

where *parentOf* is an object property [56].

Furthermore, OWL does not include non-monotonic reasoning such as negation as failure[2] and disallows to express directly the composition constructor, so it is impossible to capture relationships between a composite property and another (possibly composite) property. The standard example is the obvious relationship between the composition of the "parent" and "brother" properties and the "uncle" property. These restrictions can be onerous in some application domains, for example in describing web services, where it may be necessary to relate inputs and outputs of composite processes to the inputs and outputs of their component processes [34].

A way to overcome some of these restrictions is to extend the language with some form of rules.

Recently, a simple form of Horn-style rules has been added to OWL as a new set of axioms [70, 70]. Such extension has finally led to the definition of the Semantic Web Rule Language (SWRL) [129], the standard rule language extension of OWL which combines OWL with the Unary/Binary Datalog fragment of the Rule Markup Language (RML) [124]. SWRL allows users to write rules containing OWL constructs so as to provide more reasoning capabilities than OWL alone.

The basic idea behind SWRL is to extend OWL DL with a form of rules, while maintaining the maximum compatibility with OWL existing syntax and semantics. In order to provide a formal syntax and semantics for OWL ontologies including such rules, a new type of axioms has been added to OWL DL extending the OWL abstract syntax and the direct model-theoretic semantics for OWL DL.

The proposed form of SWRL goes beyond basic Horn clauses in allowing conjunctive consequents, class atoms admitting as predicate both simple and complex class, equalities and inequalities. Most of these features are simply "syntactic sugar", and basically do not increase the power of the language. The consistency problem for SWRL ontologies is undecidable [71]. In order to provide support to SWRL reasoning many solutions have been investigated [123, 79, 63, 137, 10] which apply restrictions in the shape of the rules. For example, DL Safe SWRL [10] rule variables bind only to explicitly named individuals in the ontology. Another basic approach is to identify the Horn-logic rules directly expressible in OWL DL (this fragment has been called Description Logic Program (DLP) [63]). In Description Logic Rules [79], rule atoms admit only OWL 2 class and property names and the normalised rule's dependency graph (undirected graph whose nodes are variables and edges are property atoms of a rule) has no cycles. It turns out that rules of the form $ordered(x, y) \wedge$

[2]We recall that a logic is non-monotonic if some conclusions can be invalidated when more knowledge is added.

$dislikes(x, y) \rightarrow Unhappy(x)$ cannot be expressed.

Datalog, instead, applies the following safety conditions:

- complex terms as arguments of predicates are disallowed,
- certain stratification restrictions on the use of negation and recursion are applied,
- every variable that appears in the head of a rule also appears in a nonarithmetic positive (i.e., not negated) literal in the body of the rule, and
- every variable appearing in a negative literal in the body of a rule also appears in some positive literal in the body of that rule.

As highlighted in [95], traditional approaches suffer from many drawbacks. The addition of rules to DL usually affects the semantics of DL. In the integration, DL and rule components should be able to contribute to the consequences of the other component. The hybrid formalism should be flexible and allow one to view the same predicate under both open- and closed-world interpretations. The formalism should be decidable, and preferably of low worst-case complexity. Even the solution proposed in [95] needs DL Safe rules.

Rule axioms have an antecedent component and a consequent component. The antecedent and consequent of a rule are both lists of atoms and are read as the conjunction of the component atoms. Atoms can be formed from unary predicates (classes), binary predicates (properties), equalities or inequalities. In a general sense, a rule is any sentence stating that if a set of premises is satisfied in a given model, then a certain conclusion must be satisfied in the same model.

An SWRL-rule $\mathtt{r}$ has the form $(\forall x_1, \ldots, x_n)(\mathtt{B} \implies \mathtt{H})$, where:

- $\mathtt{B}$ (the body of $\mathtt{r}$) and $\mathtt{H}$ (the head of $\mathtt{r}$) are conjunctions of atoms of the following types: $C(x)$, $d(y)$, $R(x, y)$, $P(x, y)$, $x = y$, $x \neq y$, with C a concept name, d a data type, R an abstract role name, P a concrete role name, and x, y either individuals or variables (in the specific case of atoms of the forms $d(y)$ and $P(x, y)$, y can be either a data type constant or a variable), and
- $\mathit{Var}(\mathtt{H}) \subseteq \mathit{Var}(\mathtt{B}) = \{x_1, \ldots, x_n\}$, where $\mathit{Var}(\mathtt{H})$ and $\mathit{Var}(\mathtt{B})$ are the sets of variables occurring in $\mathtt{H}$ and in $\mathtt{B}$, respectively.

SWRL syntax and semantics can be found in [129].

2.3 The set-theoretic fragment $\mathsf{4LQS^R}$

In this section we summarize the set-theoretic notions underpinning the DL $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ and its reasoning tasks. In order to define the fragment $\mathsf{4LQS^R}$, it is convenient to first introduce the syntax and semantics of a more general four-level quantified language, denoted by 4LQS. Then we provide some restrictions on quantified formulae of 4LQS that characterize $4LQS^R$. We recall that the satisfiability problem for $4LQS^R$ has been proved decidable in [26].

4LQS involves the four collections of variables $\mathcal{V}_0$, $\mathcal{V}_1$, $\mathcal{V}_2$, $\mathcal{V}_3$, where:

- $\mathcal{V}_0$ contains variables of sort 0, denoted by $x, y, z, ...$;
- $\mathcal{V}_1$ contains variables of sort 1, denoted by $X^1, Y^1, Z^1, ...$;
- $\mathcal{V}_2$ contains variables of sort 2, denoted by $X^2, Y^2, Z^2, ...$;
- $\mathcal{V}_3$ contains variables of sort 3, denoted by $X^3, Y^3, Z^3, ...$.

In addition to variables, 4LQS involves also *pair terms* of the form $\langle x, y\rangle$, for $x, y \in \mathcal{V}_0$. 4LQS-*atomic formulae* are classified as:

- level 0: $x = y$, $x \in X^1$, $\langle x, y\rangle = X^2$, $\langle x, y\rangle \in X^3$, where $x, y \in \mathcal{V}_0$, $\langle x, y\rangle$ is a pair term, $X^1 \in \mathcal{V}_1$, $X^2 \in \mathcal{V}_2$, X^3 in $\mathcal{V}_3$;
- level 1: $X^1 = Y^1$, $X^1 \in X^2$, with $X^1, Y^1 \in \mathcal{V}_1$, X^2 in $\mathcal{V}_2$;
- level 2: $X^2 = Y^2$, $X^2 \in X^3$, with $X^2, Y^2 \in \mathcal{V}_2$, X^3 in $\mathcal{V}_3$.

4LQS *purely universal formulae* are classified as:

- level 1: $(\forall z_1)...(\forall z_n)\varphi_0$, where $z_1, .., z_n \in \mathcal{V}_0$ and φ_0 is any propositional combination of atomic formulae of level 0;
- level 2: $(\forall Z^1_1)...(\forall Z^1_m)\varphi_1$, where $Z^1_1, .., Z^1_m \in \mathcal{V}_1$ and φ_1 is any propositional combination of atomic formulae of levels 0 and 1 and of purely universal formulae of level 1;
- level 3: $(\forall Z^2_1)...(\forall Z^2_p)\varphi_2$, where $Z^2_1, .., Z^2_p \in \mathcal{V}_2$ and φ_2 is any propositional combination of atomic formulae and of purely universal formulae of levels 1 and 2.

4LQS-formulae are all the propositional combinations of atomic formulae of levels 0, 1, 2 and of purely universal formulae of levels 1, 2, 3.

Let φ be a 4LQS-formula. Without loss of generality, we can assume that φ contains only $\neg$, $\wedge$, $\vee$ as propositional connectives. Further, let S_φ be the syntax tree for a 4LQS-formula φ,[3] and let ν be a node of S_φ. We say that a 4LQS-formula ψ occurs within φ at position ν if the subtree of S_φ rooted at ν is identical to S_ψ. In this case we refer to ν as an occurrence of ψ in φ and to the path from the root of S_φ to ν as its occurrence path. An occurrence of ψ within φ is *positive* if its occurrence path deprived by its last node contains an

[3]The notion of syntax tree for 4LQS-formulae is similar to the notion of syntax tree for formulae of first-order logic. A precise definition of the latter can be found in [40].

even number of nodes labelled by a 4LQS-formula of type $\neg\chi$. Otherwise, the occurrence is *negative*.

The variables $z_1, \dots, z_n$, $Z_1^1, \dots Z_m^1$, and $Z_1^2, \dots, Z_p^2$ are said to occur *quantified* in $(\forall z_1)\dots(\forall z_n)\varphi_0$, $(\forall Z_1^1)...(\forall Z_m^1)\varphi_1$, and $(\forall Z_1^2)...(\forall Z_p^2)\varphi_2$, respectively. A variable occurs *free* in a 4LQS-formula φ if it does not occur quantified in any subformula of φ. For $i = 0, 1, 2, 3$, we denote with $\mathtt{Var}_i(\varphi)$ the collection of variables of sort i occurring free in φ.

Given sequences of distinct variables $\vec{x}$ (in $\mathtt{Var}_0$), $\vec{X}^1$ (in $\mathtt{Var}_1$), $\vec{X}^2$ (in $\mathtt{Var}_2$), and $\vec{X}^3$ (in $\mathtt{Var}_3$), of length n, m, p, and q, respectively, and sequences of (not necessarily distinct) variables $\vec{y}$ (in $\mathtt{Var}_0$), $\vec{Y}^1$ (in $\mathtt{Var}_1$), $\vec{Y}^2$ (in $\mathtt{Var}_2$), and $\vec{Y}^3$ (in $\mathtt{Var}_3$), also of length n, m, p, and q, respectively, the 4LQS-substitution $\sigma := \{\vec{x}/\vec{y}, \vec{X}^1/\vec{Y}^1, \vec{X}^2/\vec{Y}^2, \vec{X}^3/\vec{Y}^3\}$ is the mapping $\varphi \mapsto \varphi\sigma$ such that, for any given universal quantified 4LQS-formula φ, $\varphi\sigma$ is the result of replacing in φ the free occurrences of the variables x_i in $\vec{x}$ (for $i = 1, \dots, n$) with the corresponding y_i in $\vec{y}$, of X_ℓ^1 in $\vec{X}^1$ (for $\ell = 1, \dots, m$) with Y_ℓ^1 in $\vec{Y}^1$, of X_j^2 in $\vec{X}^2$ (for $j = 1, \dots, p$) with Y_j^2 in $\vec{Y}^2$, and of X_h^3 in $\vec{X}^3$ (for $h = 1, \dots, q$) with Y_h^3 in $\vec{Y}^3$, respectively. A substitution σ is *free* for φ if the formulae φ and $\varphi\sigma$ have exactly the same occurrences of quantified variables. The *empty substitution*, denoted ϵ, satisfies $\varphi\epsilon = \varphi$, for each 4LQS-formula φ. The *support* of a substitution σ is the set of variables $x \in \mathtt{Var}_0$, $X^1 \in \mathtt{Var}_1$, $X^2 \in \mathtt{Var}_2$, and $X^3 \in \mathtt{Var}_3$ such that $x\sigma \neq x$, $X^1\sigma \neq X^1$, $X^2\sigma \neq X^2$, and $X^3\sigma \neq X^3$, respectively.

A 4LQS-*interpretation* is a pair $\boldsymbol{\mathcal{M}} = (D, M)$ where D is any non-empty collection of objects (called domain or universe of $\boldsymbol{\mathcal{M}}$) and M is an assignment over variables in $\mathcal{V}_0$, $\mathcal{V}_1$, $\mathcal{V}_2$, $\mathcal{V}_3$ such that:

- $Mx \in D$, for each $x \in \mathcal{V}_0$;
- $MX^1 \in \mathcal{P}(D)$, for each $X^1 \in \mathcal{V}_1$;[4]
- $MX^2 \in \mathcal{P}(\mathcal{P}(D))$, for each $X^2 \in \mathcal{V}_2$;
- $MX^3 \in \mathcal{P}(\mathcal{P}(\mathcal{P}(D)))$, for each $X^3 \in \mathcal{V}_3$.

We assume that pair terms are interpreted *à la* Kuratowski, and therefore we put

$$M\langle x, y\rangle =_{Def} \{\{Mx\}, \{Mx, My\}\}.$$

The presence of a pairing operator in the language is very useful for the set-theoretic representation of the logics of *shdlss* and $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ (see Chapter 3). Moreover, even though several pairing operators are available (see [53]), encoding ordered pairs à la Kuratowski turns out to be quite straightforward, at least for these purposes.

Next, let

[4]We recall that $\mathcal{P}(s)$ denotes the powerset of s.

- $\boldsymbol{\mathcal{M}} = (D, M)$ be a 4LQS-interpretation,
- $x_1, \ldots, x_n \in \mathcal{V}_0$,
- $X_1^1, \ldots, X_m^1 \in \mathcal{V}_1$,
- $X_1^2, \ldots, X_p^2 \in \mathcal{V}_2$,
- $u_1, \ldots, u_n \in D$,
- $U_1^1, \ldots, U_m^1 \in \mathcal{P}(D)$,
- $U_1^2, \ldots, U_p^2 \in \mathcal{P}(\mathcal{P}(D))$.

By

$$\boldsymbol{\mathcal{M}}[x_1/u_1, ..., x_n/u_n, X_1^1/U_1^1, \ldots, X_m^1/U_m^1, X_1^2/U_1^2, \ldots, X_p^2/U_p^2]$$

We denote the interpretation $\boldsymbol{\mathcal{M}}' = (D, M')$ such that

- $M'x_i = u_i$, for $i = 1, ..., n$,
- $M'X_j^1 = U_j^1$, for $j = 1, ..., m$,
- $M'X_k^2 = U_k^2$, for $k = 1, ..., p$,

and which otherwise coincides with M on all remaining variables.

Let φ be a 4LQS-formula and let $\boldsymbol{\mathcal{M}} = (D, M)$ be a $4LQS$-interpretation. The notion of satisfiability of φ by $\boldsymbol{\mathcal{M}}$ (denoted by $\boldsymbol{\mathcal{M}} \models \varphi$) is defined inductively over the structure of φ as follows:

- $\boldsymbol{\mathcal{M}} \models x = y$ iff $Mx = My$;
- $\boldsymbol{\mathcal{M}} \models x \in X^1$ iff $Mx \in MX^1$;
- $\boldsymbol{\mathcal{M}} \models \langle x, y \rangle = X^2$ iff $M\langle x, y \rangle = MX^2$;
- $\boldsymbol{\mathcal{M}} \models \langle x, y \rangle \in X^3$ iff $M\langle x, y \rangle \in MX^3$;
- $\boldsymbol{\mathcal{M}} \models X^1 = Y^1$ iff $MX^1 = MY^1$;
- $\boldsymbol{\mathcal{M}} \models X^1 \in X^2$ iff $MX^1 \in MX^2$;
- $\boldsymbol{\mathcal{M}} \models X^2 = Y^2$ iff $MX^2 = MY^2$;
- $\boldsymbol{\mathcal{M}} \models X^2 \in X^3$ iff $MX^2 \in MX^3$;
- $\boldsymbol{\mathcal{M}} \models (\forall z_1)...(\forall z_n)\varphi_0$ iff $\boldsymbol{\mathcal{M}}[z_1/u_1, ..., z_n/u_n] \models \varphi_0$, for all $u_1, ..., u_n \in D$;
- $\boldsymbol{\mathcal{M}} \models (\forall Z_1^1)...(\forall Z_m^1)\varphi_1$ iff $\boldsymbol{\mathcal{M}}[Z_1^1/U_1^1, ..., Z_n^1/U_n^1] \models \varphi_1$, for all $U_1^1, ..., U_m^1 \in \mathcal{P}(D)$;
- $\boldsymbol{\mathcal{M}} \models (\forall Z_1^2)...(\forall Z_m^2)\varphi_2$ iff $\boldsymbol{\mathcal{M}}[Z_1^2/U_1^2, ..., Z_n^2/U_n^2] \models \varphi_2$, for all $U_1^2, ..., U_m^2 \in \mathcal{P}(\mathcal{P}(D))$.

Propositional connectives are interpreted in the standard way, namely:

- $\mathcal{M} \models \varphi_1 \wedge \varphi_2$ iff $\mathcal{M} \models \varphi_1$ and $\mathcal{M} \models \varphi_2$;
- $\mathcal{M} \models \varphi_1 \vee \varphi_2$ iff $\mathcal{M} \models \varphi_1$ or $\mathcal{M} \models \varphi_2$;
- $\mathcal{M} \models \neg\varphi$ iff $\mathcal{M} \not\models \varphi$.

If $\mathcal{M} \models \varphi$, then $\mathcal{M}$ is said to be a 4LQS-model for φ. A 4LQS-formula is said to be *satisfiable* if it has a 4LQS-model. A 4LQS-formula is *valid* if it is satisfied by all 4LQS-interpretations.

Next, we present the fragment $4LQS^R$ of 4LQS of our interest, namely the collection of the formulae ψ of 4LQS fulfilling the restrictions:

1. for every purely universal formula $(\forall Z_1^1)...(\forall Z_m^1)\varphi_1$ of level 2 occurring in ψ and every purely universal formula $(\forall z_1)...(\forall z_n)\varphi_0$ of level 1 occurring negatively in φ_1, the condition

$$\neg\varphi_0 \rightarrow \bigwedge_{i=1}^{n} \bigwedge_{j=1}^{m} z_i \in Z_j^1$$

is a valid 4LQS-formula (in this case we say that $(\forall z_1)...(\forall z_n)\varphi_0$ is *linked to the variables* $Z_1^1, ..., Z_m^1$);

2. for every purely universal formula $(\forall Z_1^2)...(\forall Z_p^2)\varphi_2$ of level 3 in ψ:

   - every purely universal formula of level 1 occurring negatively in φ_2 and not occurring in a purely universal formula of level 2 is only allowed to be of the form

$$(\forall z_1)...(\forall z_n)\neg(\bigwedge_{i=1}^{n} \bigwedge_{j=1}^{n} \langle z_i, z_j\rangle = Y_{ij}^2),$$

   with $Y_{ij}^2 \in \mathcal{V}^2$, for $i, j = 1, ..., n$;
   - purely universal formulae $(\forall Z_1^1)...(\forall Z_m^1)\varphi_1$ of level 2 may occur only positively in φ_2.

Restriction 1 has been introduced for technical reasons concerning the decidability of the satisfiability problem for the fragment. In fact it guarantees that satisfiability is preserved in a suitable finite sub-model of ψ. Restriction 2 allows one to express binary relations and several operations on them while keeping simple, at the same time, the decision procedure (see [26]).

The reader can observe that the semantics of $4LQS^R$ plainly coincides with that of 4LQS.

2.3.1 Expressiveness of $\mathsf{4LQS^R}$

The fragment $\mathsf{4LQS^R}$ allows one to express several set-theoretic constructs, among which we mention the restricted variants of the set former over variables of sort 0 and 1, since they in turn allow one to express other significant set operators such as binary union, intersection, set difference, the singleton operator, the powerset operator, etc. More specifically, atomic formulae of type $X^i = \{X^{i-1} : \varphi(X^{i-1})\}$, for $i = 1, 2, 3$ can be expressed by the formulae

$$(\forall X^{i-1})(X^{i-1} \in X^i \leftrightarrow \varphi(X^{i-1})),$$

respectively, provided that the syntactic constraints of $\mathsf{4LQS^R}$ are satisfied. Among the other constructs of set theory which are expressible in the fragment $\mathsf{4LQS^R}$, we recall:

- literals of the form $X^2 = pow_{<h}(X^1)$, where $pow_{<h}(X^1)$ denotes the collection of subsets of X^1 with less than h elements;
- the unordered Cartesian product $X^2 = X^1_1 \otimes \ldots \otimes X^1_n$, where $X^1_1 \otimes \ldots \otimes X^1_n$ denotes the collection $\{\{x_1, \ldots, x_n\} : x_1 \in X^1_1, \ldots, x_n \in X^1_n\}$;
- literals of the form $A = pow^*(X^1_1, \ldots, X^1_n)$, where $pow^*(X^1_1, \ldots, X^1_n)$ is the variant of the powerset introduced in [16] which denotes the collection

$$\{Z : Z \subseteq \bigcup_{i=1}^{n} X^1_i \text{ and } Z \cap X^1_1 \neq \emptyset, \text{ for all } 1 \leq i \leq n\}.$$

For instance, a literal of the form $X^2 = pow_{<h}(X^1)$, with $h \geq 2$, can be expressed by the $\mathsf{4LQS^R}$-formula

$$(\forall Y^1)\big(Y^1 \in X^2 \leftrightarrow \Big((\forall z)(z \in Y^1 \rightarrow z \in X^1) \wedge (\forall z_1) \ldots (\forall z_h)\Big(\bigwedge_{i=1}^{h} z_i \in Y^1 \rightarrow \bigvee_{\substack{i,j=1 \\ i<j}}^{h} z_i = z_j\Big)\Big)\big),$$

as can be easily verified.

Within the fragment $\mathsf{4LQS^R}$, it is also possible to define binary relations over elements of a domain together with conditions on them (i.e., reflexivity, transitivity, weak connectedness, irreflexivity, intransitivity) which characterize accessibility relations of well-known modal logics as illustrated in Table 7.

Binary relation	$(\forall Z^2)(Z^2 \in X^3_R \leftrightarrow \neg(\forall z_1, z_2)\neg(\langle z_1, z_2\rangle = Z^2))$
Reflexive	$(\forall z_1)(\langle z_1, z_1\rangle \in X^3_R)$
Irreflexive	$(\forall z_1)\neg(\langle z_1, z_1\rangle \in X^3_R)$
Symmetric	$(\forall z_1, z_2)(\langle z_1, z_2\rangle \in X^3_R \rightarrow \langle z_2, z_1\rangle \in X^3_R)$
Asymmetric	$(\forall z_1, z_2)(\langle z_1, z_2\rangle \in X^3_R \rightarrow \neg(\langle z_2, z_1\rangle \in X^3_R))$
Antisymmetric	$(\forall z_1, z_2)((\langle z_1, z_2\rangle \in X^3_R \wedge \langle z_2, z_1\rangle \in X^3_R) \rightarrow (z_1 = z_2))$
Euclidean	$(\forall z_1, z_2, z_3)((\langle z_1, z_2\rangle \in X^3_R \wedge \langle z_1, z_3\rangle \in X^3_R) \rightarrow \langle z_2, z_3\rangle \in X^3_R)$
Transitive	$(\forall z_1, z_2, z_3)((\langle z_1, z_2\rangle \in X^3_R \wedge \langle z_2, z_3\rangle \in X^3_R) \rightarrow \langle z_1, z_3\rangle \in X^3_R)$
Intransitive	$(\forall z_1, z_2, z_3)((\langle z_1, z_2\rangle \in X^3_R \wedge \langle z_2, z_3\rangle \in X^3_R) \rightarrow \neg\langle z_1, z_3\rangle \in X^3_R)$
Weakly-connected	$(\forall z_1, z_2, z_3)((\langle z_1, z_2\rangle \in X^3_R \wedge \langle z_1, z_3\rangle \in X^3_R) \rightarrow (\langle z_2, z_3\rangle \in X^3_R \vee z_2 = z_3 \vee \langle z_3, z_2\rangle \in X^3_R))$

Table 7: $\mathsf{4LQS^R}$ formalization of Boolean operations over relations

Then, usual Boolean operations over relations can be defined as shown in Table 8.

Intersection $X^3_R = X^3_{R_1} \cap X^3_{R_2}$	$(\forall Z^2)(Z^2 \in X^3_R \leftrightarrow (Z^2 \in X^3_{R_1} \wedge Z^2 \in X^3_{R_2}))$
Union $X^3_R = X^3_{R_1} \cup X^3_{R_2}$	$(\forall Z^2)(Z^2 \in X^3_R \leftrightarrow (Z^2 \in X^3_{R_1} \vee Z^2 \in X^3_{R_2}))$
Complement $X^3_{R_1} = \overline{X^3_{R_2}}$	$(\forall Z^2)(Z^2 \in X^3_{R_1} \leftrightarrow \neg(Z^2 \in \overline{X^3_{R_2}}))$
Set difference $X^3_R = X^3_{R_1} \setminus X^3_{R_2}$	$(\forall Z^2)(Z^2 \in X^3_R \leftrightarrow (Z^2 \in X^3_{R_1} \wedge \neg(Z^2 \in X^3_{R_2})))$
Set inclusion $X^3_{R_1} \subseteq X^3_{R_2}$	$(\forall Z^2)(Z^2 \in X^3_R \rightarrow Z^2 \in X^3_{R_2})$

Table 8: $\mathsf{4LQS^R}$ formalization of Boolean operations over relations

In addition, since the fragment $\mathsf{4LQS^R}$ generalizes the fragment $\mathsf{3LQS^R}$ [25], it can express the two-level syllogistic 2LS [53] and, hence, the sub-fragment 3LSSP of 3LSSPU not involving the set-theoretic construct of general union [17]. We recall that 3LSSPU admits variables of three sorts and, besides the usual set-theoretical constructs, it involves the singleton set operator $\{\cdot\}$, the powerset operator *pow*, and the general union operator *Un*.

Finally, $4LQS^R$ allows one to express several modal logics, such as the normal modal logic K45 and the modal logic S5. Details can be found in [26] and [25], respectively.

2.3.2 The class of $4LQS^R$-formulae $4LQS^R_{\mathcal{DL}^{4,\times}_{\mathbf{D}}}$

In this section we restrict our attention to the class of $4LQS^R$-formulae actually involved in the set-theoretic representation of the DLs $\mathcal{DL}^4_{\mathbf{D}}$ and $\mathcal{DL}^{4,\times}_{\mathbf{D}}$, namely propositional combination of $4LQS^R$-literals ($4LQS^R$-atoms or their negations) and $4LQS^R$-purely universal formulae of the types displayed in Table 9. We refer to such class of $4LQS^R$-formulae as $4LQS^R_{\mathcal{DL}^{4,\times}_{\mathbf{D}}}$. The introduction of $4LQS^R_{\mathcal{DL}^{4,\times}_{\mathbf{D}}}$ is motivated by the fact that (1) other multi-sorted fragments such as $3LQS^R$ [25] or 3LSSP [17] do no allow one to represent DL roles and their relationships, and (2) $4LQS^R$ variables of sorts 2 are not required for the representation of the DLs $\mathcal{DL}^4_{\mathbf{D}}$ and $\mathcal{DL}^{4,\times}_{\mathbf{D}}$ (see Chapter 3).

Since the types of formulae illustrated in Table 9 do not involve variables of sort 2, notions and definitions concerning $4LQS^R_{\mathcal{DL}^{4,\times}_{\mathbf{D}}}$-formulae only regard variables of sorts $0, 1, 3$.

$4LQS^R$-literals of level 0
$x = y,\ x \in X^1,\ \langle x, y\rangle \in X^3$
$\neg(x = y),\ \neg(x \in X^1),\ \neg(\langle x, y\rangle \in X^3)$
$4LQS^R$-purely universal quantified formulae of level 1
$(\forall z_1)\ldots(\forall z_n)\varphi_0$, where $z_1, \ldots, z_n \in \mathtt{Var}_0$ and φ_0 is any propositional combination of literals of level 0.

Table 9: Types of formulae admitted in $4LQS^R_{\mathcal{DL}^{4,\times}_{\mathbf{D}}}$.

Specifically, the variables $z_1, \ldots, z_n$ are said to occur *quantified* in $(\forall z_1)\ldots(\forall z_n)\varphi_0$. A variable occurs *free* in a $4LQS^R_{\mathcal{DL}^{4,\times}_{\mathbf{D}}}$-formula φ if it does not occur quantified in any subformula of φ. For $i = 0, 1, 3$, we denote with $\mathtt{Var}_i(\varphi)$ the collections of variables of sort i occurring free in φ.

Given sequences of distinct variables $\vec{x}$ (in $\mathtt{Var}_0$), $\vec{X}^1$ (in $\mathtt{Var}_1$), and $\vec{X}^3$ (in $\mathtt{Var}_3$), of length n, m, and q, respectively, and sequences of (not necessarily distinct) variables $\vec{y}$ (in $\mathtt{Var}_0$), $\vec{Y}^1$ (in $\mathtt{Var}_1$), and $\vec{Y}^3$ (in $\mathtt{Var}_3$), also of length n, m, and q, respectively, the (level 0) $\mathsf{4LQS}^R_{\mathcal{DL}_\mathbf{D}^{4,\times}}$-substitution

$$\sigma \coloneqq \{\vec{x}/\vec{y}, \vec{X}^1/\vec{Y}^1, \vec{X}^3/\vec{Y}^3\}$$

is the mapping $\varphi \mapsto \varphi\sigma$ such that, for any given universal quantified $\mathsf{4LQS}^R_{\mathcal{DL}_\mathbf{D}^{4,\times}}$-formula φ, $\varphi\sigma$ is the $\mathsf{4LQS}^R_{\mathcal{DL}_\mathbf{D}^{4,\times}}$-formula obtained from φ by replacing the free occurrences

- of the variables x_i in $\vec{x}$ (for $i = 1, \ldots, n$) with the corresponding y_i in $\vec{y}$,
- of X^1_j in $\vec{X}^1$ (for $j = 1, \ldots, m$) with Y^1_j in $\vec{Y}^1$, and
- of X^3_h in $\vec{X}^3$ (for $h = 1, \ldots, q$) with Y^3_h in $\vec{Y}^3$, respectively.

A substitution σ is *free* for φ if the formulae φ and $\varphi\sigma$ have exactly the same occurrences of quantified variables. The *empty substitution*, denoted by ϵ, satisfies $\varphi\epsilon = \varphi$, for each $\mathsf{4LQS}^R_{\mathcal{DL}_\mathbf{D}^{4,\times}}$-formula φ. The *support* of a substitution σ is the set of variables $x \in \mathtt{Var}_0$, $X^1 \in \mathtt{Var}_1$, and $X^3 \in \mathtt{Var}_3$ such that $x\sigma \neq x$, $X^1\sigma \neq X^1$, and $X^3\sigma \neq X^3$, respectively.

A $\mathsf{4LQS}^R_{\mathcal{DL}_\mathbf{D}^{4,\times}}$-*interpretation* is a pair $\boldsymbol{\mathcal{M}} = (D, M)$, where D is a nonempty collection of objects (called *domain* or *universe* of $\boldsymbol{\mathcal{M}}$) and M is an assignment over the variables in $\mathtt{Var}_i$, for $i = 0, 1, 3$, such that:

- $MX^0 \in D$,
- $MX^1 \in \mathcal{P}(D)$,
- $MX^3 \in \mathcal{P}(\mathcal{P}(\mathcal{P}(D)))$,

where $X^i \in \mathtt{Var}_i$, for $i = 0, 1, 3$, and $\mathcal{P}(s)$ denotes the powerset of s.

Pair terms are interpreted *à la* Kuratowski, and therefore we put

$$M\langle x, y\rangle \coloneqq \{\{Mx\}, \{Mx, My\}\}.$$

Next, let

- $\boldsymbol{\mathcal{M}} = (D, M)$ be a $\mathsf{4LQS}^R_{\mathcal{DL}_\mathbf{D}^{4,\times}}$-interpretation,
- $x_1, \ldots, x_n \in \mathtt{Var}_0$, and
- $u_1, \ldots, u_n \in D$.

By $\boldsymbol{\mathcal{M}}[\vec{x}/\vec{u}]$, we denote the interpretation $\boldsymbol{\mathcal{M}}' = (D, M')$ such that $M'x_i = u_i$ (for $i = 1, \ldots, n$), and which otherwise coincides with M on all remaining variables. For a $\mathsf{4LQS}^R_{\mathcal{DL}_\mathbf{D}^{4,\times}}$-interpretation $\boldsymbol{\mathcal{M}} = (D, M)$ and a $\mathsf{4LQS}^R_{\mathcal{DL}_\mathbf{D}^{4,\times}}$-formula

φ, the satisfiability relationship $\boldsymbol{\mathcal{M}} \models \varphi$ is recursively defined over the structure of φ as follows. Literals are evaluated in a standard way according to the usual meaning of the predicates '$\in$' and '$=$', and of the propositional negation '$\neg$'. Compound formulae are interpreted according to the standard rules of propositional logic. Finally, purely universal formulae are evaluated as follows:

- $\boldsymbol{\mathcal{M}} \models (\forall z_1) \ldots (\forall z_n)\varphi_0$ iff $\boldsymbol{\mathcal{M}}[\vec{z}/\vec{u}] \models \varphi_0$, for all $\vec{u} \in D^n$;

If $\boldsymbol{\mathcal{M}} \models \varphi$, then $\boldsymbol{\mathcal{M}}$ is said to be a $\mathsf{4LQS}^R_{\mathcal{DL}^{4,\times}_{\mathbf{D}}}$-model for φ. A $\mathsf{4LQS}^R_{\mathcal{DL}^{4,\times}_{\mathbf{D}}}$-formula is said to be *satisfiable* if it has a $\mathsf{4LQS}^R_{\mathcal{DL}^{4,\times}_{\mathbf{D}}}$-model. A $\mathsf{4LQS}^R_{\mathcal{DL}^{4,\times}_{\mathbf{D}}}$-formula is *valid* if it is satisfied by all $\mathsf{4LQS}^R_{\mathcal{DL}^{4,\times}_{\mathbf{D}}}$-interpretations.

3 Description Logics via 4LQS$^{\mathsf{R}}$

In this chapter, we will introduce the DLs $\mathcal{DL}_{\mathbf{D}}^{4}$ and $\mathcal{DL}_{\mathbf{D}}^{4,\times}$, and we will prove the decidability of the consistency problem for $\mathcal{DL}_{\mathbf{D}}^{4}$- and $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KBs by reducing them to the satisfiability problem for 4LQS$^{\mathsf{R}}$-formulae.

In particular in Section 3.1, we will introduce the DL $\mathcal{DL}_{\mathbf{D}}^{4}$ and we will prove that the consistency problem for $\mathcal{DL}_{\mathbf{D}}^{4}$ is decidable by showing that by means of a suitable mapping one can construct a 4LQS$^{\mathsf{R}}$ formula $\phi_{\mathcal{KB}}$ from a $\mathcal{DL}_{\mathbf{D}}^{4}$-KB $\mathcal{KB}$ such that $\phi_{\mathcal{KB}}$ is satisfiable if and only $\mathcal{KB}$ is satisfiable. In other words, we will prove that a knowledge base of $\mathcal{DL}_{\mathbf{D}}^{4}$ and its 4LQS$^{\mathsf{R}}$ representation are equisatisfiable. Then, we will show that under certain restrictions the consistency problem for $\mathcal{DL}_{\mathbf{D}}^{4}$-KBs is **NP**-complete. Finally, we will briefly illustrate how SWRL rules can also be mapped in 4LQS$^{\mathsf{R}}$-formulae.

In Section 3.2 we will introduce the DL $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ which extends $\mathcal{DL}_{\mathbf{D}}^{4}$ and we will prove that the consistency problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ is decidable in a way analogous to the proof for $\mathcal{DL}_{\mathbf{D}}^{4}$.

3.1 The description logic $\mathcal{DL}_{\mathbf{D}}^{4}$

In this section, we will present the description logic $\mathcal{DL}(\mathbf{D})\langle$4LQS$^{\mathsf{R}}\rangle$ (shortly referred to as $\mathcal{DL}_{\mathbf{D}}^{4}$). we will first introduce the syntax and the semantics of the DL $\mathcal{DL}_{\mathbf{D}}^{4}$ and then we will prove that the consistency problem for $\mathcal{DL}_{\mathbf{D}}^{4}$-KBs is decidable by reducing it to the satisfiability problem for 4LQS$^{\mathsf{R}}$-formulae, which has been proved to be decidable in [26].

Let $\mathbf{D} = (N_D, N_C, N_F, \cdot^{\mathbf{D}})$ be a *data type map* as defined in Section 2.1, and let $\mathbf{R_A}$, $\mathbf{R_D}$, $\mathbf{C}$, $\mathbf{Ind}$ be the denumerable pairwise disjoint sets of *abstract role names*, *concrete role names*, *concept names*, and *individual names*, respectively, and $\mathbf{D}$ be a data type map.
(a) $\mathcal{DL}_{\mathbf{D}}^{4}$-data type, (b) $\mathcal{DL}_{\mathbf{D}}^{4}$-concept, (c) $\mathcal{DL}_{\mathbf{D}}^{4}$-abstract role, and (d) $\mathcal{DL}_{\mathbf{D}}^{4}$-concrete role terms are constructed according to the following syntax rules:

(a) $t \longrightarrow dr \mid \neg t_1 \mid t_1 \sqcap t_2 \mid t_1 \sqcup t_2 \mid \{e_d\}$,

(b) $C \longrightarrow A \mid \top \mid \bot \mid \neg C_1 \mid C_1 \sqcup C_2 \mid C_1 \sqcap C_2 \mid \{a\} \mid \exists R.\mathit{Self} \mid \exists R.\{a\} \mid \exists P.\{e_d\}$,

(c) $R \longrightarrow S \mid U \mid R_1^- \mid \neg R_1 \mid R_1 \sqcup R_2 \mid R_1 \sqcap R_2 \mid R_{C_1|} \mid R_{|C_1} \mid R_{C_1 \mid C_2} \mid id(C)$,

(d) $P \longrightarrow T \mid \neg P \mid P_{C_1|} \mid P_{|t_1} \mid P_{C_1|t_1}$,

where dr is a data range for $\mathbf{D}$, t, t_1, t_2 are data type terms, e_d is a constant in $N_C(d)$, a is an individual name, A is a concept name, C, C_1, C_2 are $\mathcal{DL}_{\mathbf{D}}^{4}$-concept terms, S is an abstract role name, R, R_1, R_2 are $\mathcal{DL}^4_{\mathbf{D}}$-abstract role terms, T a concrete role name, and P a $\mathcal{DL}^4_{\mathbf{D}}$-concrete role term.

A $\mathcal{DL}^4_{\mathbf{D}}$-knowledge base is a triple $\mathcal{KB} = (\mathcal{R}, \mathcal{T}, \mathcal{A})$ such that $\mathcal{R}$ is a $\mathcal{DL}^4_{\mathbf{D}}$-*RBox*, $\mathcal{T}$ is a $\mathcal{DL}^4_{\mathbf{D}}$-*TBox*, and $\mathcal{A}$ a $\mathcal{DL}^4_{\mathbf{D}}$-*ABox*. A $\mathcal{DL}^4_{\mathbf{D}}$-*RBox* is a collection of statements of the following forms:

$$\begin{array}{ccccc} R_1 \equiv R_2, & R_1 \sqsubseteq R_2, & R_1 \ldots R_n \sqsubseteq R_{n+1}, & \mathsf{Sym}(R_1), & \mathsf{Asym}(R_1), \\ \mathsf{Ref}(R_1), & \mathsf{Irref}(R_1), & \mathsf{Dis}(R_1, R_2), & \mathsf{Tra}(R_1), & \mathsf{Fun}(R_1), \\ & P_1 \equiv P_2, & P_1 \sqsubseteq P_2, & \mathsf{Fun}(P_1), & \end{array}$$

where R_1, R_2 are $\mathcal{DL}^4_{\mathbf{D}}$-abstract role terms and P_1, P_2 are $\mathcal{DL}^4_{\mathbf{D}}$-concrete role terms. A $\mathcal{DL}^4_{\mathbf{D}}$-*TBox* is a set of statements of the types:

- $C_1 \equiv C_2$, $C_1 \sqsubseteq C_2$, $C_1 \sqsubseteq \forall R_1.C_2$, $\exists R_1.C_1 \sqsubseteq C_2$, $\geq_n R_1.C_1 \sqsubseteq C_2$, $C_1 \sqsubseteq \leq_n R_1.C_2$,
- $t_1 \equiv t_2$, $t_1 \sqsubseteq t_2$, $C_1 \sqsubseteq \forall P_1.t_1$, $\exists P_1.t_1 \sqsubseteq C_1$, $\geq_n P_1.t_1 \sqsubseteq C_1$, $C_1 \sqsubseteq \leq_n P_1.t_1$,

where C_1, C_2 are $\mathcal{DL}^4_{\mathbf{D}}$-concept terms, t_1, t_2 data type terms, R_1 a $\mathcal{DL}^4_{\mathbf{D}}$-abstract role term, and P_1 a $\mathcal{DL}^4_{\mathbf{D}}$-concrete role term.

A $\mathcal{DL}^4_{\mathbf{D}}$-*ABox* is a set of assertions of the forms:

$$\begin{array}{cccc} a : C, & (a, b) : R, & (a, b) : \neg R, & a = b, \\ a \neq b, & e_d : t, & (a, e_d) : P, & (a, e_d) : \neg P, \end{array}$$

where C is a $\mathcal{DL}^4_{\mathbf{D}}$-concept term, d is a data type, t is a data type term, R is a $\mathcal{DL}^4_{\mathbf{D}}$-abstract role term, P is a $\mathcal{DL}^4_{\mathbf{D}}$-concrete role term, a, b are individual names, and e_d is a constant in $N_C(d)$.

Any expression of the type $w \sqsubseteq R$, where w is a finite string of $\mathcal{DL}^4_{\mathbf{D}}$-abstract role terms and R is an $\mathcal{DL}^4_{\mathbf{D}}$-abstract role term is called *role inclusion axiom (RIA)*.

The semantics of $\mathcal{DL}^4_{\mathbf{D}}$ is analogous to that of $\mathcal{SROIQ}(\mathbf{D})$ (see Table 1 of Section 2.1.1) and is reported in Table 10.

The semantics of $\mathcal{DL}^4_{\mathbf{D}}$ is based on interpretations $\mathbf{I} = (\Delta^{\mathbf{I}}, \Delta_{\mathbf{D}}, \cdot^{\mathbf{I}})$, where $\Delta^{\mathbf{I}}$ and $\Delta_{\mathbf{D}}$ are non-empty disjoint domains such that $d^{\mathbf{D}} \subseteq \Delta_{\mathbf{D}}$, for every $d \in N_D$, and $\cdot^{\mathbf{I}}$ is an interpretation function. Let $\mathcal{KB} = (\mathcal{R}, \mathcal{T}, \mathcal{A})$ be a $\mathcal{DL}^4_{\mathbf{D}}$-KB. An interpretation $\mathbf{I} = (\Delta^{\mathbf{I}}, \Delta_{\mathbf{D}}, \cdot^{\mathbf{I}})$ is a $\mathbf{D}$-model of $\mathcal{R}$ (and we write $\mathbf{I} \models_{\mathbf{D}} \mathcal{R}$) if $\mathbf{I}$ satisfies each axiom in $\mathcal{R}$ according to the semantic rules in Table 10. Similar definitions hold for $\mathcal{T}$ and $\mathcal{A}$ too. Then $\mathbf{I}$ satisfies $\mathcal{KB}$ (and we write $\mathbf{I} \models_{\mathbf{D}} \mathcal{KB}$) if it is a $\mathbf{D}$-model of $\mathcal{R}$, $\mathcal{T}$, and $\mathcal{A}$. A $\mathcal{DL}^4_{\mathbf{D}}$-KB $\mathcal{KB}$ is *consistent* if $\mathcal{KB}$ is satisfied by some interpretation.

Name	Syntax	Semantics
concept abstract role	A R	$A^{\mathbf{I}} \subseteq \Delta^{\mathbf{I}}$ $R^{\mathbf{I}} \subseteq \Delta^{\mathbf{I}} \times \Delta^{\mathbf{I}}$
concrete role	T	$T^{\mathbf{I}} \subseteq \Delta^{\mathbf{I}} \times \Delta_{\mathbf{D}}$
individual	a	$a^{\mathbf{I}} \in \Delta^{\mathbf{I}}$
nominal	$\{a\}$	$\{a\}^{\mathbf{I}} = \{a^{\mathbf{I}}\}$
data type	d	$d^{\mathbf{D}} \subseteq \Delta_{\mathbf{D}}$
negated data type	$\neg d$	$\Delta_{\mathbf{D}} \setminus d^{\mathbf{D}}$
data type term	t_1	$t_1^{\mathbf{D}} \subseteq \Delta_{\mathbf{D}}$
negated data type term	$\neg t_1$	$(\neg t_1)^{\mathbf{D}} = \Delta_{\mathbf{D}} \setminus t_1^{\mathbf{D}}$
data type terms intersection	$t_1 \sqcap t_2$	$(t_1 \sqcap t_2)^{\mathbf{D}} = t_1^{\mathbf{D}} \cap t_2^{\mathbf{D}}$
data type terms union	$t_1 \sqcup t_2$	$(t_1 \sqcup t_2)^{\mathbf{D}} = t_1^{\mathbf{D}} \cup t_2^{\mathbf{D}}$
constant in $N_C(d)$	e_d	$e_d^{\mathbf{D}} \in d^{\mathbf{D}}$
data range dr	dr	$dr^{\mathbf{D}} \subseteq \Delta_{\mathbf{D}}$
negated data range	$\neg dr$	$\Delta_{\mathbf{D}} \setminus dr^{\mathbf{D}}$
data range	$\{e_{d_1}, \ldots, e_{d_n}\}$	$\{e_{d_1}, \ldots, e_{d_n}\}^{\mathbf{D}} =$ $\{e_{d_1}^{\mathbf{D}}\} \cup \ldots \cup \{e_{d_n}^{\mathbf{D}}\}$
data range	ψ_d	$\psi_d^{\mathbf{D}}$
nominals	$\{a_1, \ldots, a_n\}$	$\{a_1, \ldots, a_n\}^{\mathbf{I}} = \{a_1^{\mathbf{I}}\} \cup \ldots \cup \{a_n^{\mathbf{I}}\}$
top	$\top$	$\Delta^{\mathbf{I}}$
bottom	$\bot$	$\emptyset$
negation	$\neg C$	$(\neg C)^{\mathbf{I}} = \Delta^{\mathbf{I}} \setminus C$
conjunction of concepts	$C \sqcap D$	$(C \sqcap D)^{\mathbf{I}} = C^{\mathbf{I}} \cap D^{\mathbf{I}}$
disjunction of concepts	$C \sqcup D$	$(C \sqcup D)^{\mathbf{I}} = C^{\mathbf{I}} \cup D^{\mathbf{I}}$
concept subsumption	$C_1 \sqsubseteq C_2$	$\mathbf{I} \models_{\mathbf{D}} C_1 \sqsubseteq C_2 \iff C_1^{\mathbf{I}} \subseteq C_2^{\mathbf{I}}$
concept equality	$C_1 \equiv C_2$	$C_1^{\mathbf{I}} = C_2^{\mathbf{I}}$
universal role	U	$(U)^{\mathbf{I}} = \Delta^{\mathbf{I}} \times \Delta^{\mathbf{I}}$
valued existential quantification	$\exists R.a$	$(\exists R.a)^{\mathbf{I}} = \{x \in \Delta^{\mathbf{I}} : \langle x, a^{\mathbf{I}} \rangle \in R^{\mathbf{I}}\}$
data typed existential quantification	$\exists P.e_d$	$(\exists P.e_d)^{\mathbf{I}} = \{x \in \Delta^{\mathbf{I}} : \langle x, e_d^{\mathbf{D}} \rangle \in P^{\mathbf{I}}\}$

self concept	$\exists R.Self$	$(\exists R.Self)^{\mathbf{I}} = \{x \in \Delta^{\mathbf{I}} : \langle x, x\rangle \in R^{\mathbf{I}}\}$
inverse role	R^{-}	$(R^{-})^{\mathbf{I}} = \{\langle y, x\rangle \mid \langle x, y\rangle \in R^{\mathbf{I}}\}$
abstract role complement	$\neg R$	$(\neg R)^{\mathbf{I}} = (\Delta^{\mathbf{I}} \times \Delta^{\mathbf{I}}) \setminus R^{\mathbf{I}}$
abstract role union	$R_1 \sqcup R_2$	$(R_1 \sqcup R_2)^{\mathbf{I}} = R_1^{\mathbf{I}} \cup R_2^{\mathbf{I}}$
abstract role intersection	$R_1 \sqcap R_2$	$(R_1 \sqcap R_2)^{\mathbf{I}} = R_1^{\mathbf{I}} \cap R_2^{\mathbf{I}}$
abstract role domain restriction.	$R_{C\mid}$	$(R_{C\mid})^{\mathbf{I}} = \{\langle x, y\rangle \in R^{\mathbf{I}} : x \in C^{\mathbf{I}}\}$
abstract role subsumption	$R_1 \sqsubseteq R_2$	$\mathbf{I} \models_{\mathbf{D}} R_1 \sqsubseteq R_2 \iff R_1^{\mathbf{I}} \subseteq R_2^{\mathbf{I}}$
role inclusion axiom	$R_1 \ldots R_n \sqsubseteq R$	$\mathbf{I} \models_{\mathbf{D}} R_1 \ldots R_n \sqsubseteq R \iff R_1^{\mathbf{I}} \circ \ldots \circ R_n^{\mathbf{I}} \subseteq R^{\mathbf{I}}$
abstract role equality	$R_1 \equiv R_2$	$R_1^{\mathbf{I}} = R_2^{\mathbf{I}}$
abstract role composition	$R_1 \circ R_2$	$\{(a, c) \mid \exists b.(a, b) \in R_1^{\mathbf{I}} \wedge (b, c) \in R_2^{\mathbf{I}}\}$
symmetric abstract role	$\mathsf{Sym}(R)$	$\mathbf{I} \models_{\mathbf{D}} \mathsf{Sym}(R) \iff (R^{-})^{\mathbf{I}} \subseteq R^{\mathbf{I}}$
asymmetric abstract role	$\mathsf{Asym}(R)$	$\mathbf{I} \models_{\mathbf{D}} \mathsf{Asym}(R) \iff R^{\mathbf{I}} \cap (R^{-})^{\mathbf{I}} = \emptyset$
transitive abstract role	$\mathsf{Tra}(R)$	$\mathbf{I} \models_{\mathbf{D}} \mathsf{Tra}(R) \iff R^{\mathbf{I}} \circ R^{\mathbf{I}} \subseteq R^{\mathbf{I}}$
disjoint abstract role	$\mathsf{Dis}(R_1, R_2)$	$\mathbf{I} \models_{\mathbf{D}} \mathsf{Dis}(R_1, R_2) \iff R_1^{\mathbf{I}} \cap R_2^{\mathbf{I}} = \emptyset$
reflexive abstract role	$\mathsf{Ref}(R)$	$\mathbf{I} \models_{\mathbf{D}} \mathsf{Ref}(R) \iff \{\langle x, x\rangle \mid x \in \Delta^{\mathbf{I}}\} \subseteq R^{\mathbf{I}}$
irreflexive abstract role	$\mathsf{Irref}(R)$	$\mathbf{I} \models_{\mathbf{D}} \mathsf{Irref}(R) \iff R^{\mathbf{I}} \cap \{\langle x, x\rangle \mid x \in \Delta^{\mathbf{I}}\} = \emptyset$
functional abstract role	$\mathsf{Fun}(R)$	$\mathbf{I} \models_{\mathbf{D}} \mathsf{Fun}(R) \iff (R^{-})^{\mathbf{I}} \circ R^{\mathbf{I}} \subseteq \{\langle x, x\rangle \mid x \in \Delta^{\mathbf{I}}\}$
concrete role complement	$\neg P$	$(\neg P)^{\mathbf{I}} = (\Delta^{\mathbf{I}} \times \Delta^{\mathbf{D}}) \setminus P^{\mathbf{I}}$
concrete role union	$P_1 \sqcup P_2$	$(P_1 \sqcup P_2)^{\mathbf{I}} = P_1^{\mathbf{I}} \cup P_2^{\mathbf{I}}$
concrete role intersection	$P_1 \sqcap P_2$	$(P_1 \sqcap P_2)^{\mathbf{I}} = P_1^{\mathbf{I}} \cap P_2^{\mathbf{I}}$

concrete role domain restriction	$P_{C\vert}$	$(P_{C\vert})^{\mathbf{I}} = \{\langle x,y\rangle \in P^{\mathbf{I}} : x \in C^{\mathbf{I}}\}$
concrete role range restriction	$P_{\vert t}$	$(P_{\vert t})^{\mathbf{I}} = \{\langle x,y\rangle \in P^{\mathbf{I}} : y \in t^{\mathbf{D}}\}$
concrete role restriction	$P_{C_1\vert t}$	$(P_{C_1\vert t})^{\mathbf{I}} = \{\langle x,y\rangle \in P^{\mathbf{I}} :$ $x \in C_1^{\mathbf{I}} \wedge y \in t^{\mathbf{D}}\}$
concrete role subsumption	$P_1 \sqsubseteq P_2$	$\mathbf{I} \models_{\mathbf{D}} P_1 \sqsubseteq P_2 \iff P_1^{\mathbf{I}} \subseteq P_2^{\mathbf{I}}$
disjoint concrete role	$\mathsf{Dis}(P_1, P_2)$	$\mathbf{I} \models_{\mathbf{D}} \mathsf{Dis}(P_1, P_2) \iff P_1^{\mathbf{I}} \cap P_2^{\mathbf{I}} = \emptyset$
functional concrete role	$\mathsf{Fun}(P)$	$\mathbf{I} \models_{\mathbf{D}} \mathsf{Fun}(p) \iff \langle x,y\rangle \in P^{\mathbf{I}}$ and $\langle x,z\rangle \in P^{\mathbf{I}}$ imply $y = z$
concrete role complement	$\neg P$	$(\Delta^{\mathbf{I}} \times \Delta^{D}) \setminus P^{\mathbf{I}}$
concrete role equality	$P_1 \equiv P_2$	$P_1^{\mathbf{I}} = P_2^{\mathbf{I}}$
data type terms equivalence	$t_1 \equiv t_2$	$\mathbf{I} \models_{\mathbf{D}} t_1 \equiv t_2 \iff t_1^{\mathbf{D}} = t_2^{\mathbf{D}}$
data type terms disequivalence	$t_1 \not\equiv t_2$	$\mathbf{I} \models_{\mathbf{D}} t_1 \not\equiv t_2 \iff t_1^{\mathbf{D}} \neq t_2^{\mathbf{D}}$
data type terms subsumption	$t_1 \sqsubseteq t_2$	$\mathbf{I} \models_{\mathbf{D}} (t_1 \sqsubseteq t_2) \iff t_1^{\mathbf{D}} \subseteq t_2^{\mathbf{D}}$
concept assertion	$a : C_1$	$\mathbf{I} \models_{\mathbf{D}} a : C_1 \iff (a^{\mathbf{I}} \in C_1^{\mathbf{I}})$
agreement	$a = b$	$\mathbf{I} \models_{\mathbf{D}} a = b \iff a^{\mathbf{I}} = b^{\mathbf{I}}$
disagreement	$a \neq b$	$\mathbf{I} \models_{\mathbf{D}} a \neq b \iff \neg(a^{\mathbf{I}} = b^{\mathbf{I}})$
abstract role assertion	$(a,b) : R$	$\mathbf{I} \models_{\mathbf{D}} (a,b) : R \iff \langle a^{\mathbf{I}}, b^{\mathbf{I}}\rangle \in R^{\mathbf{I}}$
concrete role assertion	$(a, e_d) : P$	$\mathbf{I} \models_{\mathbf{D}} (a, e_d) : P \iff \langle a^{\mathbf{I}}, e_d^{\mathbf{D}}\rangle \in P^{\mathbf{I}}$
negated abstract role assertion	$(a,b) : \neg R$	$\mathbf{I} \models_{\mathbf{D}} (a,b) : \neg R \iff$ $\neg(\langle a^{\mathbf{I}}, b^{\mathbf{I}}\rangle \in R^{\mathbf{I}})$
negated concrete role assertion	$(a, e_d) : \neg P$	$\mathbf{I} \models_{\mathbf{D}} (a, e_d) : \neg P \iff$ $\neg(\langle a^{\mathbf{I}}, e_d^{\mathbf{D}}\rangle \in P^{\mathbf{I}})$
negated concrete assertion	$\neg(a : C_1)$	$\neg(a : C_1)^{\mathbf{I}} = \neg(a^{\mathbf{I}} \in C_1^{\mathbf{I}})$

Table 10: Semantics of $\mathcal{DL}_{\mathbf{D}}^{4}$.

In Theorem 1, we prove the decidability of the consistency problem for $\mathcal{DL}_{\mathbf{D}}^{4}$-KBs by reducing it to the satisfiability problem for $\mathsf{4LQS^R}$-formulae, which is decidable [26]. Before introducing the theorem, it is convenient to define the

4LQS$^\mathsf{R}$-formula $\varphi_{\mathcal{KB}}$ representing the set-theoretical representation of the $\mathcal{DL}^4_{\mathbf{D}}$-KB $\mathcal{KB}$.

As a preliminary step, observe that the statements of the $\mathcal{DL}^4_{\mathbf{D}}$-KB $\mathcal{KB}$ that have to be considered are those of the following types:

- $C_1 \equiv \top$, $C_1 \equiv \neg C_2$, $C_1 \equiv C_2 \sqcup C_3$, $C_1 \equiv \{a\}$, $C_1 \sqsubseteq \forall R_1.C_2$, $\exists R_1.C_1 \sqsubseteq C_2$, $\geq_n R_1.C_1 \sqsubseteq C_2$, $C_1 \sqsubseteq \leq_n R_1.C_2$, $C_1 \sqsubseteq \forall P_1.t_1$, $\exists P_1.t_1 \sqsubseteq C_1$, $\geq_n P_1.t_1 \sqsubseteq C_1$, $C_1 \sqsubseteq \leq_n P_1.t_1$,
- $R_1 \equiv U$, $R_1 \equiv \neg R_2$, $R_1 \equiv R_2 \sqcup R_3$, $R_1 \equiv R_2^-$, $R_1 \equiv id(C_1)$, $R_1 \equiv R_{2_{C_1|}}$, $R_1 \dots R_n \sqsubseteq R_{n+1}$, $\mathsf{Ref}(R_1)$, $\mathsf{Irref}(R_1)$, $\mathsf{Fun}(R_1)$,
- $P_1 \equiv P_2$, $P_1 \equiv \neg P_2$, $P_1 \sqsubseteq P_2$, $\mathsf{Fun}(P_1)$, $P_1 \equiv P_{2_{C_1|}}$, $P_1 \equiv P_{2_{C_1|t_1}}$, $P_1 \equiv P_{2_{|t_1}}$,
- $t_1 \equiv t_2$, $t_1 \equiv \neg t_2$, $t_1 \equiv t_2 \sqcup t_3$, $t_1 \equiv \{e_d\}$,
- $a : C_1$, $(a,b) : R_1$, $(a,b) : \neg R_1$, $a = b$, $a \neq b$, $e_d : t_1$, $(a, e_d) : P_1$, $(a, e_d) : \neg P_1$.

In order to define the 4LQS$^\mathsf{R}$-formula $\varphi_{\mathcal{KB}}$, we shall make use of a mapping τ from the $\mathcal{DL}^4_{\mathbf{D}}$-statements (and their conjunctions) listed above into 4LQS$^\mathsf{R}$-formulae. To prepare for the definition of τ, we map injectively

- individuals a and constants $e_d \in N_C(d)$ into the sort 0 variables x_a and x_{e_d},
- the constant concepts $\top$ and $\bot$, data type term t, and concept term C into the sort 1 variables $X^1_\top$, $X^1_\bot$, X^1_t, X^1_C, respectively, and
- the universal relation on individuals U, abstract role term R, and concrete role term P into the sort 3 variables X^3_U, X^3_R, and X^3_P, respectively. [5]

Then the mapping τ is defined as follows:

$$\tau(C_1 \equiv \top) \coloneqq (\forall z)(z \in X^1_{C_1} \leftrightarrow z \in X^1_\top),$$

$$\tau(C_1 \equiv \neg C_2) \coloneqq (\forall z)(z \in X_{C_1} \leftrightarrow \neg(z \in X^1_{C_2})),$$

$$\tau(C_1 \equiv C_2 \sqcup C_3) \coloneqq (\forall z)(z \in X^1_{C_1} \leftrightarrow (z \in X^1_{C_2} \vee z \in X^1_{C_3})),$$

$$\tau(C_1 \equiv \{a\}) \coloneqq (\forall z)(z \in X^1_{C_1} \leftrightarrow z = x_a),$$

$$\tau(C_1 \sqsubseteq \forall R_1.C_2) \coloneqq (\forall z_1)(\forall z_2)(z_1 \in X^1_{C_1} \rightarrow (\langle z_1, z_2\rangle \in X^3_{R_1} \rightarrow z_2 \in X^1_{C_2})),$$

[5] The use of sort 3 variables to model abstract and concrete role terms is motivated by the fact that their elements, that is ordered pairs $\langle x, y\rangle$, are encoded in Kuratowski's style as $\{\{x\}, \{x, y\}\}$, namely as collections of sets of objects.

$$\tau(\exists R_1.C_1 \sqsubseteq C_2) \coloneqq (\forall z_1)(\forall z_2)(((\langle z_1, z_2\rangle \in X^3_{R_1} \wedge z_2 \in X^1_{C_1}) \rightarrow z_1 \in X^1_{C_2}),$$

$$\tau(C_1 \equiv \exists R_1.\{a\}) \coloneqq (\forall z)(z \in X^1_{C_1} \leftrightarrow \langle z, x_a\rangle \in X^3_{R_1}),$$

$$\tau(C_1 \sqsubseteq \leq_n R_1.C_2) \coloneqq (\forall z)(\forall z_1)\dots(\forall z_{n+1})(z \in X^1_{C_1} \rightarrow (\bigwedge_{i=1}^{n+1}(z_i \in X_{C_2} \wedge \langle z, z_i\rangle \in X^3_{R_1}) \rightarrow \bigvee_{i<j} z_i = z_j)),$$

$$\tau(\geq_n R_1.C_1 \sqsubseteq C_2) \coloneqq (\forall z)(\forall z_1)\dots(\forall z_n)(\bigwedge_{i=1}^{n}((z_i \in X^1_{C_1} \wedge \langle z, z_i\rangle \in X^3_{R_1}) \rightarrow \bigwedge_{i<j} z_i \neq z_j) \rightarrow z \in X^1_{C_2}),$$

$$\tau(C_1 \sqsubseteq \forall P_1.t_1) \coloneqq (\forall z_1)(\forall z_2)(z_1 \in X^1_{C_1} \rightarrow (\langle z_1, z_2\rangle \in X^3_{P_1} \rightarrow z_2 \in X^1_{t_1})),$$

$$\tau(\exists P_1.t_1 \sqsubseteq C_1) \coloneqq (\forall z_1)(\forall z_2)(((\langle z_1, z_2\rangle \in X^3_{P_1} \wedge z_2 \in X^1_{t_1}) \rightarrow z_1 \in X^1_{C_1}),$$

$$\tau(C_1 \equiv \exists P_1.\{e_d\}) \coloneqq (\forall z)(z \in X^1_{C_1} \leftrightarrow \langle z, x_{e_d}\rangle \in X^3_{P_1}),$$

$$\tau(C_1 \sqsubseteq \leq_n P_1.t_1) \coloneqq (\forall z)(\forall z_1)\dots(\forall z_{n+1})(z \in X^1_{C_1} \rightarrow (\bigwedge_{i=1}^{n+1}(z_i \in X^1_{t_1} \wedge \langle z, z_i\rangle \in X^3_{P_1}) \rightarrow \bigvee_{i<j} z_i = z_j)),$$

$$\tau(\geq_n P_1.t_1 \sqsubseteq C_1) \coloneqq (\forall z)(\forall z_1)\dots(\forall z_n)(\bigwedge_{i=1}^{n}((z_i \in X^1_{t_1} \wedge \langle z, z_i\rangle \in X^3_{P_1}) \rightarrow \bigwedge_{i<j} z_i \neq z_j) \rightarrow z \in X^1_{C_1}),$$

$$\tau(R_1 \equiv U) \coloneqq (\forall z_1)(\forall z_2)(\langle z_1, z_2\rangle \in X^3_{R_1} \leftrightarrow \langle z_1, z_2\rangle \in X^3_{U}),$$

$$\tau(R_1 \equiv \neg R_2) \coloneqq (\forall z_1)(\forall z_2)(\langle z_1, z_2\rangle \in X^3_{R_1} \leftrightarrow \neg(\langle z_1, z_2\rangle \in X^3_{R_2})),$$

$$\tau(R_1 \equiv R_2 \sqcup R_3) \coloneqq (\forall z_1)(\forall z_2)(\langle z_1, z_2\rangle \in X^3_{R_1} \leftrightarrow (\langle z_1, z_2\rangle \in X^3_{R_2} \vee \langle z_1, z_2\rangle \in X^3_{R_3})),$$

$$\tau(R_1 \equiv R_2^-) \coloneqq (\forall z_1)(\forall z_2)(\langle z_1, z_2\rangle \in X^3_{R_1} \leftrightarrow \langle z_2, z_1\rangle \in X^3_{R_2})),$$

$$\tau(R_1 \equiv id(C_1)) \coloneqq (\forall z_1)(\forall z_2)(\langle z_1, z_2\rangle \in X^3_{R_1} \leftrightarrow (z_1 \in X^1_{C_1} \wedge z_2 \in X^1_{C_1} \wedge z_1 = z_2)),$$

$$\tau(R_1 \equiv R_{2_{C_1|}}) \coloneqq (\forall z_1)(\forall z_2)(\langle z_1, z_2\rangle \in X^3_{R_1} \leftrightarrow (\langle z_1, z_2\rangle \in X^3_{R_2} \wedge z_1 \in X^1_{C_1})),$$

$$\tau(R_1 \dots R_n \sqsubseteq R_{n+1}) \coloneqq (\forall z)(\forall z_1)\dots(\forall z_n) (((\langle z, z_1\rangle \in X^3_{R_1} \wedge \dots \wedge \langle z_{n-1}, z_n\rangle \in X^3_{R_n}) \rightarrow \langle z, z_n\rangle \in X^3_{R_{n+1}}),$$

$$\tau(\mathsf{Ref}(R_1)) \coloneqq (\forall z)(\langle z, z\rangle \in X^3_{R_1}),$$

$$\tau(\mathsf{Irref}(R_1)) \coloneqq (\forall z)(\neg(\langle z, z\rangle \in X^3_{R_1})),$$

$$\tau(\mathsf{Fun}(R_1)) \coloneqq (\forall z_1)(\forall z_2)(\forall z_3)(((\langle z_1, z_2\rangle \in X^3_{R_1} \wedge \langle z_1, z_3\rangle \in X^3_{R_1}) \rightarrow z_2 = z_3),$$

$$\tau(P_1 \equiv P_2) \coloneqq (\forall z_1)(\forall z_2)(\langle z_1, z_2\rangle \in X^3_{P_1} \leftrightarrow \langle z_1, z_2\rangle \in X^3_{P_2}),$$

$$\tau(P_1 \equiv \neg P_2) \coloneqq (\forall z_1)(\forall z_2)(\langle z_1, z_2\rangle \in X^3_{P_1} \leftrightarrow \neg(\langle z_1, z_2\rangle \in X^3_{P_2})),$$

$$\tau(P_1 \sqsubseteq P_2) \coloneqq (\forall z_1)(\forall z_2)(\langle z_1, z_2\rangle \in X^3_{P_1} \rightarrow \langle z_1, z_2\rangle \in X^3_{P_2}),$$

$$\tau(\mathsf{Fun}(P_1)) \coloneqq (\forall z_1)(\forall z_2)(\forall z_3)(((\langle z_1, z_2\rangle \in X^3_{P_1} \wedge \langle z_1, z_3\rangle \in X^3_{P_1}) \rightarrow z_2 = z_3),$$

$$\tau(P_1 \equiv P_{2_{C_1|}}) \coloneqq (\forall z_1)(\forall z_2)(\langle z_1, z_2\rangle \in X^3_{P_1} \leftrightarrow (\langle z_1, z_2\rangle \in X^3_{P_2} \wedge z_1 \in X^1_{C_1})),$$

$$\tau(P_1 \equiv P_{2_{|t_1}}) \coloneqq (\forall z_1)(\forall z_2)(\langle z_1, z_2\rangle \in X^3_{P_1} \leftrightarrow (\langle z_1, z_2\rangle \in X^3_{P_2} \wedge z_2 \in X^1_{t_1})),$$

$$\tau(P_1 \equiv P_{2_{C_1|t_1}}) \coloneqq (\forall z_1)(\forall z_2)(\langle z_1, z_2\rangle \in X^3_{P_1} \leftrightarrow (\langle z_1, z_2\rangle \in X^3_{P_2} \wedge z_1 \in X^1_{C_1} \wedge z_2 \in X^1_{t_1})),$$

$$\tau(t_1 \equiv t_2) \coloneqq (\forall z)(z \in X^1_{t_1} \leftrightarrow z \in X^1_{t_2}),$$

$$\tau(t_1 \equiv \neg t_2) \coloneqq (\forall z)(z \in X^1_{t_1} \leftrightarrow \neg(z \in X^1_{t_2})),$$

$$\tau(t_1 \equiv t_2 \sqcup t_3) \coloneqq (\forall z)(z \in X^1_{t_1} \leftrightarrow (z \in X^1_{t_2} \vee z \in X^1_{t_3})),$$

$$\tau(t_1 \equiv t_2 \sqcap t_3) \coloneqq (\forall z)(z \in X^1_{t_1} \leftrightarrow (z \in X^1_{t_2} \wedge z \in X^1_{t_3})),$$

$$\tau(t_1 \equiv \{e_d\}) \coloneqq (\forall z)(z \in X^1_{t_1} \leftrightarrow z = x_{e_d}),$$

$$\tau(a : C_1) \coloneqq x_a \in X^1_{C_1},\ \tau((a,b) : R_1) \coloneqq \langle x_a, x_b\rangle \in X^3_{R_1},$$

$$\tau((a,b) : \neg R_1) \coloneqq \neg(\langle x_a, x_b\rangle \in X^3_{R_1}),$$

$$\tau(a = b) \coloneqq x_a = x_b,\ \tau(a \neq b) \coloneqq \neg(x_a = x_b),$$

$$\tau(e_d : t_1) \coloneqq x_{e_d} \in X^1_{t_1},$$

$$\tau((a, e_d) : P_1) \coloneqq \langle x_a, x_{e_d}\rangle \in X^3_{P_1},\ \tau((a, e_d) : \neg P_1) \coloneqq \neg(\langle x_a, x_{e_d}\rangle \in X^3_{P_1}),$$

$$\tau(\alpha \wedge \beta) \coloneqq \tau(\alpha) \wedge \tau(\beta).$$

Let $\mathsf{cpt}_{\mathcal{KB}}$, $\mathsf{arl}_{\mathcal{KB}}$, $\mathsf{crl}_{\mathcal{KB}}$, and $\mathsf{ind}_{\mathcal{KB}}$ be, respectively, the sets of concept, of abstract role, of concrete role, and of individual names in the $\mathcal{DL}^4_{\mathbf{D}}$-KB $\mathcal{KB}$. Moreover, let $N^{\mathcal{KB}}_D \subseteq N_D$ be the set of data types in $\mathcal{KB}$, $N^{\mathcal{KB}}_F$ a restriction of N_F assigning to every $d \in N^{\mathcal{KB}}_{\mathbf{D}}$ the set $N^{\mathcal{KB}}_F(d)$ of facets in $N_F(d)$ and in $\mathcal{KB}$. Analogously, let $N^{\mathcal{KB}}_C$ be a restriction of the function N_C associating to every $d \in N^{\mathcal{KB}}_{\mathbf{D}}$ the set $N^{\mathcal{KB}}_C(d)$ of constants contained in $N_C(d)$ and in $\mathcal{KB}$. Finally, for every data type $d \in N_D^{\mathcal{KB}}$, let $\mathsf{bf}_{\mathcal{KB}}^{\mathbf{D}}(d)$ be the set of facet expressions for d occurring in $\mathcal{KB}$ and not in $N_F(d) \cup \{\top^d, \bot_d\}$. We define the 4LQS$^{\mathsf{R}}$-formula $\varphi_{\mathcal{KB}}$ expressing the consistency of $\mathcal{KB}$ as follows:

$$\varphi_{\mathcal{KB}} \coloneqq \bigwedge_{i=1}^{12} \psi_i \wedge \bigwedge_{H \in \mathcal{KB}} \tau(H)\,,$$

where

$$\psi_1 \coloneqq (\forall z)(z \in X^1_{\mathbf{I}} \leftrightarrow \neg(z \in X^1_{\mathbf{D}})) \wedge (\forall z)(z \in X^1_{\mathbf{I}} \vee z \in X^1_{\mathbf{D}}) \wedge \neg(\forall z)\neg(z \in X^1_{\mathbf{I}}) \wedge \neg(\forall z)\neg(z \in X^1_{\mathbf{D}}),$$

$$\psi_2 \coloneqq (\forall z)(z \in X^1_{\mathbf{I}} \leftrightarrow z \in X^1_{\top}) \wedge (\forall z)\neg(z \in X_{\bot}),$$

$$\psi_3 \coloneqq \bigwedge_{A \in \mathsf{cpt}_{\mathcal{KB}}} (\forall z)(z \in X^1_A \rightarrow z \in X^1_{\mathbf{I}}),$$

$$\psi_4 \coloneqq \Big(\bigwedge_{d \in N_D^{\mathcal{KB}}} ((\forall z)(z \in X^1_d \rightarrow z \in X^1_{\mathbf{D}}) \wedge \neg(\forall z)\neg(z \in X^1_d)) \wedge (\forall z)\Big(\bigwedge_{(d_i, d_j \in N_D^{\mathcal{KB}}, i<j)} (z \in X^1_{d_i} \leftrightarrow \neg(z \in X^1_{d_j}))\Big)\Big),$$

$$\psi_5 \coloneqq \bigwedge_{d \in N_D^{\mathcal{KB}}} ((\forall z)(z \in X^1_d \leftrightarrow z \in X^1_{\top^d}) \wedge (\forall z)\neg(z \in X^1_{\bot_d})),$$

$$\psi_6 \coloneqq \bigwedge_{d \in N_D^{\mathcal{KB}}} \bigwedge_{f_d \in N_F^{\mathcal{KB}}(d)} (\forall z)(z \in X^1_{f_d} \rightarrow z \in X^1_d),$$

$$\psi_7 \coloneqq (\forall z_1)(\forall z_2)((z_1 \in X^1_{\mathbf{I}} \wedge z_2 \in X^1_{\mathbf{I}}) \leftrightarrow \langle z_1, z_2 \rangle \in X^3_U),$$

$$\psi_8 \coloneqq \bigwedge_{R \in \mathsf{arl}_{\mathcal{KB}}} (\forall z_1)(\forall z_2)(\langle z_1, z_2 \rangle \in X^3_R \rightarrow (z_1 \in X^1_{\mathbf{I}} \wedge z_2 \in X^1_{\mathbf{I}})),$$

$$\psi_9 \coloneqq \bigwedge_{T \in \mathsf{crl}_{\mathcal{KB}}} (\forall z_1)(\forall z_2)(\langle z_1, z_2 \rangle \in X^3_T \rightarrow (z_1 \in X^1_{\mathbf{I}} \wedge z_2 \in X^1_{\mathbf{D}})),$$

$$\psi_{10} \coloneqq \bigwedge_{a \in \mathsf{ind}_{\mathcal{KB}}} (x_a \in X^1_{\mathbf{I}}) \wedge \bigwedge_{d \in N_D^{\mathcal{KB}}} \bigwedge_{e_d \in N_C^{\mathcal{KB}}(d)} x_{e_d} \in X^1_d,$$

$$\psi_{11} \coloneqq \bigwedge_{e_{d_1}, \ldots, e_{d_n} \text{ in } \mathcal{KB}} (\forall z)(z \in X^1_{\{e_{d_1}, \ldots, e_{d_n}\}} \leftrightarrow \bigvee_{i=1}^n (z = x_{e_{d_i}})) \wedge \bigwedge_{a_1, \ldots, a_n \text{ in } \mathcal{KB}} (\forall z)(z \in X^1_{\{a_1, \ldots, a_n\}} \leftrightarrow \bigvee_{i=1}^n (z = x_{a_i})),$$

$$\psi_{12} \coloneqq \bigwedge_{d \in N_{\mathbf{D}}^{\mathcal{KB}}} \bigwedge_{\psi_d \in \mathsf{bf}_{\mathcal{KB}}^{\mathbf{D}}(d)} (\forall z)(z \in X^1_{\psi_d} \leftrightarrow z \in \sigma(X^1_{\psi_d})),$$

with σ the transformation function from 4LQS$^{\mathsf{R}}$-variables of sort 1 to 4LQS$^{\mathsf{R}}$-formulae recursively defined, for $d \in N_{\mathbf{D}}^{\mathcal{KB}}$, by

$$\sigma(X^1_{\psi_d}) \coloneqq \begin{cases} X^1_{\psi_d} & \text{if } \psi_d \in N_F^{\mathcal{KB}}(d) \cup \{\top^d, \bot_d\} \\ \neg\sigma(X^1_{\chi_d}) & \text{if } \psi_d = \neg\chi_d \\ \sigma(X^1_{\chi_d}) \wedge \sigma(X^1_{\varphi_d}) & \text{if } \psi_d = \chi_d \wedge \varphi_d \\ \sigma(X^1_{\chi_d}) \vee \sigma(X^1_{\varphi_d}) & \text{if } \psi_d = \chi_d \vee \varphi_d \,. \end{cases}$$

In the above formulae, the variable $X^1_{\mathbf{I}}$ denotes the set of individuals $\mathbf{I}$, X^1_d a data type $d \in N_D^{\mathcal{KB}}$, $X^1_{\mathbf{D}}$ a superset of the union of data types in $N_D^{\mathcal{KB}}$, $X^1_{\top_d}$ and $X^1_{\bot_d}$ the constants $\top_d$ and $\bot_d$, and $X^1_{f_d}$, $X^1_{\psi_d}$ a facet f_d and a facet expression ψ_d, for $d \in N_D^{\mathcal{KB}}$, respectively. In addition, X^1_A, X^3_R, X^3_T denote a concept name A, an abstract role name R, and a concrete role name T occurring in $\mathcal{KB}$, respectively. Finally, $X^1_{\{e_{d_1},\ldots,e_{d_n}\}}$ denotes a data range $\{e_{d_1},\ldots,e_{d_n}\}$ occurring in $\mathcal{KB}$, and $X^1_{\{a_1,\ldots,a_n\}}$ a finite set $\{a_1,\ldots,a_n\}$ of nominals in $\mathcal{KB}$.

Clearly, the constraints ψ_1-ψ_{12} have been introduced to guarantee that each model of $\varphi_{\mathcal{KB}}$ can be transformed into a $\mathcal{DL}^4_{\mathbf{D}}$-interpretation.

We are now ready to introduce Theorem 1 which shows that the consistency problem for $\mathcal{KB}$ is reducible to the satisfiability problem for $\varphi_{\mathcal{KB}}$.

Theorem 1. *Let $\mathcal{KB}$ be a $\mathcal{DL}^4_{\mathbf{D}}$-KB. Then, one can construct a 4LQS$^{\mathsf{R}}$-formula $\varphi_{\mathcal{KB}}$ such that $\varphi_{\mathcal{KB}}$ is satisfiable if and only if $\mathcal{KB}$ is consistent. Hence the consistency problem for $\mathcal{DL}^4_{\mathbf{D}}$-KBs is decidable.*

Proof. Let us first assume that $\varphi_{\mathcal{KB}}$ is satisfiable. It is not hard to see that $\varphi_{\mathcal{KB}}$ is satisfied by a 4LQS$^{\mathsf{R}}$-model of the form $\boldsymbol{\mathcal{M}} = (D_1 \cup D_2, M)$, where:

- D_1 and D_2 are disjoint non-empty sets and $\bigcup_{d \in N_D^{\mathcal{KB}}} d^{\mathbf{D}} \subseteq D_2$,
- $MX^1_{\mathbf{I}} \coloneqq D_1$,
- $MX^1_{\mathbf{D}} \coloneqq D_2$,
- $MX^1_d \coloneqq d^{\mathbf{D}}$, for every $d \in N_D^{\mathcal{KB}}$,
- $MX^1_{f_d} \coloneqq f_d^{\mathbf{D}}$, for every $f_d \in N_F^{\mathcal{KB}}(d)$, with $d \in N_D^{\mathcal{KB}}$.

Exploiting the fact that $\boldsymbol{\mathcal{M}}$ satisfies the constraints ψ_1-ψ_{12}, it is then possible to define a $\mathcal{DL}^4_{\mathbf{D}}$-interpretation $\mathbf{I}_{\boldsymbol{\mathcal{M}}} = (\Delta^{\mathbf{I}}, \Delta_{\mathbf{D}}, \cdot^{\mathbf{I}})$, by putting

- $\Delta^{\mathbf{I}} \coloneqq MX^1_{\mathbf{I}}$,
- $\Delta_{\mathbf{D}} \coloneqq MX^1_{\mathbf{D}}$,
- $A^{\mathbf{I}} \coloneqq MX^1_A$, for every concept name $A \in \mathsf{cpt}_{\mathcal{KB}}$,

- $S^{\mathbf{I}} := MX^3_S$, for every abstract role name $S \in \mathsf{arl}_{\mathcal{KB}}$,

- $T^{\mathbf{I}} := MX^3_T$, for every concrete role name $T \in \mathsf{crl}_{\mathcal{KB}}$, and

- $a^{\mathbf{I}} := Mx_a$, for every individual $a \in \mathsf{ind}_{\mathcal{KB}}$.

- $e_d^D := Mx_{e_d}$, for every constant $e_d \in N_C^{\mathcal{KB}}(d)$ with $d \in N_D^{\mathcal{KB}}$.

Since $\boldsymbol{\mathcal{M}} \models \bigwedge_{H\in\mathcal{KB}} \tau(H)$ and, as it can be easily checked, $\mathbf{I}_{\boldsymbol{\mathcal{M}}} \models_{\mathbf{D}} H$ if and only if $\boldsymbol{\mathcal{M}} \models \tau(H)$ for every statement $H \in \mathcal{KB}$, we plainly have $\mathbf{I}_{\boldsymbol{\mathcal{M}}} \models_{\mathbf{D}} \mathcal{KB}$, namely $\mathcal{KB}$ is consistent, as we wished to prove.

Conversely, let $\mathcal{KB}$ be a consistent $\mathcal{DL}^4_{\mathbf{D}}$-KB. Then, there is a $\mathcal{DL}^4_{\mathbf{D}}$-interpretation $\mathbf{I} = (\Delta^{\mathbf{I}}, \Delta_{\mathbf{D}}, \cdot^{\mathbf{I}})$ such that $\mathbf{I} \models_{\mathbf{D}} \mathcal{KB}$. We show how to construct, out of the data type map $\mathbf{D}$ and the $\mathcal{DL}^4_{\mathbf{D}}$-interpretation $\mathbf{I}$, a 4LQS$^{\mathsf{R}}$-interpretation $\boldsymbol{\mathcal{M}}_{\mathbf{I},\mathbf{D}} = (D_{\mathbf{I},\mathbf{D}}, M_{\mathbf{I},\mathbf{D}})$ which satisfies $\varphi_{\mathcal{KB}}$. Let us put $D_{\mathbf{I},\mathbf{D}} := \Delta^{\mathbf{I}} \cup \Delta_{\mathbf{D}}$ and define $M_{\mathbf{I},\mathbf{D}}$ by putting

- $M_{\mathbf{I},\mathbf{D}} X^1_{\mathbf{I}} := \Delta^{\mathbf{I}}$,

- $M_{\mathbf{I},\mathbf{D}} X^1_{\mathbf{D}} := \Delta_{\mathbf{D}}$,

- $M_{\mathbf{I},\mathbf{D}} X^3_U := U^{\mathbf{I}}$,

- $M_{\mathbf{I},\mathbf{D}} X^1_{dr} := dr^{\mathbf{D}}$, for every variable X^1_{dr} in $\varphi_{\mathcal{KB}}$ denoting a data range dr occurring in $\mathcal{KB}$,

- $M_{\mathbf{I},\mathbf{D}} X^1_A := A^{\mathbf{I}}$, for every X^1_A in $\varphi_{\mathcal{KB}}$ denoting a concept name in $\mathcal{KB}$, and

- $M_{\mathbf{I},\mathbf{D}} X^3_S := S^{\mathbf{I}}$, for every X^3_S in $\varphi_{\mathcal{KB}}$ denoting an abstract role name in $\mathcal{KB}$.

The variable X^3_T, denoting concrete role names, and the variables x_a, x_{e_d}, denoting individuals and data type constants, respectively, are interpreted in a similar way. From the definitions of $\mathbf{D}$ and $\mathbf{I}$, it follows easily that $\boldsymbol{\mathcal{M}}_{\mathbf{I},\mathbf{D}}$ satisfies the formulae ψ_1-ψ_{12} and $\tau(H)$, for every statement $H \in \mathcal{KB}$, and, therefore, that $\boldsymbol{\mathcal{M}}_{\mathbf{I},\mathbf{D}}$ is a model for $\varphi_{\mathcal{KB}}$. □

Remark. *For a fixed positive integer h, a $\mathcal{DL}^4_{\mathbf{D}}$-KB $\mathcal{KB}$ is said to be h-*restricted *if an atom of any of the forms*

$$R_1 \dots R_{n_1} \sqsubseteq R, \qquad \geq_{n_2} R.C_1 \sqsubseteq C_2, \qquad \geq_{n_3} P.t_1 \sqsubseteq t_2,$$
$$C_1 \sqsubseteq \leq_{n_4} R.C_2, \qquad t_1 \sqsubseteq \leq_{n_5} P.t_2,$$

occurs in $\mathcal{KB}$ and only if $n_i \leq h$, for $i = 1, \dots, 5$.

It turns out that by using the mapping function τ and some additional constraints, the consistency problem for a h-restricted $\mathcal{DL}^4_{\mathbf{D}}$-KB $\mathcal{KB}$ can be expressed by a formula $\varphi'_{\mathcal{KB}}$ such that *(i)* $\varphi'_{\mathcal{KB}}$ *belongs to the sub-language* $(\mathsf{4LQS^R})^h$ *of* $\mathsf{4LQS^R}$*, whose satisfiability problem is **NP**-complete (see [26] for details), and*

(ii) the size of $\varphi'_{\mathcal{KB}}$ *is polynomially related to that of* $\mathcal{KB}$*.*

*From (i) and (ii) above, and from the **NP**-completeness of the satisfiability problem for propositional logic, it follows immediately that the consistency problem for h-restricted* $\mathcal{DL}^4_{\mathbf{D}}$*-knowledge bases is **NP**-complete.*

Notice also that h-restricted $\mathcal{DL}^4_{\mathbf{D}}$*-KBs are quite expressive: for instance, in [108] It has been shown that the ontology* Ontoceramic*, for ceramics classification, is representable in* $(\mathsf{4LQS^R})^3$ *and, much in the same way, it can be shown that it is representable as a* 3*-restricted* $\mathcal{DL}^4_{\mathbf{D}}$*-KB.*

3.2 The description logic $\mathcal{DL}^{4,\times}_{\mathbf{D}}$

The DL $\mathcal{DL}^{\times}(\mathbf{D})\langle\mathsf{4LQS^R}\rangle$ (more simply referred to as $\mathcal{DL}^{4,\times}_{\mathbf{D}}$) is the extension of the logic $\mathcal{DL}^4_{\mathbf{D}}$ presented in Section 3.1 in which Boolean operations on concrete roles and the product of concepts are admitted. In analogy with $\mathcal{DL}^4_{\mathbf{D}}$, the logic $\mathcal{DL}^{4,\times}_{\mathbf{D}}$ supports concept constructs such as full negation, union and intersection of concepts, concept domain and range, existential quantification and minimum cardinality on the left-hand-side of inclusion axioms, role constructs such as role chains on the left-hand-side of inclusion axioms, union, intersection, and complement of roles, and properties on roles such as transitivity, symmetry, reflexivity, and irreflexivity.

Let $\mathbf{D} = (N_D, N_C, N_F, \cdot^{\mathbf{D}})$ be a *data type map* as defined in Section 2.1, and let $\mathbf{R_A}$, $\mathbf{R_D}$, $\mathbf{C}$, $\mathbf{Ind}$ be denumerable pairwise disjoint sets of abstract role names, concrete role names, concept names, and individual names, respectively. We assume that the set of abstract role names $\mathbf{R_A}$ contains a name U denoting the universal role.
(a) $\mathcal{DL}^{4,\times}_{\mathbf{D}}$-data type, (b) $\mathcal{DL}^{4,\times}_{\mathbf{D}}$-concept, (c) $\mathcal{DL}^{4,\times}_{\mathbf{D}}$-abstract role, and (d) $\mathcal{DL}^{4,\times}_{\mathbf{D}}$-concrete role terms are constructed according to the following syntax rules:

(a) $t_1, t_2 \longrightarrow dr \mid \neg t_1 \mid t_1 \sqcap t_2 \mid t_1 \sqcup t_2 \mid \{e_d\}\,,$

(b) $C_1, C_2 \longrightarrow A \mid \top \mid \bot \mid \neg C_1 \mid C_1 \sqcup C_2 \mid C_1 \sqcap C_2 \mid \{a\} \mid \exists R.Self \mid \exists R.\{a\} \mid \exists P.\{e_d\}\,,$

(c) $R_1, R_2 \longrightarrow S \mid U \mid R_1^- \mid \neg R_1 \mid R_1 \sqcup R_2 \mid R_1 \sqcap R_2 \mid R_{C_1|} \mid R_{|C_1} \mid R_{C_1\,|\,C_2} \mid id(C) \mid C_1 \times C_2\,,$

(d) $P_1, P_2 \longrightarrow T \mid \neg P_1 \mid P_1 \sqcup P_2 \mid P_1 \sqcap P_2 \mid P_{C_1|} \mid P_{|t_1} \mid P_{C_1|t_1}\,,$

where dr is a data range for $\mathbf{D}$, t_1, t_2 are data type terms, e_d is a constant in $N_C(d)$, a is an individual name, A is a concept name, C_1, C_2 are $\mathcal{DL}^{4,\times}_{\mathbf{D}}$-concept terms, S is an abstract role name, R, R_1, R_2 are $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-abstract role terms, T is a concrete role name, and P, P_1, P_2 are $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-concrete role terms.

A $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KB is a triple $\mathcal{KB} = (\mathcal{R}, \mathcal{T}, \mathcal{A})$ such that $\mathcal{R}$ is a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-*RBox*, $\mathcal{T}$ is a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-*TBox*, and $\mathcal{A}$ a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-*ABox*. These are defined as follows.

A $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-*RBox* extends the definition of $\mathcal{DL}_{\mathbf{D}}^{4}$-*RBox* by admitting also the following statements:

$$R_1 \equiv C_1 \times C_2, \qquad \mathsf{Dis}(P_1, P_2),$$

where C_1, C_2 are $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-abstract concept terms and P_1, P_2 are $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-concrete role terms, whereas definition of $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-*TBox* and of $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-*ABox* coincide with the definition of $\mathcal{DL}_{\mathbf{D}}^{4}$-*TBox* and of $\mathcal{DL}_{\mathbf{D}}^{4}$-*ABox*, respectively (see Section 3.1).

The semantics of $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ extends the semantics of $\mathcal{DL}_{\mathbf{D}}^{4}$ with the interpretation of the axioms reported in Table 11.

Name	Syntax	Semantics
concept Cartesian product	$C_1 \times C_2$	$(C_1 \times C_2)^I = C_1^I \times C_2^I$
concrete role union	$P_1 \sqcup P_2$	$P_1^{\mathbf{I}} \cup P_2^{\mathbf{I}}$
concrete role intersection	$P_1 \sqcap P_2$	$P_1^{\mathbf{I}} \cap P_2^{\mathbf{I}}$
disjoint concrete role	$\mathsf{Dis}(P_1, P_2)$	$\mathbf{I} \models_{\mathbf{D}} \mathsf{Dis}(P_1, P_2) \iff P_1^{\mathbf{I}} \cap P_2^{\mathbf{I}} = \emptyset$

Table 11: Semantics of axioms specific to $\mathcal{DL}_{\mathbf{D}}^{4,\times}$.

Decidability of the consistency problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KBs is proved in a similar way to that of $\mathcal{DL}_{\mathbf{D}}^{4}$-KBs. Before presenting the theorem, we define the 4LQS$^{\mathsf{R}}$-formula $\varphi_{\mathcal{KB}}$ representing the set-theoretical representation of the $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KB $\mathcal{KB}$.

As a preliminary step, notice that the statements of $\mathcal{KB}$ to be considered are the following:

- $C_1 \equiv \top$, $C_1 \equiv \neg C_2$, $C_1 \equiv C_2 \sqcup C_3$, $C_1 \equiv \{a\}$, $C_1 \sqsubseteq \forall R_1.C_2$, $\exists R_1.C_1 \sqsubseteq C_2$, $\geq_n R_1.C_1 \sqsubseteq C_2$, $C_1 \sqsubseteq \leq_n R_1.C_2$, $C_1 \sqsubseteq \forall P_1.t_1$, $\exists P_1.t_1 \sqsubseteq C_1$, $\geq_n P_1.t_1 \sqsubseteq C_1$, $C_1 \sqsubseteq \leq_n P_1.t_1$,
- $R_1 \equiv U$, $R_1 \equiv \neg R_2$, $R_1 \equiv R_2 \sqcup R_3$, $R_1 \equiv R_2^-$, $R_1 \equiv id(C_1)$, $R_1 \equiv R_{2_{C_1|}}$, $R_1 \dots R_n \sqsubseteq R_{n+1}$, $\mathsf{Ref}(R_1)$, $\mathsf{Irref}(R_1)$, $\mathsf{Dis}(R_1, R_2)$, $\mathsf{Fun}(R_1)$, $R_1 \equiv C_1 \times C_2$,

- $P_1 \equiv P_2$, $P_1 \equiv \neg P_2$, $P_1 \equiv P_2 \sqcup P_3$, $P_1 \sqsubseteq P_2$, $\mathsf{Fun}(P_1)$, $P_1 \equiv P_{2_{C_1|}}$, $P_1 \equiv P_{2_{C_1|t_1}}$, $P_1 \equiv P_{2_{|t_1}}$,

- $t_1 \equiv t_2$, $t_1 \equiv \neg t_2$, $t_1 \equiv t_2 \sqcup t_3$, $t_1 \equiv \{e_d\}$,

- $a : C_1$, $(a, b) : R_1$, $(a, b) : \neg R_1$, $a = b$, $a \neq b$,

- $e_d : t_1$, $(a, e_d) : P_1$, $(a, e_d) : \neg P_1$.

In order to define the 4LQS$^{\mathsf{R}}$-formula $\varphi_{\mathcal{KB}}$, we shall make use of a mapping θ from the $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-statements (and their conjunctions) listed above into 4LQS$^{\mathsf{R}}$-formulae. Specifically, θ differs from the mapping function τ defined in Section 3.1 as θ considers Boolean operations on concrete roles and the product of concepts. In addition, θ constructs 4LQS$^{\mathsf{R}}$-formulae in *Conjunctive Normal Form* (*CNF*), which are suitable for the algorithms presented in Chapter 4 to solve the main reasoning tasks for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ and for implementation purpose (see Chapter 5).

Then the mapping θ is defined as follows:

$$\theta(C_1 \equiv \top) \coloneqq (\forall z)((\neg(z \in X^1_{C_1}) \vee z \in X^1_{\top}) \wedge (\neg(z \in X^1_{\top}) \vee z \in X^1_{C_1})),$$

$$\theta(C_1 \equiv \neg C_2) \coloneqq (\forall z)((\neg(z \in X^1_{C_1}) \vee \neg(z \in X^1_{C_2})) \wedge (z \in X^1_{C_2} \vee z \in X^1_{C_1})),$$

$$\theta(C_1 \equiv C_2 \sqcup C_3) \coloneqq (\forall z)((\neg(z \in X^1_{C_1}) \vee (z \in X^1_{C_2} \vee z \in X^1_{C_3}))\wedge ((\neg(z \in X^1_{C_2}) \vee z \in X^1_{C_1}) \wedge (\neg(z \in X^1_{C_3}) \vee z \in X^1_{C_1})),$$

$$\theta(C_1 \equiv \{a\}) \coloneqq (\forall z)(\neg(z \in X^1_{C_1}) \vee z = x_a) \wedge (\neg(z = x_a) \vee z \in X^1_{C_1}),$$

$$\theta(C_1 \sqsubseteq \forall R_1.C_2) \coloneqq (\forall z_1)(\forall z_2)(\neg(z_1 \in X^1_{C_1}) \vee (\neg(\langle z_1, z_2\rangle \in X^3_{R_1})\vee z_2 \in X^1_{C_2})),$$

$$\theta(\exists R_1.C_1 \sqsubseteq C_2) \coloneqq (\forall z_1)(\forall z_2)((\neg(\langle z_1, z_2\rangle \in X^3_{R_1})\vee \neg(z_2 \in X^1_{C_1})) \vee z_1 \in X^1_{C_2}),$$

$$\theta(C_1 \equiv \exists R_1.\{a\}) \coloneqq (\forall z)((\neg(z \in X^1_{C_1}) \vee \langle z, x_a\rangle \in X^3_{R_1})\wedge (\neg(\langle z, x_a\rangle \in X^3_{R_1}) \vee z \in X^1_{C_1})),$$

$$\theta(C_1 \sqsubseteq \leq_n R_1.C_2) \coloneqq (\forall z)(\forall z_1)\ldots(\forall z_{n+1})(\neg(z \in X^1_{C_1})\vee (\bigwedge_{i=1}^{n+1}(\neg(z_i \in X_{C_2}) \vee \neg(\langle z, z_i\rangle \in X^3_{R_1}) \vee \bigvee_{i<j} z_i = z_j)),$$

$$\theta(\geq_n R_1.C_1 \sqsubseteq C_2) \coloneqq (\forall z)(\forall z_1)\dots(\forall z_n)(\bigwedge_{i=1}^{n}((\neg(z_i \in X^1_{C_1})\vee \neg(\langle z, z_i\rangle \in X^3_{R_1})) \vee \bigvee_{i<j} z_i = z_j) \vee z \in X^1_{C_2}),$$

$$\theta(C_1 \sqsubseteq \forall P_1.t_1) \coloneqq (\forall z_1)(\forall z_2)(\neg(z_1 \in X^1_{C_1}) \vee (\neg(\langle z_1, z_2\rangle \in X^3_{P_1})\vee z_2 \in X^1_{t_1})),$$

$$\theta(\exists P_1.t_1 \sqsubseteq C_1) \coloneqq (\forall z_1)(\forall z_2)((\neg(\langle z_1, z_2\rangle \in X^3_{P_1}) \vee \neg(z_2 \in X^1_{t_1}))\vee z_1 \in X^1_{C_1}),$$

$$\theta(C_1 \equiv \exists P_1.\{e_d\}) \coloneqq (\forall z)((\neg(z \in X^1_{C_1}) \vee \langle z, x_{e_d}\rangle \in X^3_{P_1})\wedge (\neg(\langle z, x_{e_d}\rangle \in X^3_{P_1}) \vee z \in X^1_{C_1})),$$

$$\theta(C_1 \sqsubseteq \leq_n P_1.t_1) \coloneqq (\forall z)(\forall z_1)\dots(\forall z_{n+1})(\neg(z \in X^1_{C_1})\vee (\bigwedge_{i=1}^{n+1}(\neg(z_i \in X_{t_1}) \vee \neg(\langle z, z_i\rangle \in X^3_{P_1}) \vee \bigvee_{i<j} z_i = z_j)),$$

$$\theta(\geq_n P_1.t_1 \sqsubseteq C_1) \coloneqq (\forall z)(\forall z_1)\dots(\forall z_n)(\bigwedge_{i=1}^{n}((\neg(z_i \in X^1_{t_1})\vee \neg(\langle z, z_i\rangle \in X^3_{P_1})) \vee \bigvee_{i<j} z_i = z_j) \vee z \in X^1_{C_1}),$$

$$\theta(R_1 \equiv U) \coloneqq (\forall z_1)(\forall z_2)((\neg(\langle z_1, z_2\rangle \in X^3_{R_1}) \vee \langle z_1, z_2\rangle \in X^3_{U})\wedge (\neg(\langle z_1, z_2\rangle \in X^3_{U}) \vee \langle z_1, z_2\rangle \in X^3_{R_1})),$$

$$\theta(R_1 \equiv \neg R_2) \coloneqq (\forall z_1)(\forall z_2)((\neg(\langle z_1, z_2\rangle \in X^3_{R_1}) \vee \neg(\langle z_1, z_2\rangle \in X^3_{R_2}))\wedge (\langle z_1, z_2\rangle \in X^3_{R_2} \vee \neg(\langle z_1, z_2\rangle \in X^3_{R_1}))),$$

$$\theta(R \equiv C_1 \times C_2) \coloneqq (\forall z_1)(\forall z_2)(\neg(\langle z_1, z_2\rangle \in X^3_{R}) \vee z_1 \in X^1_{C_1})\wedge (\neg(\langle z_1, z_2\rangle \in X^3_{R}) \vee z_2 \in X^1_{C_2}) \wedge ((\neg(z_1 \in X^1_{C_1})\vee \neg(z_2 \in X^1_{C_2})) \vee \langle z_1, z_2\rangle \in X^3_{R}))$$

$$\theta(R_1 \equiv R_2 \sqcup R_3) \coloneqq (\forall z_1)(\forall z_2)((\neg(\langle z_1, z_2\rangle \in X^3_{R_1}) \vee (\langle z_1, z_2\rangle \in X^3_{R_2}\vee \langle z_1, z_2\rangle \in X^3_{R_3})) \wedge ((\neg(\langle z_1, z_2\rangle \in X^3_{R_2}) \vee \langle z_1, z_2\rangle \in X^3_{R_1})\wedge ((\neg(\langle z_1, z_2\rangle \in X^3_{R_3}) \vee \langle z_1, z_2\rangle \in X^3_{R_1})))),$$

$$\theta(R_1 \equiv R_2^-) \coloneqq (\forall z_1)(\forall z_2)((\neg(\langle z_1, z_2\rangle \in X^3_{R_1}) \vee \langle z_2, z_1\rangle \in X^3_{R_2})\wedge (\neg(\langle z_2, z_1\rangle \in X^3_{R_2}) \vee \langle z_1, z_2\rangle \in X^3_{R_1})),$$

$$\theta(R_1 \equiv id(C_1)) \coloneqq (\forall z_1)(\forall z_2)(((\neg(\langle z_1, z_2\rangle \in X^3_{R_1}) \vee z_1 \in X^1_{C_1}) \wedge (\neg(\langle z_1, z_2\rangle \in X^3_{R_1}) \vee z_2 \in X^1_{C_1}) \wedge (\neg(\langle z_1, z_2\rangle \in X^3_{R_1}) \vee z_1 = z_2)) \wedge ((\neg(z_1 \in X^1_{C_1}) \vee \neg(z_2 \in X^1_{C_1}) \vee z_1 \neq z_2) \vee \langle z_1, z_2\rangle \in X^3_{R_1})),$$

$$\theta(R_1 \equiv R_{2_{C_1|}}) \coloneqq (\forall z_1)(\forall z_2)(((\neg(\langle z_1, z_2\rangle \in X^3_{R_1}) \vee \langle z_1, z_2\rangle \in X^3_{R_2}) \wedge (\neg(\langle z_1, z_2\rangle \in X^3_{R_1}) \vee z_1 \in X^1_{C_1})) \wedge ((\neg(\langle z_1, z_2\rangle \in X^3_{R_2}) \vee \neg(z_1 \in X^1_{C_1})) \vee \langle z_1, z_2\rangle \in X^3_{R_1})),$$

$$\theta(R_1 \dots R_n \sqsubseteq R_{n+1}) \coloneqq (\forall z)(\forall z_1)\dots(\forall z_n)((\neg(\langle z, z_1\rangle \in X^3_{R_1}) \vee \dots \vee \neg(\langle z_{n-1}, z_n\rangle \in X^3_{R_n})) \vee \langle z, z_n\rangle \in X^3_{R_{n+1}}),$$

$$\theta(\mathsf{Ref}(R_1)) \coloneqq (\forall z)(\langle z, z\rangle \in X^3_{R_1}),$$

$$\theta(\mathsf{Irref}(R_1)) \coloneqq (\forall z)(\neg(\langle z, z\rangle \in X^3_{R_1})),$$

$$\theta(\mathsf{Fun}(R_1)) \coloneqq (\forall z_1)(\forall z_2)(\forall z_3)((\neg(\langle z_1, z_2\rangle \in X^3_{R_1}) \vee \neg(\langle z_1, z_3\rangle \in X^3_{R_1})) \vee z_2 = z_3),$$

$$\theta(P_1 \equiv P_2) \coloneqq (\forall z_1)(\forall z_2)((\neg(\langle z_1, z_2\rangle \in X^3_{P_1}) \vee \langle z_1, z_2\rangle \in X^3_{P_2}) \wedge (\neg(\langle z_1, z_2\rangle \in X^3_{P_2}) \vee \langle z_1, z_2\rangle \in X^3_{P_1})),$$

$$\theta(P_1 \equiv \neg P_2) \coloneqq (\forall z_1)(\forall z_2)((\neg(\langle z_1, z_2\rangle \in X^3_{P_1}) \vee \neg(\langle z_1, z_2\rangle \in X^3_{P_2})) \wedge (\langle z_1, z_2\rangle \in X^3_{P_2} \vee \langle z_1, z_2\rangle \in X^3_{P_1})),$$

$$\theta(P_1 \sqsubseteq P_2) \coloneqq (\forall z_1)(\forall z_2)(\neg(\langle z_1, z_2\rangle \in X^3_{P_1}) \vee \langle z_1, z_2\rangle \in X^3_{P_2}),$$

$$\theta(\mathsf{Fun}(P_1)) \coloneqq (\forall z_1)(\forall z_2)(\forall z_3)((\neg(\langle z_1, z_2\rangle \in X^3_{P_1}) \vee \neg(\langle z_1, z_3\rangle \in X^3_{P_1}) \vee z_2 = z_3),$$

$$\theta(P_1 \equiv P_{2_{C_1|}}) \coloneqq (\forall z_1)(\forall z_2)((\neg(\langle z_1, z_2\rangle \in X^3_{P_1}) \vee \langle z_1, z_2\rangle \in X^3_{P_2}) \wedge (\neg(\langle z_1, z_2\rangle \in X^3_{P_1}) \vee z_1 \in X^1_{C_1}) \wedge ((\neg\langle z_1, z_2\rangle \in X^3_{P_2}) \vee \neg(z_1 \in X^1_{C_1}) \vee \langle z_1, z_2\rangle \in X^3_{P_1})),$$

$$\theta(P_1 \equiv P_{2_{|t_1}}) \coloneqq (\forall z_1)(\forall z_2)((\neg(\langle z_1, z_2\rangle \in X^3_{P_1}) \vee \langle z_1, z_2\rangle \in X^3_{P_2}) \wedge (\neg(\langle z_1, z_2\rangle \in X^3_{P_1}) \vee z_2 \in X^1_{t_1}) \wedge ((\neg(\langle z_1, z_2\rangle \in X^3_{P_2}) \vee \neg(z_2 \in X^1_{t_1})) \vee \langle z_1, z_2\rangle \in X^3_{P_1})),$$

$$\theta(P_1 \equiv P_{2_{C_1|t_1}}) \coloneqq (\forall z_1)(\forall z_2)((\neg(\langle z_1, z_2\rangle \in X^3_{P_1}) \vee \langle z_1, z_2\rangle \in X^3_{P_2})\wedge (\neg(\langle z_1, z_2\rangle \in X^3_{P_1}) \vee z_1 \in X^1_{C_1}) \wedge (\neg(\langle z_1, z_2\rangle \in X^3_{P_1}) \vee z_2 \in X^1_{t_1})\wedge (\neg(\langle z_1, z_2\rangle \in X^3_{P_2}) \vee \neg(z_1 \in X^1_{C_1}) \vee \neg(z_2 \in X^1_{t_1}) \vee \langle z_1, z_2\rangle \in X^3_{P_1})),$$

$$\theta(t_1 \equiv t_2) \coloneqq (\forall z)((\neg(z \in X^1_{t_1}) \vee z \in X^1_{t_2}) \wedge (\neg(z \in X^1_{t_2}) \vee z \in X^1_{t_1})),$$

$$\theta(t_1 \equiv \neg t_2) \coloneqq (\forall z)((\neg(z \in X^1_{t_1}) \vee \neg(z \in X^1_{t_2})) \wedge (z \in X^1_{t_2} \vee z \in X^1_{t_1})),$$

$$\theta(t_1 \equiv t_2 \sqcup t_3) \coloneqq (\forall z)((\neg(z \in X^1_{t_1}) \vee (z \in X^1_{t_2} \vee z \in X^1_{t_3}))\wedge ((\neg(z \in X^1_{t_2}) \vee z \in X^1_{t_1}) \wedge (\neg(z \in X^1_{t_3}) \vee z \in X^1_{t_1}))),$$

$$\theta(t_1 \equiv t_2 \sqcap t_3) \coloneqq (\forall z)((\neg(z \in X^1_{t_1}) \vee (z \in X^1_{t_2} \wedge z \in X^1_{t_3}))\wedge (((\neg(z \in X^1_{t_2}) \vee \neg(z \in X^1_{t_3})) \vee z \in X^1_{t_1})),$$

$$\theta(t_1 \equiv \{e_d\}) \coloneqq (\forall z)((\neg(z \in X^1_{t_1}) \vee z = x_{e_d}) \wedge (\neg(z = x_{e_d}) \vee z \in X^1_{t_1})),$$

$$\theta(a : C_1) \coloneqq x_a \in X^1_{C_1},$$

$$\theta(a : \neg C_1) \coloneqq \neg(x_a \in X^1_{C_1}),$$

$$\theta((a,b) : R_1) \coloneqq \langle x_a, x_b\rangle \in X^3_{R_1},$$

$$\theta((a,b) : \neg R_1) \coloneqq \neg(\langle x_a, x_b\rangle \in X^3_{R_1}),$$

$$\theta(a = b) \coloneqq x_a = x_b,\ \theta(a \neq b) \coloneqq \neg(x_a = x_b),$$

$$\theta(e_d : t_1) \coloneqq x_{e_d} \in X^1_{t_1},$$

$$\theta((a, e_d) : P_1) \coloneqq \langle x_a, x_{e_d}\rangle \in X^3_{P_1},$$

$$\theta((a, e_d) : \neg P_1) \coloneqq \neg(\langle x_a, x_{e_d}\rangle \in X^3_{P_1}),$$

$$\theta(\alpha \wedge \beta) \coloneqq \theta(\alpha) \wedge \theta(\beta).$$

Let $\mathsf{cpt}_{\mathcal{KB}}$, $\mathsf{arl}_{\mathcal{KB}}$, $\mathsf{crl}_{\mathcal{KB}}$, $\mathsf{ind}_{\mathcal{KB}}$, $N_D^{\mathcal{KB}} \subseteq N_D$, $N_C^{\mathcal{KB}}$, and $\mathsf{bf}^{\mathbf{D}}_{\mathcal{KB}}(d)$ be as defined in Section 3.1. We assume without loss of generality that the facet expressions in $\mathsf{bf}^{\mathbf{D}}_{\mathcal{KB}}(d)$ are in *CNF*. In a similar way to Section 3.1, we define the 4LQS$^{\mathsf{R}}$-formula $\phi_{\mathcal{KB}}$ expressing the consistency of $\mathcal{KB}$ by putting:

$$\phi_{\mathcal{KB}} \coloneqq \bigwedge_{H \in \mathcal{KB}} \theta(H) \wedge \bigwedge_{i=1}^{12} \xi_i\,,$$

where

$$\xi_1 \coloneqq (\forall z)((\neg(z \in X^1_{\mathbf{I}}) \vee \neg(z \in X^1_{\mathbf{D}})) \wedge (z \in X^1_{\mathbf{D}} \vee z \in X^1_{\mathbf{I}}))\wedge (\forall z)(z \in X^1_{\mathbf{I}} \vee z \in X^1_{\mathbf{D}}) \wedge \neg(\forall z)\neg(z \in X^1_{\mathbf{I}}) \wedge \neg(\forall z)\neg(z \in X^1_{\mathbf{D}}),$$

$$\xi_2 \coloneqq (\forall z)((\neg(z \in X^1_{\mathbf{I}}) \vee z \in X^1_{\top}) \wedge (\neg(z \in X^1_{\top}) \vee z \in X^1_{\mathbf{I}}))\wedge (\forall z)\neg(z \in X_{\bot}),$$

$$\xi_3 \coloneqq \bigwedge_{A \in \mathsf{cpt}_{\mathcal{KB}}} (\forall z)(\neg(z \in X^1_A) \vee z \in X^1_{\mathbf{I}}),$$

$$\xi_4 \coloneqq (\bigwedge_{d \in N^{\mathcal{KB}}_D}((\forall z)(\neg(z \in X^1_d) \vee z \in X^1_{\mathbf{D}}) \wedge \neg(\forall z)\neg(z \in X^1_d)) \wedge (\forall z) (\bigwedge_{(d_i, d_j \in N^{\mathcal{KB}}_D, i<j)}((\neg(z \in X^1_{d_i}) \vee \neg(z \in X^1_{d_j})) \wedge (z \in X^1_{d_j} \vee z \in X^1_{d_i})))),$$

$$\xi_5 \coloneqq \bigwedge_{d \in N^{\mathcal{KB}}_D}((\forall z)((\neg(z \in X^1_d) \vee z \in X^1_{\top_d}) \wedge (\neg(z \in X^1_{\top_d}) \vee z \in X^1_d)\wedge (\forall z)\neg(z \in X^1_{\bot_d})),$$

$$\xi_6 \coloneqq \bigwedge_{\substack{f_d \in N^{\mathcal{KB}}_F(d),\\ d \in N^{\mathcal{KB}}_D}} (\forall z)(\neg(z \in X^1_{f_d}) \vee z \in X^1_d),$$

$$\xi_7 \coloneqq (\forall z_1)(\forall z_2)((\neg(z_1 \in X^1_{\mathbf{I}}) \vee \neg(z_2 \in X^1_{\mathbf{I}}) \vee \langle z_1, z_2\rangle \in X^3_U)\wedge ((\neg(\langle z_1, z_2\rangle \in X^3_U) \vee z_1 \in X^1_{\mathbf{I}}) \wedge (\neg(\langle z_1, z_2\rangle \in X^3_U) \vee z_2 \in X^1_{\mathbf{I}}))),$$

$$\xi_8 \coloneqq \bigwedge_{R \in \mathsf{arl}_{\mathcal{KB}}} (\forall z_1)(\forall z_2)((\neg(\langle z_1, z_2\rangle \in X^3_R) \vee z_1 \in X^1_{\mathbf{I}})\wedge (\neg(\langle z_1, z_2\rangle \in X^3_R) \vee z_2 \in X^1_{\mathbf{I}}))),$$

$$\xi_9 \coloneqq \bigwedge_{T \in \mathsf{crl}_{\mathcal{KB}}} (\forall z_1)(\forall z_2)(\neg(\langle z_1, z_2\rangle \in X^3_T) \vee z_1 \in X^1_{\mathbf{I}}) \wedge (\neg(\langle z_1, z_2\rangle \in X^3_T)\vee z_2 \in X^1_{\mathbf{D}}))),$$

$$\xi_{10} \coloneqq \bigwedge_{a \in \mathsf{ind}_{\mathcal{KB}}} (x_a \in X^1_{\mathbf{I}}) \wedge \bigwedge_{\substack{d \in N^{\mathcal{KB}}_D,\\ e_d \in N^{\mathcal{KB}}_C(d)}} x_{e_d} \in X^1_d,$$

$$\xi_{11} \coloneqq \bigwedge_{\{e_{d_1},\ldots,e_{d_n}\} \text{ in } \mathcal{KB}} (\forall z)((\neg(z \in X^1_{\{e_{d_1},\ldots,e_{d_n}\}})\vee \bigvee_{i=1}^{n}(z = x_{e_{d_i}}))\wedge(\bigwedge_{i=1}^{n}(z \neq x_{e_{d_i}} \vee z \in X^1_{\{e_{d_1},\ldots,e_{d_n}\}}))) \wedge \bigwedge_{\{a_1,\ldots,a_n\} \text{ in } \mathcal{KB}} (\forall z)((\neg(z \in X^1_{\{a_1,\ldots,a_n\}})\vee \bigvee_{i=1}^{n}(z = x_{a_i})) \wedge (\bigwedge_{i=1}^{n}(z \neq x_{a_i} \vee z \in X^1_{\{a_1,\ldots,a_n\}}))),$$

$$\xi_{12} \coloneqq \bigwedge_{\substack{d \in N^{\mathcal{KB}}_{\mathbf{D}},\\ \psi_d \in \mathsf{bf}^{\mathbf{D}}_{\mathcal{KB}}(d)}} (\forall z)(\neg(z \in X^1_{\psi_d}) \vee z \in \zeta(X^1_{\psi_d})) \wedge (\neg(z \in \zeta(X^1_{\psi_d}))\vee z \in X^1_{\psi_d}),$$

with ζ the transformation function from 4LQS$^{\mathsf{R}}$-variables of sort 1 to 4LQS$^{\mathsf{R}}$-formulae recursively defined, for $d \in N_{\mathbf{D}}^{\mathcal{KB}}$, by

$$\zeta(X^1_{\psi_d}) := \begin{cases} X^1_{\psi_d} & \text{if } \psi_d \in N_F^{\mathcal{KB}}(d) \cup \{\top^d, \bot_d\} \\ \neg\zeta(X^1_{\chi_d}) & \text{if } \psi_d = \neg\chi_d \\ \zeta(X^1_{\chi_d}) \wedge \zeta(X^1_{\varphi_d}) & \text{if } \psi_d = \chi_d \wedge \varphi_d \\ \zeta(X^1_{\chi_d}) \vee \zeta(X^1_{\varphi_d}) & \text{if } \psi_d = \chi_d \vee \varphi_d\,. \end{cases}$$

The constraints ξ_1-ξ_{12}, which represent the *CNF* translation of the constraints ψ_1-ψ_{12} defined in Section 3.1, are introduced to guarantee that each model of $\phi_{\mathcal{KB}}$ can easily be transformed in a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-interpretation. We are now ready to present Theorem 2, which shows that the consistency problem for $\mathcal{KB}$ is reducible to the satisfiability problem for $\varphi_{\mathcal{KB}}$.

Theorem 2. *Let $\mathcal{KB}$ be a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-knowledge base. Then, one can construct a 4LQS$^{\mathsf{R}}$-formula $\varphi_{\mathcal{KB}}$ such that $\varphi_{\mathcal{KB}}$ is satisfiable if and only if $\mathcal{KB}$ is consistent. Hence the consistency problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KBs is decidable.*

Proof. To prove the statement, we show that if $\boldsymbol{\mathcal{M}}$ is a 4LQS$^{\mathsf{R}}$-interpretation such that $\boldsymbol{\mathcal{M}} \models \phi_{\mathcal{KB}}$, we can construct a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-interpretation $\mathbf{I}_{\boldsymbol{\mathcal{M}}}$ such that $\mathbf{I}_{\boldsymbol{\mathcal{M}}} \models_{\mathbf{D}} \mathcal{KB}$ and, conversely, if $\mathbf{I}$ is a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-interpretation such that $\mathbf{I} \models_{\mathbf{D}} \mathcal{KB}$, we can construct a 4LQS$^{\mathsf{R}}$-interpretation $\boldsymbol{\mathcal{M}}_{\mathbf{I}}$ such that $\boldsymbol{\mathcal{M}}_{\mathbf{I}} \models \phi_{\mathcal{KB}}$.

Thus, let $\boldsymbol{\mathcal{M}}$ be any 4LQS$^{\mathsf{R}}$-interpretation $\boldsymbol{\mathcal{M}}$ such that $\boldsymbol{\mathcal{M}} \models \phi_{\mathcal{KB}}$. Reasoning as in the proof of Theorem 1 in Section 3.1, it is not hard to see that such $\boldsymbol{\mathcal{M}}$ is a 4LQS$^{\mathsf{R}}$-interpretation of the form $\boldsymbol{\mathcal{M}} = (D_1 \cup D_2, M)$.

Exploiting the fact that $\boldsymbol{\mathcal{M}}$ satisfies the constrains ξ_1-ξ_{12}, it is then possible to define a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-interpretation $\mathbf{I}_{\mathcal{M}} = (\Delta^I, \Delta_{\mathbf{D}}, \cdot^I)$ in an analogous way to the $\mathcal{DL}_{\mathbf{D}}^{4}$-interpretation in Theorem 1.

Since $\boldsymbol{\mathcal{M}} \models \underset{H\in\mathcal{KB}}{\theta(H)} \wedge \bigwedge_{i=1}^{12} \xi_i$, and, as it can be easily checked, $\mathbf{I}_{\mathcal{M}} \models_{\mathbf{D}} H$, iff $\boldsymbol{\mathcal{M}} \models \theta(H)$, for every statement $H \in \mathcal{KB}$, we plainly have that $\mathbf{I}_{\mathcal{M}} \models_{\mathbf{D}} \mathcal{KB}$.

Conversely, let $\mathbf{I} = (\Delta^{\mathbf{I}}, \Delta_{\mathbf{D}}, \cdot^{\mathbf{I}})$ be a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-interpretation such that $\mathbf{I} \models_{\mathbf{D}} \mathcal{KB}$. We show how to construct, out of the data type map $\mathbf{D}$ and the $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-interpretation $\mathbf{I}$, a 4LQS$^{\mathsf{R}}$-interpretation $\boldsymbol{\mathcal{M}}_{\mathbf{I},\mathbf{D}} = (D_{\mathbf{I},\mathbf{D}}, M_{\mathbf{I},\mathbf{D}})$ which satisfies $\phi_{\mathcal{KB}}$. Let us put $D_{\mathbf{I},\mathbf{D}} := \Delta^{\mathbf{I}} \cup \Delta_{\mathbf{D}}$ and define $M_{\mathbf{I},\mathbf{D}}$ as in Theorem 1.

From the definitions of $\mathbf{D}$ and $\mathbf{I}$, it follows easily that $\boldsymbol{\mathcal{M}}_{\mathbf{I},\mathbf{D}}$ satisfies the formulae ξ_1-ξ_{12} and $\theta(H)$, for every statement $H \in \mathcal{KB}$, and, therefore, that $\boldsymbol{\mathcal{M}}_{\mathbf{I},\mathbf{D}}$ is a model for $\phi_{\mathcal{KB}}$. $\square$

3.2.1 Expressiveness of $\mathcal{DL}_{\mathbf{D}}^{4}$ and of $\mathcal{DL}_{\mathbf{D}}^{4,\times}$

Some considerations on the expressive power of the DL $\mathcal{DL}_{\mathbf{D}}^{4,\times}$, which also apply to the DL $\mathcal{DL}_{\mathbf{D}}^{4}$, are in order.

Despite $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ allowing one to express existential quantification and at-least number restriction only on the left-hand-side of inclusion axioms and universal quantification and at-most number restriction only on the right-hand-side of inclusion axioms, it is more liberal than $\mathcal{SROIQ}(\mathbf{D})$ [69] in the construction of role inclusion axioms, since

- the roles involved are not restricted by any ordering relationship,
- the notion of simple role is not needed, and
- Boolean operations on roles and role constructs such as the product of concepts are admitted.

For example, the role hierarchy $\{RS \sqsubseteq S, RT \sqsubseteq R, VT \sqsubseteq T, VS \sqsubseteq V\}$ presented in [69] and not expressible in $\mathcal{SROIQ}(\mathbf{D})$ is admitted in the language of $\mathcal{DL}_{\mathbf{D}}^{4,\times}$. Moreover, the notion of simple role is not needed in the definition of role inclusion axioms and of axioms involving number restrictions. Also, Boolean operators on roles are admitted and can be introduced in inclusion axioms such as, for instance, $R_1 \sqsubseteq R_2 \sqcap R_3$ and $R_1 \sqsubseteq \neg R_2 \sqcup R_3$. Finally, $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ treats derived data types by admitting data type terms constructed from data ranges by means of a finite number of applications of the Boolean operators. Basic and derived data types can be used inside inclusion axioms involving concrete roles.

Moreover, $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ supports more OWL constructs than the DLs underpinning the profiles OWL QL, OWL RL, and OWL EL [78], such as disjoint union of concepts and union of data ranges. A deep comparison among the standard OWL 2 profiles and the profile identified by $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ is illustrated in Table 12. Furthermore, basic and derived data types can be used inside inclusion axioms involving concrete roles. Concerning the expressiveness of rules, the set-theoretic fragment $\mathsf{4LQS^R}$ underpinning $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ allows one also to express the disjunctive Datalog fragment admitting negation, equality and constraints, subject to no safety condition, and supporting for data types (see Section 3.2.2).

	Profile Support			
OWL Construct	**EL**	**QL**	**RL**	$\mathcal{DL}_{\mathbf{D}}^{4,\times}$
ObjectSomeValuesFrom	Y	SUB to THING and SUP	SUB to THING	SUB
DataSomeValuesFrom	Y	Y	SUB	SUB

ObjectHasValue	Y	N	SUB	SUB
DataHasValue	Y	N	SUB	SUB
ObjectHasSelf	Y	N	SUB	Y
ObjectAllValuesFrom	N	N	SUP	SUP
DataAllValuesFrom	N	N	SUP	SUP
ObjectMaxCardinality	N	N	SUP to 0/1	SUP
DataMaxCardinality	N	N	SUP to 0/1	SUP
ObjectMinCardinality	N	N	N	SUB
DataMinCardinality	N	N	N	SUB
ObjectExactCardinality	N	N	N	N
DataExactCardinality	N	N	N	N
ObjectOneOf	Y	N	SUB	Y
DataOneOf	Y	N	N	Y
ObjectInsersectionOf	Y	SUP	Y	Y
DataIntersectionOf	Y	Y	Y	Y
SubClassOf	Y	Y	Y	Y
EquivalentClasses	Y	Y	Class & HasValue	Y
Construct	**EL**	**QL**	**RL**	$\mathcal{DL}_{\mathbf{D}}^{4,\times}$
DisjointClasses	Y	Y	SUB	Y
ObjectUnionOf	N	N	SUB	Y
DataUnionOf	N	N	N	Y
DisjointUnion	N	N	N	Y
ObjectComplementOf	N	SUP	SUP	Y
SameIndividual	Y	N	N	Y
DifferentIndividuals	Y	Y	Y	Y
ClassAssertion	Y	Y	Y	Y
ObjectPropertyAssertion	Y	Y	Y	Y
DataPropertyAssertion	Y	Y	Y	Y
NegativeObjectPropertyAssertion	Y	Y	N	Y
NegativeDataPropertyAssertion	Y	Y	N	Y
SubObjectPropertyOf	Y	Y	Y	Y
SubDataPropertyOf	Y	Y	Y	Y
EquivalentObjectProperties	Y	Y	Y	Y
EquivalentDataProperties	Y	Y	Y	Y
TransitiveObjectProperty	Y	N	Y	Y

ReflexiveObjectProperty	Y	Y	Y	Y
FunctionalDataProperty	N	N	Y	Y
FunctionalObjectProperty	N	N	Y	Y
DisjointObjectProperties	N	Y	Y	Y
DisjointDataProperties	N	Y	Y	Y
IrreflexiveObjectProperties	N	Y	Y	Y
InverseObjectProperties	N	Y	Y	Y
InverseFunctionalObjectProperties	N	N	Y	Y
SymmetricObjectProperty	N	Y	Y	Y
AsymmetricObjectProperty	N	Y	Y	Y
ObjectPropertyDomain	Y	Y	SUP	Y
DataPropertyDomain	Y	Y	SUP	Y
ObjectPropertyRange	Y	Y	SUP	Y
DataPropertyRange	Y	Y	SUP	Y
HasKey	Y	N	SUP	SUP
ObjectPropertyChain	Y	N	N	Y

Table 12: OWL 2 profiles comparison.

For what concerns complexity issues, in [26] it is calculated that the small model constructed by the decision procedure for the satisfiability problem for $\mathsf{4LQS}^{\mathsf{R}}$ is exponential in the size of the input formula. This means that the decision procedure for the satisfiability problem for $\mathsf{4LQS}^{\mathsf{R}}$ is in **N2Exptime**. Such complexity result also holds for the decision procedure for consistency problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KBs, as the latter problem is polynomially reducible to the satisfiability problem for $\mathsf{4LQS}^{\mathsf{R}}$.

Finally, for what concerns the expressiveness of rules, we remark that the set-theoretic fragment $\mathsf{4LQS}^{\mathsf{R}}$ underpinning $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ allows one to express the disjunctive Datalog fragment with negation, equality, constraints and data types, which is not subjected to any safety conditions.

Next, we define the OWL 2 profile identified by the DL $\mathcal{DL}_{\mathbf{D}}^{4,\times}$. The grammar of such a profile includes the general definitions of the OWL 2 Specification (see [133, Section 13.1]), with the exception of the productions concerning *ClassExpression* that are reported in what follows:

$$ClassExpression := subClassExpression \mid superClassExpression$$

$$
\begin{aligned}
subClassExpression \coloneqq\ & Class \mid ObjectIntersectionOf \mid \\
& ObjectUnionOf \mid ObjectComplementOf \mid \\
& ObjectOneOf \mid subObjectSomeValuesFrom \mid \\
& ObjectHasValue \mid ObjectHasSelf \mid \\
& subObjectMinCardinality \mid \\
& subDataSomeValuesFrom \mid DataHasValue \mid \\
& subDataMinCardinality
\end{aligned}
$$

$$
\begin{aligned}
subObjectSomeValuesFrom \coloneqq\ & 'ObjectSomeValuesFrom' \\
& '(' \ ObjectPropertyExpression \\
& \quad [\ subClassExpression\ ]\ ')' \mid \\
& 'ObjectSomeValuesFrom' \\
& '(' \ ObjectPropertyExpression\ owl:Thing\ ')'
\end{aligned}
$$

$$
\begin{aligned}
subDataSomeValuesFrom \coloneqq\ & 'DataSomeValuesFrom' \\
& '(' \ DataPropertyExpression\ [\ DataRange\ ] \\
& \ ')' \mid' \ DataSomeValuesFrom' \\
& '(' \ DataPropertyExpression\ [\ DataRange\ ]\ ')'
\end{aligned}
$$

$$
\begin{aligned}
subObjectMinCardinality \coloneqq\ & 'ObjectMinCardinality'\ '('\ n \\
& ObjectPropertyExpression \\
& \quad [\ subClassExpression\ ]\ ')' \mid \\
& 'ObjectMinCardinality'\ '('\ n \\
& \ ObjectPropertyExpression\ owl:Thing\ ')'
\end{aligned}
$$

$$
\begin{aligned}
subDataMinCardinality \coloneqq\ & 'DataMinCardinality'\ '('\ n \\
& \ DataPropertyExpression\ [\ DataRange\ \ ]\ ')'
\end{aligned}
$$

$$
\begin{aligned}
superClassExpression \coloneqq\ & Class \mid ObjectIntersectionOf \mid \\
& ObjectUnionOf \mid ObjectComplementOf \mid \\
& ObjectOneOf \mid superObjectAllValuesFrom \mid
\end{aligned}
$$

$$ObjectHasValue \mid ObjectHasSelf \mid$$
$$superObjectMaxCardinality \mid$$
$$superDataAllValuesFrom \mid DataHasValue \mid$$
$$superDataMaxCardinality$$

$$superObjectAllValuesFrom := 'ObjectAllValuesFrom' \; '(' \; ObjectPropertyExpression \; [\; supClassExpression \;] \; ')'$$

$$superDataAllValuesFrom := 'DataAllValuesFrom' \; '(' \; DataPropertyExpression \; [\; DataRange \;] \; ')' \; | \; 'DataAllValuesFrom' \; '(' \; DataPropertyExpression \; [\; DataRange \;] \; ')'$$

$$superObjectMaxCardinality := 'ObjectMaxCardinality' \; n \; '(' \; ObjectPropertyExpression \; [\; supClassExpression \;] \; ')' \; | \; 'ObjectMaxCardinality' \; '(' \; n \; ObjectPropertyExpression \; owl:Thing \; ')'$$

$$superDataMaxCardinality := 'DataMaxCardinality' \; '(' \; n \; DataPropertyExpression \; [\; DataRange \;] \; ')'$$

3.2.2 Mapping SWRL-rules into 4LQS^R-formulae

In Table 13, we provide some examples showing how SWRL rules can be expressed by 4LQS^R-formulae. We do not provide here a formal translation function since it is not hard to see that it could easily be constructed by resorting to the mapping function τ introduced in Section 3.1, and hence to the mapping θ introduced in Section 3.2.

Type of Rule	Rule
SWRL-rule	$hasParent(X,Y), hasBrother(Y,Z) : -hasUncle(X,Z).$
4LQS^R-rule	$(\forall x)(\forall y)(\forall z)(\langle x,y\rangle \in X^3_{hasParent} \wedge \langle y,z\rangle \in X^3_{hasBrother} \rightarrow \langle x,z\rangle \in X^3_{hasUncle})$
SWRL-rule	$Location(X), Trauma(Y), isLocationOf(X,Y), isPartOf(X,Z)$:- $isLocationOf(Z,Y)$

4LQS^R-rule	$(\forall x)(\forall y)(\forall z)(x \in X^1_{Location} \wedge y \in X^1_{Trauma} \wedge \langle x, z\rangle \in X^3_{isPartOf} \rightarrow \langle z, y\rangle \in X^3_{isLocationOf})$
SWRL-rule 4LQS^R-rule	$loc(X), sameLoc(Y, X), lat(Y, Z), long(Y, T) : -lat(X, Z), long(X, T)$ $\forall(x, y, z, t)(x \in X^3_{loc} \wedge \langle x, y\rangle \in X^3_{sameLoc} \wedge \langle y, z\rangle \in X^3_{lat} \wedge \langle y, t\rangle \in X^3_{long}) \rightarrow \langle x, z\rangle \in X^3_{lat} \wedge \langle x, t\rangle \in X^3_{long}$
SWRL-rule 4LQS^R-rule	$Person(X), hasAge(X, Y), (Y \geq 18) : -Adult(X)$ $(\forall x)(\forall y)(x \in X^3_{Person} \wedge \langle x, y\rangle \in X^3_{hasAge} \wedge y \in X^1_{\geq 18} \rightarrow x \in X^1_{Adult})$
SWRL-rule 4LQS^R-rule	$Region(Y), hasLocation(X, Y) : -hasRegion(X, Y)$ $(\forall x)(\forall y)(y \in X^3_{Region} \wedge \langle x, y\rangle \in X^3_{hasLocation} \rightarrow \langle x, y\rangle \in X^3_{hasRegion})$

Table 13: Examples of SWRL rules translation.

In addition, the 4LQS^R-fragment admits formulae of the types:

- $(\forall x)(\forall y)(x \in X^1_A \vee \langle x, y\rangle \in X^3 \rightarrow x \in X^1_B \vee x \in X^1_C)$,
- $(\forall x)(\forall y)(x \in X^1_A \wedge \neg(y \in X^1_A) \rightarrow \neg(\langle x, y\rangle \in X^3_R))$

that can be seen as rules admitting classical negation and disjunctive atoms both in their head and in their body, which are not allowed in SWRL as consequence of its monotonicity (i.e., it cannot be used to modify existing information). It turns out that in SWRL one can not deal directly with incomplete information.

The following 4LQS^R-formula φ is an example of such type of rules:

$$\varphi_1 = (\forall x)(\forall y)(\forall z)(\forall w)\Big(\big(\langle x, y\rangle \in X^3_{hasFather} \wedge \langle x, z\rangle \in X^3_{hasMother} \wedge \neg(\langle y, w\rangle \in X^3_{hasSibling}) \wedge \neg(\langle z, w\rangle \in X^3_{hasSibling})\big) \rightarrow \neg(\langle x, w\rangle) \in X^3_{hasUncle})\Big).$$

Since φ contains negated atoms, namely

$$\neg(\langle y, w\rangle \in X^3_{hasSibling}), \quad \neg(\langle z, w\rangle \in X^3_{hasSibling}), \quad \neg(\langle x, w\rangle) \in X^3_{hasUncle}),$$

it represents a rule not expressible in the SWRL language.

4 Automated Reasoning in $\mathcal{DL}_{\mathbf{D}}^{4,\times}$

The most important feature of a knowledge representation system is the capability of providing reasoning services. Depending on the type of the application domain, there is a wide variety of implicit knowledge that it is desirable to infer from what is explicitly mentioned in the knowledge base. For instance, one can query the knowledge base in order to retrieve the individuals belonging to a specific class, the sub-properties of a particular property, or to check if two individuals are related to each other through a set of properties. These are all common ABox reasoning tasks. In this section, we shall introduce the reasoning services provided by the description logic $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ and we shall present sound, complete, and terminating procedures that solve them. In particular, we will focus on two families of reasoning tasks. The first one concerns $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-TBox reasoning, which includes the problem of deciding if a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KB is consistent, whereas the second one is related to problems concerning $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-ABox reasoning.

One of the most classical reasoning task related to TBox reasoning is *concept subsumption*, that is the problem to decide, given two (complex) concepts C and D, whether or not D is a more general concept than C (in which we can write $C \subseteq D$). Another important problem, strictly related to the first one, is *concept satisfiability*, that is the problem to decide whether or not a concept contains some instance.

The problems of subsumption and satisfiability are also extended to abstract (concrete) roles. The problem of *role subsumption* consists in establishing whether or not a role S is more general than a role T (in which we can write $T \subseteq S$). The problem of *role satisfiability* for a role T consists in deciding whether or not there exists two individuals related to each other through T. Notice that concept (role) equivalence and disjunction can easily be reduced to concept (role) subsumption. As we will show, the reasoning services concerning $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-TBox can be reduced to the problem of $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KB consistency.

The second family of reasoning tasks consists in querying the $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KBs and includes:

(1) *Instance checking*: the problem of deciding whether or not an individual a is an instance of a concept C;

(2) *Instance retrieval*: the problem of retrieving all the individuals that are instances of a given concept;

(3) *Role filler retrieval*: the problem of retrieving all the fillers x such that the pair (a, x) is an instance of a role R;

(4) *Concept retrieval*: the problem of retrieving all concepts that an individual is an instance of;

(5) *Role instance retrieval*: the problem of retrieving all roles that a pair of individuals (a, b) is an instance of;

(6) *Conjunctive Query Answering* (CQA): the problem of finding the answer set of a conjunctive query.

This chapter is organized as follows.

In Section 4.1, we shall formally define the CQA problem and we shall provide an algorithm which solves the CQA problem and the reasoning tasks concerning $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-TBox.

In Section 4.2, we shall show that problems (1-6) can be reduced to a generalization of the CQA problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$, called *Higher-Order Conjunctive Query Answering* for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$, and we shall provide an algorithm to solve the latter problem.

Finally, in Section 4.2.2, we shall provide an improved version of the algorithm introduced in Section 4.2.

4.1 Conjunctive Query Answering for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$

We now formally introduce the problem of CQA.

Let $\mathcal{V} = \{v_1, v_2, \ldots\}$ and $\mathcal{D} = \{d_1, d_2, \ldots\}$ be denumerable infinite sets of variables disjoint from **Ind** and from $\bigcup\{N_C(d) : d \in N_{\mathbf{D}}\}$. A $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-*atomic formula* is an expression of the following types:

$$R(w_1, w_2), \quad P(w_1, u), \quad C(w_1), \quad w_1 = w_2, \quad t(u),$$

where

- $w_1, w_2 \in \mathcal{V} \cup \mathbf{Ind}$,
- $u_1, u_2 \in \mathcal{D} \cup \bigcup\{N_C(d) : d \in N_{\mathbf{D}}\}$,
- R is a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-abstract role term,
- P is a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-concrete role term,
- C is a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-concept term, and
- t is a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-data type term.

A $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-atomic formula containing no variables is said to be *closed.* A $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-*literal* is a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-atomic formula or its negation. A $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-*conjunctive query* is a conjunction of $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-literals.

Let

- $v_1, \ldots, v_n \in \mathcal{V}$,

- $d_1, \ldots, d_l \in \mathcal{D}$,

- $o_1, \ldots, o_n \in \mathbf{Ind}$, and

- $e_1, \ldots, e_l \in \bigcup\{N_C(d) : d \in N_{\mathbf{D}}\}$.

A substitution

$$\sigma \coloneqq \{v_1/o_1, \ldots, v_n/o_n, d_1/e_1, \ldots, d_l/e_l,\}$$

is a map such that, for every $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-literal L, $L\sigma$ is obtained from L by replacing

- the occurrences of $v_1, \ldots, v_n$ in L with $o_1, \ldots, o_n$, respectively and

- the occurrences of $d_1, \ldots, d_l$ in L with $e_1, \ldots, e_l$, respectively.

Substitutions can be extended to $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive queries in the usual way.

Let $Q \coloneqq (L_1 \wedge \ldots \wedge L_m)$ be a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive query, and $\mathcal{KB}$ a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KB. A substitution σ involving *exactly* the variables occurring in Q is a *solution for Q w.r.t. $\mathcal{KB}$* if there exists a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-interpretation $\mathbf{I}$ such that

$$\mathbf{I} \models_{\mathbf{D}} \mathcal{KB} \text{ and } \mathbf{I} \models_{\mathbf{D}} Q\sigma.$$

The collection Σ of the solutions for Q w.r.t. $\mathcal{KB}$ is the *answer set of Q w.r.t. $\mathcal{KB}$*. Then, the *Conjuntive Query Answering* problem for Q w.r.t. $\mathcal{KB}$ consists in finding the answer set Σ of Q w.r.t. $\mathcal{KB}$.

We shall solve the *CQA* problem just stated by reducing it to the analogous problem formulated in the context of the fragment $\mathsf{4LQS^R}$ (and in turn to the decision procedure for $\mathsf{4LQS^R}$ presented in [26]). The *CQA* problem for $\mathsf{4LQS^R}$-formulae can be stated as follows. Let ϕ be a $\mathsf{4LQS^R}$-formula and let ψ be a conjunction of $\mathsf{4LQS^R}$-quantifier-free atomic formulae of level 0 of the types

$$x = y, \qquad x \in X^1, \qquad \langle x, y\rangle \in X^3,$$

or their negations. The *CQA problem for ψ w.r.t. ϕ* consists in computing the *answer set of ψ w.r.t. ϕ*, namely the collection Σ' of all the substitutions

$$\sigma' \coloneqq \{x_1/y_1, \ldots, x_n/y_n\}$$

such that

(a) the support of $\theta(\sigma)$ is a subset of the variables (of sort 0) in ψ;

(b) $\boldsymbol{\mathcal{M}} \models \phi \wedge \psi\sigma'$, for some $\mathsf{4LQS^R}$-interpretation $\boldsymbol{\mathcal{M}}$.

In view of the decidability of the satisfiability problem for $\mathsf{4LQS^R}$-formulae, the *CQA* problem for $\mathsf{4LQS^R}$-formulae is decidable as well. Indeed, given two

$\mathsf{4LQS^R}$-formulae ϕ and ψ satisfying the above requirements, to compute the answer set of ψ w.r.t. ϕ, for each candidate substitution σ' defined as above one has just to test the $\mathsf{4LQS^R}$-formula $\phi \wedge \psi\sigma'$ for satisfiability. Since the number of possible candidate substitutions is $|\mathtt{Var}_0(\phi)|^{|\mathtt{Var}_0(\psi)|}$ and the satisfiability problem for $\mathsf{4LQS^R}$-formulae is decidable, the answer set of ψ w.r.t. ϕ can be computed effectively. Summarizing,

Lemma 1. *The CQA problem for* $\mathsf{4LQS^R}$*-formulae is decidable.* □

The *CQA* problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ can be solved via an effective reduction to the *CQA* problem for $\mathsf{4LQS^R}$-formulae, and then exploiting Lemma 1. The reduction is accomplished through the function θ (see Section 3.2) that maps injectively

- individuals a, constants $e_d \in N_C(d)$, variables $w \in \mathsf{V_i}$, and variables $u \in \mathsf{V_e}$ into sort 0 variables x_a, x_{e_d}, x_w, x_u, respectively,
- the constant concepts $\top$ and $\bot$, data type terms t, and concept terms C, into sort 1 variables $X^1_\top$, $X^1_\bot$, X^1_t, X^1_C, respectively,
- the universal relation U, abstract role terms R, and concrete role terms P, into sort 3 variables X^3_U, X^3_R, X^3_P, respectively,
- $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-*TBoxes*, $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-*RBoxes*, and $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-*ABoxes*, into $\mathsf{4LQS^R}$-formulae in *CNF*,

and which we exploit to map instances of the *CQA* problem from the $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-context to the $\mathsf{4LQS^R}$-context.

The mapping function θ, which has been used to map the $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KB $\mathcal{KB}$ in the $\mathsf{4LQS^R}$-formula ϕ_{KB}, will be extended to map also

- $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive query Q in the $\mathsf{4LQS^R}$-formula ψ_Q and
- the answer set Σ of Q w.r.t. $\mathcal{KB}$ in a set Σ' of $\mathsf{4LQS^R}$-substitutions (of level 0).

More specifically, let the mapping function θ be defined as in Section 3.2. θ is extended to $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive query literals as follows.

- $\theta(R_1(w_1, w_2)) := \langle x_{w_1}, x_{w_2} \rangle \in X^3_{R_1}$,
- $\theta(P_1(w_1, u)) := \langle x_{w_1}, x_u \rangle \in X^3_{P_1}$,
- $\theta(C_1(w_1)) := x_{w_1} \in X^1_{C_1}$,
- $\theta(w_1 = w_2) := x_{w_1} = x_{w_2}$,

- $\theta(t(u)) \coloneqq x_{u_1} \in X^1_t$,
- $\theta(\neg R_1(w_1, w_2)) \coloneqq \neg(\langle x_{w_1}, x_{w_2}\rangle \in X^3_{R_1})$,
- $\theta(\neg P_1(w_1, u)) \coloneqq \neg(\langle x_{w_1}, x_u\rangle \in X^3_{P_1})$,
- $\theta(\neg C_1(w_1)) \coloneqq \neg(x_{w_1} \in X^1_{C_1})$,
- $\theta(\neg(w_1 = w_2)) \coloneqq \neg(x_{w_1} = x_{w_2})$,
- $\theta(\neg t(u)) \coloneqq \neg(x_{u_1} \in X^1_t)$.

To complete, we extend the mapping θ on substitutions

$$\sigma \coloneqq \{x_1/o_1, \ldots, x_n/o_n, d_1/e_1, \ldots, d_l/e_l\},$$

where

- $x_1, \ldots x_n \in \mathcal{V}$,
- $d_1, \ldots d_l \in \mathcal{D}$,
- $o_1, \ldots, o_n \in \mathbf{Ind}$, and
- $e_1, \ldots, e_l \in \bigcup\{N_C(d) : d \in N_{\mathbf{D}}\}$,

by putting

$$\begin{aligned}\theta(\sigma) &:= \theta(\{x_1/o_1, \ldots, x_n/o_n, d_1/e_1, \ldots, d_l/e_l\}) \\ &= \{x_{x_1}/x_{o_1}, \ldots, x_{x_n}/x_{o_n}, x_{d_1}/x_{e_1}, \ldots, x_{d_l}/x_{e_l}\} \\ &= \sigma',\end{aligned}$$

where

$$x_{x_1}, \ldots, x_n, \qquad x_{d_1}, \ldots, x_{d_l}, \qquad x_{o_1}, \ldots, x_{o_n}, \qquad x_{e_1}, \ldots, x_{e_l},$$

are $\mathsf{4LQS^R}$-variables of sort 0. Finally, θ is extended to $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive queries as usual.

Given a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KB $\mathcal{KB}$ and a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive query Q, using the function θ we can effectively construct the following $\mathsf{4LQS^R}$-formulae in *CNF*:

$$\begin{aligned}\phi_{\mathcal{KB}} &\coloneqq \bigwedge_{H \in \mathcal{KB}} \theta(H) \wedge \bigwedge_{i=1}^{12} \xi_i\,, \\ \psi_Q &\coloneqq \theta(Q)\,.\end{aligned} \tag{1}$$

The following theorem states that the *CQA* problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ is decidable.

Theorem 3. *Given a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-knowledge base $\mathcal{KB}$ and a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive query Q, the CQA problem for Q w.r.t. $\mathcal{KB}$ is decidable.*

Proof. We solve the problem of *CQA* for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ via a reduction to the problem of *CQA* for $\mathsf{4LQS^R}$, and then by exploiting the decidability result provided by Lemma 1.

We have proved in Theorem 2 in Section 3.1 that if $\boldsymbol{\mathcal{M}}$ is a $\mathsf{4LQS^R}$-interpretation such that $\boldsymbol{\mathcal{M}} \models \phi_{\mathcal{KB}}$, we can construct a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-interpretation $\mathbf{I}_{\boldsymbol{\mathcal{M}}}$ such that $\mathbf{I}_{\boldsymbol{\mathcal{M}}} \models_{\mathbf{D}} \mathcal{KB}$ and, if $\mathbf{I}$ is a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-interpretation such that $\mathbf{I} \models_{\mathbf{D}} \mathcal{KB}$, we can construct a $\mathsf{4LQS^R}$-interpretation $\boldsymbol{\mathcal{M}}_{\mathbf{I}}$ such that $\boldsymbol{\mathcal{M}}_{\mathbf{I}} \models \phi_{\mathcal{KB}}$.

To prove the theorem, we have to show that for all substitutions σ, $\sigma \in \Sigma$ iff $\theta(\sigma) \in \Sigma'$. To prove the first part of the theorem, let us assume for the sake of contradiction that there exists a substitution σ such that $\sigma \in \Sigma$ and $\theta(\sigma) \notin \Sigma'$.

Since $\theta(\sigma) \notin \Sigma'$, by the definition of Σ' (answer set of ψ_Q w.r.t $\phi_{\mathcal{KB}}$) at least one of the following two conditions (a) and (b) must not hold for $\theta(\sigma)$:

(a) the support of $\theta(\sigma)$ is a subset of the variables (of sorts 0) in ψ_Q;

(b) $\boldsymbol{\mathcal{M}} \models \phi_{\mathcal{KB}} \wedge \psi_Q\theta(\sigma)$, for some $\mathsf{4LQS^R}$-interpretation $\boldsymbol{\mathcal{M}}$.

It is easy to check that by the construction of the mapping θ, condition (a) is fulfilled. Thus condition (b) must not hold and hence $\boldsymbol{\mathcal{M}} \not\models \psi_Q\theta(\sigma)$, for every $\mathsf{4LQS^R}$-interpretation $\boldsymbol{\mathcal{M}}$ such that $\boldsymbol{\mathcal{M}} \models \phi_{\mathcal{KB}}$.

Since $\sigma \in \Sigma$, there is a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-interpretation $\mathbf{I}$ such that $\mathbf{I} \models_D \mathcal{KB}$ and $\mathbf{I} \models_D Q\sigma$. Thus, by Theorem 2 in Section 3.1, we can define a $\mathsf{4LQS^R}$-interpretation $\boldsymbol{\mathcal{M}}_{\mathbf{I}}$ such that $\boldsymbol{\mathcal{M}}_{\mathbf{I}} \models \phi_{\mathcal{KB}}$ and $\boldsymbol{\mathcal{M}}_{\mathbf{I}} \models \psi_Q\theta(\sigma)$, which leads to contradiction as desired.

The converse part of the theorem can be proved in an analogous way. □

4.1.1 A procedure for the *CQA* problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$

In this section, I will first introduce a KE-tableau-based algorithm that also serves as decision procedure for the consistency problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KBs. I recall that, since the mapping θ considers a proper subset of the formulae of $\mathsf{4LQS^R}$, i.e., it does not involve $\mathsf{4LQS^R}$-variables of sort 2, $\mathsf{4LQS^R}$-atomic formulae of levels 1 and 2, and $\mathsf{4LQS^R}$-purely universal quantified formulae of levels 2 and 3, we can focus our attention to the fragment $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$ which is actually involved in the set-theoretic representation of $\mathcal{DL}_{\mathbf{D}}^{4,\times}$. I will illustrate a procedure that, given a KE-tableau representing the saturation of the $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-formula $\phi_{\mathcal{KB}}$ corresponding to a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KB and a $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-formula ψ_Q corresponding to a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive query Q, yields all the (level 0) substitutions

$$\sigma := \{x_1/y_1, \ldots, x_n/y_n\},$$

with $\{x_1, \ldots, x_n\} \in \mathtt{Vars}(\psi)$ and $\{y_1, \ldots, y_n\} \in \mathtt{Var}_0(\phi)$ belonging to the answer set Σ' of ψ_Q w.r.t. $\phi_{\mathcal{KB}}$.

To prepare for the procedure *Saturate-KB* to be described next, a brief introduction on the KE-tableau system is in order (see [35] for a detailed overview of KE-tableau). The KE-system is a refutation system inspired to Smullyan's semantic tableaux [113]. The main characteristic distinguishing KE-tableaux from the latter is the introduction of an analytic cut rule (PB-rule) that permits to reduce inefficiencies of semantic tableaux. In fact, firstly, the classic tableau system cannot represent the use of auxiliary lemmas in proofs; secondly, it cannot express the bivalence of classical logic. Thirdly, it is extremely inefficient, as witnessed by the fact that it cannot polynomially simulate the truth-tables. None of these anomalies occurs if the cut rule is permitted.

We now introduce some notions and notations useful for the presentation of the procedure *Saturate-KB*.

Let $\overline{\phi}_{\mathcal{KB}}$ be the formula obtained from the formula $\phi_{\mathcal{KB}}$ representing the $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KB (see Section 4.1) by:

- moving universal quantifiers in $\phi_{\mathcal{KB}}$ as inwards as possible according to the rule $(\forall z)(A(z) \wedge B(z)) \longleftrightarrow ((\forall z)A(z) \wedge (\forall z)B(z))$,
- renaming universally quantified variables so as to make them pairwise distinct.

Let $F_1, \ldots, F_k$ be the conjuncts of $\overline{\phi}_{\mathcal{KB}}$ that are $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-literals (we recall that a $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-literal is of one of the types $x = y$, $x \in X^1$, $\langle x, y\rangle \in X^3$ and their complements) and let $S_1, \ldots, S_m$ be the conjuncts of $\overline{\phi}_{\mathcal{KB}}$ that are $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-purely universal formulae.
For every

$$S_i := (\forall z_1^i) \ldots (\forall z_{n_i}^i)\chi_i,\ i = 1, \ldots, m,$$

I put

$$Exp(S_i) := \bigwedge_{\{x_{a_1}, \ldots, x_{a_{n_i}}\} \subseteq \mathtt{Var}_0(\overline{\phi}_{\mathcal{KB}})} S_i\{z_1^i / x_{a_1}, \ldots, z_{n_i}^i / x_{a_{n_i}}\}.$$

Let

$$\Phi_{\mathcal{KB}} := \{F_j : j = 1, \ldots, k\} \cup \bigcup_{i=1}^{m} Exp(S_i) \qquad (2)$$

and let

$$\Phi = \{C_1, \ldots, C_p\}$$

be a collection of disjunctions of $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-literals.

$\mathcal{T}$ is a KE-tableau for Φ if there exists a finite sequence $\mathcal{T}_1, \ldots, \mathcal{T}_t$ such that

(i) $\mathcal{T}_1$ is a one-branch tree consisting of the sequence $C_1, \ldots, C_p$,

(ii) $\mathcal{T}_t = \mathcal{T}$, and

(iii) for each $i < t$, $\mathcal{T}_{i+1}$ is obtained from $\mathcal{T}_i$ by an application of one of the rules in Figure 1.

$$\frac{\beta_1 \vee \ldots \vee \beta_n \qquad \mathcal{S}_i^{\overline{\beta}}}{\beta_i}\ \textbf{E-Rule} \qquad\qquad \frac{}{A \mid \overline{A}}\ \textbf{PB-Rule}$$

where $\mathcal{S}_i^{\overline{\beta}} := \{\overline{\beta}_1, ..., \overline{\beta}_n\} \setminus \{\overline{\beta}_i\}$, for $i = 1, ..., n$

with A a literal

Figure 1: Expansion rules for the KE-tableau.

The set of formulae $\mathcal{S}_i^{\overline{\beta}} = \{\overline{\beta}_1, \ldots, \overline{\beta}_n\} \setminus \{\overline{\beta}_i\}$ occurring as premise in the E-rule contains the complements of all the components, with the exception of the component β_i.

Let $\mathcal{T}$ be a KE-tableau. A branch ϑ of $\mathcal{T}$ is *closed* if it contains both A and $\neg A$, for some formula A. Otherwise, the branch is *open*. A formula $\beta_1 \vee \ldots \vee \beta_n$ is *fulfilled* in a branch ϑ, if β_i is in ϑ, for some $i = 1, \ldots, n$. A branch ϑ is *complete* if every formula $\beta_1 \vee \ldots \vee \beta_n$ occurring in ϑ is fulfilled. A KE-tableau is *complete* if all its branches are complete. A formula, or a branch, or a KE-tableau that is not fulfilled is said to be *unfulfilled*.

Next, we introduce the procedure *Saturate-KB* that takes as input the set $\Phi_{\mathcal{KB}}$ constructed from a $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-formula $\phi_{\mathcal{KB}}$ representing a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KB $\mathcal{KB}$ as shown above, and yields a complete KE-tableau $\mathcal{T}_{\mathcal{KB}}$ for $\Phi_{\mathcal{KB}}$.

Procedure 1. *Saturate-KB($\Phi_{\mathcal{KB}}$)*

1. *$\mathcal{T}_{\mathcal{KB}} := \Phi_{\mathcal{KB}}$;*
2. *Select an open branch ϑ of $\mathcal{T}_{\mathcal{KB}}$ that is not yet complete.*
   (a) *Select a formula $\beta_1 \vee \ldots \vee \beta_n$ on ϑ that is not fulfilled.*
   (b) *If $\mathcal{S}_j^{\overline{\beta}}$ is in ϑ, for some $j \in \{1, \ldots, n\}$, apply the E-Rule to $\beta_1 \vee \ldots \vee \beta_n$ and $\mathcal{S}_j^{\overline{\beta}}$ on ϑ and go to step 2.*
   (c) *If $\mathcal{S}_j^{\overline{\beta}}$ is not in ϑ, for every $j = 1, \ldots, n$, let $B^{\overline{\beta}}$ be the collection of formulae $\overline{\beta}_1, \ldots, \overline{\beta}_n$ present in ϑ and let $\overline{\beta}_h$ be the lowest index formula such that $\overline{\beta}_h \in \{\{\overline{\beta}_1, \ldots, \overline{\beta}_n\} \setminus B^{\overline{\beta}}\}$, then apply the PB-rule to $\overline{\beta}_h$ on ϑ, and go to step 2.*
3. *Return $\mathcal{T}_{\mathcal{KB}}$.*

Soundness of the procedure *Saturate-KB* can easily be proved in a standard way and its completeness can be shown much along the lines of Proposition

36 in [35]. Concerning termination of the procedure *Saturate-KB*, our proof is based on the following two facts.

The rules in Figure 1 are applied only to unfulfilled formulae on open branches and tend to reduce the number of unfulfilled formulae occurring on the considered branch. In particular, when the E-Rule is applied on a branch ϑ, the number of unfulfilled formulae on ϑ decreases. In case of application of the PB-Rule on a formula $\beta = \beta_1 \vee \ldots \vee \beta_n$ on a branch, the rule generates two branches. In one of the branches the number of unfulfilled formulae decreases (because β becomes fulfilled), whereas in the other one the number of unfulfilled formulae stays constant but the subset $B^{\overline{\beta}}$ of $\{\overline{\beta}_1, \ldots, \overline{\beta}_n\}$ occurring on the branch gains a new element. Once $|B^{\overline{\beta}}|$ gets equal to $n-1$, namely after at most $n-1$ applications of the PB-rule, the E-rule is applied and the formula $\beta = \beta_1 \vee \ldots \vee \beta_n$ becomes fulfilled, thus decrementing the number of unfulfilled formulae on the branch. Since the number of unfulfilled formulae on each open branch gets equal to zero after a finite number of steps and the rules of Figure 1 can be applied only to unfulfilled formulae on open branches, the procedure terminates.

By the completeness of Procedure 1, each open branch ϑ of $\mathcal{T}_{\mathcal{KB}}$ induces a $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-interpretation $\boldsymbol{\mathcal{M}}_\vartheta$ such that $\boldsymbol{\mathcal{M}}_\vartheta \models \Phi_{\mathcal{KB}}$. Specifically, $\boldsymbol{\mathcal{M}}_\vartheta = (D_\vartheta, M_\vartheta)$ is defined as follows. We put

- $D_\vartheta \coloneqq \{x \in \mathcal{V}_0 : x \text{ occurs in } \vartheta\}$;
- $M_\vartheta x \coloneqq x$, for every $x \in D_\vartheta$;
- $M_\vartheta X^1_C = \{x : x \in X^1_C \text{ is in } \vartheta\}$, for every $X^1_C \in \mathcal{V}_1$ occurring in ϑ;
- $M_\vartheta X^3_R = \{\langle x, y\rangle : \langle x, y\rangle \in X^3_R \text{ is in } \vartheta\}$, for every $X^3_R \in \mathcal{V}_3$ occurring in ϑ.

It is easy to check that $\boldsymbol{\mathcal{M}}_\vartheta \models \overline{\phi}_{\mathcal{KB}}$ and thus, plainly, $\boldsymbol{\mathcal{M}}_\vartheta \models \phi_{\mathcal{KB}}$ holds. Next, we provide some complexity results. Let r be the maximum number of universal quantifiers in S_i, and $k \coloneqq |\mathtt{Var}_0(\overline{\phi}_{\mathcal{KB}})|$. Then, each S_i generates k^r expansions. Since the number of such formulae in the knowledge base is m, then the number of disjunctions in the initial branch of the KE-tableau is $m \cdot k^r$. Next, let ℓ be the maximum number of literals in S_i, for $i = 1, \ldots, m$. Then, the maximum depth of the KE-tableau, namely the maximum size of the models of $\Phi_{\mathcal{KB}}$ constructed as illustrated above, is $\mathcal{O}(\ell m k^r)$. Additionally, the number of leaves of the tableau, that is the number of such models of $\Phi_{\mathcal{KB}}$, is $O(2^{\ell m k^r})$.

Given a KE-tableau constructed by the procedure *Saturate-KB* and a $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-formula ψ_Q representing a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive query Q, we now describe how to yield all the substitutions σ' in the answer set Σ' of ψ_Q w.r.t. $\phi_{\mathcal{KB}}$. By the soundness of the procedure *Saturate-KB*, we can limit ourself to consider only the models $\boldsymbol{\mathcal{M}}_\vartheta$ of $\phi_{\mathcal{KB}}$ induced by each open branch ϑ of $\mathcal{T}_{\mathcal{KB}}$. For every open and complete branch ϑ of $\mathcal{T}_{\mathcal{KB}}$, we construct a decision tree $\mathcal{D}_\vartheta$ in which every maximal branch of $\mathcal{D}_\vartheta$ defines a substitution σ' such that $\boldsymbol{\mathcal{M}}_\vartheta \models \psi_Q\sigma'$.

Let d be the number of literals in ψ_Q. $\mathcal{D}_\vartheta$ is a finite labelled tree of depth $d+1$ whose labelling satisfies the following conditions, for $i = 0, \ldots, d$:

(i) every node of $\mathcal{D}_\vartheta$ at level i is labelled with $(\sigma_i, \psi_Q\sigma_i)$; in particular, the root is labelled with $(\sigma'_0, \psi_Q\sigma'_0)$, where σ'_0 is the empty substitution;

(ii) if a node at level i is labelled with $(\sigma'_i, \psi_Q\sigma'_i)$, then its s successors, with $s > 0$, are labelled with

$$\big(\sigma'_i\varrho_1^{q_i+1}, \psi_Q(\sigma'_i\varrho_1^{q_i+1})\big), \ldots, \big(\sigma'_i\varrho_s^{q_i+1}, \psi_Q(\sigma'_i\varrho_s^{q_i+1})\big),$$

where q_{i+1} is the $(i+1)$-st conjunct of $\psi_Q\sigma'_i$ and $\mathcal{S}_{q_{i+1}} = \{\varrho_1^{q_i+1}, \ldots, \varrho_s^{q_i+1}\}$ is the collection of the substitutions $\varrho = \{x_1/y_1, \ldots, x_j/y_j\}$ with $\{x_1, \ldots, x_j\} = \mathtt{Var}_0(q_{i+1})$ such that $p = q_{i+1}\varrho$, for some literal p on ϑ. If $s = 0$, the node labelled with $(\sigma'_i, \psi_Q\sigma'_i)$ is a leaf node and, if $i = d$, σ'_i is added to Σ'.

It is easy to verify that the leaves of the maximal branches of $\mathcal{D}_\vartheta$ defines a substitution σ' such that $\boldsymbol{\mathcal{M}}_\vartheta \models \psi_Q\sigma'$.

Let $\delta(\mathcal{T}_{\mathcal{KB}})$ and $\lambda(\mathcal{T}_{\mathcal{KB}})$ be, respectively, the maximum depth of $\mathcal{T}_{\mathcal{KB}}$ and the number of leaves of the tableau $\mathcal{T}_{\mathcal{KB}}$ computed above. Plainly, $\delta(\mathcal{T}_{\mathcal{KB}}) = \mathcal{O}(\ell m k^r)$ and $\lambda(\mathcal{T}_{\mathcal{KB}}) = \mathcal{O}(2^{\ell m k^r})$. It is easy to verify that $s = 2^k$ is the maximum branching of $\mathcal{D}_\vartheta$. Since $\mathcal{D}_\vartheta$ is a s-ary tree of depth $d+1$, where d is the number of literals in ψ_Q, and the s successors of a node are computed in $\mathcal{O}(\delta(\mathcal{T}_{\mathcal{KB}}))$ time, the number of leaves in $\mathcal{D}_\vartheta$ is $\mathcal{O}(s^{(d+1)}) = \mathcal{O}(2^{k(d+1)})$ and they are computed in $\mathcal{O}(2^{k(d+1)}\delta(\mathcal{T}_{\mathcal{KB}}))$ time. Finally, since we have $\lambda(\mathcal{T}_{\mathcal{KB}})$ of such decision trees, the answer set of ψ_Q w.r.t. $\phi_{\mathcal{KB}}$ is computed in time

$$\mathcal{O}(2^{k(d+1)}\delta(\mathcal{T}_{\mathcal{KB}})\lambda(\mathcal{T}_{\mathcal{KB}})) = \mathcal{O}(2^{k(d+1)} \cdot \ell m k^r \cdot 2^{\ell m k^r}) = \mathcal{O}(\ell m k^r 2^{k(d+1)+\ell m k^r})\ .$$

As the sizes of $\phi_{\mathcal{KB}}$ and of ψ_Q are polynomially related to those of $\mathcal{KB}$ and of Q, respectively (see Section 3.2 for details on the reduction), the construction of the answer set of Q with respect to $\mathcal{KB}$ can be done in double-exponential time (**2EXPTIME**). When $\mathcal{KB}$ contains neither role chain axioms nor qualified cardinality restrictions, the complexity of the *CQA* problem is in **EXPTIME**, since in this case the maximum number of universal quantifiers in $\phi_{\mathcal{KB}}$ is a constant (in particular $r = 3$).

We now provide an example of application of the procedure *Saturate-KB* to a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KB. Let $\mathcal{KB}$ be the following knowledge base:

$$\begin{aligned}\mathcal{KB} :=\{&\{\mathit{Student} \sqcup \mathit{Italian} \sqsubseteq \mathit{PizzaLover},\\ &\mathit{PizzaLover} \sqcap \mathit{EatLowCalFood} \sqsubseteq \mathit{Slim}\},\\ &\{\mathit{Student}(\mathit{Clara})\}, \{\emptyset\}\}\,.\end{aligned} \tag{3}$$

We map the individual *Clara* to the $4\mathsf{LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-variable of sort 0 x_C and the concepts *Student*, *Italian*, *PizzaLover*, *EatLowCalFood*, and *Slim* to the $4\mathsf{LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-variables of sort 1 X^1_{ST}, X^1_I, X^1_{PL}, X^1_{ELCF}, X^1_{SL}, respectively. Then, by means of the mapping θ in Section 3.2, we define the following $4\mathsf{LQS}^\mathsf{R}$-formula representing $\mathcal{KB}$ in set-theoretic terms:

$$\begin{aligned}\phi_{\mathcal{KB}} =&(\forall x)((\neg(x \in X^1_{\mathsf{ST}}) \vee x \in X^1_{\mathsf{PL}}) \wedge (\neg(x \in X^1_{\mathsf{I}}) \vee x \in X^1_{\mathsf{PL}}))\wedge\\ &(\forall x)(\neg(x \in X^1_{\mathsf{PL}}) \vee \neg(x \in X^1_{\mathsf{ELCF}}) \vee x \in X^1_{\mathsf{SL}})\wedge\\ &x_C \in X^1_{\mathsf{ST}}\,.\end{aligned}$$

From $\phi_{\mathcal{KB}}$, we define the formula $\bar{\phi}_{\mathcal{KB}}$ by moving universal quantifiers as inward as possible and renaming universally quantified variables so as to make them pairwise distinct.

$$\begin{aligned}\bar{\phi}_{\mathcal{KB}} =&(\forall x)(\neg(x \in X^1_{\mathsf{ST}}) \vee x \in X^1_{\mathsf{PL}})\wedge\\ &(\forall y)(\neg(y \in X^1_{\mathsf{I}}) \vee y \in X^1_{\mathsf{PL}})\wedge\\ &(\forall z)(\neg(z \in X^1_{\mathsf{PL}}) \vee \neg(z \in X^1_{\mathsf{ELCF}}) \vee z \in X^1_{\mathsf{SL}})\wedge\\ &x_C \in X^1_{\mathsf{ST}}\,.\end{aligned}$$

The set $\Phi_{\mathcal{KB}}$ constructed from $\bar{\phi}_{\mathcal{KB}}$ by applying (2) is the following:

$$\begin{aligned}\Phi_{\mathcal{KB}} := \{&\neg(x_C \in X^1_{\mathsf{ST}}) \vee x_C \in X^1_{\mathsf{PL}},\\ &\neg(x_C \in X^1_{\mathsf{I}}) \vee x_C \in X^1_{\mathsf{PL}},\\ &\neg(x_C \in X^1_{\mathsf{PL}}) \vee \neg(x_C \in X^1_{\mathsf{ELCF}}) \vee x_C \in X^1_{\mathsf{SL}},\\ &x_C \in X^1_{\mathsf{ST}}\}\,.\end{aligned}$$

The set $\Phi_{\mathcal{KB}}$ in now given in input to the procedure *Saturate-KB* that returns the KE-tableau illustrated in Figure 2.

We are now ready to evaluate $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive queries over the KE-tableau $\mathcal{T}_{\mathcal{KB}}$ built by the procedure *Saturate-KB*. For example, let us consider the $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive query $Q = \mathit{Slim}(x) \wedge \mathit{Student}(x)$. The $4\mathsf{LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-formula representing Q is $\psi_Q = x_z \in X^1_{ST} \wedge x_z \in X^1_{SL}$. The forest of decision trees whose maximal branches represent the answer set of ψ_Q w.r.t. $\phi_{\mathcal{KB}}$ is illustrated in Figure 3.

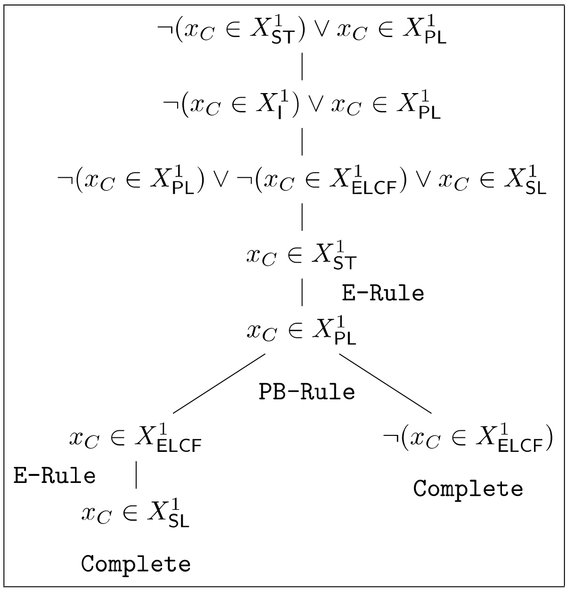

Figure 2: KE-tableau for $\Phi_{\mathcal{KB}}$.

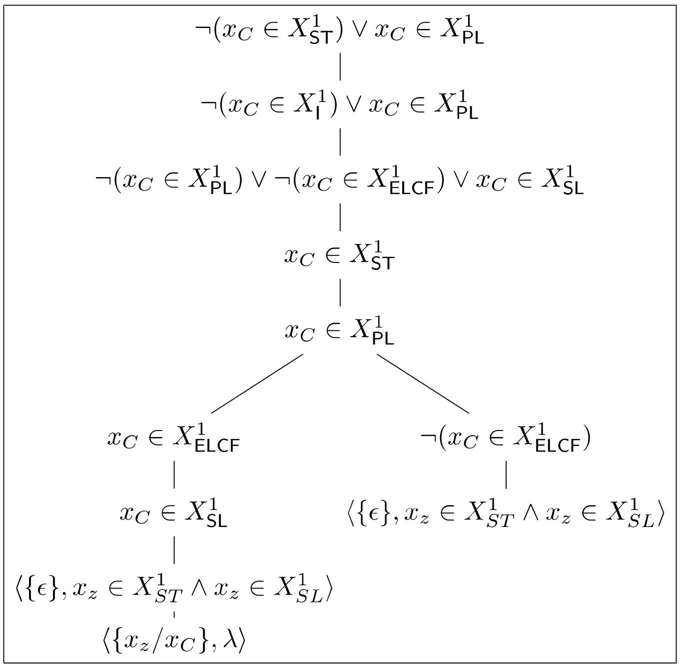

Figure 3: Decision trees for the evaluation of the conjunctive query Q.

Hence, by the mapping θ defined in Section 3.2 and extended in Section 4.1, and by Theorem 3 in Section 4.1, the answer set of Q w.r.t. $\mathcal{KB}$ by the forest of decision trees is:

$$\Sigma := \{\sigma^1_{\vartheta_1} := \{x/Clara\}, \sigma^1_{\vartheta_2} := \{\epsilon\}\}.$$

4.2 ABox Reasoning services for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$

In this section, we will study the decidability for the most widespread ABox reasoning tasks relative to the DL $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ by resorting to a general problem, called *Higher-Order Conjunctive Query Answering* (*HOCQA*) problem, which can be instantiated to each of them.

Specifically, let

- $\mathsf{V_i} = \{\mathsf{v}_1, \mathsf{v}_2, \ldots\}$,
- $\mathsf{V_e} = \{\mathsf{e}_1, \mathsf{e}_2, \ldots\}$,
- $\mathsf{V_d} = \{\mathsf{t}_1, \mathsf{t}_2, \ldots\}$,
- $\mathsf{V_c} = \{\mathsf{c}_1, \mathsf{c}_2, \ldots\}$,
- $\mathsf{V_{ar}} = \{\mathsf{r}_1, \mathsf{r}_2, \ldots\}$, and
- $\mathsf{V_{cr}} = \{\mathsf{p}_1, \mathsf{p}_2, \ldots\}$

be pairwise disjoint denumerable infinite sets of variables disjoint from **Ind**, $\bigcup\{N_C(d) : d \in N_{\mathbf{D}}\}$, **C**, $\mathbf{R_A}$, and $\mathbf{R_D}$. HO-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$-*atomic formulae* are expressions of the following types:

$$\begin{array}{ccc} R(w_1, w_2), & P(w_1, u), & C(w_1), \\ t(u), & \mathsf{r}(w_1, w_2), & \mathsf{p}(w_1, u), \\ \mathsf{c}(w_1), & \mathsf{t}(u), & w_1 = w_2, \end{array}$$

and their negations, where

- $w_1, w_2 \in \mathsf{V_i} \cup \mathbf{Ind}$,
- $u \in \mathsf{V_e} \cup \bigcup\{N_C(d) : d \in N_{\mathbf{D}}\}$,
- R is a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-abstract role term,
- P is a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-concrete role term,
- C is a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-concept term,
- t is a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-data type term,
- $\mathsf{r} \in \mathsf{V_{ar}}$,

- $\mathsf{p} \in \mathsf{V_{cr}}$,
- $\mathsf{c} \in \mathsf{V_c}$, and
- $\mathsf{t} \in \mathsf{V_d}$.

A HO-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$-atomic formula containing no variables is said to be *ground*. A HO-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$-*literal* is a HO-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$-atomic formula or its negation. A HO $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-*conjunctive query* is a conjunction of HO-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$-literals. We denote with λ the *empty* HO-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive query.

Let

$\mathsf{v}_1, \ldots, \mathsf{v}_n \in \mathsf{V_i}$,

$\mathsf{e}_1, \ldots, \mathsf{e}_g \in \mathsf{V_e}$,

$\mathsf{t}_1, \ldots, \mathsf{t}_l \in \mathsf{V_d}$,

$\mathsf{c}_1, \ldots, \mathsf{c}_m \in \mathsf{V_c}$,

$\mathsf{r}_1, \ldots, \mathsf{r}_k \in \mathsf{V_{ar}}$,

$\mathsf{p}_1, \ldots, \mathsf{p}_h \in \mathsf{V_{cr}}$,

$o_1, \ldots, o_n \in \mathbf{Ind}$,

$e_{d_1}, \ldots, e_{d_g} \in \bigcup\{N_C(d) : d \in N_{\mathbf{D}}\}$,

$t_1, \ldots, t_l \in \bigcup\{N_C(d) : d \in N_{\mathbf{D}}\}$,

$C_1, \ldots, C_m \in \mathbf{C}$,

$R_1, \ldots, R_k \in \mathbf{R_A}$, and

$P_1, \ldots, P_h \in \mathbf{R_D}$.

A substitution

$$\sigma \coloneqq \{\mathsf{v}_1/o_1, \ldots, \mathsf{v}_n/o_n, \mathsf{e}_1/e_{d_1}, \ldots, \mathsf{e}_g/e_{d_g}, \mathsf{t}_1/t_1, \ldots, \mathsf{t}_l/t_l, \mathsf{c}_1/C_1, \ldots, \mathsf{c}_m/C_m, \\ \mathsf{r}_1/R_1, \ldots, \mathsf{r}_k/R_k, \mathsf{p}_1/P_1, \ldots, \mathsf{p}_h/P_h\}$$

is a map such that, for every HO-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$-literal L, $L\sigma$ is obtained from L by replacing:

- the occurrences of v_i in L with o_i, for $i = 1, \ldots, n$;
- the occurrences of e_b in L with d_b, for $b = 1, \ldots, g$;
- the occurrences of t_s in L with t_s, for $s = 1, \ldots, l$;

- the occurrences of c_j in L with C_j, for $j = 1, \ldots, m$;
- the occurrences of r_ℓ in L with R_ℓ, for $\ell = 1, \ldots, k$;
- the occurrences of p_t in L with P_t, for $t = 1, \ldots, h$.

Substitutions can be extended to HO-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive queries in the usual way.

Let $Q := (L_1 \wedge \ldots \wedge L_m)$ be a HO-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive query, and $\mathcal{KB}$ a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KB. A substitution σ involving *exactly* the variables occurring in Q is a *solution for Q w.r.t. $\mathcal{KB}$* if there exists a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-interpretation $\mathbf{I}$ such that

$$\mathbf{I} \models_{\mathbf{D}} \mathcal{KB} \text{ and } \mathbf{I} \models_{\mathbf{D}} Q\sigma.$$

The collection Σ of the solutions for Q w.r.t. $\mathcal{KB}$ is the *Higher-Order Answer Set* of Q w.r.t. $\mathcal{KB}$. Then the *Higher-Order Conjunctive Query Answering problem* for Q w.r.t. $\mathcal{KB}$ consists in finding the HO-answer set Σ of Q w.r.t. $\mathcal{KB}$.

As remarked above, the *HOCQA* problem can be instantiated to significant ABox reasoning problems such as the ones illustrated in the beginning of Chapter 4.

Specifically, the instance checking problem (1.) is a specialization of the *HOCQA* problem admitting HO-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive queries of the form $Q_{IC} = C(w_1)$, with $w_1 \in \mathbf{Ind}$. The instance retrieval problem (2.) is a particular case of the *HOCQA* problem in which HO-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive queries have the form $Q_{IR} = C(w_1)$, where w_1 is a variable in V_i. The *HOCQA* problem can be instantiated to the role filler retrieval problem (3.) by admitting HO-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive queries $Q_{RF} = R(w_1, w_2)$, with $w_1 \in \mathbf{Ind}$ and w_2 a variable in V_i. The concept retrieval problem (4.) is a specialization of the *HOCQA* problem allowing HO-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive queries of the form $Q_{QR} = c(w_1)$, with $w_1 \in \mathbf{Ind}$ and c a variable in V_c. Finally, the role instance retrieval problem (5.) is a particularization of the *HOCQA* problem, where HO-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive queries have the form $Q_{RI} = r(w_1, w_2)$, with $w_1, w_2 \in \mathbf{Ind}$ and r a variable in V_cr.

In particular, the *CQA* problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ is an instance of the *HOCQA* problem with HO-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive queries of the form $(L_1 \wedge \ldots \wedge L_m)$, where the conjuncts L_i are atomic formulae of any of the types

$$R(w_1, w_2),\ C(w_1),\ P(w_1, u_1), \text{ and } w_1 = w_2 \text{ (or their negations),}$$

with $w_1, w_2 \in (\mathbf{Ind} \cup \mathsf{V}_\mathsf{i})$ and $u_1 \in \mathsf{V}_\mathsf{d} \cup \bigcup\{N_C(d) : d \in N_{\mathbf{D}}\}$. Notice also that problems (1.), (2.), and (3.) for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ are instances of the *CQA* problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$, whereas problems (4.) and (5.) for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ fall outside the definition of the *CQA* problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$.

We shall solve the *HOCQA* problem just stated by reducing it to the analogous problem formulated in the context of the set-theoretic fragment $\mathsf{4LQS^R}$ (and in turn to the decision procedure for $\mathsf{4LQS^R}$ presented in [26]). The *HOCQA* problem for $\mathsf{4LQS^R}$-formulae can be stated as follows. Let ϕ be a $\mathsf{4LQS^R}$-formula and let ψ be a conjunction of $\mathsf{4LQS^R}$-quantifier-free atomic formulae of level 0 of the types

$$x = y, \qquad x \in X^1, \qquad \langle x, y\rangle \in X^3,$$

or their negations.

The *HOCQA problem for ψ w.r.t. ϕ* consists in computing the HO *answer set of ψ w.r.t. ϕ*, namely the collection Σ' of all the substitutions σ' such that

(a) the support of σ' is a subset of the variables (of sorts 0, 1, and 3) in ψ, and

(b) $\phi \wedge \psi\sigma'$ is satisfiable.

In view of the decidability of the satisfiability problem for $\mathsf{4LQS^R}$-formulae, the *HOCQA* problem for $\mathsf{4LQS^R}$-formulae is decidable as well. Decidability of the *HOCQA* problem is proved in analogous way to the decidability of the *CQA* problem for $\mathsf{4LQS^R}$-formulae (see Section 4.1).

Lemma 2. *The HOCQA problem for* $\mathsf{4LQS^R}$*-formulae is decidable.* □

In an analogous way to the *CQA* problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$, the *HOCQA* problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ can be solved by a reduction to the *HOCQA* problem for $\mathsf{4LQS^R}$-formulae, and then by exploiting Lemma 2. The reduction is accomplished by means of the mapping function θ introduced in Section 3.2 and in Section 4.1 which we now extend in order to map also HO-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive queries into $\mathsf{4LQS^R}$-formulae in *CNF*. We exploit the resulting mapping function θ in order to map effectively the *HOCQA* problem from the $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-context into the $\mathsf{4LQS^R}$-context. Given a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KB $\mathcal{KB}$ and a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-HO-conjunctive query Q, by means of the mapping function θ the following $\mathsf{4LQS^R}$-formulae in *CNF* can be effectively constructed.

$$\begin{aligned} \phi_{\mathcal{KB}} &\coloneqq \bigwedge_{H \in \mathcal{KB}} \theta(H) \wedge \bigwedge_{i=1}^{12} \xi_i \\ \psi_Q &\coloneqq \theta(Q)\,. \end{aligned} \tag{4}$$

Then, it is sufficient to show that, denoting by Σ the HO-answer set of Q w.r.t. $\mathcal{KB}$ and by Σ' the HO-answer set of ψ_Q w.r.t. $\phi_{\mathcal{KB}}$, Σ consists of all the substitutions σ involving exactly the variables occurring in Q, and such that

$\theta(\sigma) \in \Sigma'$. Then, since by Lemma 2 Σ' can be computed effectively, Σ can be computed effectively too.

The mapping θ is extended to HO-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive queries as follows.

- $\theta(t(u)) \coloneqq x_u \in X_t^1$,
- $\theta(\mathsf{c}_1(w_1)) \coloneqq x_{w_1} \in X_{\mathsf{c}_1}^1$,
- $\theta(\mathsf{r}_1(w_1, w_2)) \coloneqq \langle x_{w_1}, x_{w_2} \rangle \in X_{\mathsf{r}_1}^3$,
- $\theta(\mathsf{p}_1(w_1, u_1)) \coloneqq \langle x_{w_1}, x_{u_1} \rangle \in X_{\mathsf{p}_1}^3$,
- $\theta(\mathsf{t}(u)) \coloneqq x_u \in X_{\mathsf{t}}^1$,
- $\theta(\neg \mathsf{c}_1(w_1)) \coloneqq \neg(x_{w_1} \in X_{\mathsf{c}_1}^1)$,
- $\theta(\neg \mathsf{r}_1(w_1, w_2)) \coloneqq \neg(\langle x_{w_1}, x_{w_2} \rangle \in X_{\mathsf{r}_1}^3)$,
- $\theta(\neg \mathsf{p}_1(w_1, u_1)) \coloneqq \neg(\langle x_{w_1}, x_{u_1} \rangle \in X_{\mathsf{p}_1}^3)$,
- $\theta(\neg \mathsf{t}(u)) \coloneqq \neg(x_u \in X_{\mathsf{t}}^1)$.

We also extend the mapping θ to the substitutions

$$\begin{aligned}\sigma \coloneqq \quad & \{\mathsf{v}_1/o_1, \ldots, \mathsf{v}_n/o_n, \mathsf{e}_1/e_{d_1}, \ldots, \mathsf{e}_g/e_{d_g}, \mathsf{c}_1/C_1, \ldots, \mathsf{c}_m/C_m, \\ & \mathsf{t}_1/t_1, \ldots, \mathsf{t}_l/t_l, \mathsf{r}_1/R_1, \ldots, \mathsf{r}_k/R_k, \mathsf{p}_1/P_1, \ldots, \mathsf{p}_h/P_h\}\end{aligned}$$

with

- $\mathsf{v}_1, \ldots, \mathsf{v}_n \in \mathsf{V_i}$,
- $\mathsf{e}_1, \ldots, \mathsf{e}_g \in \mathsf{V_e}$,
- $\mathsf{c}_1, \ldots, \mathsf{c}_m \in \mathsf{V_c}$,
- $\mathsf{r}_1, \ldots, \mathsf{r}_k \in \mathsf{V_{ar}}$,
- $\mathsf{p}_1, \ldots, \mathsf{p}_h \in \mathsf{V_{cr}}$,
- $\mathsf{t}_1, \ldots, \mathsf{t}_l \in \mathsf{V_d}$, $o_1, \ldots, o_n \in \mathbf{Ind}$,
- $e_{d_1}, \ldots e_{d_g}, t_1, \ldots, t_l \in \bigcup\{N_C(d) : d \in N_{\mathbf{D}}\}$,
- $C_1, \ldots, C_m \in \mathbf{C}$,
- $R_1, \ldots, R_k \in \mathbf{R_A}$, and
- $P_1, \ldots, P_h \in \mathbf{R_D}$,

by putting

$$
\begin{aligned}
\theta(\sigma) :=& \\
&\theta(\{\mathsf{v}_1/o_1, \ldots \mathsf{v}_n/o_n, \mathsf{e}_1/e_{d_1}, \ldots, \mathsf{e}_g/e_{d_g}, \mathsf{c}_1/C_1, \ldots, \mathsf{c}_m/C_m, \\
&\mathsf{r}_1/R_1, \ldots, \mathsf{r}_k, /R_k, \mathsf{p}_1/P_1, \ldots \mathsf{p}_h/P_h, \mathsf{t}_1/t_1, \ldots, \mathsf{t}_l/t_l\}) \\
=& \{x_{\mathsf{v}_1}/x_{o_1}, \ldots, x_{\mathsf{v}_n}/x_{o_n}, x_{\mathsf{e}_1}/x_{e_{d_1}}, \ldots, x_{\mathsf{e}_g}/x_{e_{d_g}}, X^1_{\mathsf{c}_1}/X^1_{C_1}, \ldots, X^1_{\mathsf{c}_m}/X^1_{C_m}, \\
&X^1_{\mathsf{t}_1}/X^1_{t_1}, \ldots, X^1_{\mathsf{t}_l}/X^1_{t_l}, X^3_{\mathsf{r}_1}/X^3_{R_1}, \ldots, X^3_{\mathsf{r}_k}, /X^3_{R_k}, X^3_{\mathsf{p}_1}/X^3_{P_1}, \ldots, X^3_{\mathsf{p}_h}/X^3_{P_h}, \} \\
=& \sigma'
\end{aligned}
\tag{5}
$$

where

- $x_{\mathsf{v}_1}, \ldots x_{\mathsf{v}_n}$, $x_{o_1}, \ldots, x_{o_n}$, $x_{\mathsf{t}_1}, \ldots x_{\mathsf{t}_l}$, $x_{d_1}, \ldots, x_{d_l}$ are variables of sort 0 in $\mathsf{4LQS^R}$,
- $X^1_{\mathsf{c}_1}, \ldots, X^1_{\mathsf{c}_m}$, $X^1_{C_1}, \ldots, X^1_{C_m}$ are variables of sort 1 in $\mathsf{4LQS^R}$,
- $X^3_{\mathsf{r}_1}, \ldots, X^3_{\mathsf{r}_k}$, $X^3_{\mathsf{p}_1}, \ldots, X^3_{\mathsf{p}_h}$, $X^3_{R_1}, \ldots, X^3_{R_k}$, and $X^3_{P_1}, \ldots, X^3_{P_h}$ are variables of sort 3 in $\mathsf{4LQS^R}$.

Theorem 4. *Given a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-knowledge base $\mathcal{KB}$ and a HO-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive query Q, the HOCQA problem for Q w.r.t. $\mathcal{KB}$ is decidable.*

Proof. Theorem 4 can be proved much along the same lines of the proof of Theorem 3. □

4.2.1 An algorithm for the *HOCQA* problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$

In this section, we will introduce an effective set-theoretic procedure to compute the answer set of a HO-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive query Q w.r.t. a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KB $\mathcal{KB}$. Such procedure, called *HOCQA-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$*, takes as input the $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-translation of $\mathcal{KB}$ $\phi_{\mathcal{KB}}$ and the $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-formula representing the HO-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive query Q ψ_Q, and returns a KE-tableau $\mathcal{T}_{\mathcal{KB}}$, representing the saturation of $\mathcal{KB}$, and the answer set Σ' of ψ_Q w.r.t. $\phi_{\mathcal{KB}}$ (namely the collection of all substitutions σ' such that $\boldsymbol{\mathcal{M}} \models \phi_{\mathcal{KB}} \wedge \psi_Q\sigma'$, for some $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-interpretation $\boldsymbol{\mathcal{M}}$).

Specifically, procedure *HOCQA-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$* constructs a complete KE-tableau $\mathcal{T}_{\mathcal{KB}}$ for the expansion $\Phi_{\mathcal{KB}}$ of $\phi_{\mathcal{KB}}$ (cf. (2)), representing the saturation of the $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KB $\mathcal{KB}$. Then, it constructs for each open branch of $\mathcal{T}_{\mathcal{KB}}$ a decision tree whose leaves are labelled with elements of Σ'.

In order to deal with literals of type $x = y$, we slightly modify the definitions concerning the KE-tableau introduced in Section 4.1.1.

Specifically, let $\mathcal{T}$ be a KE-tableau as defined in Section 4.1.1. A branch ϑ of $\mathcal{T}$ is *closed* if it contains either both A and $\neg A$, for some formula A, or a literal of type $\neg(x = x)$. Otherwise, the branch is *open*. A KE-tableau is *closed* if all its branches are closed. A formula $\beta_1 \vee \ldots \vee \beta_n$ is *fulfilled* in a branch ϑ, if β_i is in ϑ, for some $i = 1, \ldots, n$. A branch ϑ is *fulfilled* if every formula $\beta_1 \vee \ldots \vee \beta_n$ occurring in ϑ is fulfilled. A KE-tableau is *fulfilled* if all its branches are fulfilled. A formula, or a branch, or a KE-tableau that is not fulfilled is said to be *unfulfilled*. A branch ϑ is *complete* if either it is closed or it is open, fulfilled, and it does not contain any literal of type $x = y$, with x and y distinct variables. A KE-tableau is *complete* (resp., *fulfilled*) if all its branches are complete (resp., fulfilled or closed).

In addition, let ϑ be a branch of a KE-tableau. We denote with $<_\vartheta$ an arbitrary but fixed total order on the variables in $\mathsf{Var}_0(\vartheta)$.

Procedure *HOCQA*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ takes care of literals of type $x = y$ occurring in the branches of $\mathcal{T}_{\mathcal{KB}}$ by constructing, for each open and fulfilled branch ϑ of $\mathcal{T}_{\mathcal{KB}}$ a substitution σ_ϑ such that $\vartheta\sigma_\vartheta$ does not contain literals of type $x = y$ with distinct x, y. Then, for every open and complete branch $\vartheta' := \vartheta\sigma_\vartheta$ of $\mathcal{T}_{\mathcal{KB}}$, procedure *HOCQA*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ constructs a decision tree $\mathcal{D}_{\vartheta'}$ such that every maximal branch of $\mathcal{D}_{\vartheta'}$ induces a substitution σ' such that $\sigma_\vartheta\sigma'$ belongs to the answer set of ψ_Q with respect to $\phi_{\mathcal{KB}}$. The decision tree $\mathcal{D}_{\vartheta'}$ built by procedure *HOCQA*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ significantly differs from the definition given in Section 4.1.1 as it allows one to deal with HO-conjunctive queries and with literals of type $x = y$.

Specifically, $\mathcal{D}_{\vartheta'}$ is defined as follows.

Let d be the number of literals in ψ_Q. Then $\mathcal{D}_{\vartheta'}$ is a finite labelled tree of depth $d+1$ whose labelling satisfies the following conditions, for $i = 0, \ldots, d$:

(i) every node of $\mathcal{D}_{\vartheta'}$ at level i is labelled with $(\sigma'_i, \psi_Q\sigma_\vartheta\sigma'_i)$; in particular, the root is labelled with $(\sigma'_0, \psi_Q\sigma_\vartheta\sigma'_0)$, where σ'_0 is the empty substitution;

(ii) if a node at level i is labelled with $(\sigma'_i, \psi_Q\sigma_\vartheta\sigma'_i)$, then its s successors, with $s > 0$, are labelled with
$$\big(\sigma'_i\varrho_1^{q_{i+1}}, \psi_Q\sigma_\vartheta(\sigma'_i\varrho_1^{q_{i+1}})\big), \ldots, \big(\sigma'_i\varrho_s^{q_{i+1}}, \psi_Q\sigma_\vartheta(\sigma'_i\varrho_s^{q_{i+1}})\big),$$
where q_{i+1} is the $(i+1)$-st conjunct of $\psi_Q\sigma_\vartheta\sigma'_i$ and $\mathcal{S}_{q_{i+1}} = \{\varrho_1^{q_{i+1}}, \ldots, \varrho_s^{q_{i+1}}\}$ is the collection of the substitutions

$$\begin{aligned}\varrho = \{&x_{v_1}/x_{o_1}, \ldots, x_{v_n}/x_{o_n}, X^1_{c_1}/X^1_{C_1}, \ldots, X^1_{c_m}/X^1_{C_m},\\ &X^3_{r_1}/X^3_{R_1}, \ldots, X^3_{r_k}/X^3_{R_k}, X^3_{p_1}/X^3_{P_1}, \ldots, X^3_{p_h}/X^3_{P_h}\},\end{aligned} \qquad (6)$$

with

– $\{x_{v_1}, \ldots, x_{v_n}\} = \mathtt{Var}_0(q_{i+1})$,

– $\{X^1_{c_1}, \ldots, X^1_{c_m}\} = \mathtt{Var}_1(q_{i+1})$, and

– $\{X^3_{p_1}, \ldots, X^3_{p_h}, X^3_{r_1}, \ldots, X^3_{r_k}\} = \mathtt{Var}_3(q_{i+1})$

such that $t = q_{i+1}\varrho$, for some literal t on ϑ'. If $s = 0$, the node labelled with $(\sigma'_i, \psi_Q\sigma_\vartheta\sigma'_i)$ is a leaf node and, if $i = d$, $\sigma_\vartheta\sigma'_i$ is added to Σ'. In this case, the leaf node is contained in a non failing-branch and the substitution $\sigma_\vartheta\sigma'_i$ is a match for the query ψ.

We are now ready to present the procedure *HOCQA*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$.

procedure *HOCQA*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$($\psi_Q$,$\phi_{\mathcal{KB}}$);
  $\Sigma' := \emptyset$;
  - let $\Phi_{\mathcal{KB}}$ be the expansion of $\phi_{\mathcal{KB}}$ (cf. (2));
  $\mathcal{T}_{\mathcal{KB}} := \Phi_{\mathcal{KB}}$;
  while $\mathcal{T}_{\mathcal{KB}}$ is unfulfilled **do**
    - select an unfulfilled open branch ϑ of $\mathcal{T}_{\mathcal{KB}}$ and an unfulfilled formula $\beta_1 \vee \ldots \vee \beta_n$ in ϑ;
    if $\mathcal{S}_j^{\overline{\beta}}$ is in ϑ, for some $j \in \{1, \ldots, n\}$ **then**
      - apply the E-Rule to $\beta_1 \vee \ldots \vee \beta_n$ and $\mathcal{S}_j^{\overline{\beta}}$ on ϑ;
    else
      - let $B^{\overline{\beta}}$ be the collection of the formulae $\overline{\beta}_1, \ldots, \overline{\beta}_n$ present in ϑ and h the lowest index such that $\overline{\beta}_h \notin B^{\overline{\beta}}$;
      - apply the PB-rule to $\overline{\beta}_h$ on ϑ;
    end if;
  end while;
  while $\mathcal{T}_{\mathcal{KB}}$ has open branches containing literals of type $x = y$, with distinct x and y **do**
    - select such an open branch ϑ of $\mathcal{T}_{\mathcal{KB}}$;
    $\sigma_\vartheta := \epsilon$ (where ϵ is the empty substitution);
    $\mathsf{Eq}_\vartheta :=$ {literals of type $x = y$, occurring in ϑ};
    while Eq_ϑ contains $x = y$, with distinct x, y **do**
      - select a literal $x = y$ in Eq_ϑ, with distinct x, y;
      $z := \min_{<_\vartheta}(x, y)$;
      $\sigma_\vartheta := \sigma_\vartheta \cdot \{x/z, y/z\}$;
      $\mathsf{Eq}_\vartheta := \mathsf{Eq}_\vartheta\sigma_\vartheta$;
    end while;
    $\vartheta := \vartheta\sigma_\vartheta$;
    if ϑ is open **then**
      - initialize $\mathcal{S}$ to the empty stack;
      - push $(\epsilon, \psi_Q\sigma_\vartheta)$ in $\mathcal{S}$;
      while $\mathcal{S}$ is non-empty **do**
        - pop $(\sigma', \psi_Q\sigma_\vartheta\sigma')$ from $\mathcal{S}$;
        if $\psi_Q\sigma_\vartheta\sigma' \neq \lambda$ **then**

31: - let q be the leftmost conjunct of $\psi_Q\sigma_\vartheta\sigma'$;
32: $\psi_Q\sigma_\vartheta\sigma' := \psi_Q\sigma_\vartheta\sigma'$ deprived of q;
33: $Lit_Q^M := \{t \in \vartheta : t = q\rho$, for some substitution $\rho\}$;
34: **while** Lit_Q^M is non-empty **do**
35: - let $t \in Lit_Q^M$, $t = q\rho$;
36: $Lit_Q^M := Lit_Q^M \setminus \{t\}$;
37: - push $(\sigma'\rho, \psi_Q\sigma_\vartheta\sigma'\rho)$ in $\mathcal{S}$;
38: **end while**;
39: **else**
40: $\Sigma' := \Sigma' \cup \{\sigma_\vartheta\sigma'\}$;
41: **end if**;
42: **end while**;
43: **end if**;
44: **end while**;
45: **return** $(\mathcal{T}_{\mathcal{KB}}, \Sigma')$;
46: **end procedure**;

For each open branch ϑ of $\mathcal{T}_{\mathcal{KB}}$, the procedure *HOCQA-*$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ computes the corresponding $\mathcal{D}_\vartheta$ by constructing a stack of its nodes. Initially the stack contains the root node $(\epsilon, \psi_Q\sigma_\vartheta)$ of $\mathcal{D}_\vartheta$, as defined in condition (i). Then, iteratively, the following steps are executed. An element $(\sigma', \psi_Q\sigma_\vartheta\sigma')$ is popped out of the stack. If the last literal of the query ψ_Q has not been reached, the successors of the current node are computed according to condition (ii) and inserted in the stack. Otherwise the current node must have the form (σ', λ) and the substitution $\sigma_\vartheta\sigma'$ is inserted in Σ'. Notice that, in case of a failing query match, the set Lit_Q^M computed at step 33 is empty. Since the while-loop 34-38 is not executed, no successor node is pushed in the stack. Thus, the failing branch is abandoned and the procedure selects another branch by popping one of its nodes from the stack.

We now provide an example of execution of the procedure *HOCQA-*$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ with input a HO-conjunctive query and a knowledge base. Let us consider the knowledge base $\mathcal{KB}$ (3) used in the example of Section 4.1.1. The KE-tableau for $\mathcal{KB}$ constructed by the procedure *HOCQA-*$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ is the same as the one illustrated in Figure 2. We give in input to the procedure *HOCQA-*$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ the HO-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive query $Q = c(x)$, whose corresponding $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-representation is the formula $\psi_Q = x_z \in X_c^1$. The forest of decision trees constructed by the procedure *HOCQA-*$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ is shown in Figure 4.

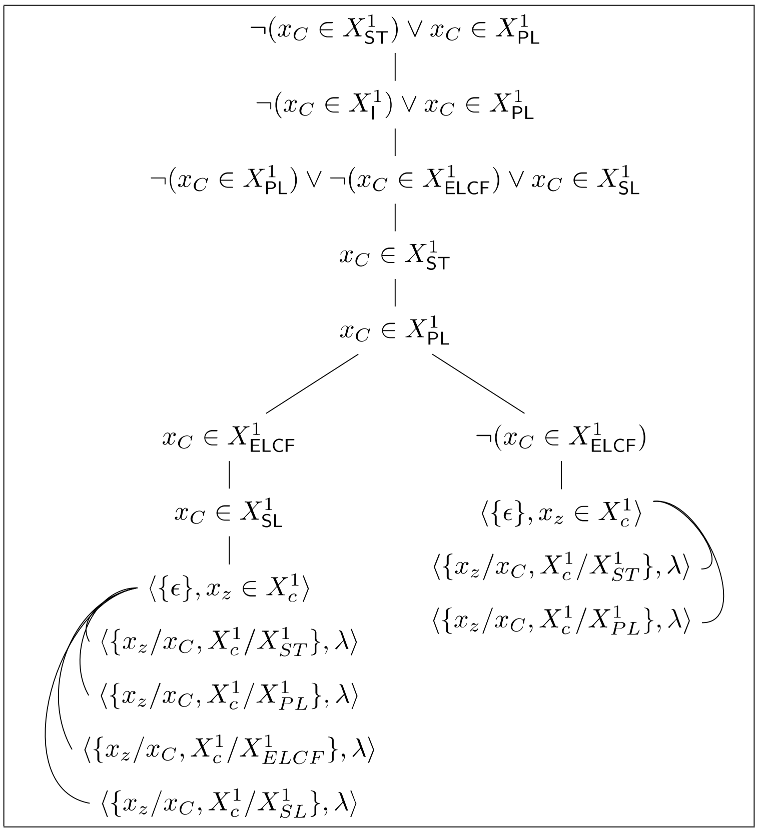

Figure 4: Decision trees for the evaluation of the HO-conjunctive query Q.

Finally, as shown in Figure 4, by the mapping θ defined in Section 3.2 and extended in Section 4.1 and in Section 4.2, and by Theorem 4 in Section 4.2, the answer set of Q w.r.t. $\mathcal{KB}$ computed by the procedure *HOCQA*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ is the following:

$$\begin{aligned}\Sigma := \{&\sigma^1_{\vartheta_1} := \{x/Clara, c/Studente\},\ \sigma^2_{\vartheta_1} := \{x/Clara, c/PizzaLover\},\\ &\sigma^3_{\vartheta_1} := \{x/Clara, c/EatLowCalFood\},\ \sigma^4_{\vartheta_1} := \{x/Clara, c/Slim\},\\ &\sigma^1_{\vartheta_2} := \{x/Clara, c/Studente\},\ \sigma^2_{\vartheta_2} := \{x/Clara, c/PizzaLover\}\}\,.\end{aligned}$$

4.2.1.1 Correctness of the procedure *HOCQA*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$

Correctness of the procedure *HOCQA*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ follows from Theorems 5 and 6, which show that $\phi_{\mathcal{KB}}$ is satisfiable if and only if $\mathcal{T}_{\mathcal{KB}}$ is a non-closed KE-tableau, and from Theorem 7, which shows that the set Σ' coincides with the HO-answer set of ψ_Q w.r.t. $\phi_{\mathcal{KB}}$. Theorems 5, 6, and 7 are stated below. In particular, Theorem 5 requires the following technical lemmas.

Lemma 3. *Let ϑ be a branch of $\mathcal{T}_{\mathcal{KB}}$ selected at step* 15 *of Procedure HOCQA-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ (ψ_Q,$\phi_{\mathcal{KB}}$), let σ_ϑ be the associated substitution constructed during the execution of the while-loop 18–23, and let $\boldsymbol{\mathcal{M}} = (D, M)$ be a $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-interpretation satisfying ϑ. Then*

$$Mx = Mx\sigma_\vartheta, \text{ for every } x \in \mathsf{Var}_0(\vartheta), \tag{7}$$

is an invariant of the while-loop 18–23.

Proof. We prove the thesis by induction on the number i of iterations of the while loop 18–23 of the procedure *HOCQA-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$* ($\psi_Q$,$\phi_{\mathcal{KB}}$). For simplicity, we indicate with $\sigma_\vartheta^{(i)}$ and with $Eq_{\sigma_\vartheta}^{(i)}$ the substitution σ_ϑ and the set Eq_{σ_ϑ} calculated at iteration $i \geq 0$, respectively.

If $i = 0$, then $\sigma_\vartheta^{(0)}$ is the empty substitution ϵ and thus (7) trivially holds.

Assume by inductive hypothesis that (7) holds at iteration $i \geq 0$. We want to prove that (7) holds at iteration $i + 1$.

At iteration $i+1$, $\sigma_\vartheta^{(i+1)} = \sigma_\vartheta^{(i)} \cdot \{x/z, y/z\}$, where $z = \min_{<_\vartheta}\{x, y\}$ and $x = y$ is a literal in $Eq_{\sigma_\vartheta}^{(i)}$, with distinct x, y. We assume, without loss of generality, that z is the variable x (an analogous proof can be carried out assuming that z is the variable y). By inductive hypothesis, $Mw = Mw\sigma_\vartheta^{(i)}$ for every $w \in \mathsf{Var}_0(\vartheta)$. If $w\sigma_\vartheta^{(i)} \in \mathsf{Var}_0(\vartheta) \setminus \{y\}$, plainly $w\sigma_\vartheta^{(i)}$ and $w\sigma_\vartheta^{(i+1)}$ coincide and thus $Mw\sigma_\vartheta^{(i)} = Mw\sigma_\vartheta^{(i+1)}$. Since $Mw = Mw\sigma_\vartheta^{(i)}$, it follows that $Mw = Mw\sigma_\vartheta^{(i+1)}$.

If $w\sigma_\vartheta^{(i)}$ coincides with y, we reason as follows. At iteration $i + 1$ variables x, y are considered because the literal $x = y$ is selected from $Eq_{\sigma_\vartheta}^{(i)}$. If $x = y$ is a literal belonging to ϑ, then $Mx = My$. Since $w\sigma_\vartheta^{(i)}$ coincides with y, $w\sigma_\vartheta^{(i+1)}$ coincides with x, $My = Mx$, and $Mw = Mw\sigma_\vartheta^{(i)}$ (by inductive hypothesis), then $Mw = Mw\sigma_\vartheta^{(i+1)}$. If $x = y$ is not a literal occurring in ϑ, then ϑ must contain a literal $x' = y'$ such that, at iteration i, x coincides with $x'\sigma_\vartheta^{(i)}$ and y coincides with $y'\sigma_\vartheta^{(i)}$. Since $Mx' = My'$, $Mx' = Mx'\sigma_\vartheta^{(i)}$ (by inductive hypothesis), and $My' = My'\sigma_\vartheta^{(i)}$, then $Mx = My$ and thus, by reasoning as above, $Mw = Mw\sigma_\vartheta^{(i+1)}$. Hence, (7) is an invariant of the loop, as we wished to prove, since it holds at each iteration of the while-loop. □

Lemma 4. *Let $\mathcal{T}_0, \ldots, \mathcal{T}_h$ be a sequence of KE-tableaux such that $\mathcal{T}_0 = \Phi_{\mathcal{KB}}$ and where $\mathcal{T}_{i+1}$ is obtained from $\mathcal{T}_i$ by applying either the rule of step 8, or the rule of step 11, or the substitution of step 24 of Procedure HOCQA-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ (ψ_Q,$\phi_{\mathcal{KB}}$), for $i = 1, \ldots, h-1$. If $\mathcal{T}_i$ is satisfied by a $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-interpretation $\boldsymbol{\mathcal{M}}$, then $\mathcal{T}_{i+1}$ is satisfied by $\boldsymbol{\mathcal{M}}$ as well, for $i = 1, \ldots, h-1$.*

Proof. Let $\boldsymbol{\mathcal{M}} = (D, M)$ be a $\mathsf{4LQS}^\mathsf{R}$-interpretation satisfying $\mathcal{T}_i$. Then $\boldsymbol{\mathcal{M}}$ satisfies a branch $\bar{\vartheta}$ of $\mathcal{T}_i$. In case the branch $\bar{\vartheta}$ is different from the branch selected at step 6 (if the E-rule (step 8) or the PB-rule (step 11) is applied), or at step 3 (if a substitution for handling equalities at step 24 is applied), $\bar{\vartheta}$ belongs to $\mathcal{T}_{i+1}$ and therefore $\mathcal{T}_{i+1}$ is satisfied by $\boldsymbol{\mathcal{M}}$. In case $\bar{\vartheta}$ is the branch selected and modified to obtain $\mathcal{T}_{i+1}$, we have to consider the following distinct cases.

- The branch $\bar{\vartheta}$ has been selected at step 6 and thus it is an open not yet fulfilled branch. Then, if step 8 is executed, the E-rule is applied to an unfulfilled formula $\beta_1 \vee \ldots \vee \beta_n$ and to the set of formulae $\mathcal{S}_j^{\overline{\beta}}$ on the branch $\bar{\vartheta}$ generating the new branch $\bar{\vartheta}' := \bar{\vartheta}; \beta_i$. Plainly, if $\boldsymbol{\mathcal{M}} \models \bar{\vartheta}$, then $\boldsymbol{\mathcal{M}} \models \beta_1 \vee \ldots \vee \beta_n$, $\boldsymbol{\mathcal{M}} \models \mathcal{S}_j^{\overline{\beta}}$ and, as a consequence, $\boldsymbol{\mathcal{M}} \models \beta_i$. Thus $\boldsymbol{\mathcal{M}} \models \bar{\vartheta}'$, so that $\boldsymbol{\mathcal{M}}$ satisfies $\mathcal{T}_{i+1}$. If step 10 is performed, the PB-rule is applied on $\bar{\vartheta}$ originating the branches (belonging to $\mathcal{T}_{i+1}$) $\bar{\vartheta}' := \bar{\vartheta}; \overline{\beta}_h$ and $\bar{\vartheta}'' := \bar{\vartheta}; \beta_h$. Since either $\boldsymbol{\mathcal{M}} \models \beta_h$ or $\boldsymbol{\mathcal{M}} \models \overline{\beta}_h$, then $\boldsymbol{\mathcal{M}} \models \bar{\vartheta}'$ or $\boldsymbol{\mathcal{M}} \models \bar{\vartheta}''$. Thus $\boldsymbol{\mathcal{M}}$ satisfies $\mathcal{T}_{i+1}$, as we wished to prove.
- The branch $\bar{\vartheta}$ has been selected at step 14 and thus it is an open and fulfilled branch not yet complete. Once step 24 is executed, the new branch $\bar{\vartheta}\sigma_{\bar{\vartheta}}$ is generated. Since $\boldsymbol{\mathcal{M}} \models \bar{\vartheta}$ and, by Lemma 3, $Mx = Mx\sigma_{\bar{\vartheta}}$, for every $x \in \mathsf{Var}_0(\bar{\vartheta})$, then $\boldsymbol{\mathcal{M}} \models \bar{\vartheta}\sigma_{\bar{\vartheta}}$, so that $\boldsymbol{\mathcal{M}}$ satisfies $\mathcal{T}_{i+1}$. Thus the thesis follows. □

Then we have:

Theorem 5. *If $\phi_{\mathcal{KB}}$ is satisfiable, then $\mathcal{T}_{\mathcal{KB}}$ is not closed.*

Proof. Let us assume by contradiction that $\mathcal{T}_{\mathcal{KB}}$ is closed. Since $\Phi_{\mathcal{KB}}$ is satisfiable, there exists a $\mathsf{4LQS^R}$-interpretation $\boldsymbol{\mathcal{M}}$ satisfying every formula of $\Phi_{\mathcal{KB}}$. Thanks to Lemma 4, any KE-tableau for $\Phi_{\mathcal{KB}}$ obtained by applying either step 8, or step 11, or step 24 of the procedure *HOCQA*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ is satisfied by $\boldsymbol{\mathcal{M}}$. Thus $\mathcal{T}_{\mathcal{KB}}$ is satisfied by $\boldsymbol{\mathcal{M}}$ as well. In particular, there exists a branch ϑ_c of $\mathcal{T}_{\mathcal{KB}}$ satisfied by $\boldsymbol{\mathcal{M}}$. Since by hypothesis $\mathcal{T}_{\mathcal{KB}}$ is closed, then the branch ϑ_c is closed as well and thus, by definition, it contains either both A and $\neg A$, for some formula A, or a literal of type $\neg(x = x)$. As the branch ϑ is satisfied by $\boldsymbol{\mathcal{M}}$, then either $\boldsymbol{\mathcal{M}} \models A$ and $\boldsymbol{\mathcal{M}} \models \neg A$, or $\boldsymbol{\mathcal{M}} \models \neg(x = x)$, contradicting the hypothesis. Thus, the KE-tableau $\mathcal{T}_{\mathcal{KB}}$ is not closed. □

Theorem 6. *If $\mathcal{T}_{\mathcal{KB}}$ is not closed, then $\phi_{\mathcal{KB}}$ is satisfiable.*

Proof. Since $\mathcal{T}_{\mathcal{KB}}$ is not closed, there exists a branch ϑ' of $\mathcal{T}_{\mathcal{KB}}$ which is open and complete. The branch ϑ' is obtained during the execution of the procedure *HOCQA*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ from an open fulfilled branch ϑ by applying to ϑ the substitution σ_ϑ constructed during the execution of step 14 of the procedure. Thus, $\vartheta' = \vartheta\sigma_\vartheta$. Since each formula of $\Phi_{\mathcal{KB}}$ occurs in ϑ, showing that ϑ is satisfiable is enough to prove that $\Phi_{\mathcal{KB}}$ is satisfiable.

We now construct a $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-interpretation $\boldsymbol{\mathcal{M}}_\vartheta$ satisfying every formula X occurring in ϑ and thus $\Phi_{\mathcal{KB}}$. The interpretation $\boldsymbol{\mathcal{M}}_\vartheta = (D_\vartheta, M_\vartheta)$ is defined as follows.

- $D_\vartheta \coloneqq \{x\sigma_\vartheta : x \in \mathsf{Var}_0(\vartheta)\}$;
- $M_\vartheta x \coloneqq x\sigma_\vartheta$, $x \in \mathsf{Var}_0(\vartheta)$;
- $M_\vartheta X^1 \coloneqq \{x\sigma_\vartheta : x \in X^1 \text{ occurs in } \vartheta\}$, $X^1 \in \mathsf{Var}_1(\vartheta)$;
- $M_\vartheta X^3 \coloneqq \{\langle x\sigma_\vartheta, y\sigma_\vartheta\rangle : \langle x, y\rangle \in X^3 \text{ occurs in } \vartheta\}$, $X^3 \in \mathsf{Var}_3(\vartheta)$.

We will show that $\boldsymbol{\mathcal{M}}_\vartheta$ satisfies each formula in ϑ. The proof is carried out by induction on the formula structure and cases distinction. Let us consider, at first, a literal $x = y$ occurring in ϑ. By the construction of σ_ϑ described in the procedure, $x\sigma_\vartheta$ and $y\sigma_\vartheta$ have to coincide. Thus $M_\vartheta x = x\sigma_\vartheta = y\sigma_\vartheta = M_\vartheta y$ and then $\boldsymbol{\mathcal{M}}_\vartheta \models x = y$.

Next, we consider a literal $\neg(z = w)$ occurring in ϑ. If $z\sigma_\vartheta$ and $w\sigma_\vartheta$ coincide, namely they are the same variable, then the branch $\vartheta' = \vartheta\sigma_\vartheta$ must be a closed branch, against the hypothesis. Thus $z\sigma_\vartheta$ and $w\sigma_\vartheta$ are distinct variables and therefore $M_\vartheta z = z\sigma_\vartheta \neq w\sigma_\vartheta = M_\vartheta w$. Hence $\boldsymbol{\mathcal{M}}_\vartheta \not\models z = w$, and finally $\boldsymbol{\mathcal{M}}_\vartheta \models \neg(z = w)$, as we wished to prove.

Let $x \in X^1$ be a literal occurring in ϑ. By the definition of M_ϑ, we have $x\sigma_\vartheta \in M_\vartheta X^1$, namely $M_\vartheta x \in M_\vartheta X^1$. Thus $\boldsymbol{\mathcal{M}}_\vartheta \models x \in X^1$, as desired. If $\neg(y \in X^1)$ occurs in ϑ, then $y\sigma_\vartheta \notin M_\vartheta X^1$. Assume, by contradiction that $y\sigma_\vartheta \in M_\vartheta X^1$. Then there is a literal $z \in X^1$ in ϑ such that $z\sigma_\vartheta$ and $y\sigma_\vartheta$ coincide. In this case the branch ϑ', obtained from ϑ by applying the substitution σ_ϑ, would be closed, contradicting the hypothesis. Thus $y\sigma_\vartheta \notin M_\vartheta X^1$ implies that $M_\vartheta y \notin M_\vartheta X^1$, so that $\boldsymbol{\mathcal{M}}_\vartheta \not\models y \in X^1$, i.e., $\boldsymbol{\mathcal{M}}_\vartheta \models \neg(y \in X^1)$.

If $\langle x, y\rangle \in X^3$ is a literal on ϑ, then by the very definition of M_ϑ, $\langle x\sigma_\vartheta, y\sigma_\vartheta\rangle \in M_\vartheta X^3$, that is $\langle M_\vartheta x, M_\vartheta y\rangle \in M_\vartheta X^3$, so that $\boldsymbol{\mathcal{M}}_\vartheta \models \langle x, y\rangle \in X^3$.

Let $\neg(\langle z, w\rangle \in X^3)$ be a literal occurring on ϑ, and assume that $\langle z\sigma_\vartheta, w\sigma_\vartheta\rangle \in M_\vartheta X^3$. Then there is a literal $\langle z', w'\rangle \in X^3$ in ϑ such that $z\sigma_\vartheta$ coincides with $z'\sigma_\vartheta$ and $w\sigma_\vartheta$ coincides with $w'\sigma_\vartheta$. But then, the branch $\vartheta' = \vartheta\sigma_\vartheta$ would be closed, contradicting the hypothesis. Thus, $\langle z\sigma_\vartheta, w\sigma_\vartheta\rangle \notin M_\vartheta X^3$, that is $\langle M_\vartheta z, M_\vartheta w\rangle \notin M_\vartheta X^3$. Hence $\boldsymbol{\mathcal{M}}_\vartheta \not\models \langle x, y\rangle \in X^3$, and finally $\boldsymbol{\mathcal{M}}_\vartheta \models \neg(\langle x, y\rangle \in X^3)$.

Let $\beta = \beta_1 \vee \ldots \vee \beta_k$ be a disjunction of literals in ϑ. Since ϑ is fulfilled, β is fulfilled too and, therefore, ϑ contains a disjunct β_i of β, for some $i \in \{1, \ldots, k\}$. By inductive hypothesis, $\boldsymbol{\mathcal{M}}_\vartheta \models \beta_i$, hence $\boldsymbol{\mathcal{M}}_\vartheta \models \beta$.

We have shown that $\boldsymbol{\mathcal{M}}_\vartheta$ satisfies each formula in ϑ and, in particular the formulae in $\Phi_{\mathcal{KB}}$. It turns out that $\Phi_{\mathcal{KB}}$ is satisfiable, as we wished to prove. $\square$

It is easy to check that the $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-interpretation $\boldsymbol{\mathcal{M}}_\vartheta$ defined in Theorem 6 satisfies $\phi_{\mathcal{KB}}$, a collection of $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-purely universal formulae and of $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-literals corresponding to a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KB $\mathcal{KB}$ and, therefore, that the following corollary holds.

Corollary 6.1. *If $\mathcal{T}_{\mathcal{KB}}$ is not closed, then $\phi_{\mathcal{KB}}$ is satisfiable.*

In what follows, we state and prove a technical lemma which is needed in the proof of Theorem 7.

Lemma 5. *Let $\psi_Q \coloneqq q_1 \wedge \ldots \wedge q_d$ be a HO-$\mathsf{4LQS}^{\mathsf{R}}$-conjunctive query, let $(\mathcal{T}_{\mathcal{KB}}, \Sigma')$ be the output of HOCQA-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$, and let ϑ' be an open and complete branch of $\mathcal{T}_{\mathcal{KB}}$. Then, for any substitution σ', we have*

$$\sigma' \in \Sigma' \iff \{q_1\sigma', \ldots, q_d\sigma'\} \subseteq \vartheta' \,.$$

Proof. If $\sigma' \in \Sigma'$, then $\sigma' = \sigma_\vartheta \sigma'_d$ and the decision tree $\mathcal{D}_{\vartheta'}$ contains a branch η of length $d+1$ having as leaf (σ'_d, λ). Specifically, the branch η comprises the nodes

$(\epsilon, q_1\sigma_\vartheta \wedge \ldots \wedge q_d\sigma_\vartheta)$, $(\rho^{(1)}, q_2\sigma_\vartheta\rho^{(1)} \wedge \ldots \wedge q_d\sigma_\vartheta\rho^{(1)})$, ..., $(\rho^{(1)} \ldots \rho^{(d)}, \lambda)$,

and hence $\sigma' = \sigma_\vartheta \rho^{(1)} \ldots \rho^{(d)}$.
Consider the node

$(\rho^{(1)} \ldots \rho^{(i+1)}, q_{i+2}\sigma_\vartheta\rho^{(1)} \ldots \rho^{(i+1)} \wedge \ldots \wedge q_d\sigma_\vartheta\rho^{(1)} \ldots \rho^{(i+1)})$

obtained from the father node

$(\rho^{(1)} \ldots \rho^{(i)}, q_{i+1}\sigma_\vartheta\rho^{(1)} \ldots \rho^{(i)} \wedge \ldots \wedge q_d\sigma_\vartheta\rho^{(1)} \ldots \rho^{(i)})$

since $q_{i+1}\sigma_\vartheta\rho^{(1)} \ldots \rho^{(i)} = t$, for some $t \in \vartheta'$. The literal $q_{i+1}\sigma_\vartheta\rho^{(1)} \ldots \rho^{(i)}$ is ground, therefore it coincides with $q_{i+1}\sigma'$. Thus $q_{i+1}\sigma' = t$, and hence $q_{i+1}\sigma' \in \vartheta'$. Given the generality of $i = 0, \ldots, d-1$, we have $\{q_1\sigma', \ldots, q_d\sigma'\} \subseteq \vartheta'$, as we wished to prove.

To prove that if $\{q_1\sigma', \ldots, q_d\sigma'\} \subseteq \vartheta'$ then $\sigma' \in \Sigma'$, we have to show that the decision tree $\mathcal{D}_{\vartheta'}$ constructed by the procedure *HOCQA-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$* has a branch η of length $d+1$ having as leaf a node (σ'_d, λ) such that $\sigma' = \sigma_\vartheta\sigma'_d$ (where σ_ϑ is the the substitution determined by the procedure *HOCQA-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$* such that $\vartheta' = \vartheta\sigma_\vartheta$). Let $(\epsilon, q_1\sigma_\vartheta \wedge \ldots \wedge q_d\sigma_\vartheta)$ be the root of the decision tree $\mathcal{D}_{\vartheta'}$. At step 29 of the procedure, the node $(\epsilon, q_1\sigma_\vartheta \wedge \ldots \wedge q_d\sigma_\vartheta)$ is popped out from the stack and the conjunct $q = q_1\sigma_\vartheta$ is selected. Then, all the elements of the set Lit_Q^M are considered, namely all the literals t in ϑ' such that $t = q_1\sigma_\vartheta\rho$, for some substitution ρ. Among them, we have also the substitution $q_1\sigma' = q_1\sigma_\vartheta\sigma'_d$. Thus, we put $\rho^{(1)} = \sigma'_d$ and at step 37 the procedure pushes the node $(\sigma'_d, q_2\sigma_\vartheta\sigma'_d \wedge \ldots \wedge q_d\sigma_\vartheta\sigma'_d)$ in the stack. Then, also the conjuncts $q_2\sigma_\vartheta\sigma'_d, \ldots, q_d\sigma_\vartheta\sigma'_d$ are processed sequentially. Since each of them coincides with a literal on ϑ', we have that $\rho^{(2)} = \ldots = \rho^{(d)} = \epsilon$. Considering that $\sigma'_d\epsilon = \sigma'_d$, the procedure builds the sequence of nodes

$$(\epsilon, q_1\sigma_\vartheta \wedge \ldots \wedge q_d\sigma_\vartheta)$$

$$(\sigma'_d, q_2\sigma_\vartheta\sigma'_d \wedge \ldots \wedge q_d\sigma_\vartheta\sigma'_d)$$

$$\vdots$$

$$(\sigma'_d, q_d\sigma_\vartheta\sigma'_d)$$

$$(\sigma'_d, \lambda)$$

constituting a branch η of length $d+1$ of $\mathcal{D}_{\vartheta'}$. Since η has as leaf node (σ'_d, λ), then $\sigma_\vartheta\sigma'_d = \sigma' \in \Sigma'$, as we wished to prove. □

Theorem 7. *Let Σ' be the set of substitutions returned by the procedure HOCQA-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ (ψ_Q, $\phi_{\mathcal{KB}}$). Then Σ' is the HO answer set of ψ_Q w.r.t. $\phi_{\mathcal{KB}}$.*

Proof. To prove the theorem, we show that the following two assertions hold.

1. If $\sigma' \in \Sigma'$, then σ' is an element of the HO answer set of ψ_Q w.r.t. $\phi_{\mathcal{KB}}$.

2. If σ' is a substitution of the HO answer set of ψ_Q w.r.t. $\phi_{\mathcal{KB}}$, then $\sigma' \in \Sigma'$.

We prove assertion (1.) as follows. Let $\sigma' \in \Sigma'$ and let $\vartheta' = \vartheta\sigma_\vartheta$ be an open and complete branch of $\mathcal{T}_{\mathcal{KB}}$ such that $\mathcal{D}_{\vartheta'}$ contains a branch η of $d+1$ nodes whose leaf is labelled $\langle\sigma'_d, \lambda\rangle$, where σ'_d is a substitution such that $\sigma' = \sigma_\vartheta\sigma'_d$. By Lemma 5, $\{q_1\sigma', \ldots, q_d\sigma'\} \subseteq \vartheta'$. Let $\boldsymbol{\mathcal{M}}_\vartheta$ be a $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-interpretation constructed as shown in Theorem 6. We have that $\boldsymbol{\mathcal{M}}_\vartheta \models q_i\sigma'$, for $i = 1, \ldots, d$ because $\{q_1\sigma', \ldots, q_d\sigma'\} \subseteq \vartheta'$ holds. Thus $\boldsymbol{\mathcal{M}}_\vartheta \models \psi_Q\sigma'$, and since $\boldsymbol{\mathcal{M}}_\vartheta \models \phi_{\mathcal{KB}}$, $\boldsymbol{\mathcal{M}}_\vartheta \models \phi_{\mathcal{KB}} \wedge \psi_Q\sigma'$ holds. Hence σ' is a substitution of the answer set of ψ_Q w.r.t. $\phi_{\mathcal{KB}}$. To show that assertion (2.) holds, let us consider a substitution σ' belonging to the answer set of ψ_Q w.r.t. $\phi_{\mathcal{KB}}$. Then there exists a $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-interpretation $\boldsymbol{\mathcal{M}} \models \phi_{\mathcal{KB}} \wedge \psi_Q\sigma'$. Assume by contradiction that $\sigma' \notin \Sigma'$. Then, by Lemma 5, we have $\{q_1\sigma, \ldots, q_d\sigma'\} \not\subseteq \vartheta'$, for every open and complete branch ϑ' of $\mathcal{T}_{\mathcal{KB}}$. In particular, given any open complete branch ϑ' of $\mathcal{T}_{\mathcal{KB}}$, there is an $i \in \{1, \ldots, d\}$ such that $q_i\sigma' \notin \vartheta' = \vartheta\sigma_\vartheta$ and thus $\boldsymbol{\mathcal{M}}_\vartheta \not\models q_i\sigma'$.

By the generality of $\vartheta' = \vartheta\sigma_\vartheta$, every $\boldsymbol{\mathcal{M}}_\vartheta$ satisfying $\mathcal{T}_{\mathcal{KB}}$ (and therefore also $\phi_{\mathcal{KB}}$) does not satisfy $\psi_Q\sigma'$. Since we can prove that $\boldsymbol{\mathcal{M}} \models \phi_{\mathcal{KB}} \wedge \psi_Q\sigma'$, for some $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-interpretation $\boldsymbol{\mathcal{M}}$, by restricting our interest to the interpretations $\boldsymbol{\mathcal{M}}_\vartheta$ of $\phi_{\mathcal{KB}}$ defined in the proof of Theorem 6, it turns out that σ' is not a substitution belonging to the answer set of ψ_Q w.r.t. $\phi_{\mathcal{KB}}$, a contradiction. Thus, the assertion (2.) holds. Finally, since the assertions (1.) and (2.) hold, Σ' and the answer set of ψ_Q w.r.t. $\phi_{\mathcal{KB}}$ coincide and the thesis holds. □

4.2.1.2 Termination of the procedure *HOCQA*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$

Termination of the procedure *HOCQA*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ is based on the fact that the *while-loops* 5–13 and 14–44 terminate.

Concerning termination of the *while-loop* 5–13, the proof is based on the following two facts. The E-Rule and PB-Rule are applied only to unfulfilled formulae on open branches and tend to reduce the number of unfulfilled formulae occurring on the considered branch. In particular, when the E-Rule is applied on a branch ϑ, the number of unfulfilled formulae on ϑ decreases. In case of applications of the PB-Rule to a formula $\beta = \beta_1 \vee \ldots \vee \beta_n$ on a branch, the rule generates two branches. In one of them the number of unfulfilled formulae decreases (because β becomes fulfilled). In the other one the number of unfulfilled formulae remains constant but the subset $B^{\overline{\beta}}$ of $\{\overline{\beta}_1, \ldots, \overline{\beta}_n\}$ occurring on the branch gains a new element. Once $|B^{\overline{\beta}}|$ gets equal to $n-1$, namely after at most $n-1$ applications of the PB-rule, the E-rule is applied and the formula $\beta = \beta_1 \vee \ldots \vee \beta_n$ becomes fulfilled, thus decrementing the number of unfulfilled formulae on the branch. Since the number of unfulfilled formulae on each open branch gets equal to zero after a finite number of steps and the E-rule and PB-rule can be applied only to unfulfilled formulae on open branches, the *while-loop* 5–13 terminates.

Concerning the *while-loop* 14–44, its termination can be proved by observing that the number of branches of the KE-tableau resulting from the execution of the previous *while-loop* 5–13 is finite and then showing that the internal *while-loops* 18–23 and 28–42 always terminate. Indeed, initially the set Eq_ϑ contains a finite number of literals of type $x = y$, and σ_ϑ is the empty substitution. It is then enough to show that the number of literals of type $x = y$ in Eq_ϑ, with distinct x and y, strictly decreases during the execution of the internal *while-loop* 18–23. But this follows immediately, since at each of its iterations one puts $\sigma_\vartheta := \sigma_\vartheta \cdot \{x/z, y/z\}$, with $z := \min_{<_\vartheta}(x, y)$, according to a fixed total order $<_\vartheta$ over the variables of $\mathsf{Var}_0(\vartheta)$ and then the application of σ_ϑ to Eq_ϑ replaces a literal of type $x = y$ in Eq_ϑ, with distinct x and y, with a literal of type $x = x$. The *while-loop* 28–42 terminates when the stack $\mathcal{S}$ of the nodes of the decision tree gets empty. Since the query ψ_Q contains a finite number of conjuncts and the number of literals on each open and complete branch of $\mathcal{T}_{\mathcal{KB}}$ is finite, the number of possible matches (namely the size of the set Lit_Q^M) computed at step (C) is finite as well. Thus, in particular, the internal *while-loop* 34–38 terminates at each execution. Once the procedure has processed the last conjunct of the query, the set Lit_Q^M of possible matches is empty and thus no element gets pushed in the stack $\mathcal{S}$ anymore. Since the first instruction of the *while-loop* at step (i) removes an element from $\mathcal{S}$, the stack gets empty after a finite number of "pops". Hence, the procedure *HOCQA*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ terminates, as we wished to prove.

4.2.1.3 Complexity of the procedure *HOCQA*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$

We now provide some complexity results.

Let r be the maximum number of universal quantifiers in each S_i ($i = 1, \ldots, m$), and put $k := |\mathtt{Var}_0(\phi_{\mathcal{KB}})|$. Then, each S_i generates at most k^r expansions. Since the knowledge base contains m such formulae, the number of disjunctions in the initial branch of the KE-tableau is bounded by $m \cdot k^r$. Next, let ℓ be the maximum number of literals in each S_i. Then, the height of the KE-tableau (which corresponds to the maximum size of the models of $\Phi_{\mathcal{KB}}$ constructed as illustrated above) is $\mathcal{O}(\ell m k^r)$ and the number of leaves of the tableau, namely the number of such models of $\Phi_{\mathcal{KB}}$, is $\mathcal{O}(2^{\ell m k^r})$. Notice that the construction of Eq_ϑ and of σ_ϑ in the lines 16–23 of the procedure *HOCQA*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ takes $\mathcal{O}(\ell m k^r)$ time, for each branch ϑ.

Let $\eta(\mathcal{T}_{\mathcal{KB}})$ and $\lambda(\mathcal{T}_{\mathcal{KB}})$ be, respectively, the height of $\mathcal{T}_{\mathcal{KB}}$ and the number of leaves of $\mathcal{T}_{\mathcal{KB}}$ computed by the procedure *HOCQA*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$. Plainly, $\eta(\mathcal{T}_{\mathcal{KB}}) = \mathcal{O}(\ell m k^r)$ and $\lambda(\mathcal{T}_{\mathcal{KB}}) = \mathcal{O}(2^{\ell m k^r})$, as computed above. It is easy to verify that $s = \mathcal{O}(\ell m k^r)$ is the maximum branching of $\mathcal{D}_\vartheta$. Since the height of $\mathcal{D}_\vartheta$ is h, where h is the number of literals in ψ_Q, and the successors of a node are computed in $\mathcal{O}(\ell m k^r)$ time, it follows that the number of leaves in $\mathcal{D}_\vartheta$ is $\mathcal{O}(s^h) = \mathcal{O}((\ell m k^r)^h)$, so that they are computed in $\mathcal{O}(s^h \cdot \ell k^r \cdot h) = \mathcal{O}(h \cdot (\ell m k^r)^{(h+1)})$ time. Finally, since we have $\lambda(\mathcal{T}_{\mathcal{KB}})$ of such decision trees, the answer set of ψ_Q w.r.t. $\phi_{\mathcal{KB}}$ is computed in time

$$\mathcal{O}(h \cdot (\ell m k^r)^{(h+1)} \cdot \lambda(\mathcal{T}_{\mathcal{KB}})) = \mathcal{O}(h \cdot (\ell m k^r)^{(h+1)} \cdot 2^{\ell m k^r}).$$

Since the size of $\phi_{\mathcal{KB}}$ and of ψ_Q are related to those of $\mathcal{KB}$ and of Q, respectively (see the proof of Theorem 4), the construction of the HO-answer set of Q with respect to $\mathcal{KB}$ can be done in double-exponential time. In case $\mathcal{KB}$ contains neither role chain axioms nor qualified cardinality restrictions, the complexity of the *HOCQA* problem is in **EXPTIME**, since the maximum number of universal quantifiers in $\phi_{\mathcal{KB}}$ is a constant (in particular $r = 3$). The latter complexity result is a clue of the fact that the *HOCQA* problem is intrinsically more difficult than the consistency problem (the latter proved to be **NP**-complete in Section 3.1). This is motivated by the fact that the consistency problem simply requires to guess a model of the knowledge base whereas the *HOCQA* problem forces the construction of all the models of the knowledge base and to compute a decision tree for each of them.

We remark that such result compares favourably to the complexity of the usual *CQA* problem for a wide collection of description logics such as the Horn fragment of $\mathcal{SHOIQ}$ and of $\mathcal{SROIQ}$, respectively, **EXPTIME**- and **2EXPTIME**-complete in combined complexity (see [101] for details).

4.2.2 An efficient variant of the KE$^\gamma$-tableau system

In this section, we will present an efficient variant of the KE-tableau system for sets of universally quantified clauses, where the KE-tableau-elimination rule is generalized in such a way as to incorporate the γ-rule. The novel system, called KE$^\gamma$-tableau, turns out to be an improvement of the system introduced in Section 4.2.1 and of the standard first-order KE-tableau [35].

Before presenting the novel procedures *Consistency*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ and *HOCQA*$^\gamma$-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ which solve the consistency problem of $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KBs and the *HOCQA* problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$, respectively, it is convenient to give various useful definitions and notations.

The procedure *Consistency*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ takes as input a $\mathsf{4LQS^R}$-formula $\phi_{\mathcal{KB}}$, representing a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KB, and checks its consistency. If $\phi_{\mathcal{KB}}$ is consistent, the procedure *Consistency*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ builds a KE-tableau $\mathcal{T}_{\mathcal{KB}}$, whose distinct open and complete branches induce the models of $\phi_{\mathcal{KB}}$. Then the procedure *HOCQA*$^\gamma$-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ computes the answer set of a given $\mathsf{4LQS^R}$-formula ψ_Q, representing a HO-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive query Q with respect to $\phi_{\mathcal{KB}}$ by means of a forest of decision trees based on the branches of the KE-tableau $\mathcal{T}_{\mathcal{KB}}$, computed by the procedure *Consistency*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ with input $\phi_{\mathcal{KB}}$.

Before presenting the procedures to be given next, we shortly introduce the variant of KE-tableau called KE$^\gamma$-tableau.

Let $\Phi \coloneqq \{C_1, \ldots, C_p\}$, where each C_i is either a $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-literal of the types illustrated in Table 9 or a $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-purely universal quantified formula of the form $(\forall x_1)\ldots(\forall x_m)(\beta_1 \vee \ldots \vee \beta_n)$, with $\beta_1, \ldots, \beta_n$ $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-literals. A tableau $\mathcal{T}$ is a KE$^\gamma$-tableau for Φ if there exists a finite sequence $\mathcal{T}_1, \ldots, \mathcal{T}_t$ such that:

(i) $\mathcal{T}_1$ is the one-branch tree consisting of the sequence $C_1, \ldots, C_p$,

(ii) $\mathcal{T}_t = \mathcal{T}$, and

(iii) for each $i < t$, $\mathcal{T}_{i+1}$ is obtained from $\mathcal{T}_i$ either by an application of one of the rules (E$^\gamma$-Rule or PB-rule) in Figure 5, or by applying a substitution σ to a branch ϑ of $\mathcal{T}_i$ (in particular, the substitution σ is applied to each formula X of ϑ, and the resulting branch will be denoted with $\vartheta\sigma$).

$$\frac{\psi \qquad \mathcal{S}^{\overline{\beta}_i\tau}}{\beta_i\tau}\ \textbf{E}^\gamma\textbf{-rule}$$

where
$\psi = (\forall x_1)\dots(\forall x_m)(\beta_1\vee\dots\vee\beta_n)$,
$\tau := \{x_1/x_{o_1}\dots x_m/x_{o_m}\}$,
and $\mathcal{S}^{\overline{\beta}_i\tau} := \{\overline{\beta}_1\tau,...,\overline{\beta}_n\tau\} \setminus \{\overline{\beta}_i\tau\}$, for $i = 1,...,n$

$$\frac{}{A \ \mid \ \overline{A}}\ \textbf{PB-rule}$$

where A is a literal

Figure 5: Expansion rules for the KE$^\gamma$-tableau.

In the definition of the E$^\gamma$-Rule reported in Figure 5, we have:

(a) $\tau := \{x_1/x_{o_1}\dots x_m/x_{o_m}\}$ is a substitution such that $x_1,\dots,x_m$ are the quantified variables in ψ and $x_{o_1},\dots,x_{o_m} \in \mathtt{Var}_0(\Phi)$;

(b) $\mathcal{S}^{\overline{\beta}_i\tau} := \{\overline{\beta}_1\tau,\dots,\overline{\beta}_n\tau\} \setminus \{\overline{\beta}_i\tau\}$ is a set containing the complements of all the disjuncts $\beta_1,\dots,\beta_n$ to which the substitution τ is applied, with the exception of the disjunct β_i.

Let $\mathcal{T}$ be a KE$^\gamma$-tableau. A branch ϑ of $\mathcal{T}$ is *closed* if either it contains both A and $\neg A$, for some formula A, or a literal of type $\neg(x = x)$. Otherwise, the branch is *open*. A KE$^\gamma$-tableau is *closed* if all its branches are closed. A formula $\psi = (\forall x_1)\dots(\forall x_m)(\beta_1 \vee \dots \vee \beta_n)$ is *fulfilled* in a branch ϑ, if ϑ contains $\beta_i\tau$, for some $i = 1,\dots,n$ and for all τ having as domain the set $Q\mathtt{Var}_0(\psi) = \{x_1,\dots,x_m\}$ of the quantified variables occurring in ψ and as range the set $\mathtt{Var}_0(\vartheta)$ of the variables of sort 0 occurring free in ϑ. Notice that, since the procedure *Consistency*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ to be defined next does not introduce any new variable, $\mathtt{Var}_0(\vartheta)$ coincides with $\mathtt{Var}_0(\phi_{\mathcal{KB}})$, for every branch ϑ. A branch ϑ is *fulfilled* if every formula $\psi = (\forall x_1)\dots(\forall x_m)(\beta_1\vee\dots\vee\beta_n)$ occurring in ϑ is fulfilled.

A KE$^\gamma$-tableau is *fulfilled* if all its branches are fulfilled. A formula, or a branch, or a KE$^\gamma$-tableau that is not fulfilled is said to be *unfulfilled*. A branch ϑ is *complete* if either it is closed or it is open, fulfilled, and it does not contain any literal of type $x = y$, with x and y distinct variables. A KE$^\gamma$-tableau is *complete* (resp., *fulfilled*) if all its branches are complete (resp., fulfilled or closed).

A $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-interpretation $\boldsymbol{\mathcal{M}}$ *satisfies* a branch ϑ of a KE$^\gamma$-tableau (or, equivalently, ϑ *is satisfied* by $\boldsymbol{\mathcal{M}}$), and we write $\boldsymbol{\mathcal{M}} \models \vartheta$, if $\boldsymbol{\mathcal{M}} \models X$, for every formula X occurring in ϑ.

A $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-interpretation $\boldsymbol{\mathcal{M}}$ satisfies a KE$^\gamma$-tableau $\mathcal{T}$ (or, equivalently, $\mathcal{T}$ *is satisfied* by $\boldsymbol{\mathcal{M}}$), and we write $\boldsymbol{\mathcal{M}} \models \mathcal{T}$, if $\boldsymbol{\mathcal{M}}$ satisfies a branch ϑ of $\mathcal{T}$.

A branch ϑ of a KE$^\gamma$-tableau $\mathcal{T}$ is *satisfiable* if there exists a $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-interpretation $\boldsymbol{\mathcal{M}}$ that satisfies ϑ. A KE$^\gamma$-tableau is satisfiable if at least one of its branches is satisfiable.

The procedures *Consistency*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ and $HOCQA^{\gamma}$-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ are shown next.

procedure *Consistency*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}(\phi_{\mathcal{KB}})$
  $\Phi_{\mathcal{KB}} := \{\phi : \phi \text{ is a conjunct of } \phi_{\mathcal{KB}}\}$;
  $\mathcal{T}_{\mathcal{KB}} := \Phi_{\mathcal{KB}}$;
  $\mathcal{E} := \emptyset$;
  while $\mathcal{T}_{\mathcal{KB}}$ is unfulfilled **do**
    - select an unfulfilled open branch ϑ of $\mathcal{T}_{\mathcal{KB}}$ and an unfulfilled formula $\psi = (\forall x_1) \ldots (\forall x_m)(\beta_1 \vee \ldots \vee \beta_n)$ in ϑ;
    $\Sigma_{\psi}^{\mathcal{KB}} := \{\tau : \tau = \{x_1/x_{o_1}, \ldots, x_m/x_{o_m}\}\}$, where $\{x_1, \ldots, x_m\} = Q\mathtt{Var}_0(\psi)$ and $\{x_{o_1}, \ldots, x_{o_m}\} \in \mathtt{Var}_0(\phi_{\mathcal{KB}})$;
    for $\tau \in \Sigma_{\psi}^{\mathcal{KB}}$ **do**
      if $\beta_i\tau \notin \vartheta$, for every $i = 1, \ldots, n$ **then**
        if $\mathcal{S}_j^{\overline{\beta}\tau}$ is in ϑ, for some $j \in \{1, \ldots, n\}$ **then**
          - apply the E^{γ}-Rule to ψ and $\mathcal{S}_j^{\overline{\beta}\tau}$ on ϑ;
        else
          - let $B^{\overline{\beta}\tau}$ be the collection of literals $\overline{\beta}_1\tau, \ldots, \overline{\beta}_n\tau$ present in ϑ and let h be the lowest index such that $\overline{\beta}_h\tau \notin B^{\overline{\beta}\tau}$;
          - apply the PB-rule to $\overline{\beta}_h\tau$ on ϑ;
        end if;
      end if;
    end for;
  end while;
  for ϑ in $\mathcal{T}_{\mathcal{KB}}$ **do**
    if ϑ is an open branch **then**
      $\sigma_{\vartheta} := \epsilon$ (where ϵ is the empty substitution);
      $\mathsf{Eq}_{\vartheta} := \{\text{literals of type } x = y, \text{ occurring in } \vartheta\}$;
      while Eq_{ϑ} contains $x = y$, with distinct x, y **do**
        - select a literal $x = y$ in Eq_{ϑ}, with distinct x, y;
        $z := min_{<_{x_0}}(x, y)$ (with $<_{x_0}$ an arbitrary but fixed total order on $\mathsf{Var}_0(\phi_{\mathcal{KB}})$);
        $\sigma_{\vartheta} := \sigma_{\vartheta} \cdot \{x/z, y/z\}$;
        $\mathsf{Eq}_{\vartheta} := \mathsf{Eq}_{\vartheta}\sigma_{\vartheta}$;
      end while;
      $\mathcal{E} := \mathcal{E} \cup \{(\vartheta, \sigma_{\vartheta})\}$;
      $\vartheta := \vartheta\sigma_{\vartheta}$;
    end if;
  end for;
  return $(\mathcal{T}_{\mathcal{KB}}, \mathcal{E})$;
end procedure;

procedure $HOCQA^{\gamma}$-$\mathcal{DL}_{\mathbf{D}}^{4,\times}(\psi_Q, \mathcal{E})$
  $\Sigma' := \emptyset$;
  while $\mathcal{E} \neq \emptyset$ **do**
    - let $(\vartheta, \sigma_\vartheta) \in \mathcal{E}$;
    - $\vartheta := \vartheta\sigma_\vartheta$;
    - initialize $\mathcal{S}$ to the empty stack;
    - push $(\epsilon, \psi_Q\sigma_\vartheta)$ in $\mathcal{S}$;
    while $\mathcal{S}$ is not empty **do**
      - pop $(\sigma', \psi_Q\sigma_\vartheta\sigma')$ from $\mathcal{S}$;
      if $\psi_Q\sigma_\vartheta\sigma' \neq \lambda$ **then**
        - let q be the leftmost conjunct of $\psi_Q\sigma_\vartheta\sigma'$;
        $\psi_Q\sigma_\vartheta\sigma' := \psi_Q\sigma_\vartheta\sigma'$ deprived of q;
        $Lit_q^\vartheta := \{t \in \vartheta : t = q\rho$, for some substitution $\rho\}$;
        while Lit_q^ϑ is not empty **do**
          - let $t \in Lit_q^\vartheta$, $t = q\rho$;
          $Lit_q^\vartheta := Lit_q^\vartheta \setminus \{t\}$;
          - push $(\sigma'\rho, \psi_Q\sigma_\vartheta\sigma'\rho)$ in $\mathcal{S}$;
        end while;
      else
        $\Sigma' := \Sigma' \cup \{\sigma_\vartheta\sigma'\}$;
      end if;
    end while;
    $\mathcal{E} := \mathcal{E} \setminus \{(\vartheta, \sigma_\vartheta)\}$;
  end while;
  return Σ';
end procedure;

As example of execution of the procedure *Consistency*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$, we consider the knowledge base $\mathcal{KB}$ (3) used in the example of Section 4.1.1. We recall that the $4\mathsf{LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-formula representing $\mathcal{KB}$ in set-theoretic terms is the following:

$$\begin{aligned}\phi_{\mathcal{KB}} = &(\forall x)(\neg(x \in X^1_{\mathsf{ST}}) \vee x \in X^1_{\mathsf{PL}}) \wedge \\ &(\forall y)(\neg(y \in X^1_{\mathsf{I}}) \vee y \in X^1_{\mathsf{PL}}) \wedge \\ &(\forall z)(\neg(z \in X^1_{\mathsf{PL}}) \vee \neg(z \in X^1_{\mathsf{ELCF}}) \vee z \in X^1_{\mathsf{SL}}) \wedge \\ &x_C \in X^1_{\mathsf{ST}}\,.\end{aligned}$$

The KE$^\gamma$-tableau returned by the procedure *Consistency*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ with input $\phi_{\mathcal{KB}}$ is illustrated in Figure 6.

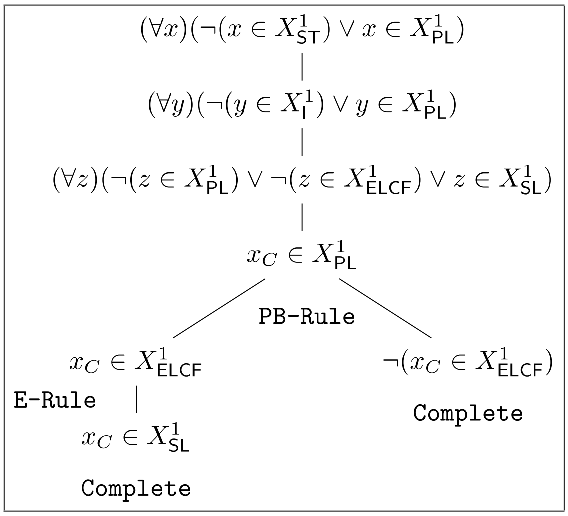

Figure 6: KE$^\gamma$-tableau for $\phi_{\mathcal{KB}}$.

4.2.2.1 Correctness of the procedures *Consistency-*$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ and *HOCQA*$^\gamma$*-*$\mathcal{DL}_{\mathbf{D}}^{4,\times}$

Correctness of the procedure *Consistency-*$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ follows from Theorems 8 and 9, which show that $\phi_{\mathcal{KB}}$ is satisfiable if and only if $\mathcal{T}_{\mathcal{KB}}$ is a non-closed KE$^\gamma$-tableau, whereas correctness of the procedure *HOCQA*$^\gamma$*-*$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ is proved by Theorem 10, which shows that the output set Σ' is the HO-answer set of ψ_Q w.r.t. $\phi_{\mathcal{KB}}$.

Before stating (and proving) Theorems 8, 9, and 10, we prove the following technical lemmas, which are needed for the proof of Theorem 8.

Lemma 6. *Let ϑ be a branch of $\mathcal{T}_{\mathcal{KB}}$ selected at step* 19 *of procedure Consistency-*$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ *($\phi_{\mathcal{KB}}$), σ_ϑ the associated substitution constructed during the execution of the* while*-loop 23–28, and $\boldsymbol{\mathcal{M}} = (D, M)$ a* $\mathsf{4LQS^R}$*-interpretation satisfying ϑ. Then,*

$$Mx = Mx\sigma_\vartheta, \text{ for every } x \in \mathsf{Var}_0(\vartheta), \tag{8}$$

is an invariant of the while*-loop 23–28.*

Proof. The thesis is proved by induction on the number i of iterations of the while-loop 23–28 of the procedure *HOCQA*$^\gamma$*-*$\mathcal{DL}_{\mathbf{D}}^{4,\times}$$(\psi_Q, \phi_{\mathcal{KB}})$ in an analogous way to the proof of Lemma 3 in Section 4.2.1.1. □

Lemma 7. *Let $\mathcal{T}_0, \ldots, \mathcal{T}_h$ be a sequence of* KE$^\gamma$*-tableaux, where $\mathcal{T}_0 = \phi_{\mathcal{KB}}$, and $\mathcal{T}_{i+1}$ is obtained from $\mathcal{T}_i$ by applying either the rule of step 11, or the rule of step 14, or the substitution of step 30 of procedure Consistency-*$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ *($\phi_{\mathcal{KB}}$), for*

$i = 1, \ldots, h-1$. *If* $\mathcal{T}_i$ *is satisfied by a* $4\mathsf{LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$*-interpretation* $\boldsymbol{\mathcal{M}}$*, then* $\mathcal{T}_{i+1}$ *is satisfied by* $\boldsymbol{\mathcal{M}}$ *as well, for* $i = 1, \ldots, h-1$.

Proof. Let $\boldsymbol{\mathcal{M}} = (D, M)$ be a $4\mathsf{LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-interpretation satisfying $\mathcal{T}_i$. Then $\boldsymbol{\mathcal{M}}$ satisfies a branch $\bar{\vartheta}$ of $\mathcal{T}_i$. In case the branch $\bar{\vartheta}$ is different from the branch selected at step 5, if the E^γ-Rule (step 11) or the PB-rule (step 14) is applied, or if a substitution for handling equalities (step 30) is applied, $\bar{\vartheta}$ belongs to $\mathcal{T}_{i+1}$ and therefore $\mathcal{T}_{i+1}$ is satisfied by $\boldsymbol{\mathcal{M}}$. In case $\bar{\vartheta}$ is the branch selected and modified to obtain $\mathcal{T}_{i+1}$, we have to consider the following two cases.

The branch $\bar{\vartheta}$ has been selected at step 6 (and thus it is an open branch not yet fulfilled): Let $\psi = (\forall x_1) \ldots (\forall x_m)(\beta_1 \vee \ldots \vee \beta_n)$ be the unfulfilled formula selected in $\bar{\vartheta}$, and $\tau = \{x_1/x_{o_1}, \ldots, x_m/x_{o_m}\}$ the substitution in $\Sigma_\psi^{\mathcal{KB}}$ chosen at step 8. If $\beta_i\tau \in \theta$, for some $i = i, \ldots, n$, then step 8 proceeds with the next iteration. Otherwise, if step 10 is executed, the E^γ-Rule is applied to the formula $\psi = (\forall x_1) \ldots (\forall x_m)(\beta_1 \vee \ldots \vee \beta_n)$ and to the set of formulae $\mathcal{S}_j^{\overline{\beta}\tau}$ on the branch $\bar{\vartheta}$, generating the new branch $\bar{\vartheta}' := \bar{\vartheta}; \beta_i\tau$. Since $\boldsymbol{\mathcal{M}} \models \bar{\vartheta}$, we plainly have that $\boldsymbol{\mathcal{M}} \models \psi = (\forall x_1) \ldots (\forall x_m)(\beta_1 \vee \ldots \vee \beta_n)$ and $\boldsymbol{\mathcal{M}} \models \mathcal{S}_j^{\overline{\beta}\tau}$.

Since $\boldsymbol{\mathcal{M}} \models \overline{\beta}_j\tau$, we have that $\boldsymbol{\mathcal{M}}[x_1/Mx_{o_1}, \ldots, x_m/Mx_{o_m}] \models \overline{\beta}_j$, for $j \in \{1, \ldots, n\} \setminus \{i\}$.

Considering that

$$\boldsymbol{\mathcal{M}} \models \psi = (\forall x_1) \ldots (\forall x_m)(\beta_1 \vee \ldots \vee \beta_n),$$

then

$$\boldsymbol{\mathcal{M}}[x_1/Mx_{o_1}, \ldots, x_m/Mx_{o_m}] \models \beta_1 \vee \ldots \vee \beta_n,$$

so that $\boldsymbol{\mathcal{M}}[x_1/Mx_{o_1}, \ldots, x_m/Mx_{o_m}] \models \beta_i$, namely, $\boldsymbol{\mathcal{M}} \models \beta_i\tau$, as we wished to prove.

If step 14 is performed, the PB-rule is applied on $\bar{\vartheta}$, originating the branches $\bar{\vartheta}' := \bar{\vartheta}; \overline{\beta}_h$ and $\bar{\vartheta}'' := \bar{\vartheta}; \beta_h$ (belonging to $\mathcal{T}_{i+1}$). Since either $\boldsymbol{\mathcal{M}} \models \beta_h$ or $\boldsymbol{\mathcal{M}} \models \overline{\beta}_h$, then either $\boldsymbol{\mathcal{M}} \models \bar{\vartheta}'$ or $\boldsymbol{\mathcal{M}} \models \bar{\vartheta}''$. Thus $\boldsymbol{\mathcal{M}}$ satisfies $\mathcal{T}_{i+1}$, as we wished to prove.

The branch $\bar{\vartheta}$ has been selected at step 19 (and thus it is an open and fulfilled branch not yet complete: Once step 30 is executed, the new branch $\bar{\vartheta}\sigma_{\bar{\vartheta}}$ is generated. Since $\boldsymbol{\mathcal{M}} \models \bar{\vartheta}$ and, by Lemma 6, $Mx = Mx\sigma_{\bar{\vartheta}}$, for every $x \in \mathsf{Var}_0(\bar{\vartheta})$, then $\boldsymbol{\mathcal{M}} \models \bar{\vartheta}\sigma_{\bar{\vartheta}}$, and therefore $\boldsymbol{\mathcal{M}}$ satisfies $\mathcal{T}_{i+1}$, completing the proof of the lemma. □

Theorem 8. *If* $\phi_{\mathcal{KB}}$ *is satisfiable, then* $\mathcal{T}_{\mathcal{KB}}$ *is not closed.*

Proof. Let us assume, for contradiction, that $\mathcal{T}_{\mathcal{KB}}$ is closed. Since $\phi_{\mathcal{KB}}$ is satisfiable, there exists a $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-interpretation $\boldsymbol{\mathcal{M}}$ satisfying every formula of $\phi_{\mathcal{KB}}$. Thanks to Lemma 7, any KE$^\gamma$-tableau for $\phi_{\mathcal{KB}}$ obtained by applying either step 11, or step 14, or step 30 of the procedure *Consistency*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ is satisfied by $\boldsymbol{\mathcal{M}}$. Thus, $\mathcal{T}_{\mathcal{KB}}$ is satisfied by $\boldsymbol{\mathcal{M}}$ as well. In particular, there exists a branch ϑ_c of $\mathcal{T}_{\mathcal{KB}}$ satisfied by $\boldsymbol{\mathcal{M}}$. From the initial assumption that $\mathcal{T}_{\mathcal{KB}}$ is closed, it follows that the branch ϑ_c is closed as well and thus it must contain either both A and $\neg A$, for some formula A, or a literal of type $\neg(x = x)$. But ϑ_c is satisfied by $\boldsymbol{\mathcal{M}}$; hence, either $\boldsymbol{\mathcal{M}} \models A$ and $\boldsymbol{\mathcal{M}} \models \neg A$ or $\boldsymbol{\mathcal{M}} \models \neg(x = x)$, which are clearly impossible. Thus, the KE$^\gamma$-tableau $\mathcal{T}_{\mathcal{KB}}$ is not closed, proving the theorem. □

Theorem 9. *If $\mathcal{T}_{\mathcal{KB}}$ is not closed, then $\phi_{\mathcal{KB}}$ is satisfiable.*

Proof. Since $\mathcal{T}_{\mathcal{KB}}$ is not closed, there must exist a branch ϑ' in $\mathcal{T}_{\mathcal{KB}}$ which is open and complete. The branch ϑ' is obtained during the execution of the procedure *Consistency*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ from an open fulfilled branch ϑ by applying to it the substitution σ_ϑ constructed during the execution of the while-loop at step 19 of the procedure. Thus, $\vartheta' = \vartheta\sigma_\vartheta$. Since each formula of $\phi_{\mathcal{KB}}$ occurs in ϑ, to prove that $\phi_{\mathcal{KB}}$ is satisfiable, it is enough to show that ϑ is satisfiable.

Let us construct a $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-interpretation $\boldsymbol{\mathcal{M}}_\vartheta = (D_\vartheta, M_\vartheta)$ satisfying every formula X occurring in ϑ and thus $\phi_{\mathcal{KB}}$. We put:

$$
\begin{aligned}
D_\vartheta &:= \{x\sigma_\vartheta : x \in \mathsf{Var}_0(\vartheta)\}; \\
M_\vartheta x &:= x\sigma_\vartheta, && \text{for every } x \in \mathsf{Var}_0(\vartheta); \\
M_\vartheta X^1 &:= \{x\sigma_\vartheta : x \in X^1 \text{ occurs in } \vartheta\}, && \text{for every } X^1 \in \mathsf{Var}_1(\vartheta); \\
M_\vartheta X^3 &:= \{\langle x\sigma_\vartheta, y\sigma_\vartheta\rangle : \langle x, y\rangle \in X^3 \text{ occurs in } \vartheta\}, && \text{for every } X^3 \in \mathsf{Var}_3(\vartheta).
\end{aligned}
$$

Next, we show that $\boldsymbol{\mathcal{M}}_\vartheta$ satisfies each formula in ϑ. We shall proceed by structural induction and case distinction. To begin with, we consider the case in which the literal $x = y$ occurs in ϑ. By the very construction of σ_ϑ, as described in procedure *Consistency*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$, $x\sigma_\vartheta$ and $y\sigma_\vartheta$ must coincide. Thus, $M_\vartheta x = x\sigma_\vartheta = y\sigma_\vartheta = M_\vartheta y$ and then $\boldsymbol{\mathcal{M}}_\vartheta \models x = y$.

Next, let us assume that the literal $\neg(z = w)$ occurs in ϑ. If $z\sigma_\vartheta$ and $w\sigma_\vartheta$ coincide, namely they are the same variable, then the branch $\vartheta' = \vartheta\sigma_\vartheta$ must be closed, contradicting the initial hypothesis. Thus, $z\sigma_\vartheta$ and $w\sigma_\vartheta$ must be distinct variables and therefore $M_\vartheta z = z\sigma_\vartheta \neq w\sigma_\vartheta = M_\vartheta w$. It follows that $\boldsymbol{\mathcal{M}}_\vartheta \not\models z = w$ and, therefore, $\boldsymbol{\mathcal{M}}_\vartheta \models \neg(z = w)$, as we wished to prove.

If $x \in X^1$ occurs in ϑ, then, by the very definition of M_ϑ, we have $x\sigma_\vartheta \in M_\vartheta X^1$, namely $M_\vartheta x \in M_\vartheta X^1$. Thus, $\boldsymbol{\mathcal{M}}_\vartheta \models x \in X^1$, as desired.

If $\neg(y \in X^1)$ occurs in ϑ, then $y\sigma_\vartheta \notin M_\vartheta X^1$. Assume, for contradiction, that $y\sigma_\vartheta \in M_\vartheta X^1$. Then there is a literal $z \in X^1$ in ϑ such that $z\sigma_\vartheta$ and $y\sigma_\vartheta$ coincide. In this case, the branch ϑ' obtained from ϑ by applying the substitution σ_ϑ would be closed, contradicting the initial hypothesis. Thus, we have $y\sigma_\vartheta \notin M_\vartheta X^1$, which implies $M_\vartheta y \notin M_\vartheta X^1$. Hence, $\boldsymbol{\mathcal{M}}_\vartheta \not\models y \in X^1$, so that $\boldsymbol{\mathcal{M}}_\vartheta \models \neg(y \in X^1)$.

If $\langle x, y\rangle \in X^3$ occurs in ϑ, then, by the very definition of M_ϑ, we have $\langle x\sigma_\vartheta, y\sigma_\vartheta\rangle \in M_\vartheta X^3$, that is, $\langle M_\vartheta x, M_\vartheta y\rangle \in M_\vartheta X^3$, so that $\boldsymbol{\mathcal{M}}_\vartheta \models \langle x, y\rangle \in X^3$.

Next, assume that $\neg(\langle z, w\rangle \in X^3)$ occurs in ϑ, but $\langle z\sigma_\vartheta, w\sigma_\vartheta\rangle \in M_\vartheta X^3$. Then a literal $\langle z', w'\rangle \in X^3$ occurs in ϑ such that $z\sigma_\vartheta$ coincides with $z'\sigma_\vartheta$ and $w\sigma_\vartheta$ coincides with $w'\sigma_\vartheta$. But then, the branch $\vartheta' = \vartheta\sigma_\vartheta$ would be closed, a contradiction. Thus, we must have $\langle z\sigma_\vartheta, w\sigma_\vartheta\rangle \notin M_\vartheta X^3$, that is $\langle M_\vartheta z, M_\vartheta w\rangle \notin M_\vartheta X^3$. Hence, $\boldsymbol{\mathcal{M}}_\vartheta \not\models \langle x, y\rangle \in X^3$, yielding $\boldsymbol{\mathcal{M}}_\vartheta \models \neg(\langle x, y\rangle \in X^3)$.

Finally, let $\psi \coloneqq (\forall x_1)\dots(\forall x_m)(\beta_1 \vee \dots \vee \beta_n)$ be a $\mathsf{4LQS^R}$-purely universal quantified formula of level 1 occurring in ϑ. Since ϑ is fulfilled, then ψ is fulfilled too, so that ϑ must contain the formula $\beta_i\tau$, for some $i = 1, \dots, n$ and for all τ in $\Sigma_\psi^{\mathcal{KB}}$. Let $\tau = \{x_1/x_{o_1}, \dots, x_m/x_{o_m}\}$ be any substitution in $\Sigma_\psi^{\mathcal{KB}}$. By inductive hypothesis, we have $\boldsymbol{\mathcal{M}}_\vartheta \models \beta_i\tau$, for some $i \in \{1, \dots, n\}$. Thus,

$$\boldsymbol{\mathcal{M}}_\vartheta[x_1/Mx_{o_1}, \dots, x_m/Mx_{o_m}] \models \beta_i$$

and, *a fortiori*,

$$\boldsymbol{\mathcal{M}}_\vartheta[x_1/Mx_{o_1}, \dots, x_m/Mx_{o_m}] \models \beta_1 \vee \dots \vee \beta_n.$$

From the generality of τ, it follows that $\boldsymbol{\mathcal{M}}_\vartheta \models (\forall x_1)\dots(\forall x_m)(\beta_1 \vee \dots \vee \beta_n)$, namely $\boldsymbol{\mathcal{M}}_\vartheta \models \psi$.

In conclusion, we have shown that $\boldsymbol{\mathcal{M}}_\vartheta$ satisfies each formula in ϑ and, in particular, all the formulae in $\phi_{\mathcal{KB}}$, as we wished to prove. □

Lemma 8. *Let $\psi_Q \coloneqq q_1 \wedge \dots \wedge q_d$ be a $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-HO-conjunctive query, Σ' the output of HOCQA$^\gamma$-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$($\psi_Q$, $\mathcal{E}$), and ϑ' an open and complete branch of $\mathcal{T}_{\mathcal{KB}}$. Then, for any substitution σ', we have:*

$$\sigma' \in \Sigma' \iff \{q_1\sigma', \dots, q_d\sigma'\} \subseteq \vartheta'\,.$$

Proof. Lemma 8 is proved in a similar way to Lemma 5 in Section 4.2.1.1. □

Theorem 10. *Let Σ' be the set of substitutions returned by the call to the procedure HOCQA$^\gamma$-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$($\psi_Q$, $\mathcal{E}$). Then Σ' is the HO-answer set of ψ_Q w.r.t. $\phi_{\mathcal{KB}}$.*

Proof. The theorem is proved much along the same lines of the proof of Theorem 7 in Section 4.2.1.1. □

4.2.2.2 Termination of the procedures *Consistency*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ and *HOCQA*$^{\gamma}$-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$

Termination of the procedure *Consistency*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ is based on the fact that the *while-loops* 5–18 and 19–32 terminate. In addition, the procedure *HOCQA*$^{\gamma}$-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ terminates, provided that the *while-loop* 8–22 terminates.

Concerning termination of the *while-loop* 5–18 of *Consistency*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$, the proof is grounded on the following facts. The loop selects iteratively an unfulfilled branch ϑ and a $\mathsf{4LQS}^{R}_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-purely universal quantified formula of level 1 $\psi = (\forall x_1)\ldots(\forall x_m)(\beta_1 \vee \ldots \vee \beta_n)$ occurring in it. Since the sets $Q\mathtt{Var}_0(\psi)$ and $\mathtt{Var}_0(\phi_{\mathcal{KB}})$ are finite, line 7 builds a finite set $\Sigma_{\psi}^{\mathcal{KB}}$ containing finite substitutions τ. The internal *for-loop* 8–17 selects iteratively an element τ in $\Sigma_{\psi}^{\mathcal{KB}}$. The E$^{\gamma}$-Rule and PB-rule are applied only if $\beta_i\tau \notin \vartheta$, for all $i = 1, \ldots, n$. In particular, if the E$^{\gamma}$-Rule is applied on ϑ, the procedure adds $\beta_i\tau$ in ϑ, for some $i = 1, \ldots, n$. In case the PB-rule is applied on ϑ, two branches are generated. On one branch the procedure adds $\beta_i\tau$, for some $i = 1, \ldots, n$, whereas on the other one it adds $\overline{\beta}_i\tau$, so that the set $B^{\overline{\beta}\tau}$ gains $\overline{\beta}_i\tau$ as a new element. After at most $n-1$ applications of the PB-rule, $\left|B^{\overline{\beta}\tau}\right|$ gets equal to $n-1$ and the E$^{\gamma}$-Rule is applied. Since the set $\Sigma_{\psi}^{\mathcal{KB}}$ is finite, the *for-loop* 8–17 terminates after a finite number of steps. After the last iteration of the *for-loop*, ϑ contains $\beta_i\tau$, for some $i = 1, \ldots, n$ and for all τ, thus ψ gets fulfilled. Since $\phi_{\mathcal{KB}}$ contains a finite number of formulae ψ, the *while-loop* 5–18 terminates in a finite number of steps, as we wanted to prove.

Proofs of termination of the *while-loop* 19–32 of the procedure *Consistency*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ and of the *while-loop* 8–22 of the procedure *HOCQA*$^{\gamma}$-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ follow along the same lines of the proof of termination of the *while-loop* 14–44 of the procedure *HOCQA*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$.

Termination of the *while-loop* 19–32 of the procedure *Consistency*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ can be proved by observing that the number of branches of the KE$^{\gamma}$-tableau resulting from the execution of the previous *while-loop* 5–18 is finite and then showing that the *while-loops* 19–32 and 23–28 always terminate.

Indeed, initially the set Eq_{ϑ} contains a finite number of literals of type $x = y$, and σ_{ϑ} is the empty substitution. It is then enough to show that the number of literals of type $x = y$ in Eq_{ϑ}, with distinct x and y, strictly decreases during the execution of the internal *while-loop* 23–28. But this follows immediately, since at each of its iterations one put $\sigma_{\vartheta} := \sigma_{\vartheta} \cdot \{x/z, y/z\}$, with $z := \min_{<_{\vartheta}}(x, y)$, according to a fixed total order $<_{\vartheta}$ over the variables of $\mathsf{Var}_0(\vartheta)$ and then the application of σ_ϑ to Eq_ϑ replaces a literal of type $x = y$ in Eq_ϑ, with distinct x and y, with a literal of type $x = x$.

The *while-loop* 8–22 of the procedure *HOCQA*$^\gamma$-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ terminates when the stack $\mathcal{S}$ of the nodes of the decision tree gets empty. Since the query ψ_Q contains a finite number of conjuncts and the number of literals on each open and complete branch of $\mathcal{T}_{\mathcal{KB}}$ is finite, the number of possible matches (namely the size of the set Lit_Q^M) computed at step (13) is finite as well. Thus, in particular, the internal *while-loop* 14–18 terminates at each execution. Once the procedure has processed the last conjunct of the query, the set Lit_Q^M of possible matches is empty and thus no element gets pushed in the stack $\mathcal{S}$ any more. Since the first instruction of the *while-loop* at step (i) removes an element from $\mathcal{S}$, the stack gets empty after a finite number of "pops". Hence the procedure *HOCQA*$^\gamma$-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ terminates, as we wished to prove.

4.2.2.3 Complexity issues

Next, we provide some complexity results.

Concerning the procedure *Consistency*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$, we reason as in the case of procedure *HOCQA*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$. Let ψ be any $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-purely universal quantified formula of level 1 in $\Phi_{\mathcal{KB}}$ (see line 2 of the procedure *Consistency*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ for the definition of $\Phi_{\mathcal{KB}}$). Let r be the maximum number of universal quantifiers in ψ, ℓ the maximum number of literals in ψ, and $k := |\mathtt{Var}_0(\Phi_{\mathcal{KB}})|$. It easily follows that $|\Sigma_\psi^{\mathcal{KB}}| = k^r$. Since the maximum number of literals contained in ψ is ℓ, the procedure applies $\ell - 1$ times the PB-Rule and one time the E$^\gamma$-Rule to $\psi\tau_j$, for $j = 1, \ldots, k^r$. Thus ψ generates a KE$^\gamma$-tableau of height $\mathcal{O}(\ell k^r)$.

Assuming that m is the number of $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-purely universal quantified formulae of level 1 in $\Phi_{\mathcal{KB}}$, the maximum height of the KE$^\gamma$-tableau (which corresponds to the maximum size of the models of $\phi_{\mathcal{KB}}$ that are constructed as illustrated in Theorem 9) is $\mathcal{O}(m\ell k^r)$ and the maximum number of leaves of the KE$^\gamma$-tableau, i.e., the maximum number of such models of $\phi_{\mathcal{KB}}$ is $\mathcal{O}(2^{m\ell k^r})$. Notice that the construction of Eq_ϑ and of σ_ϑ in the lines 19–32 of procedure *Consistency*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ takes $\mathcal{O}(m\ell k^r)$-time, for each branch ϑ.

Let $\eta(\mathcal{T}_{\mathcal{KB}})$ and $\lambda(\mathcal{T}_{\mathcal{KB}})$ be, respectively, the height of $\mathcal{T}_{\mathcal{KB}}$ and the number of leaves of $\mathcal{T}_{\mathcal{KB}}$ computed by the procedure *Consistency*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$. Plainly, $\eta(\mathcal{T}_{\mathcal{KB}}) = \mathcal{O}(\ell m k^r)$ and $\lambda(\mathcal{T}_{\mathcal{KB}}) = \mathcal{O}(2^{\ell m k^r})$, as computed above.

It is easy to verify that $s = \mathcal{O}(\ell m k^r)$ is the maximum branching of $\mathcal{D}_\vartheta$. Since the height of $\mathcal{D}_\vartheta$ is h, where h is the number of literals in ψ_Q, and the successors of a node are computed in $\mathcal{O}(\ell m k^r)$ time, it follows that the number of leaves in $\mathcal{D}_\vartheta$ is $\mathcal{O}(s^h) = \mathcal{O}((\ell m k^r)^h)$ and that they can be computed in

$$\mathcal{O}(s^h \cdot \ell m k^r \cdot h) = \mathcal{O}(h \cdot (\ell m k^r)^{(h+1)})\text{-time.}$$

Finally, since we have $\lambda(\mathcal{T}_{\mathcal{KB}})$ of such decision trees, the answer set of ψ_Q w.r.t. $\phi_{\mathcal{KB}}$ can be computed in time

$$\mathcal{O}(h \cdot (\ell m k^r)^{(h+1)} \cdot \lambda(\mathcal{T}_{\mathcal{KB}})) = \mathcal{O}(h \cdot (\ell m k^r)^{(h+1)} \cdot 2^{\ell m k^r}).$$

In consideration of the fact that the sizes of $\phi_{\mathcal{KB}}$ and ψ_Q are polynomially related to those of $\mathcal{KB}$ and of Q, respectively (see the proof of Theorem 8 for details on the reduction), the HO-answer set of Q with respect to $\mathcal{KB}$ can be computed in double-exponential time. It turns out that the asymptotic complexity of the procedures *Consistency*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ and *HOCQA*$^{\gamma}$-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ coincides with the asymptotic complexity of the procedure *HOCQA*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$. As we will show in the next section, the introduction of the KE$^{\gamma}$-tableau has therefore remarkable aftermaths since its implementation is very efficient.

5 A C++ reasoner for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$

In this chapter, we will introduce the C++ implementation of two reasoners based on the decision procedures introduced in this dissertation.

The first one is the KE-tableau-based reasoner (referred to as KE-system) and is based on the procedures *HOCQA*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$, whereas the second one is the KE$^\gamma$-tableau-based reasoner (referred to as KE$^\gamma$-system) and is based on the procedure *Consistency*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ and *HOCQA*$^\gamma$-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$. The two reasoners have been realized by exploiting the same data structures.

In this chapter, we will also provide a benchmarking of the two mentioned systems and of a system based on the FO KE-tableau [35] (referred to as FO KE-system). The latter system is realized much along the same lines of the KE$^\gamma$-system and implements the expansion rules presented in [35].

The source code of the implementation of the three reasoners is available via the GitHub repository in [30].

The rest of the chapter is organized as follows. In Section 5.1 and Section 5.2, we will provide a general overview of the KE-system and of the KE$^\gamma$-system, respectively. In Section 5.3 we will introduce some technical details concerning the two mentioned systems. In Section 5.4, we will provide some examples of reasoning with the KE-system and the KE$^\gamma$-system. Finally, in Section 5.5, we will describe how we performed a benchmarking of the KE-system, of the KE$^\gamma$-systemand of the FO-KE-system, and we will show that the KE$^\gamma$-system is more efficient than the other two.

5.1 Overview of the KE-system

We now give a general overview of the KE-system.

The input of the reasoner is an OWL ontology serialized in the OWL/XML syntax. The working flow of the reasoner is illustrated in Figure 7.

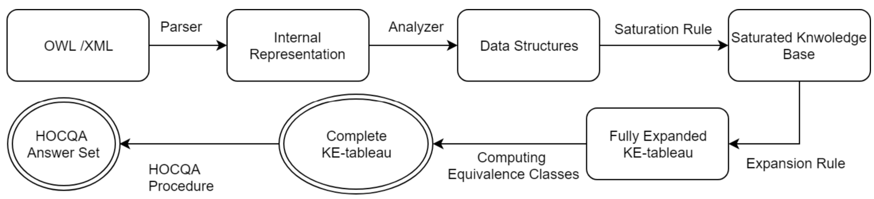

Figure 7: Execution cycle of the KE-tableau-system.

If the ontology meets the requirements of the DL $\mathcal{DL}_{\mathbf{D}}^{4,\times}$, an ad hoc parser produces the internal representation of all axioms and assertions of the ontology in set-theoretic terms as a list of strings. Such a translation exploits the function θ used in Section 4.2 to map a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KB $\mathcal{KB}$ into a set of $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-formulae. Each such string represents either a $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-literal or a $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-purely universally quantified formula (see Table 9 in Section 2.3.2). The internal representation turns out to be useful whenever one would include additional serializations, since the data structures have not to be modified. In the subsequent step, the reasoner builds the data structures required for the execution of the reasoning procedure, then it constructs the expansion of each $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-purely universally quantified formula in $\mathcal{KB}$ according to the procedure *HOCQA*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ in Section 4.2.1, yielding an expanded (ground) knowledge base, $\Phi_{\mathcal{KB}}$. Then a KE-tableau $\mathcal{T}_{\mathcal{KB}}$, representing the saturation of $\mathcal{KB}$, is constructed.

Initially a one-branch KE-tableau $\mathcal{T}_{\mathcal{KB}}$ for $\Phi_{\mathcal{KB}}$ is constructed. Then, the tableau $\mathcal{T}_{\mathcal{KB}}$ is expanded by systematically applying the E-Rule (elimination rule) and the PB-Rule (principle of bivalence rule) in Figure 1 in Section 4.1.1 to formulae of type $\beta_1 \vee \ldots \vee \beta_n$ till saturation, giving priority to the application of the E-Rule. Once such rules are no longer applicable, for each open branch ϑ of the resulting KE-tableau, atomic formulae of type $x = y$ occurring in ϑ are analysed in order to build for each ϑ the equivalence class of x and y. The system constructs the equivalence classes regarding the individuals involved in formulae of type $x = y$ occurring in ϑ and substitutes each individual x on ϑ with the representative of the equivalence class of x. Such step returns the complete KE-tableau.

The equivalence class is obtained by computing for each open branch ϑ of $\mathcal{T}_{\mathcal{KB}}$ the substitution σ_ϑ such that $\vartheta\sigma_\vartheta$ does not contain literals of type $x = y$, with distinct x, y. The resulting set of pair $(\vartheta, \sigma_\vartheta)$ is stored in the set $\mathcal{E}$. Finally, the reasoner is ready to evaluate a query ψ_Q given in input as a string coded in the internal representation.

5.2 Overview of the KE$^\gamma$-system

We now give a general overview of the KE$^\gamma$-system which has been implemented much along the same lines of the KE-system. The working flow of the KE$^\gamma$-system is reported in Figure 8.

As in the case of the KE-system, the input of the KE$^\gamma$-system is an OWL ontology serialized in the OWL/XML syntax and admitting SWRL rules.

If the ontology meets the requirements of the DL $\mathcal{DL}_{\mathbf{D}}^{4,\times}$, the parser produces the internal coding of all axioms and assertions of the input knowledge base $\mathcal{KB}$ in set-theoretic terms as a list of strings. Then, as in the KE-system, the KE$^\gamma$-system builds the data structures required to execute the algorithm, which are implemented in a similar way. In the two subsequent steps, the reasoner constructs a complete KE$^\gamma$-tableau $\mathcal{T}_{\mathcal{KB}}$ whose open branches represent all

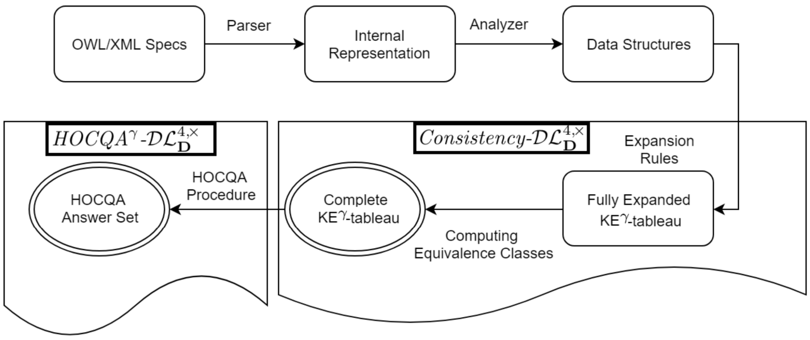

Figure 8: Execution cycle of the KE$^{\gamma}$-system.

possible models for the set-theoretic representation $\phi_{\mathcal{KB}}$ of $\mathcal{KB}$ (see Section 4.2.2 for the definition of KE$^{\gamma}$-tableau). The tableau $\mathcal{T}_{\mathcal{KB}}$ is constructed in two steps. In the first step, the KE$^{\gamma}$-system applies systematically the two expansion rules (see Figure 5 in Section 4.2.2): the first one is a generalization of the KE-elimination rule that incorporates the gamma rule and the second one is the principle of bivalence rule (PB-rule), thus constructing all branches of the KE$^{\gamma}$-tableau (see Figure 5 in Section 4.2.2). In the second step, the KE$^{\gamma}$-system processes each open branch ϑ of $\mathcal{T}_{\mathcal{KB}}$ as follows. As in the KE-system, the KE$^{\gamma}$-system constructs the equivalence classes regarding the individuals involved in formulae of type $x = y$ occurring in ϑ and substitutes each individual x on ϑ with the representative of the equivalence class of x. Such step returns the complete KE$^{\gamma}$-tableau. Finally, as in the KE-system, the reasoner takes as input the internal coding of ψ_Q, the set-theoretic representation of a query Q, and computes the HO-answer set of ψ_Q with respect to $\phi_{\mathcal{KB}}$. The task of computing the complete KE$^{\gamma}$-tableau for $\phi_{\mathcal{KB}}$ is performed by the implementation of the procedure *Consistency*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ illustrated in Section 4.2.2, whereas the task of computing the *HOCQA* answer set of a given query w.r.t. the knowledge base is performed by the implementation of the procedure *HOCQA*$^{\gamma}$-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$.

We recall that the procedure *Consistency*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ initially constructs a one-branch KE$^{\gamma}$-tableau $\mathcal{T}_{\mathcal{KB}}$ for the set $\Phi_{\mathcal{KB}}$ of conjuncts of $\phi_{\mathcal{KB}}$. Then, it expands $\mathcal{T}_{\mathcal{KB}}$ by systematically applying the E$^{\gamma}$-Rule and the PB-rule in Figure 5 of Section 4.2.2 to formulae of type $\psi = (\forall x_1) \ldots (\forall x_m)(\beta_1 \vee \ldots \vee \beta_n)$ till they are all fulfilled, giving priority to the application of the E$^{\gamma}$-Rule. Once such rules are no longer applicable, for each open branch ϑ of the resulting KE$^{\gamma}$-tableau, atomic formulae of type $x = y$ occurring in ϑ are used to compute the equivalence class of x and y. For each open branch ϑ of $\mathcal{T}_{\mathcal{KB}}$, the equivalence class of each variable occurring in ϑ is obtained by computing the substitution σ_ϑ such that $\vartheta\sigma_\vartheta$ does not contain literals of type $x = y$, with distinct x, y. The resulting pair $(\vartheta, \sigma_\vartheta)$ is stored in the set $\mathcal{E}$.

As explained in Section 4.2.2, the procedure *HOCQA*$^\gamma$*-*$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ takes as input a query ψ_Q and the set $\mathcal{E}$ yielded by the procedure *Consistency-*$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ and returns the answer set Σ' of ψ_Q w.r.t. $\phi_{\mathcal{KB}}$. For each open and complete branch ϑ of $\mathcal{T}_{\mathcal{KB}}$, the procedure *HOCQA*$^\gamma$*-*$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ constructs a decision tree $\mathcal{D}_\vartheta$ such that every maximal branch of $\mathcal{D}_\vartheta$ induces a substitution σ' such that $\sigma_\vartheta\sigma'$ belongs to the answer set of ψ_Q w.r.t. to $\phi_{\mathcal{KB}}$.

The procedure *HOCQA*$^\gamma$*-*$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ is applied to a given query ψ_Q and to the set $\mathcal{E}$ built by the procedure *Consistency-*$\mathcal{DL}_{\mathbf{D}}^{4,\times}$. For each open and complete branch ϑ of $\mathcal{T}_{\mathcal{KB}}$, the procedure *HOCQA*$^\gamma$*-*$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ constructs a decision tree $\mathcal{D}_\vartheta$ such that every maximal branch of $\mathcal{D}_\vartheta$ induces a substitution σ' such that $\sigma_\vartheta\sigma'$ belongs to the answer set of ψ_Q w.r.t. to $\phi_{\mathcal{KB}}$.

5.3 Implementation of the reasoners

We first show how the internal coding of $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KBs are represented in terms of $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-formulae, the data structures used by both reasoners for representing formulae, nodes, and how KE-system and KE$^\gamma$-system are implemented. Next, we describe the functions that implement the procedure *HOCQA-*$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ used by the KE-system and the functions that implement the procedures *Consistency-*$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ and *HOCQA*$^\gamma$*-*$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ used by the KE$^\gamma$-system. For the sake of conciseness, in this section we often refer to (1) the construction of the equivalence classes regarding the individuals involved in formulae of type $x = y$ occurring in a branch of either a KE-tableau or a KE$^\gamma$-tableau as to simply the construction of equivalence classes and to (2) the computation of the HO-answer set of ψ_Q with respect to $\phi_{\mathcal{KB}}$ as the construction of the answer set.

Since the KE-system and the KE$^\gamma$-system implement in an analogous way the data structures and the functions that compute the equivalence classes and the answer set, we organize this section as follows. We first introduce the data structures used by both reasoners, then we describe the function that implements the procedure *HOCQA-*$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ for what concerns the saturation of $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KBs and the function that implements the procedures *Consistency-*$\mathcal{DL}_{\mathbf{D}}^{4,\times}$. Finally, we describe the functions that implement the computation of equivalence classes and of the answer set.

To begin with, variables, quantifiers, Boolean operators, set-theoretic relators, and pairs of $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$ are mapped into strings as follows. Variables of type X^i_{name} are mapped into strings of the form $Vi\{name\}$. For the sake of uniformity, variables of sort 0 are denoted with $X^0, Y^0, \ldots$, whereas an individual a, a concept C, and a role R of a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KB are respectively mapped into the variables X^0_a, X^1_C, and X^3_R, according to the function θ described in Section 4.2. The symbols $\forall$, $\wedge$, $\vee$, $\neg\wedge$, $\neg\vee$ are mapped into the strings `$FA`, `$AD`, `$OR`,

`$DA`, `$RO`, respectively. The relators $\in$, $\notin$, $=$, $\neq$ are mapped into the strings `$IN`, `$NI`, `$EQ`, `$QE`, respectively. A pair $\langle X_1^0, X_2^0 \rangle$ is mapped in the string `$OA V01 $CO V02 $AO`, where `$OA` represents the bracket "$\langle$", `$AO` the bracket "$\rangle$", and `$CO` the comma symbol.

Data structures for representing $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KBs are the following.

$\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-variables are implemented by means of the class `Var` that has four fields. The field `type` of type integer defines the sort of the $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-variables, the field `name` of type string represents the name of the variable, and the field `var` of type integer represents a free variable when set to 0, or a purely universally quantified (bound) variable, when set to 1. The field `index` stores the position of the variable in the vector `VVL`, delegated to collect free variables.

Purely universal quantified variables and free variables are collected in the vectors `VQL` and `VVL` respectively, which provide a subvector for each sort of variable. The access to `VQL` and `VVL` is masked by the class `VariableSet`, introduced to protect the elements of the two vectors from direct manipulations.

The operators admitted in $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$ and internally coded as strings are mapped in three vectors that are fields of the class `Operator`. Specifically, we identify the vector `boolOp` with values `$OR`, `$AD`, `$RO`, `$DA`, the vector `setOp` with values `$IN`, `$EQ`, `$NI`, `$QE`, `$OA`, `$AO`, `$CO`, and the vector `qutOp` with values `$FA`.

$\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-literals are stored using the class `Lit` that has two fields. The field `litOp` of type integer represents the operator of the formula and corresponds to the index of one of the first four elements of the vector `setOp`. The field `components` is a vector whose elements point to the variables involved in the literal and stored in `VQL` and `VVL`.

$\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-formulae are represented by the class `Formula` having a binary tree structure, whose nodes contain an object of the class `Lit`.

The left and right children contain the left and right subformula, respectively. The class `Formula` contains the following fields. The field `lit` of type pointer to `Lit` represents the literal. The field `operand` of type integer represents the propositional operator and its value is the index of the corresponding element of the vector `boolOp`. The field `psubformula` of type pointer to `Formula` is the pointer to the father node, whereas the field `lsubformula` and the field `rsubformula` contain the pointers to the nodes representing the left and the right component of the formula, respectively.

We are now ready to describe the functions that implement the reasoning procedures of the KE-system.

As stated above, the first step of the KE-system consists in parsing the ontology from the OWL/XML file to the internal representation. Such a task is performed by the function `readOWLXMLontology` that takes in input the ontology serialized in the OWL/XML format and returns a vector of strings representing the internal coding of the KB. The elements of the vector so obtained are analysed and parsed by the function `insertFormulaKB` that returns an object of type `Formula` representing the input formula. At this step the formula is transformed in *CNF*. Additionally, universal quantifiers, if present, are moved as inward as possible and renamed in such a way as to be pairwise distinct. The function `insertFormulaKB` builds also the vectors `VVL` and `VQL`.

Once all input formulae have been parsed, the reasoner constructs the expansion of the KB by means of the procedure `expandKB` that yields the vector of the output formulae (`out`) from the vector of the input formulae (`inpf`).

In order to instantiate all the quantified variables, `expandKB` exploits a stack and the vectors `VVL` and `VQL`. After this step, the reasoner checks for atomic clashes in the expanded KB by means of the procedure `checkNodeClash`.

The construction of the KE-tableau is performed by the procedure `expandTableau` that exploits two stacks of type vector of pointers to `Node`. The first stack, named `noncomBranches`, keeps track of the non-complete branches, whereas the second one, called `unfulFormula`, keeps track of the unfulfilled disjunctive formulae. Initially, `expandTableau` attempts to empty the stack `unfulFormula` by selecting iteratively its elements and applying either the procedure `ERule` or the procedure `PBRule`, respectively implementing the E-Rule and the PB-Rule described in Figure 1 in Section 4.1.1, according to the procedure *HOCQA*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ in Section 4.2.1. The disjuncts of the current formula are stored in a temporary vector and selected iteratively. If a disjunct has its negation on the branch, it is removed from the vector. Once all disjuncts of the formula have been selected, if there is only an element in the stack, then the procedure `ERule` is applied to the disjunctive formula. If there is more than one element in the vector, then the procedure `PBRule` is applied. In case the stack is empty, a contradiction is found and the branch is closed. Clash checks are performed at each insertion of a formula, when a branch gets closed, it is added to `closedbranches`.

The procedure `expandTableau` terminates when either `noncomBranches` or `unfulFormula` are empty. When the procedure terminates with some elements in `noncomBranches`, such branches are added to the vector `openbranches`.

The next two phases consist in building the equivalence classes and in evaluating the input queries, respectively. Since the KE-system and KE$^{\gamma}$-system implement such steps in the same way, we first introduce functions that implement the KE$^{\gamma}$-system and then the functions that implement the construction of the equivalence classes and of the answer set.

As mentioned above, the KE$^{\gamma}$-system takes as input an OWL ontology compatible with the $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ requirements, also admitting SWRL rules and serialized in the OWL/XML syntax. As a first step, the function `readOWLXMLontology` produces the internal coding of all the axioms and assertions of the ontology, yielding a list of strings. Then the reasoner builds from the output of `readOWLXMLontology` the objects of type `Formula` that implement the $\mathsf{4LQS}^{R}_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-formulae representing the knowledge base, and stores them in the field `root` of an object of type `Tableau`. In this phase, the formulae are transformed in *CNF* and universal quantifiers are moved as inward as possible and renamed in such a way as to be pairwise distinct.

The object of type `Tableau` representing the KE$^{\gamma}$-tableau is the input of the procedure `expandGammaTableau` that expands the KE$^{\gamma}$-tableau by iteratively selecting and fulfilling the purely universal quantified formulae given in input. Once a purely universal quantified formula has been selected, `expandGammaTableau` builds iteratively the set of substitutions τ to be applied to the purely universal quantified formula selected. A substitution τ is a map from the quantified variables of the formula, selected in order of appearance, to the elements of the vector `VVL`. Such a map is a vector whose indices represent the quantified variables occurring in the selected formula and whose values are the indices of the elements in the vector `VVL` selected by the τ. The substitution τ is constructed by applying standard techniques for computing the *variations with repetition* of the set of indices of the elements of `VVL` taken k by k, where k is the number of quantified variables occurring in the selected formula.

The procedure `expandGammaTableau` fulfils the formula selected by systematically applying the functions `EGrule` with the current τ and `PBrule`, respectively implementing the E$^{\gamma}$-Rule and the PB-rule. More precisely, it works as follows. The disjuncts of the current formula to which τ is applied are stored in a temporary vector and selected iteratively. If a disjunct has its negation on the branch, it is removed from the temporary vector. Once all the elements of the temporary vector have been selected, if the last one does not have its negation on the branch, then `EGrule` is applied to the formula inserting the last element of the temporary vector in the branch according to Figure 5. If there is more than one element left in the temporary vector, then the procedure `PBrule` is applied. In case the stack is empty, a contradiction is found and the branch gets closed and inserted in the vector `closedbranches`.

If the procedure `expandGammaTableau` terminates with some elements in `openbranches`, then the reasoner builds the set of equivalence classes of the variables involved in formulae of type $X^0 = Y^0$, for each element of `openbranches` by means of the procedure `buildsEqSet`. The latter procedure updates the field `EqSet` of the object of type `Tableau` with the new information concerning the set of equivalence classes. After the execution of `buildsEqSet`, if `openbranches` contains some elements, a consistent knowledge base is returned.

We now describe the function that evaluates queries in both the KE-system and KE$^\gamma$-system. Such function, called `performQuery`, implements the procedure *HOCQA*$^\gamma$-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ (see Section 4.2.2) and the *while-loop* of lines $28 - 42$ of the procedure *HOCQA*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ (see Section 4.2.1).

The function `performQuery` takes as input the object of type `Tableau` returned by `buildsEqSet` and a string representing the internal coding of the input query ψ_Q, and yields an object of type `QueryManager` storing, among other information, the answer set of ψ_Q w.r.t. $\phi_{\mathcal{KB}}$. The execution of `performQuery` is based on an object of type `QueryManager` that stores the input query ψ_Q as a string, the data structure representing ψ_Q, namely an object of type `Formula`, and, for each element of `openbranches`, the answer set of ψ_Q w.r.t. $\phi_{\mathcal{KB}}$.

The answer set is implemented by endowing the object of type `QueryManager` with the pair of vectors `VarMatch`. The first vector of `VarMatch` contains an integer for each element in `openbranches`, which is set to 1 if the corresponding branch has solution, 0 otherwise. The second (three-dimensional) vector contains for each element in `openbranches` a vector of solutions, each one constituted by a vector of pairs of pointers to `Var`. The first `Var` of such a pair is a variable belonging to the query, whereas the second `Var` is the matched individual.

The function `performQuery` implements a decision tree for each element in `openbranches` by means of a stack that keeps track of the partial solutions of the query, in the nodes of the decision tree. Such a stack, called `matchSet` is constituted by a vector of pairs of objects of type `Var` such that the first object represents the query variable and the second one the matched element. Initially, `matchSet` is empty. At first step, the procedure selects the first conjunct of the query, and for each match found, it pushes in `matchSet` a vector of pairs representing the match. The procedure selects iteratively the conjuncts of the query and then applies to the selected conjunct the substitution that is currently at the top of `matchSet`. If the literal obtained by the application of such partial solution has one or more matches in the branch, the resulting substitutions are pushed in `matchSet`. Once all the literals of the query have been processed, if `matchSet` is not empty, then it contains the leaves of the maximal branches of the decision tree and hence its elements are all added to `VarMatch`. In this case, the element of the first vector of `VarMatch`, whose index corresponds to the selected branch, is set to 1.

5.4 Example of reasoning in $\mathcal{DL}_{\mathbf{D}}^{4,\times}$

In this section we will provide some examples of reasoning in $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ both with the KE-tableau and KE$^\gamma$-tableau. We will also show the results provided by the

KE-system and by the KE$^{\gamma}$-system when they are executed on the knowledge bases considered.

5.4.1 Example of reasoning with the KE-system

We now show how the KE-tableau is constructed and how the KE-system works. Let us consider the simple OWL ontology illustrated in Figure 9.

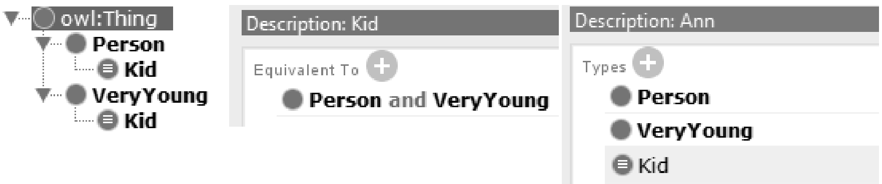

Figure 9: OWL ontology of the example.

Then, the representation of the ontology in terms of the DL $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ is the following knowledge base:

$$\begin{aligned}\mathcal{KB} :=& \{\{\emptyset\}, \{\mathsf{Kid} \equiv \mathsf{Person} \sqcap \mathsf{VeryYoung}\},\\ & \{\mathsf{Person}(Ann)\}\} \,.\end{aligned}$$

As mentioned above, the mapping function θ (see Section 4.2) is applied to $\mathcal{KB}$ thus yielding the following $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-formula $\phi_{\mathcal{KB}}$ representing $\mathcal{KB}$:

$$\begin{aligned}\phi_{\mathcal{KB}} = (\forall x)\big((\neg(x \in X^1_{\mathsf{Kid}}) \vee x \in X^1_{\mathsf{Person}})\\ \wedge (\neg(x \in X^1_{\mathsf{Kid}}) \vee x \in X^1_{\mathsf{VeryYoung}})\\ \wedge (\neg(x \in X^1_{\mathsf{Person}}) \vee \neg(x \in X^1_{\mathsf{VeryYoung}}) \vee x \in X^1_{\mathsf{Kid}})\big)\\ \wedge x_{Ann} \in X^1_{\mathsf{Person}} \,.\end{aligned}$$

In the subsequent step, the formula $\phi_{\mathcal{KB}}$ is transformed in *CNF*, universal quantifiers are moved as inward as possible, and universally quantified variables are renamed so as to make them pairwise distinct. The resulting $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-formula $\bar{\phi}_{\mathcal{KB}}$ is the following:

$$\begin{aligned}\bar{\phi}_{\mathcal{KB}} = &(\forall x)((\neg(x \in X^1_{\mathsf{Kid}}) \vee x \in X^1_{\mathsf{Person}})\wedge\\ &(\forall y)(\neg(y \in X^1_{\mathsf{Kid}}) \vee y \in X^1_{\mathsf{VeryYoung}})\wedge\\ &(\forall z)(\neg(z \in X^1_{\mathsf{Person}}) \vee \neg(z \in X^1_{\mathsf{VeryYoung}}) \vee z \in X^1_{\mathsf{Kid}}))\wedge\\ &x_{Ann} \in X^1_{\mathsf{Person}}\,.\end{aligned}$$

The internal representation of $\bar{\phi}_{\mathcal{KB}}$ computed by the KE-system is illustrated in Figure 10, whereas the vectors `VVL` and `VQL` are shown in Figure 11.

```
(V0{Ann} $IN V1{Person})
( (V0{x} $NI V1{Kid})  $OR (V0{x} $IN V1{Person}) )
( (V0{y} $NI V1{Kid})  $OR (V0{y} $IN V1{VeryYoung}) )
( ( (V0{z} $NI V1{Person})  $OR (V0{z} $NI V1{VeryYoung}) ) $OR (V0{z} $IN V1{Kid}) )
```

Figure 10: Internal representation of $\mathcal{KB}$.

```
Check VVL          Check VQL
Vector 0           Vector 0
V0{Ann}            V0{x}
Vector 1           V0{y}
V1{Person}         V0{z}
V1{VeryYoung}      Vector 1
V1{Kid}            Vector 3
Vector 3
```

Figure 11: The vectors `VVL` and `VQL` of $\mathcal{KB}$.

Then, the $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-formula $\Phi_{\mathcal{KB}}$ representing the expansion of $\bar{\phi}_{\mathcal{KB}}$ is computed. The formula $\Phi_{\mathcal{KB}}$ consists in the following collection of $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-literals:

$$\begin{aligned}\Phi_{\mathcal{KB}} = &\{\neg(x_A \in X^1_{\mathsf{Kid}}) \vee x_{Ann} \in X^1_{\mathsf{Person}},\\ &\neg(x_A \in X^1_{\mathsf{Kid}}) \vee x_A \in X^1_{\mathsf{VeryYoung}},\\ &\neg(x_A \in X^1_{\mathsf{Person}}) \vee \neg(x_A \in X^1_{\mathsf{VeryYoung}}) \vee x_{Ann} \in X^1_{\mathsf{Kid}},\\ &x_{Ann} \in X^1_{\mathsf{Person}}\}\,.\end{aligned}$$

The reasoner computes $\Phi_{\mathcal{KB}}$ by means of the function `expandKB`, yielding the result shown in Figure 12, where each line of the console output is the internal representation of an $\mathsf{4LQS}^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-formula mapped in an object of type `Formula`. According to the procedure *HOCQA*-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$ in Section 4.2.1, the initial KE-tableau $\mathcal{T}_{\mathcal{KB}}$ computed by the expansion function `expandKB` is constituted by the set of formulae $\Phi_{\mathcal{KB}}$. Specifically, $\Phi_{\mathcal{KB}}$ is stored in the field `setFormula` of an object of type `Node`, representing the root node of the class `Tableau`.

```
( ( (V0{Ann} $NI V1{Person})  $OR (V0{Ann} $NI V1{VeryYoung}) ) $OR (V0{Ann} $IN V1{Kid}) )
( (V0{Ann} $NI V1{Kid})  $OR (V0{Ann} $IN V1{VeryYoung}) )
( (V0{Ann} $NI V1{Kid})  $OR (V0{Ann} $IN V1{Person}) )
(V0{Ann} $IN V1{Person})
```

Figure 12: Expansion of $\mathcal{KB}$ as computed by the reasoner.

Then, the tableau $\mathcal{T}_{\mathcal{KB}}$ is expanded by systematically applying the E-Rule and the PB-Rule in Figure 1 in Section 4.1.1 to formulae of type $\beta_1 \vee \ldots \vee \beta_n$, till all β-formulae have been analysed. The resulting KE-tableau that consists of two complete open branches is illustrated in Figure 13. The complete open branches computed by the reasoner are shown in Figure 14.

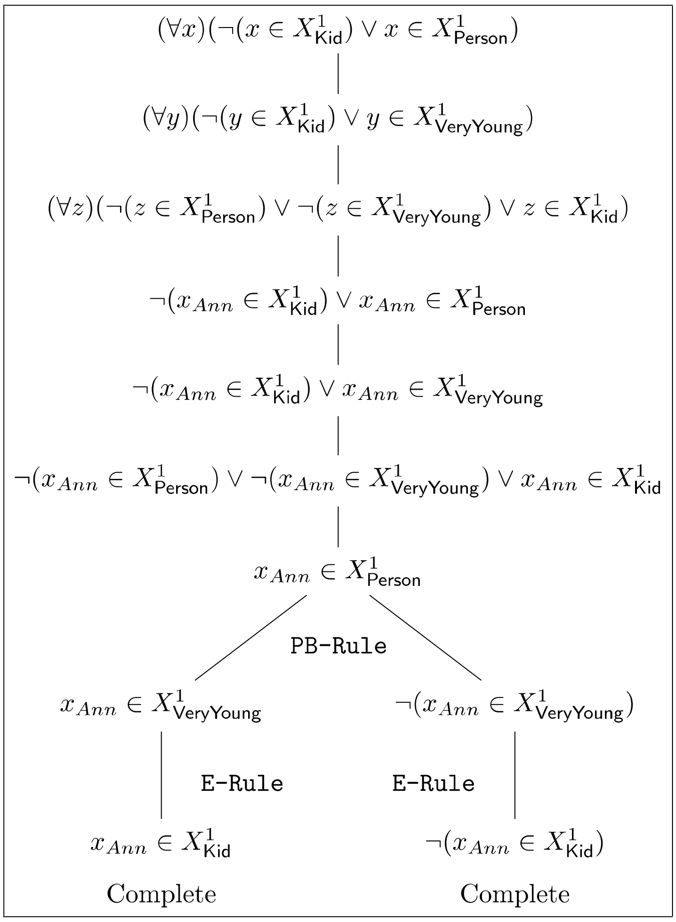

Figure 13: KE-tableau for $\phi_{\mathcal{KB}}$.

```
Branch: 0
(V0{Ann} $NI V1{VeryYoung})
(V0{Ann} $NI V1{Kid})
----
( ( (V0{Ann} $NI V1{Person})  $OR (V0{Ann} $NI V1{VeryYoung}) )
   $OR (V0{Ann} $IN V1{Kid}) )
( (V0{Ann} $NI V1{Kid})  $OR (V0{Ann} $IN V1{VeryYoung}) )
( (V0{Ann} $NI V1{Kid})  $OR (V0{Ann} $IN V1{Person}) )
(V0{Ann} $IN V1{Person})
----
Branch: 1
(V0{Ann} $IN V1{VeryYoung})
(V0{Ann} $IN V1{Kid})
----
( ( (V0{Ann} $NI V1{Person})  $OR (V0{Ann} $NI V1{VeryYoung}) )
   $OR (V0{Ann} $IN V1{Kid}) )
( (V0{Ann} $NI V1{Kid})  $OR (V0{Ann} $IN V1{VeryYoung}) )
( (V0{Ann} $NI V1{Kid})  $OR (V0{Ann} $IN V1{Person}) )
(V0{Ann} $IN V1{Person})
----
```

Figure 14: The open branches of the KE-tableau computed by the reasoner.

In the last step, for each open complete branch, the reasoner computes the set of equivalence classes for the individuals involved in formulae of type $x = y$ and checks for inconsistency. For instance, let us consider the following knowledge base $\mathcal{KB}_2$.

$$\begin{aligned}\mathcal{KB}_2 :=& \{\{\emptyset\},\\ & \{Person(Ann), Person(Paul), Person(John), Person(Carl),\\ & Annet \neq Ann, Ann = Anna, Paul = Paolo, Carl = Carlo\},\\ & \{\emptyset\}\}\,.\end{aligned}$$

The tableau $\mathcal{T}_{\mathcal{KB}_2}$ is the consistent one-branch KE-tableau shown in Figure 15. For the single branch of $\mathcal{T}_{\mathcal{KB}_2}$, the three equivalence classes computed by the reasoner are shown in Figure 16.

```
(V0{Annet} $QE V0{Anna})
(V0{Ann} $EQ V0{Anna})
(V0{Paul} $EQ V0{Paolo})
(V0{Carl} $EQ V0{Carlo})
(V0{Carl} $IN V1{Person})
(V0{John} $IN V1{Person})
(V0{Paul} $IN V1{Person})
(V0{Ann} $IN V1{Person})
```

Figure 15: The one-branch KE-tableau $\mathcal{T}_{\mathcal{KB}_2}$.

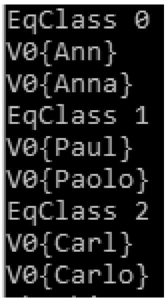

Figure 16: Set of equivalence classes for $\mathcal{T}_{\mathcal{KB}_2}$.

5.4.2 Example of reasoning with the KE$^\gamma$-system

We now show an example of execution of the KE$^\gamma$-system. Let us consider the ontology in Figure 17 and the query:

$$\psi_Q = \langle x_z, x_{Eva} \rangle \in X^3_r.$$

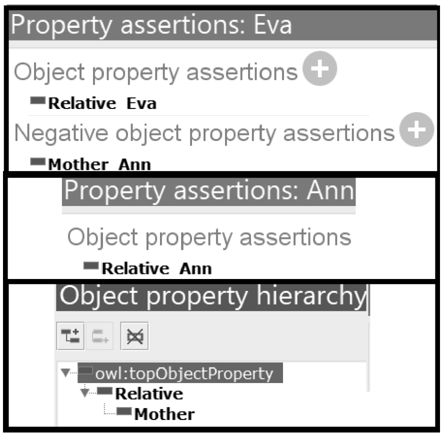

Figure 17: OWL ontology of the example.

The 4LQS$^R_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-formula representing the knowledge base in Figure 17 is the following:

$$\begin{aligned}\phi_{\mathcal{KB}} = \quad & \neg(\langle x_{Eva}, x_{Ann} \rangle \in X^3_{Mother}) \wedge \langle x_{Ann}, x_{Ann} \rangle \in X^3_{Relative} \wedge \\ & \langle x_{Eva}, x_{Eva} \rangle \in X^3_{Relative} \wedge \\ & (\forall z_1)(\forall z_2)(\neg(\langle z_1, z_2 \rangle \in X^3_{Mother}) \vee \langle z_1, z_2 \rangle \in X^3_{Relative}) \,.\end{aligned}$$

The KE$^\gamma$-tableau for $\phi_{\mathcal{KB}}$ together with the decision trees for the evaluation of ψ_Q w.r.t $\phi_{\mathcal{KB}}$ is shown in Figure 18.

As a first step, the reasoner produces the internal code of the OWL ontology as shown in Figure 19.

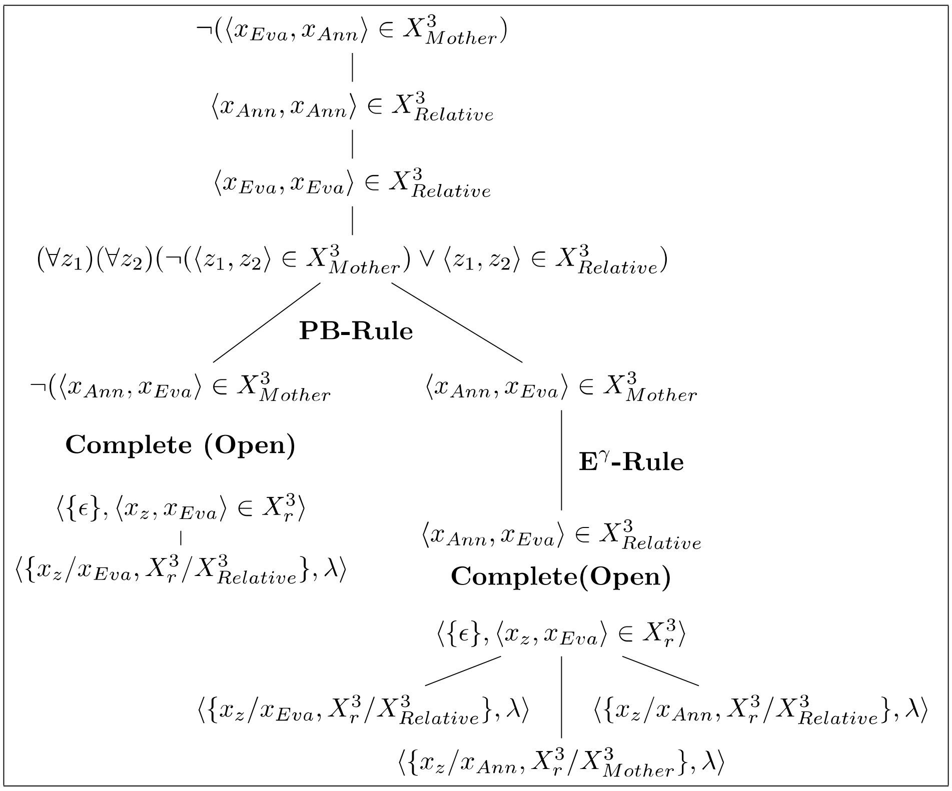

Figure 18: KE$^\gamma$-tableau for $\phi_{\mathcal{KB}}$ and decision trees for the evaluation of ψ_Q.

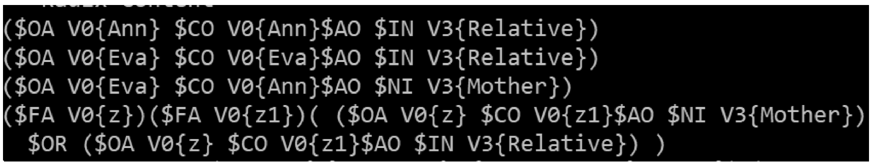

Figure 19: Internal representation of the ontology.

The vectors `VVL` and `VQL` built by the reasoner are shown in Figure 20.

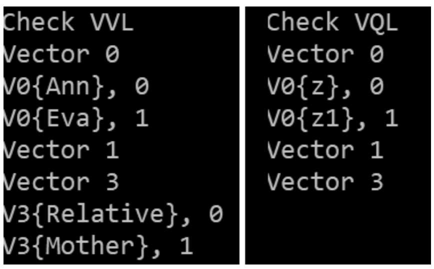

Figure 20: The vectors `VVL` and `VQL`.

The KE$^{\gamma}$-tableau computed by the reasoner is shown in Figure 21.

```
Branch: 0
($OA V0{Ann} $CO V0{Eva}$AO $NI V3{Mother})
----
($OA V0{Ann} $CO V0{Ann}$AO $IN V3{Relative})
($OA V0{Eva} $CO V0{Eva}$AO $IN V3{Relative})
($OA V0{Eva} $CO V0{Ann}$AO $NI V3{Mother})
($FA V0{z})($FA V0{z1})( ($OA V0{z} $CO V0{z1}$AO $NI V3{Mother})
  $OR ($OA V0{z} $CO V0{z1}$AO $IN V3{Relative}) )
----
Branch: 1
($OA V0{Ann} $CO V0{Eva}$AO $IN V3{Mother})
($OA V0{Ann} $CO V0{Eva}$AO $IN V3{Relative})
----
($OA V0{Ann} $CO V0{Ann}$AO $IN V3{Relative})
($OA V0{Eva} $CO V0{Eva}$AO $IN V3{Relative})
($OA V0{Eva} $CO V0{Ann}$AO $NI V3{Mother})
($FA V0{z})($FA V0{z1})( ($OA V0{z} $CO V0{z1}$AO $NI V3{Mother})
  $OR ($OA V0{z} $CO V0{z1}$AO $IN V3{Relative}) )
----
```

Figure 21: The KE$^{\gamma}$-tableau computed by the reasoner.

The execution of the query HO-conjunctive query ψ_Q is illustrated in Figure 22.

```
Printing query results ...
Tableau branch number: 0
Solution number: 0
V3{r},V3{Relative}; V0{z},V0{Eva};
Tableau branch number: 1
Solution number: 0
V3{r},V3{Mother}; V0{z},V0{Ann};
Solution number: 1
V3{r},V3{Relative}; V0{z},V0{Ann};
Solution number: 2
V3{r},V3{Relative}; V0{z},V0{Eva};
Printing Y/N results ...
Branch number: 0 Answer:1
Branch number: 1 Answer:1
```

Figure 22: Results of the execution of the query ψ_Q.

5.5 Remarks on the different versions of the reasoners

The C++ implementation of the KE^{γ}-system described in Section 5.2 is more efficient than the implementation of the KE-system reported in Section 5.1. The motivation behind such performance improvement relies on the introduction of the E^{γ}-Rule (see Figure 5 in Section 4.2.2) that acts on the $\mathsf{4LQS}^{R}_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-purely universal quantified formulae in the knowledge base by systematically instantiating them and applying the standard E-rule (elimination rule) on-the-fly. The E^{γ}-Rule takes the place of the preliminary phase of systematic expansion of the $\mathsf{4LQS}^{R}_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-purely universal quantified formulae in the KB and of the application of the E-rule implemented by the KE-system. The KE^{γ}-system turns out to be also more efficient than the implementation of the FO-KE-tableau in [36] that applies the standard γ- and E-rule.

In fact, it turns out that the KE-system and the FO-KE-system have similar performances.

We point out that $\mathsf{4LQS}^{R}_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-quantified variables and $\mathsf{4LQS}^{R}_{\mathcal{DL}_{\mathbf{D}}^{4,\times}}$-quantifier-free variables are collected in two separate vectors and stored in order of appearance in the KB. These vectors ensure that the individuals used for the expansion of the universally quantified formulae are selected in the same order for all the three systems. This fact guarantees that the number of branches of the three systems is the same, a key-aspect in the evaluation of their performances. In fact, in a KE-tableau system, the number of branches coincides with the number of distinct models that each system computes in order to saturate the KB. Since the number of distinct branches is the same and the PB-rule is the same in all the three systems, the difference of performance among them is due just to the expansion rules.

With the following example we can make a comparison on the tableau construction among the three systems. Let

$$\begin{aligned}\phi_{\mathcal{KB}} = & \neg(\langle x_{Italy}, x_{Rome}\rangle \in X^3_{locatedIn}) \wedge \\ & (\forall z_1)(\langle z_1, z_1\rangle \in X^3_{isPartOf}) \wedge \\ & (\forall z_2)(\forall z_3)(\neg(\langle z_2, z_3\rangle \in X^3_{LocatedIn}) \vee \langle z_2, z_3\rangle \in X^3_{isPartOf})\end{aligned}$$

be a $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KB, and let

$$\psi_Q = \langle x_{Rome}, x_{Italy}\rangle \in X^3_r$$

be a HO-$\mathcal{DL}_{\mathbf{D}}^{4,\times}$-conjunctive query. In Figures 23 and 24 we show the KE^{γ}-tableau and the KE-tableau, respectively, calculating the answer set of ψ_Q w.r.t. ϕ_{KB}. Since the FO KE-tableau can be represented along the same lines of the KE-tableau we refrain from reporting it.

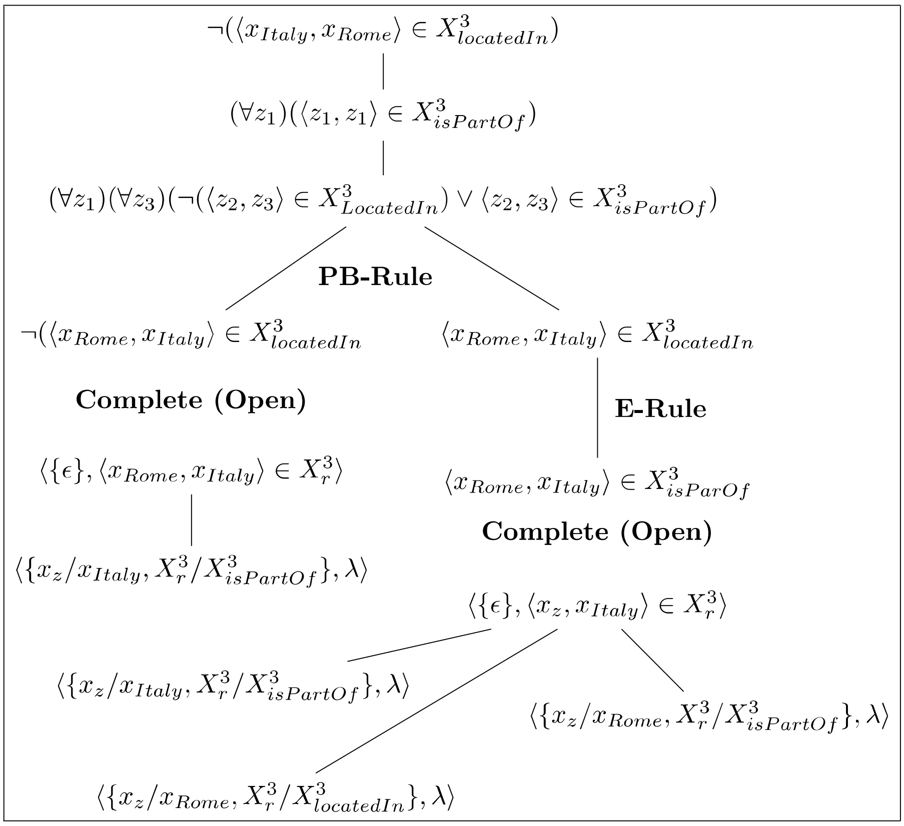

Figure 23: KE$^\gamma$-tableau for $\phi_{\mathcal{KB}}$ and decision trees for the evaluation of ψ_Q.

The metric used in the benchmarking is the number of models of the input KB computed by the reasoners and the time required to compute such models. In addition to some real-word ontologies (see Chapter 6), the examples applied are simple, ad hoc constructed $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KBs of the following form:

$$\begin{aligned}\Phi_{\mathcal{KB}} = \{&x_a \in X^1_D,\, x_b \in X^1_D,\, x_c \in X^1_D,\, x_d \in X^1_D,\\ &(\forall z)(\forall z_1)((z \in X^1_A \wedge \langle z, z_1\rangle \in X^3_P \wedge z_1 \in X^1_B \wedge \langle z, z_1\rangle \in X^3_{P_1} \rightarrow)\\ &z_1 \in X^1_C)\}\,.\end{aligned}$$

For instance, $\Phi_{\mathcal{KB}}$ generates more than 10^6 open branches. These are computed in about 2 seconds using the KE$^\gamma$-system and in about 6 seconds using the other systems. As shown in Figure 25, the KE$^\gamma$-system has a better performance than the other ones up to about 400%, even if in some cases (lowest part of the plot) the performances of the three systems are comparable. Thus the KE$^\gamma$-system is always convenient, also because the expansions of $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-purely universal quantified formulae of level 1 are not stored in memory. As remarked above, we also considered the ontologies `ArchivioMuseoFabbrica`, `Ontoluoghi` in Chapter 6 and `Ontoceramic` (see [27]), suitably deprived of some of the OWL constructs not expressible in $\mathcal{DL}_{\mathbf{D}}^{4,\times}$. For instance, since the reasoner does not implement a data type checker, we pruned the ontologies from data properties.

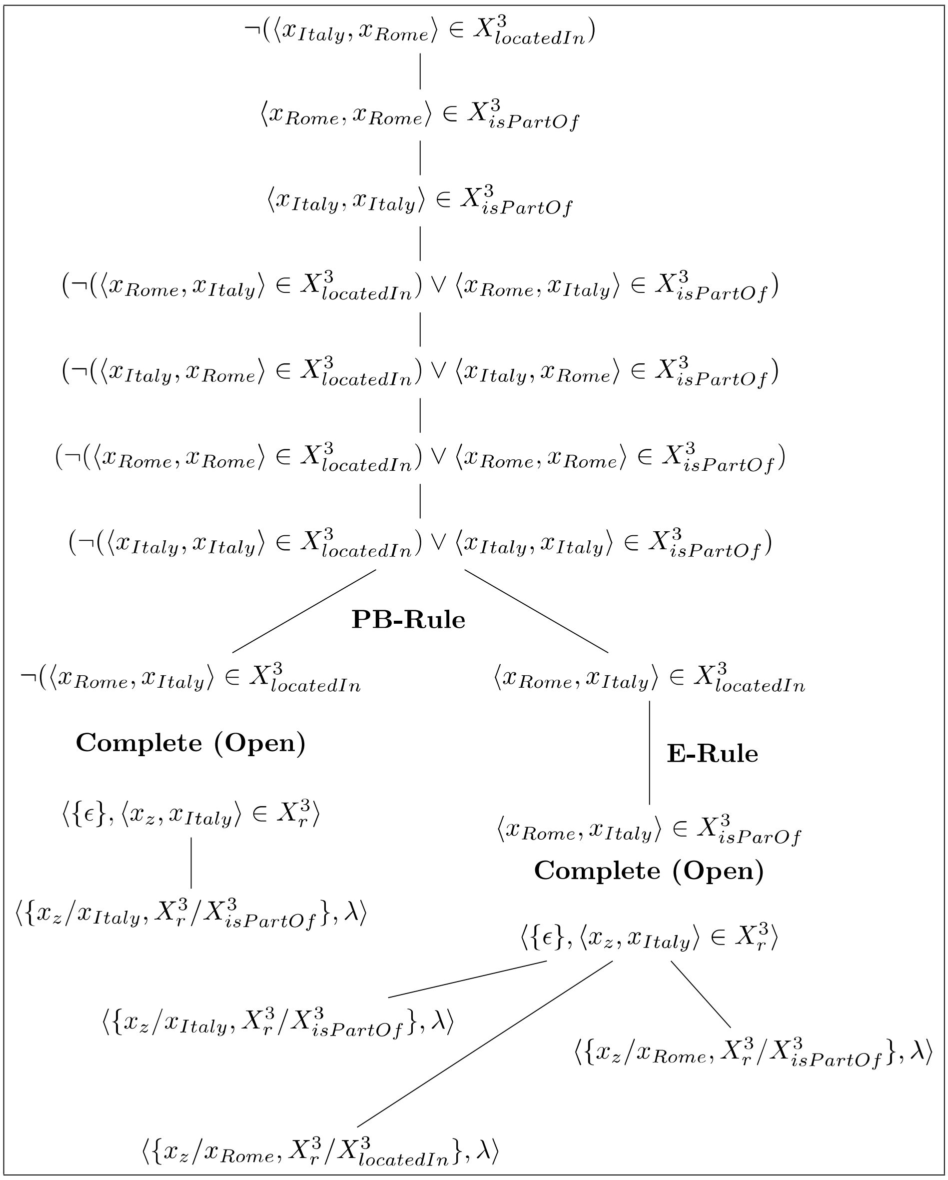

Figure 24: KE-tableau for $\phi_{\mathcal{KB}}$ and decision trees for the evaluation of ψ_Q.

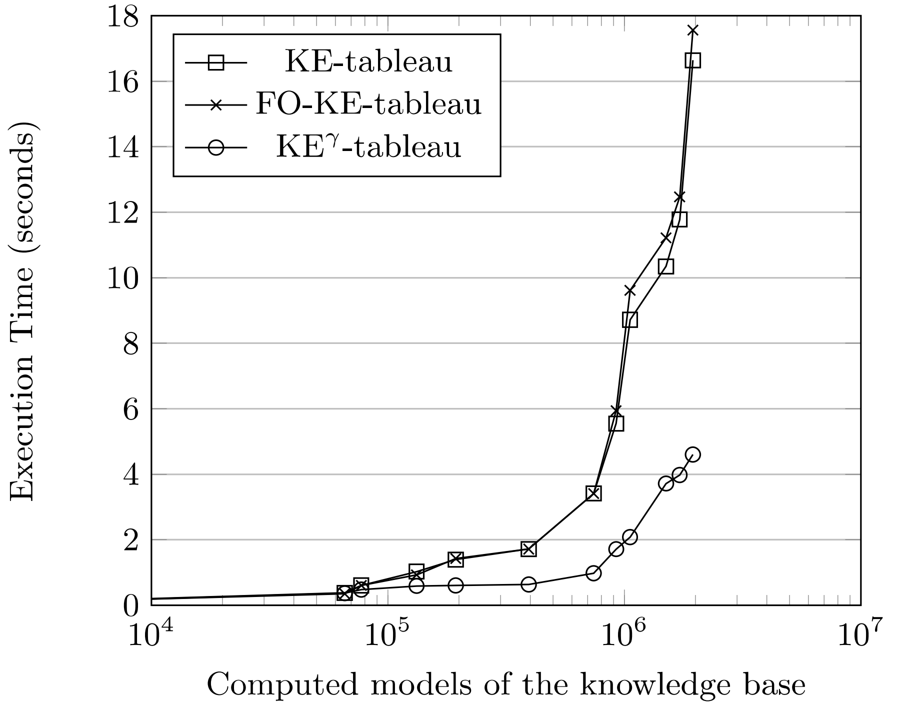

Figure 25: Comparison among the KE-system, FO KE-system, and KE$^{\gamma}$-system.

6 Semantic Web Ontologies

In this chapter, we will present three ontologies for real-world problems: one concerning the problem of knowledge extraction from non-structured text and two concerning the human and social science field. Specifically, concerning the former problem, we will present an ontology for recognizing geographical places which is used as a knowledge representation and reasoning support for an extended version of a rule-based location extraction algorithm introduced in [31]. The ontology, called `OntoLocEstimation`, can be represented in the DL $\mathcal{DL}_{\mathbf{D}}^{4,\times}$. Concerning the latter problem, we will present an ontology expressible in $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ for the renovation of the Benedictine Monastery of San Nicolò l'Arena in Catania, called `ArchivioMuseoFabbrica`, and an ontology for representing an ideal Benedictine monastery described by the Saint Gall plan, called `SaintGall`. Since the latter ontology makes a broad use of the OWL construct *SomeValues-From*, it cannot be expressed in $\mathcal{DL}_{\mathbf{D}}^{4,\times}$. The rest of this chapter is organized as follows. In Section 6.1, we will introduce the ontology `OntoLocEstimation`, in Section 6.2, the ontology `ArchivioMuseoFabbrica`. Finally, in Section 6.3, we will present the `SaintGall` ontology.

6.1 The ontology *OntoLuoghi*

6.1.1 Introduction

Recognizing location names of geographical places and of public or private buildings inside non-structured text documents is an important issue with several practical applications. For instance, in the investigative field, it is important to reveal a place named in the transcription of an interception (i.e., by means of wire-tapping) and in the social media context to reveal the places visited by users to provide targeted advertisements.

This problem falls both in the category of *information extraction* from unlabeled texts and of *named entity recognition*, where most of the existing approaches are focused on the English language and are in general not applicable to the Italian language, or they lead to unsatisfactory results due to the peculiarities of the language.

The problem of location name recognition has been addressed in various ways [109], e.g., using maximum entropy models [98], or with *Conditional Random Fields* [82], or with automatic learning techniques to infer the rules for the named entities identification inside free texts [2], or, in the last decade, also with linked data and ontologies [76] (a survey of the main geographical ontologies and datasets can be found in [8]). However, many of such approaches have been tested, or developed, vertically on top of the English language, making them hard to generalize to the Italian language. In addition, as witnessed by some preliminary experiments, widespread applications such as *Stanbol* [117] turned out to be unsatisfactory from the point of view of success ratio when applied to Italian places drawn from non-structured text written in Italian.

Only in the last decade, however, linked data and ontologies have been used to face the question. For instance, in [76] the authors present an ontology constructed on top of the *GNIS* [121] and *GEONet* [48] gazetteers, and of the *WordNet* [92] vocabulary, and provided with a statistical ranking algorithm to correctly identify geographical locations.

Here we focus on the problem of recognising names of geographical places in the Italian country, which appear in non-structured Italian text documents. The approach consists in extracting location names from Italian texts according to an extended version of the algorithm presented in [31], and then storing data and making inferences, even in presence of name ambiguities, with semantic web tools such as geographic linked datasets, *OWL* ontologies, and *SWRL* rules.

We recall that the algorithm presented in [31] relies on a set of three finite state machines, each designed to recognise several sentence patterns for the Italian language, in which location names are typically found. Such an analysis takes as input a non-structured text written in Italian and yields as output a HTML text, where candidate location names have been automatically marked by a label.

We have unified the finite state machines presented in [31] in a single automaton, reducing the overall number of states involved during the extraction of location names from the text; each location detected by the algorithm is then searched in the *OpenStreetMap* dataset [119].

Such a search yields a list of possible matches of real places, each with its degree of reliability. Then, the data retrieved by *OpenStreetMap* are inserted in a novel ontology, named *OntoLocEstimation*, to handle ambiguous geographical names. *OntoLocEstimation* uses the ontology *OntoLuoghi*, introduced in [27], that contains a detailed description of the administrative model of Italian places.

The use of open datasets, such as *OpenStreetMap*, allows a widespread and detailed coverage of Italian geographical places and provides a high precision in the detection of real places. In addition, the introduction of SWRL rules [129] allows inferences on knowledge, implicitly contained in the novel ontologies, which are more refined than other currently available ontologies on geographical places.

6.1.2 Rule-Based Location Extraction

In this section we briefly illustrate the approach we used to extract geographical and administrative places from Italian non-structured text. Unlike machine learning approaches, both unsupervised and supervised, we proposed a rule-based approach built from simple grammar rules of the Italian language complemented by a dictionary, where each rule identifies a different pattern that characterises sentences at the end of which we usually find a location name. The devised rules are supported by a specifically compiled Italian lexicon, containing the classes of words *Articles*, *Verbs*, and *Descriptors*, and a list of *Non-places* words that are known false positives; in order to improve the overall accuracy of the extraction tool, the lexicon can also be extended by other, user-detected false positives [31]. As shown in [31], these rules provide a large coverage of the Italian grammar for what concerns statements about places.

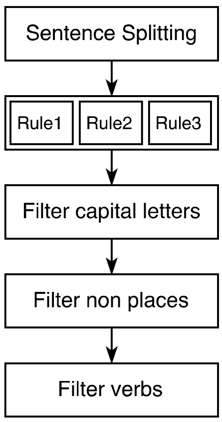

Figure 26: Pipe and filter workflow for location extraction.

Figure 26 depicts the entire workflow used for the location extraction: in the first *sentence splitting* step, an input text T is separated into a list of sentences using occurrences of punctuation marks, i.e., *full-stop*, *ellipsis*, *exclamation-mark*, *question-mark*. Any other non-letter symbol is ignored, e.g., dollar sign, percent sign, etc. Each sentence is further segmented into words using the space character as a separator.

The tokens (words) are then fed to a finite state machine implementing three different *rules*, i.e., grammar cases possibly implying the use of a place name at the end of a sentence: if the accepting state *6* of the automaton in Figure 27 is reached, the current token is marked as a location candidate. The result is a list of candidate words that must pass through a configurable sequence of *filters* before being actually labelled as a place name.

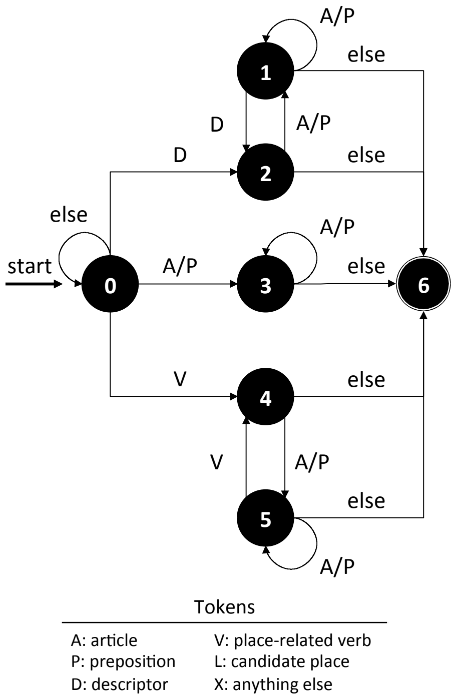

Figure 27: Unified automaton of the three Italian grammar rules.

Figure 27 shows the unified automaton used to implement the three grammar rules defined in [31]; those rules were devised to accommodate the mentioning of a place name in sentences such as the rule names suggest, namely, *Da Roma* (from Rome), *Vicino a Roma* (Near Rome) and *Andando a Roma* (Going to Rome). To give a simple detection example, Figure 28 shows an excerpt of the unified automaton giving a finite state machine expressing only the first rule, (*Da Roma*), whose name recalls the grammar pattern responsible for the matching. The automaton scans the tokens of a given sentence and remains in state *0* until a preposition (P) or an article (A) is found, at which point the automaton changes its internal state to *1*. Subsequently, the automaton remains in state *1* until a different kind of word is encountered, in which case the final state is reached and a new candidate word is found. However, several candidates will be actually dropped afterwards by the filters, e.g., known false positives or conjugated verbs.

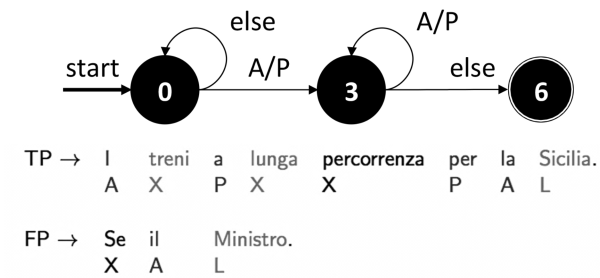

Figure 28: Examples of the *Da Roma* rule.

All the candidates found by the automaton are then given as input to a sequence of filters in order to remove trivial false positives that may have been selected:

- Filter0: the candidate for a location name must begin with a capital letter; even if it may considerably improve the detection accuracy in several cases, this is not a mandatory filter. In fact it is applied only when we can assume that location names are written with a leading capital letter (e.g., if we are analyzing newspaper articles);
- Filter1: remove all known false positives using the devised lexicon of non-places;
- Filter2: remove conjugated verbs.

Any word surviving the above filters is labelled as a place name, and is given to the ontological support to store the results and automatically retrieve the information concerning the algorithm and dataset used, and the spotted place.

6.1.3 An ontology for reasoning with places

Next, we first describe the ontology `OntoLocEstimation`, developed with the purpose of reasoning with geographical and administrative places. Subsequently, we show how the ontology is populated by means of a Java framework. The ontology `OntoLocEstimation` is associated to the algorithm introduced in Section 6.1.2 and it allows us to manage the identification of a specific location even in presence of uncertainty.

We illustrate how the ontology `OntoLocEstimation` is structured. `OntoLocEstimation` extends the ontology `OntoLuoghi` [27] with OWL constructs allowing us to deal with the administration of Italian places and with the algorithm described in Section 6.1.2. In its turn, `OntoLuoghi` reuses some concepts and properties of `LinkedGeoData` [114], an ontology developed by the *AKSW* research group (Agile Knowledge Engineering and Semantic Web) with the aim of adding a spatial dimension to the Semantic Web.

Locations are modeled by means of a taxonomy of OWL classes. Association among locations is performed through a taxonomy of object properties. The path allowed, namely the hierarchy of such classes, is shown in Figure 30. Double-hoop entities in Figure 30 are considered as optional. Names of the OWL classes and object properties involved are shown in Figure 29.[6] Reasoning capabilities concerning locations are strengthened using the SWRL rules shown in Figure 31.

Several entities defined in `LinkedGeodata` are reused in `OntoLuoghi` by defining an equivalent entity which extends its definition. For instance, the entity `Localisation` is equivalent to `LinkedGeoData:Place` and inserted in the taxonomy of Italian places (see [27] for details). The object property `hasLocalisation` is defined as an OWL transitive property. Thus, if the pairs of objects (x, y) and (y, z) are in the property `hasLocalisation`, then the pair (x, z) is included in the property `hasLocalisation` too. The property `hasLocalisation` can be used together with its subproperties to infer the administrative hierarchy of a location, providing only the top level of a place (i.e., the superproperty). For instance, if we write the statements (Sicily hasState Italy), (Catania hasRegion Italy), (Acireale hasProvince Catania), then the statements (Sicily hasLocalisation Italy), (Catania hasLocalisation Italy), (Acireale hasLocalisation Catania), (Acireale hasLocalisation Sicily), (Acireale hasLocalisation Italy) are inferred thanks to the subproperty relationships and to the transitivity of `hasLocalisation`. In addition, the set of SWRL rules depicted in Figure 31 allows one to infer the statements (Catania hasState Italy), (Acireale hasRegion Sicilia), (Acireale hasState Italy). Moreover, subproperties are defined as functional properties so as to guarantee that places cannot be associated to different locations. For instance, Acireale cannot be associated to the province of Catania and Palermo at the same time.

Usually, an algorithm recognising a geographical location from a keyword in a non-structured text yields several candidates, each in combination with a *degree of belief*. The degree of belief indicates how much the association between the keyword and the candidate geographical place is considered reliable by the approach in use. The ontology should model this situation and also manage additional information such as the algorithm applied for extracting location names from the text and the dataset adopted. Such information is relevant if one wishes to compare the accuracy that different algorithms and datasets

[6] Images in Figures 29, 32, 33, 34, and 35 are drawn from the interface of the editor Protégé [120]

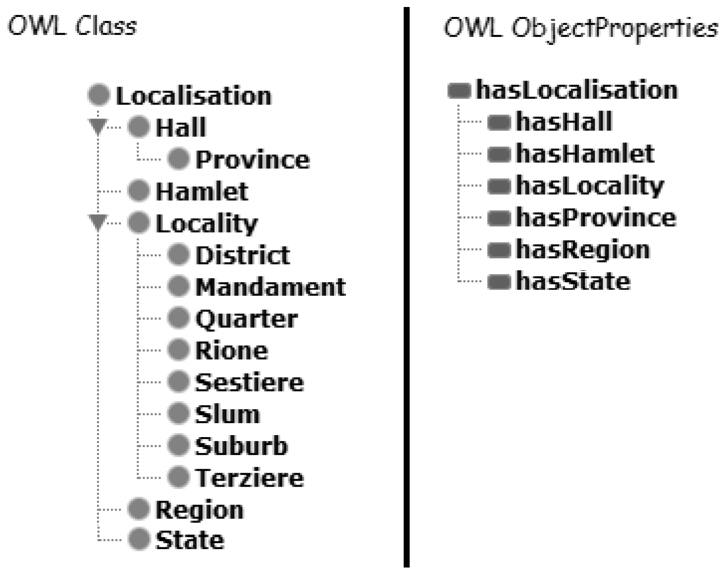

Figure 29: Classes and properties for locations.

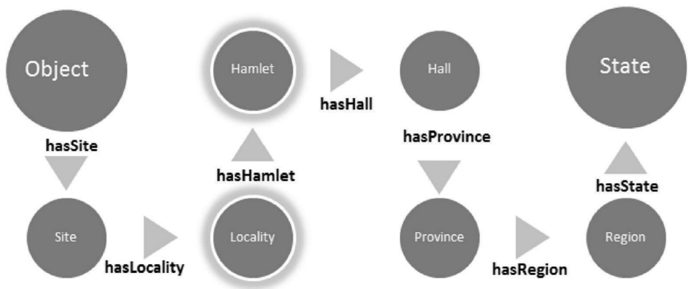

Figure 30: Allowed path for locations.

provide for the task of recognising location names.

Our approach works as follows. Every time the algorithm finds a keyword, an instance of the class `TextKey` is added to the dataset. The class `LocEstimation` models the fact that the algorithm of Section 6.1.2 is executed on the keyword and some results are provided. The instances of `Textkey` and `LocEstimation` are related to each other by the object property `hasLocEstimation`. For every estimation there may be zero, one, or more matches. Each match of the algorithm is modeled by the `GuessedLocation` class. Instances of the class `GuessedLocation` are associated to the ones of the class `LocEstimation` by the object property `hasGuessedLocation`. Each instance of the class `GuessedLocation` provides information concerning the candidate geographical place and the relative degree of belief. The degree of belief is introduced by means of the data property `hasGuessedValue` having as

OntoCeramic Rules
Locality(y), hasLocalisation(?x,?y) → hasLocality(?x,?y)
Region(y), hasLocalisation(?x,?y) → hasRegion(?x,?y)
State(y), hasLocalisation(?x,?y) → hasState(?x,?y)
Hamlet(y), hasLocalisation(?x,?y) → hasHamlet(?x,?y)
Hall(y), hasLocalisation(?x,?y) → hasHall(?x,?y)
Province(y), hasLocalisation(?x,?y) → hasProvince(?x,?y)

Figure 31: SWRL rules for reasoning with locations.

range the data type double. The geographical place is specified with the object property `hasReferredLoction` having as range every instance of the class `Localisation`. Figure 32 illustrates classes and properties introduced in order to model the recognition of places; an example is shown in Figure 33.

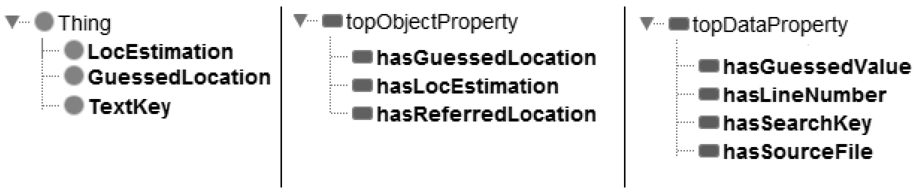

Figure 32: Classes and properties for locations recognition.

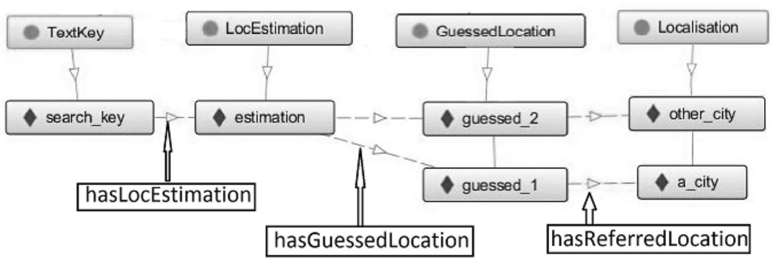

Figure 33: Example of location estimation.

The ontology provided takes also into account information concerning the algorithm for the extraction of the location and the dataset used. Classes and object properties defined for this purpose are shown in Figure 34.

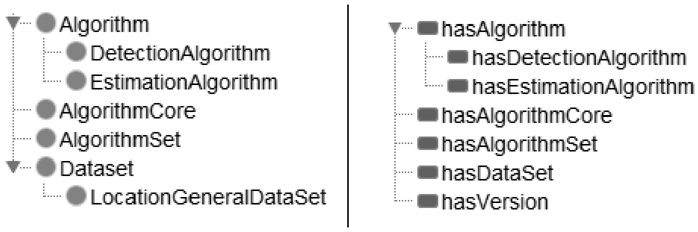

Figure 34: Classes and object property for algorithms.

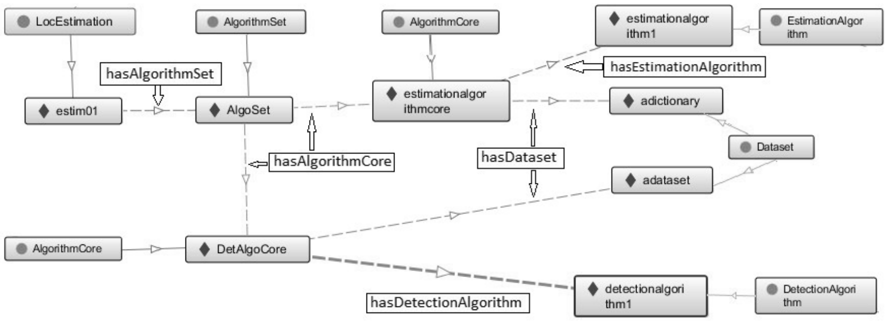

Figure 35: Example of describing algorithms.

A class `Algorithm` provides information about the algorithm applied. Currently, two types of algorithms have been identified, but others can be added. The class `DetectionAlgorithm` includes all the algorithms used to establish in a non-structured text whether a word represents a geographical place, whereas the class `EstimationAlgorithm` includes information about the algorithm used to choose a set of possible geographical places and to assign them a degree of belief. In addition, it is possible to specify the dataset used by the algorithm by means of the class `Dataset`. The subclasses of the class `Dataset` describe the type of the dataset. For instance, the class `LocationGeneralDataset` is used to represent the datasets containing general geographical information.

In order to keep track of the algorithms used, we provide the class `AlgorithmSet`, which is associated to an instance of the class `LocEstimation` by means of the object property `hasAlgorithmSet`. To each instance of the class `AlgorithmSet`, one or more instances of the class `AlgorithmCore` are associated by means of the object property `hasAlgorithmCore`. The latter class relates an algorithm to the datasets used. In order to indicate the datasets used, the `hasDataSet` object property is provided, having as range the `DataSet` class. Analogously, to associate an algorithm to an instance of the class `DataCore`, the object property `hasAlgorithm` is provided, having as range the class `Algorithm`. In particular, two subproperties of the class `hasAlgorithm` are provided: the property `hasDetectionAlgorithm`, that associates an instance of the class `AlgorithmCore` to an instance of the class `DetectionAlgorithm`, and the property `hasEstimationAlgorithmCore` for the instances of the class `EstimationAlgorithm`. How these classes and relations are used is shown in the example of Figure 35. Note that, by means of the class `AlgorithmCore`, additional information about why, how, and when the algorithm set is used can be tracked.

We describe shortly how the ontology `OntoLocEstimation` is populated by means of a Java framework built ad hoc. The first step of the process is to retrieve an open dataset of locations, that is as rich as possible, and to map such knowledge inside the ontology. We take into account the `OpenStreetMap` dataset and implement a Java parser in order to populate the ontology with the `OpenStreetMap` entries. The parser exploits the OWL API library [67] together with Jena Ontology API [118] that we adopted to perform SPARQL [130] queries. We used the reasoner Pellet [112] to carry out inferences on the ontology. As far as we know, Pellet provides the best deal between efficiency and data type reasoning capabilities.

In a preliminary phase of this work, we also considered the `LinkedGeoData` dataset that provides a semi-automated conversion of a subset of the `OpenStreetMap` dataset in RDF format. However conversion of RDF in OWL is not straightforward. Thus, being mainly interested in OWL reasoning, we opted to provide the mapping of `OpenStreetMap` data in OWL statements by means of a simple parser.

6.2 The ontology *ArchivioMuseoFabbrica*

In this section we present an OWL ontology concerning the history of the renovation of the Benedictine Monastery of San Nicolò l'Arena in Catania by the architect Giancarlo De Carlo. Specifically, we consider a wide subset of public and private documents collected from 1977 to 2006, during the process of restoration and adaptation of the monastery to a campus for the University of Catania. The task of modeling and populating the ontology have been carried out from the analysis of documents stored in the *Archivio del Museo della Fabbrica*, in the new archive of professor Giuseppe Giarrizzo, in the private collection of Antonino Leonardi, and from the conceptual map of the locations of the monastery.

6.2.1 Introduction

The renovation of the Benedictine Monastery of San Nicolò L'Arena in Catania and its adaptation as university campus performed by the architect Giancarlo De Carlo is an important project of cultural interest in Sicily [15]. The monastery is in fact one of the largest in Europe and has been declared national monument in 1869. The history of its foundation is as much intricate as the history of its recovery and readaptation as a university building. This is why it is worth sketching its outlines, even if on a general scale. Founded in 1558 on the flat surface of Cipriana di Collina di Montevergine, around a cloister with a square plan, the Monastery of San Nicolò L'Arena was surrounded by the lava flow coming from Etna on the North and West walls in 1669, and then destroyed by the earthquake in 1693. In 1702, the *fabrica nova* building yard was opened again and went on for the whole 18th century.[7] In 1866, the whole building was confiscated and declared national monument with Royal Decree dated 15th August 1869. Because of *leggi eversive*,[8] the Monastery became Royal state property. As a result, a substantial part of it was given away to the *Municipality of Catania* for common benefit purposes, namely defense and education. During that period, the building was modified to receive schools, public gyms, and military barracks: each of these institutions modified the locations according to their needs. From the 1950s on, the Monastery was progressively emptied from civil uses to be almost completely released in the second half of the 1970s. At the beginning of the 1970s, when the task of regenerating the old town became urgent and mandatory (see, for instance, Commissione Franceschini, Declaration XL, 1967), other possible uses for the Monastery were considered. Meanwhile, the *Facoltà di Lettere e Filosofia* was looking for a new building more suitable to its needs, because of the increased number of enrolled students. Thus, in 1977 the University of Catania became the owner of almost the whole Monastery of Benedictines thanks to a donation in its favour made by the Municipality of Catania.[9] By the end of the 1970s, the arrangement of the first rooms was started by *Ufficio Tecnico Universitario* in order to set out the establishment of the first university institutions. Such an initial restoration work brought to light

[7]Legge 15 agosto 1867, n. 3848. art. 20 "I fabbricati dei conventi soppressi da questa e dalle precedenti leggi, quando siano sgombri dai religiosi, saranno conceduti ai comuni ed alle provincie, [...], e sia giustificato il bisogno e l'uso di scuole, di asili infantili, di ricoveri di mendicità, di ospedali, o di altre opere di beneficenza, e di pubblica utilità nel rapporto dei comuni e delle provincie [...]"

[8]Legge 15 agosto 1867, n. 3848. art. 20 "I fabbricati dei conventi soppressi da questa e dalle precedenti leggi, quando siano sgombri dai religiosi, saranno conceduti ai comuni ed alle provincie, [...], e sia giustificato il bisogno e l'uso di scuole, di asili infantili, di ricoveri di mendicità, di ospedali, o di altre opere di beneficenza, e di pubblica utilità nel rapporto dei comuni e delle provincie [...]"

[9]However, the taking of possession of the entire building will be postponed until 2001 for several reasons.

a rich palimpsest of information able to tell the history of the Monastery and of the order which commissioned its construction centuries ago, besides that one of the ancient city. The drawing up of *Progetto Guida*[10] (De Carlo, 1988) at the heart of the recovery of the Monastery was a long process because of its shape, its new usage, and the deep intertwining of historical stratifications. During his engagement with the University of Catania, Giancarlo De Carlo for the first time intensively tested the *progettazione tentativa.*[11]Thanks to this particular operational method, the *Progetto Guida* was recognized by *Regione Siciliana* to be of "important artistic interest" [89].[12] Some of the solutions adopted were often slightly and, sometimes, substantially different if compared to the reading of *Progetto Guida*. The reasons that pushed Giancarlo De Carlo and the *Ufficio Tecnico Universitario* to modify the projects in progress are traceable in the numerous archaeological findings, in the permissions by Superintendence, and in the decisions made by the committing university. As a result, the reconstruction of the history of the Benedictine Monastery restoration and of the political, economic, and social implications on the Catania area turns out to be complicated. According to the Distant Reading approach,[13] We use semantic web tools such as OWL in order to reconstruct such history and to provide the researchers with a powerful and efficient tool that can be used to study and integrate the data. Specifically, we define an OWL ontology representing the locations and architectonic elements of the Monastery involved in the renovation activities, and a conceptual map of related documents written during the restoration period, stored in private and public archives. A related work is the project Mapping the *Republic of Letters* [49] using semantic web technologies to describe the Grand Tour Travelers and their correspondence from the XVIII to the XX Century. The latter approach, however, differs from that used in this dissertation because it covers a longer period of time and because it is mainly based on well established databases.

---

[10] In 1983, Giancarlo De Carlo was in charge of the drawing up of a project, the *Progetto Guida* indeed, for the recovery of the whole monastery. After the approval of such project, the architect was asked to work as consultant and, sometimes, as supervisor.

[11] <<Tentativa: meaning the attempt to reach the solution by proceeding through tests and checks, but also to tempt the situation one deals with, in order to bring to light its imbalances and to understand how and to what extent it can be changed without being distorted and to reach new balances >>(DE CARLO, 1996).

[12] DECREE 23rd May 2008, Gazzetta Ufficiale della Regione Siciliana, 20th June 2008, Year 62 N. 28: <<[...] la ristrutturazione [...] effettuata su progetto redatto nel 1986 dall'architetto Giancarlo De Carlo determina di fatto il riuso del monastero attraverso una serie di interventi che vanno dalla manutenzione ordinaria e straordinaria di alcune delle parti esistenti, all'inserimento di elementi nuovi nel contesto storico, [...] le opere di ristrutturazione del complesso sopraddetto costituiscono pregevole esempio di opera di architettura contemporanea e rivestono importante interesse artistico>>.

[13] Introduced by Franco Moretti in 2005, the Distant Reading approach consists in reading literary texts by means of geographical and conceptual maps, graphs, and trees. Such approach differs from the traditional close reading because it focuses on relationships among entities by means of processes of abstraction and reduction.

6.2.2 Background tools: Letters, documents, and data

The documents used to construct the ontological dataset, produced from 1977 to 2006, are stored in *Archivio del Museo della Fabbrica*,[14] in the new archive of the Department Head Giuseppe Giarrizzo, and in the personal collection of documents of the surveyor Antonino Leonardi.[15] The juxtaposition of collections of documents of different nature is motivated by the fact that the Monastery of the Benedictines is "an exclusive architectonic event for Sicily" [38] for both its past and contemporary age. Its restoration, indeed, has been a monumental action lasted more than 25 years (1977-2005), facing different vacillating moments both of speed and optimism and complete arrest and pessimism. The reconstruction of the relationships among the documents is grounded on the building yard diaries,[16] official documents drafted by the architect to inform the rector of the University of Catania about the progress of works, and addressed in copy to the Department Head Giuseppe Giarrizzo and the surveyor Antonino Leonardi. Each report was accompanied by an introductory letter sent to the three persons mentioned above. However, only letters addressed to Leonardi[17] and to professor Giarrizzo were classified. Each report, when possible, is associated to one or two letters. Letters are meaningful sources of information because they provide indications and ideas by De Carlo which are not contained in the reports. The building yard diaries, the letters addressed to professor Giarrizzo, and the letters addressed to Leonardi are stored in the *Archivio del Museo della Fabbrica*, in the new archive of the department head, and in the personal archive of the surveyor Leonardi, respectively. In particular, data drawn from such documents are classified according to the categories belonging to the *Archivio Museo della Fabbrica*. These are listed in what follows:

- Year: date of contract stipulation or commitment act;

[14]By the end of the 90s, the idea that documents collected for administrative reasons could establish a resource for the creation of a contemporary archive took shape. This archive would be linked to the activities of the new *Museo dell'Edificio dei Benedettini* created thanks to the "Progetto Coordinato Catania-Lecce". Its main goal was the adjustment of university buildings, having substantial lacks in terms of cataloguing and of inventory of library collections and census of scientific, archaeological, and artistic collections. The initiative 6 of the project was referred to the institution of Museo dell'Edificio dei Benedettini.

[15]The big intervention of requalification of the Monastery of the Benedictines is to be attributed to the department head Giuseppe Giarrizzo (1927-2015). He was the Department Head until 1998, personally following all the stages of the building yard. The surveyor Antonino Leonardi (1937-2016) was the Responsabile dell'Ufficio Tecnico Universitario - Benedictine section from 1980 to 2006.

[16]Giancarlo De Carlo's diaries begin in 1989 when the *Progetto Guida* became executive. It was intention of the Department Head Giuseppe Giarrizzo and his collaborators to publish these diaries, which unfortunately never saw the light.

[17]Giancarlo De Carlo's letters mentioned here refer to the correspondence the architect had with the surveyor Leonardi from 1983 on issues concerning the recovery of the monastery. One hundred five of such letters have been selected from the whole corpus and commented by Leonardi through interviews for the publication of "La rabbia e la gentilezza. 105 lettere di Giancarlo De Carlo sul recupero del Monastero dei Benedettini" [84].

- Category: typology of works to be distinguished between building works and/or alike, restoration works; structure (supplies and/or works), supply and/or decor execution, various (optional performances);
- Place: internal/external location of the Monastery where the work has been carried out.
- Object: activities and/or performances afferent to operational and handcrafted qualifications within each type of category;
- Work Supervision: reporting the indication of the individual responsible for the project and/or works direction of the single intervention.

In addition to letters and reports, we also considered a dataset from the press review stored in the *Archivio del Museo della Fabbrica*, and concerning the renovation of the Monastery. In fact, this material could help the reader to better understand the media and political mood that characterized such interventions. In the ontology, gathered data have been linked to notions of space and time. As far as space is concerned, the Benedictine Monastery is conceptually divided into five levels, as illustrated later.

6.2.3 Description of the ontology

The ontology uses the following external ontologies: `CIDOC-CRM` (Comitato Internazionale per La Documentazione - Conceptual Relational Model) [41], that provides generic definitions for describing concepts and relationships typically used in cultural heritage documentation, *LinkedGeoData*, the ontology for geographical places, *DBpedia* [83] and *Wikidata* [127], vocabularies for general knowledge. *CIDOC-CRM* has been developed in order to create a general data model for museums, with a particular focus on information interchange (becoming an ISO standard only recently). The primary role of the *CIDOC-CRM* is to provide a basis for mediation of cultural heritage information and hence the semantic glue needed to transform today's disparate, localised information sources into a coherent and valuable global resource. *Wikidata* is a free and open knowledge base that can be read and edited by both humans and machines and acts as central storage for the structured data of its *Wikimedia* sister projects including *Wikipedia*, *Wikivoyage*, *Wikisource*, and others. *DBpedia* is a crowd-sourced community effort to extract structured information from *Wikipedia* and make this information available on the Web. As illustrated in the ontology import closure described in Figure 36, the main ontology `ArchivioMuseoFabbrica` is the access point to the knowledge base and aggregator of the ontologies presented in this section. In fact, as depicted in Figure 36, `ArchivioMuseoFabbrica` links together the ontology `MonasteroBenedettini` designed to represent the spaces inside the Benedictine Monastery, importing in its turn `EdificiStorici`, a more general ontology describing historical buildings, and `DocumentoAF` that describes documents in the *Archivio del Museo della Fabbrica*, in the new archive of professor Giarrizzo, and in the private collection of Leonardi. The latter ontology imports in its turn the ontologies `Professioni`, representing people and jobs involved in the renovation of the monastery, and `OntoLuoghi`, cataloguing Italian geographical places. Before introducing the entities contained in `ArchivioMuseoFabbrica`, we first present its imported ontologies.

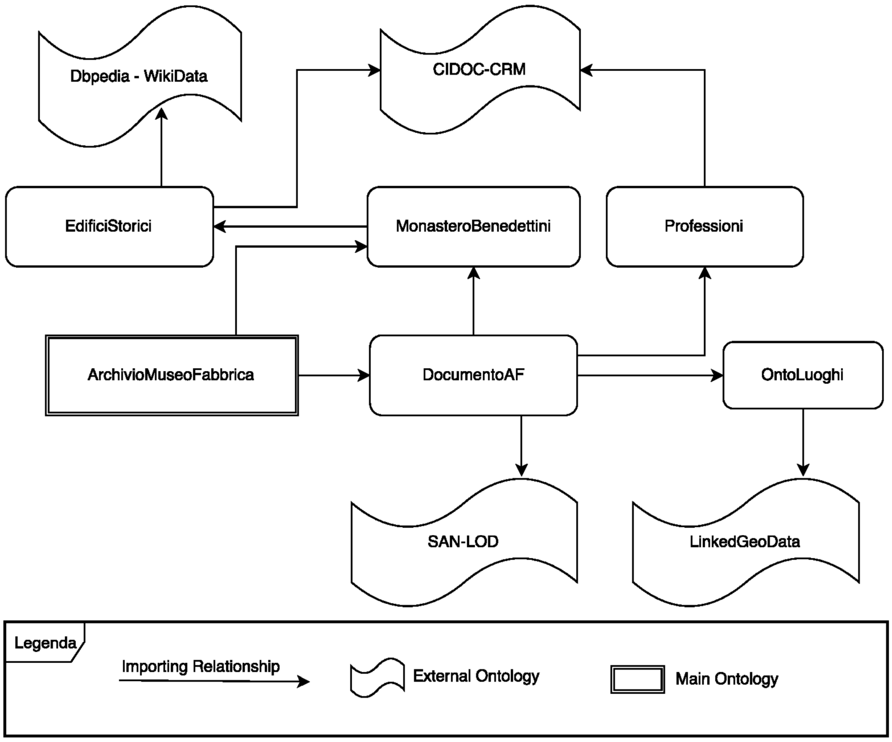

Figure 36: Import closure of the ontology.

The ontology `MonasteroBenedettini` depicts the monastery by suitably populating classes belonging to the more general ontology `EdificiStorici` with individuals representing areas and elements of the monastery, also introducing relations among them, as shown in Figures 37.

The ontology `EdificiStorici` describes the structure of a Benedictine monastery. It contains two main classes: the class `Area` which represents a generic area inside the monastery, is a subclass of the `CIDOC-CRM` class `E53_Place`, and the class `Elementi`, which represents decorative, natural, or architectonic structures.

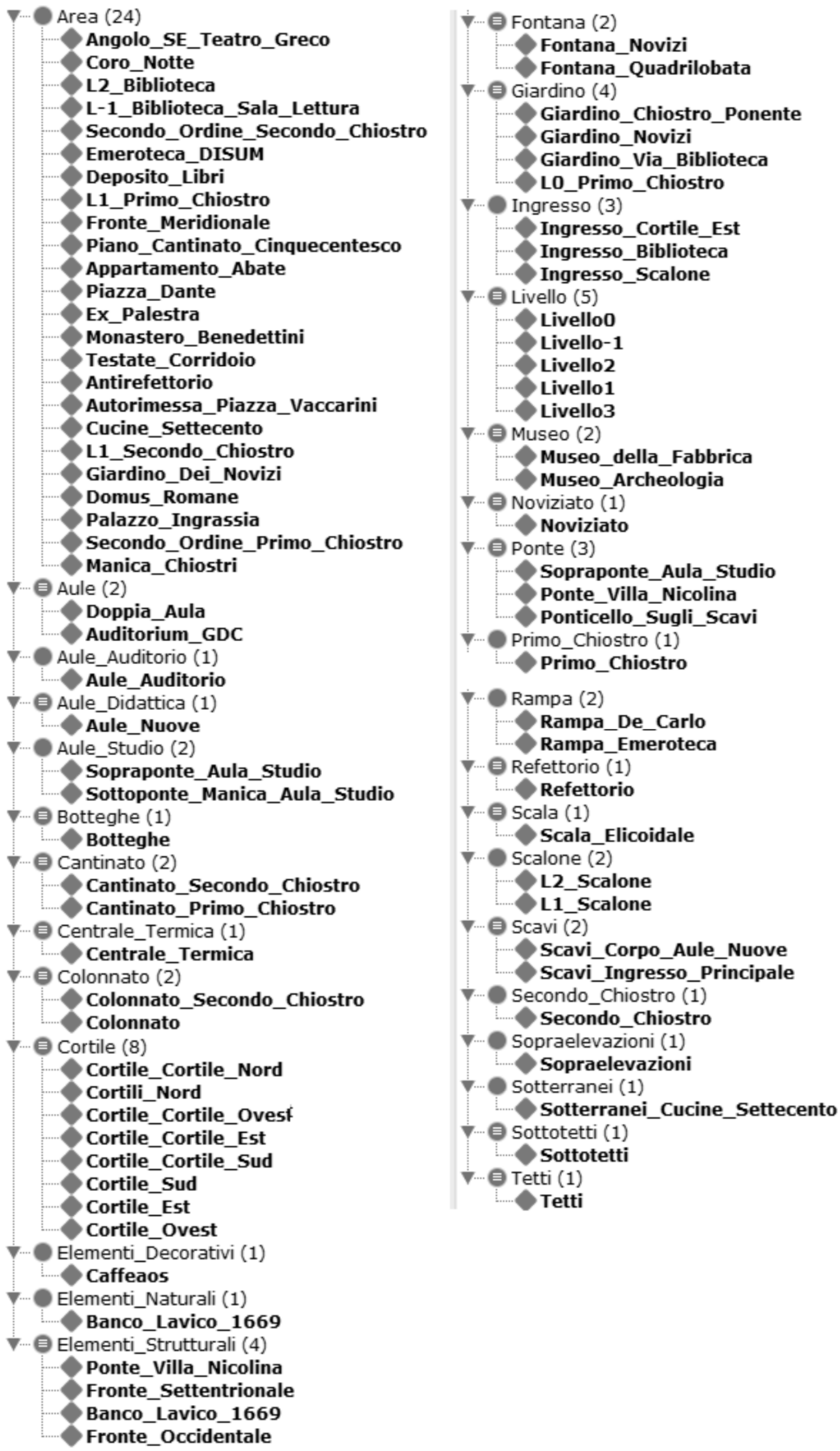

Figure 37: Description of the Benedictine Monastery.

Furthermore, the ontology contains, as a subclass of `Area`, the class `Livello`, representing layers of the monastery. The ontology identifies several layers, partitioning the whole space of the building, each corresponding to an instance of `Livello`. Each instance uses the data property `numeroLivello` having as range the integer corresponding to the level. Each area of the monastery converges to a layer representing its distance from the ground layer, namely layer 0. In the case of the Catania's Benedictine Monastery, the ontology identifies five layers, from -1 to 3. Elements described by `EdificiStorici` are shown in Figure 38. Figure 39 and Figure 40 show the levels of the Catania's Benedictine Monastery and their usage, respectively.

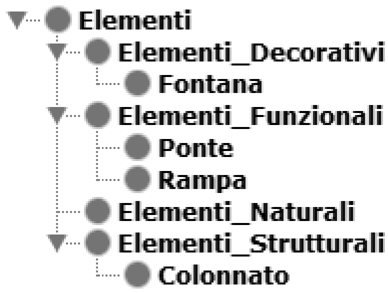

Figure 38: Elements of the Benedictine monastery.

Figure 39: Layers of the Benedictine Monastery.

Figure 40: Example of the usage of levels of the Benedictine Monastery.

Rooms and locations of the monastery are organized in a taxonomy as shown in Figure 41. In addition to the hierarchy of the areas of the monastery, we are also interested in describing relations among areas. Such a conceptualization is defined by means of the object property `contiene`, representing the general containment relationship among areas and of its inverse relation, `faParteDi`, subproperty of the `CIDOC-CRM` entity `89.falls_within`.

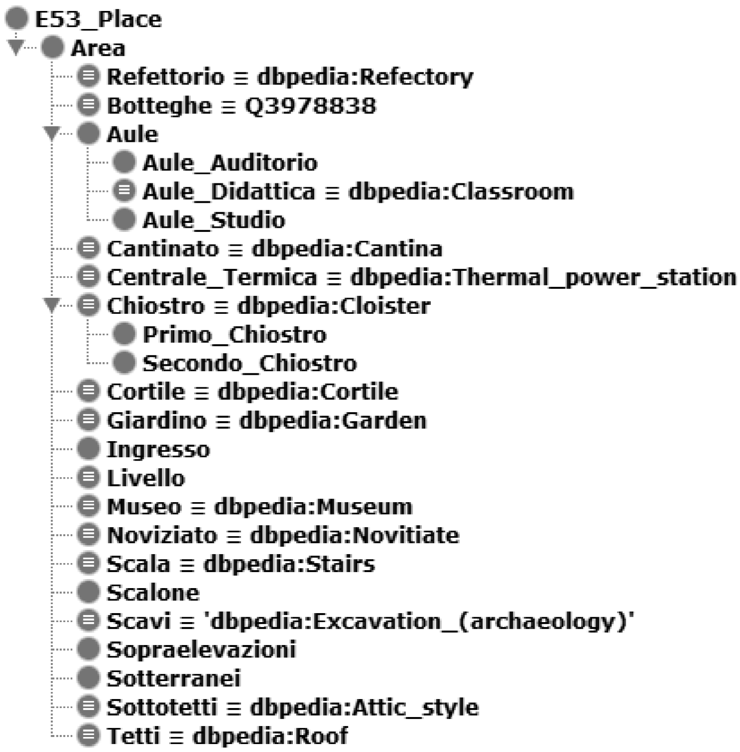

Figure 41: Rooms and locations of the Benedictine Monastery.

The object properties `contiene` and `faParteDi` are both defined as transitive properties. The ontology also contains the object property `contieneDirettamente`, a subproperty of `contiene`, representing the fact that an area directly contains an element or another area. Indeed, the superproperty `contiene` infers all the indirect inclusions among areas. The inverse of `contieneDirettamente` is the object property `faParteDirettamenteDi`. The data properties `haNomeCompleto` and `haNomeInUso` associate a `xsd:string` representing, respectively, official names and commonly used names to members of the class `Area` and of the class `Elementi`. The ontology `DocumentoAF` identifies one main class `Documenti`, a subclass of the `CIDOC-CRM` class `E31_Document`, which contains three subclasses `DocumentoPersonale`, `DocumentoIstituzionale`, and `DocumentoMuseoFabbrica`, each describing a document type.

Specifically, the class DocumentoPersonale describes generic personal letters, the class DocumentoIstituzionale represents documents sent to public institutions, such as the University of Catania, and the class DocumentoMuseoFabbrica describes the archives of the Museo della Fabbrica. The latter includes two further types of documents: the letters represented by the class Lettere and the building yard reports identified by the class VerbaleDiCantiere. Each member of the class Document is associated to its author by the object property haAutore and to its recipient by the object property haDestinatario. Each member of the class DocumentoMuseoFabbrica is related to the area of the monastery cited in the document by the object property haRiferimentoCantiere. The range of the first two object properties is constituted by the union of the classes Persona and Azienda, belonging to the ontology Professioni (see Figure 36), whereas the range of the latter object property is the class Area of the ontology EdificiStorici. Members of the class Lettere are related to the provenance place of the letter, represented by instances of the class Place of the ontology Ontoluoghi (see Figure 36), by means of the object property haLuogoAutore. In addition, they are associated to the building yard report cited by the letter, represented by members of the class VerbaleDiCantiere, by means of the object property haRiferimentoDocumentoAllegato.

The instances of DocumentoMuseoFabbrica can specify the date and number of the document by means of the data properties haDataDocumento and haNumeroDocumento, respectively. An instance of class Lettere can specify the relation number by means of the data property haNumeroRelazione, which is subproperty of haNumeroDocumento having as range the data type xsd:int. By means of the data property haURLDocumentoFisico, it is possible to refer to the URL of the original document published on the web providing as domain an instance of the class Documento_Digitale. An instance of such a class is related to an instance of the class Document by means of the object property haDocumentoDigitale. Moreover, dates of inspection cited by a document can be specified by means of the data property haDataSopralluogo using as range the data type xsd:date. One can also describe newspaper articles somehow related to events concerning the monastery by means of the class ArticoloGiornale. The relative newspaper can be specified by means of the object property haTestataGiornalistica, having as range an instance of the class TestataGiornalistica. The topic of the newspaper is specified by means of the object property haRubricaGiornalistica, having as range an instance of the class RubricaGiornalistica. Moreover, for a newspaper article, one can specify authors, publication date, page number, title, and number of images by means of the object properties haAutore, having as range an instance of the classes `Persona`, `dataPubblicazione`, `haPaginaRiferimento`, `haNumeroImmagini`, respectively. Classes and object properties of the ontology `DocumentiAF` are shown in Figure 42, data properties in Figure 43.

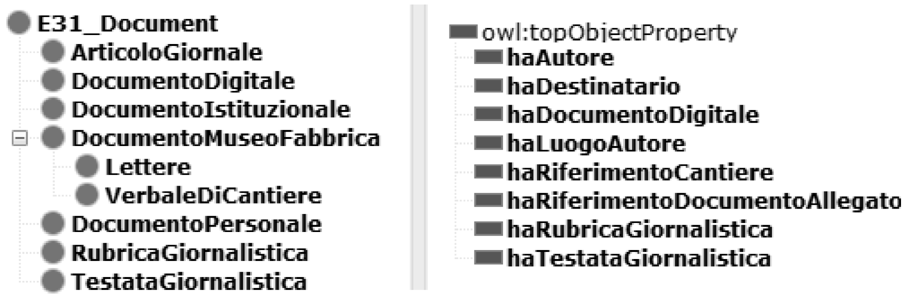

Figure 42: Classes and object properties of the ontology `DocumentiAF`.

Figure 43: Data properties of the ontology `DocumentiAF`.

The ontology OntoLuoghi has been introduced in [27] in order to handle and reason on geographical places, especially in the Italian country. In the context of the Museo della Fabbrica, we are interested in referencing the provenance place of the letters. The ontology extends `LinkedGeoData`, providing a taxonomy of geographical Italian places together with object properties that associate a place with its own administrative locations. The ontology `Professioni` describes people and companies cited in the documents of Museo della Fabbrica. The ontology represents persons and companies by means of the classes `Persona` and `Azienda`, respectively. The former is equivalent to the `CIDOC-CRM` class `E21_Person`, whereas the latter is a subclass of the `CIDOC-CRM` class `E74_Group`. An instance of `Persona` can be a member of the class `Professionista`, which describes the profession of the person. The ontology also provides a taxonomy for the jobs a person can carry out. The superclass of the taxonomy is `Mansioni`. One can specify the first name and the last name of each element of the class `Persona` by means of the data properties `haNome` and `haCognome`, respectively. The class Incarico describes a task performed by a professional. We distinguish two kinds of tasks, IncaricoAffidato, which refers to tasks effectively assigned, and IncaricoIndicato, which refers to tasks suggested for a professional. The person (company) designated to perform the task is an element of the class Persona (Azienda) and the job specified in the task is an instance of the class Mansione. Persona and Azienda are related to the classes IncaricoAffidato and IncaricoIndicato by means of the object properties haIncaricoAffidato and haIncaricoIndicato respectively, both subproperties of the object property haIncarico. People and job assigned are related by the object property haQualificaIncarico, having as domain subclasses of the class Incarico and as range the class Mansione. These object properties are subproperties of the CIDOC-CRM relation P14i_performed. The classes Mansione and Incarico are subclasses of the CIDOC-CRM class E7_Activity. The start and the end dates of the commission are introduced by the data properties haDataInizioIncarico and haDataFineIncarico, respectively, having as range the data type xsd:date and as domain an instance of the class Durata_Incarico. Such instance is associated to an instance of Incarico by means of the object property haDurataIncarico. The classes of the ontology Professioni are shown in Figure 44.

Figure 44: Classes of the ontology Professioni.

As outlined in the first part of this section, ArchivioMuseoFabbrica is the aggregator of the ontology presented in this section. The usage schema of the ontology `ArchivioMuseoFabbrica` is shown in Figure 45.

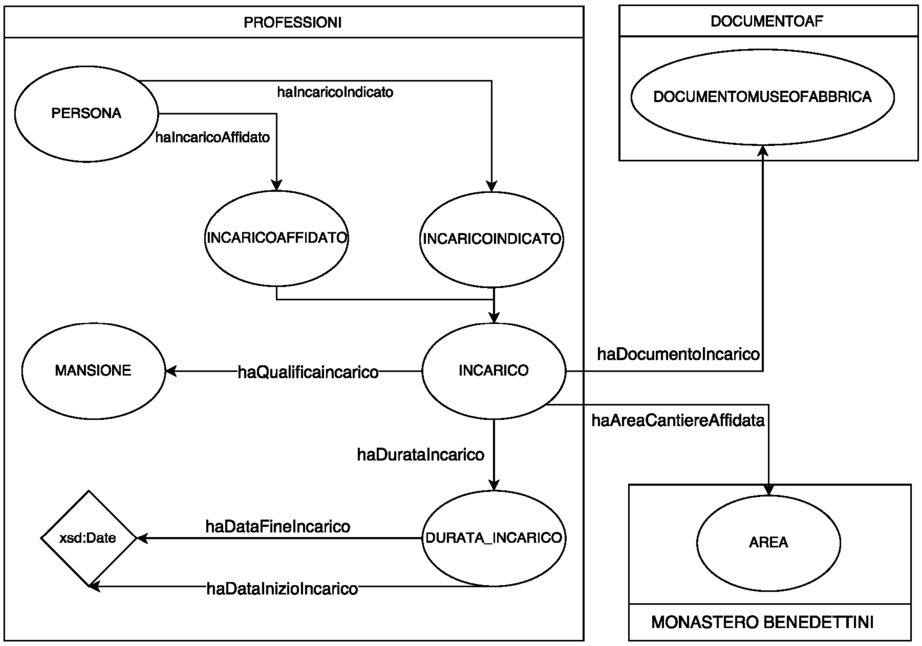

Figure 45: Schema of the ontology `ArchivioMuseoFabbrica`.

Specifically, it contains the object property `haAreaCantiereAffidata`, having as domain the class `Incarico` and as range the class `Area`. It provides the link between tasks from the ontology `Professioni` to the areas of the monastery (ontology `EdificiStorici`) involved in the task. The second object property is `haDocumentoIncarico`, having as domain the class `Incarico` and as range the class `Documenti`, associating a task performed by a professional with the archive of the Museo della Fabbrica (ontology `DocumentiAF`) that documented the task. The latter object property is a subclass of the `CIDOC-CRM` relation `P70_documents`. The task of populating the ontologies has been accomplished by means of several ad hoc scripts written in Java to read the source data, mainly provided as Excel sheets, and to return the ontological dataset. The main library used is the OWL-API, a Java library for manipulating ontologies. Some examples of source data and ontological mapping are shown below. Figure 46 shows the input data of the Giarrizzo's letters, whereas Figure 47 shows the ontological data displayed by the editor Protégé. An example of the input data of the press review is illustrated in Figure 48, whereas an example of OWL mapping in Figure 49.

	A	B	C	D	E
1	data Luogo	n. relazione	data sopralluogo	Mittente	Destinatorio
2	Milano 27 marzo 1991	13	20, 21,22,23 marzo 1991	Giancarlo De Carlo	Giuseppe Giarrizzo
3	Milano 26 luglio 1991	15	19/20 luglio 1991	Giancarlo De Carlo	Giuseppe Giarrizzo
4	Assente	59	29/30/31 gennaio 1999	Giancarlo De Carlo	Giuseppe Giarrizzo
5	Milano 10 settembre 1991	16	5/6/7 settembre 1991	Giancarlo De Carlo	Giuseppe Giarrizzo
6	Milano 4 ottobre 1991	17	1/2/3 ottobre 1991	Giancarlo De Carlo	Giuseppe Giarrizzo
7	Milano 7 novembre 1991	18	3/4 novemre 1991	Giancarlo De Carlo	Giuseppe Giarrizzo
8	Milano 21 novembre 1991	19	19/20 novembre 1991	Giancarlo De Carlo	Giuseppe Giarrizzo
9	Milano 20 dicembre 1991	20	14/ 15/ 16/ 17 dicembre 1991	Giancarlo De Carlo	Giuseppe Giarrizzo
10	Milano, 28 gennaio 1992	21	19/20/21 gennaio 1992	Giancarlo De Carlo	Giuseppe Giarrizzo

Figure 46: Dataset example of professor Giarrizzo's letters.

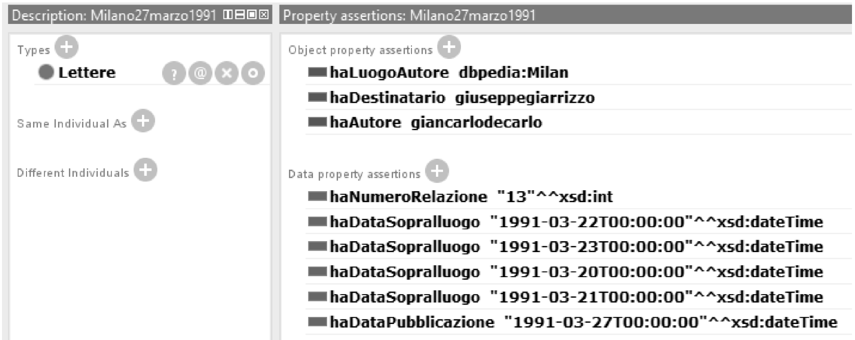

Figure 47: Example of mapping of the professor Giarrizzo's letters.

1	data	testata	pagina	n. pagina	ruubrica	titolo	autore	pre
2	03/01/1996	la sicilia	assente	13	avviato il restauro	ripristino di tutti i dettagli originali	luigi prestinenza	0
3	21/02/1996	la sicilia	assente	16	catania e l'unione europea	ospiterà due importati appuntamenti internazionali	pinella leocata	1
4	28/02/1996	la sicilia	il fatto	2	[redacted]		Salvatore la rocca	2
5	28/02/1996	la sicilia	il fatto	2	l'amarezza di grande Bommarito	"aspettiamo l'esito dell'inchiesta, chi ha sbagliato pagherà"	pinella leocata	1

Figure 48: Dataset example of press review.

Figure 49: Example of mapping of the press review.

6.3 The ontology *SaintGall*

6.3.1 Introduction

The project of valorisation of the Benedictine Monastery of San Nicola L'Arena in Catania, currently carried on by the University of Catania, required the realization of an information system allowing the different agents involved in the process to manipulate information concerning the Sicilian late-baroque building nowadays hosting the Department of Humanities, and, in particular, to query, integrate, share, understand, and modify data concerning its contemporary renovation (1977–2005). In view of the project requirements, we proposed an ontology, called ArchivioMuseoFabbrica, for the most important features of the Monastery (see Section 6.2).

The ontology ArchivioMuseoFabbrica has been developed to value the rich archival heritage arisen directly from De Carlo's building site and storing pictures, diaries, projects, letters, photos, architectural plastics, and so on.[18] The ontology of the Monastery is very specific and focused on the contemporary aspect and re-adaptation of the building. Indeed, it has been conceived to reconstruct relationships between areas involved in the renovation work and people and companies entrusted with the performance of specific tasks. Besides information on documents and people, the ontology also includes a section devoted to the internal description of the building. The latter has been defined from the *Progetto Guida*– designed by the architect De Carlo for the renovation of the Benedictine Monastery (1988)– in order to associate each area of the Monastery with the relative documents stored inside the building site. In light of such characteristics, it is very useful for managing knowledge about the specific monastery and refurbishment activities. Such work and the value it has as a support for interventions on this specific architectural heritage opens a new path of research: to build the ontology of an ideal or generic monastery. It was therefore necessary to introduce a more generic tool permitting to describe the building in an atomic way, namely considering the smallest characterizing part of it and combining them appropriately in a larger landscape. From this point of view, the ontology turns out to be excessively focused on the renovation work and, consequently, it does not permit to analyse the Sicilian building as an ideal or generic monastery and to study its architectonic history through the eras preceding the Seventies of the XX Century. In view of that, the Saint Gall plan has been used to define a novel ontology concerning the generic "concept" of Benedictine monastery.[19] In fact, the scroll of the Saint Gall plan stored in the *Stiftsbibliothek Sankt Gallen* and representing the Swiss Monastery in plant

[18]https://goo.gl/evgpB9
[19]http://www.stgallplan.org/

is a precious document not only from historical and artistic points of view – it presents itself as a conceptual map of an "ideal monastery" –, but also because of the organization of the information easily readable due to its high-resolution digitalization.[20] Thanks to the schematic nature of the Saint Gall plan and since it has been conceived with the purpose of organizing the spaces according to the monastic life-style, leaded by the San Benedict rule, it is a perfect source for the development of an ontology for the "ideal monastery".

In this section, we present an OWL 2 ontology, called SaintGall, representing the monastery described in the Saint Gall plan. The ontology SaintGall has been developed by taking into account structural, architectonic, and functional details of the buildings included in the plan, and information provided by [128].

The major problems we faced in considering information provided by [128] are the correct interpretation of the socio-cultural context and the lack of encyclopaedic and architectural knowledge of the author. Our interpretation of the text, however, is confirmed by the studies of more recent authors [58], [39], [80].

The entities of the ontology have been created from scratch. The usage of foundational ontologies has been avoided in this phase, since we preferred to analyse horizontally the domain and to postpone the phase of ontological interoperability at a later stage. The ontology consists of more than 400 classes, almost 60 object properties, and more than 1000 logical axioms, and it exploits OWL 2 constructs such as existential restriction and qualified cardinality restriction (further structural details are summarized in Table 14).

We used the Fact++ reasoner for the classification of the ontology, as it performs well with ontologies admitting axioms containing existential ("Some values from") and universal ("All values from") OWL restrictions.[21]

Metric type	SaintGall metric value
DL expressivity	SHOIQ(D)
Axioms count	1665
Logical axioms count	1103
Declaration axioms count	477
Classes count	401
Total classes count	401
Object properties count	58
Data properties count	3

[20]The project of digitalization and metadatation of the Saint Gall plan lasted approximately six years and gave to the research community the possibility to remotely vision the scroll in every detail, from front to back, and to zoom and rotate it. There is also a search tool for the metadata included in the map. The project, available at http://www.stgallplan.org/, also includes the Manuscripts of the Reichenau Monastery.

[21]http://owl.cs.manchester.ac.uk/tools/fact/

Individuals count	13
Sub-class axioms count	972
Equivalent classes axioms count	18
Disjoint classes axioms count	16
Hidden GCI count	8
Sub-object property axioms count	40
Inverse object properties axioms count	23
Transitive object properties axioms count	8
Symmetric object properties axioms count	2
Object property range axioms count	2
Data property domain axioms count	3
Data property range axioms count	3
Class assertion axioms count	13
Same individual axioms count	2
Different individuals axioms count	1
Annotation assertion axioms count	86

Table 14: Structural metrics of the ontology SaintGall

6.3.2 Historical Notes

Monasteries were conceived by the Benedictine monastic order, founded by Saint Benedict of Nursia, after the collapse of the Western Roman Empire. The monastic shape aims at preserving the European Christianity inside small self-sustaining communities where to lead a life of mystic and religious contemplation and introspection. The main principle is to protect and shield Christian religion and tradition from barbarian invasions. In the tradition related to Saint Benedict, the theme of Goths is recurring. For instance, we recall the famous episode of the meeting between the God's servant Benedict and the king Totila, commander of the Goth's army.

Monasteries differ from convents primarily because of their purpose. Monasteries are inhabited by monks belonging to some monastic order such as the Benedictine one, having an ascetic and solitary lifestyle. Convents, originating later with the mendicant orders such as the Franciscan one, are more dependent by the outside world. The two religious constructions arose in different historical periods, carrying out different functions inside the religious community. Monasteries are often also abbeys that are spaces where the *nullius diocesis* is effective. Such norm, in the canon law, represents the independence of a church and of the related monastery from the diocese in which the building is located.

Therefore, the abbot substitutes for the bishop inside the Benedictine "village".

<<Monasticism has its root in the interpretation of the Christian faith developed in the theology of the 6th century firstly in Orient. Analogously to theology and architecture, it is subjected to a deep transformation in Occident. [...] The Benedictine Order remains for a long time the principal one. Hundreds of convents and monasteries are spread across the Christian Europe and represent cells of Christian tradition and faith, of science, and of culture>> [80].

Since the foundation of the Monastery of Montecassino, the first Benedictine monastery, in the first half of the 6th century, the Benedictine Order has been subjected to incessant internal reforms, especially when periods of spiritual and moral weakening took over the strictness of the Order. Reforms often influenced the housing architecture of the monks.

Starting from the 7th century, Western Europe is characterized by a capillary network of monasteries, preserved more than castles, public and laic works, precisely because of their religious function [58]. Their shape in Occident remained largely unchanged in its characteristics during the whole Middle Age and in all Christian countries, with differences in the volume of some spaces. For example, in more recent periods, largest libraries are built in order to host bigger amounts of books.

The plan of Saint Gall, illustrated in Figure 50, is a model of monastery better representing the Benedictine architecture since it is strongly inspired by the rule of Saint Benedict, and can be considered a fixed-type for the Middle Age monasteries [39].

The plan has been designed in the context of Pre-Romanesque Carolingian art and architecture, in which a varied and complicated partition of the space is preferred: base arches separate high and wide spaces, ribbed vaults are used to emphasize the complexity of the environment, and walls are treated so that <<it is given greater importance to the compactness of the masonry blocks until the 11th century.

[...] The planimetry of Saint Gall Monastery provides the ideal scheme adopted until the late-Gothic period: the monks dwellings are located on three sides of a rectangular or quadratic courtyard (cloister), while on the other side (on north or on south) runs one side of the church. The internal court is entirely surrounded by a quadrangular arcade that links the buildings. Usually, beside the oriental wing of the cloister, we find two small rooms assigned to the sacristy and armarium (book storage), and one room assigned to the monks reunions where they discuss all the community matters under the supervision of the abbot or the prior.

On the east side of the church, there is the monks dormitory, located on the upper floor for space connectivity reasons and because of tasks to be performed inside the church. Often, in front of the church there is the refectory, the dinner room, and the kitchen of the monastery, while on the occidental wing there are others buildings, services, and the entrance.>> [80]. Thus, rooms of common use such as dormitories turn out to be of great relevance. They are usually located beside a cloister containing a little garden or a green area, and, in the middle, a fountain or a water source for the ablutions. The cloister was devoted to contemplation of beauty and meditation, being an open area in a seclusion space. Inside the fortification, right out the cloister area, there are locations assigned to the farms and the dwelling of animals.

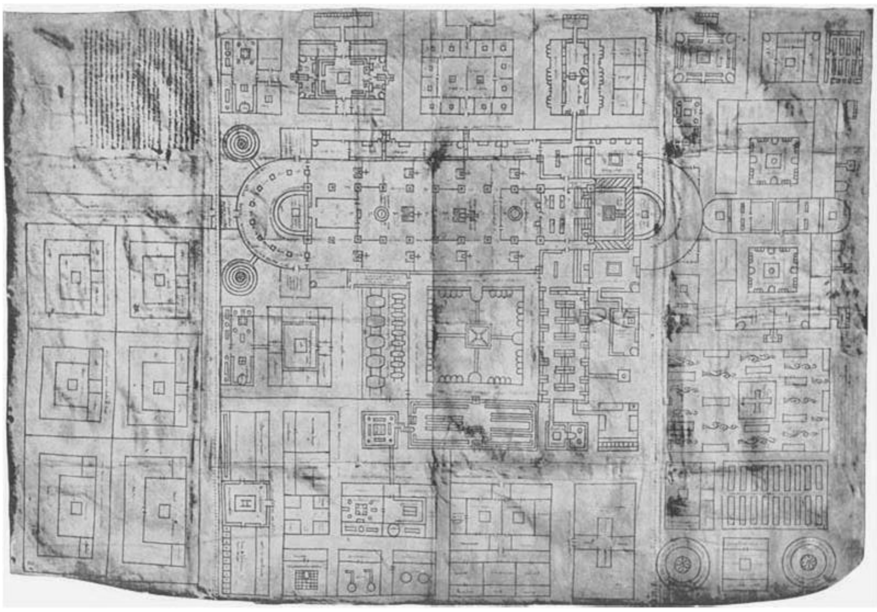

Figure 50: The plan of Saint Gall.

As showed in Figure 50, the space arrangement is inspired by that of a small village: a large church in the centre with annexed cloister and dormitory; at East one can find the novitiate, the infirmary, the cemetery, and the orchards; at South the factories; at North the guest houses, schools, and the dwelling of the abbot; and at West the farms. Thanks to its shape, the Monastery accomplishes the *nullius diocesis*, being self-sustained and independent from the outside world, and the *ora et labora* rule professed by San Benedict.

Since it is one of the most ancient descriptions of primitive Benedictine monastery arrived intact today, it turns out to be an important structural, architectonic, and functional landmark for the Benedictine monasteries. In the plan, the Saint Gall Monastery is idealized together with its essential components. In fact, as it often happens in the context of architectural history, buildings realized in a long temporal window are subject to change with respect to the original idea because of historical, economical, practical, and morphological reasons. Many European monasteries are inspired by the Saint Gall plan even if for practical and technical reasons they deviated from it.

For instance, the Benedictine Monastery of San Nicolò l'Arena in Catania [38] contains most of the elements of the Saint Gall plan with the exception of some locations such as the brewery that, for cultural reasons, has been replaced by a distillery. Moreover, the Benedictine Monastery in Catania is a urban monastery and therefore the structure of the animal farms is also slightly modified.

6.3.3 Applied ontology for architecture heritage: a focus about Types in Architecture

Literature is rich in research about the effort of applying ontologies to the design for architectural heritage. Ontologies indeed can describe the conservation process or the redesign of architectural artefacts in order to support both investigation and design phases [1].

Recently, documentation processes concerning the conservation of the architectural heritage and of the architectural design *ex novo* have been carried out focusing efforts and contributions on knowledge representation and management of existing elements, training memories, and personal remembrances [115], [116], [88].

Thus a representation of knowledge that suits the richness and specificity of information related to historical architecture is needed so to be a support for the decision-making of design phases [1].

One of the fields where ontologies have been introduced is architecture because of its intrinsic wideness and variety of knowledge (historical, technical, artistic, subjective knowledge, and so on). In facts the theoretical luggage of architecture fits well to be explored and modeled via ontological methods. On the other hand, ontological representation of knowledge can be extremely useful to manage, organize and share this luggage in an unambiguous way. In addition it should be available even as a tool for the single agent that has to unravel the knowledge, data, and information concerning a single design task.

Design in architecture is an expression of critical conscience detached from reality and derives from the special personal universe of the architect and her/his "inventive memory". The analysis of memories (disaggregated knowledge) can be possible through ontological tools [66], [12].

According to some literature, the architectural type is a kind of epistemological foundational knowledge. Specifically, an architectural type is a logical statement that describes a formal structure (i.e., a shape) [91]. Types regulate relational and functional systems, a kind of "mental projects" that can be considered pre-representations of the space preceding the design stage [97].

An ontological analysis can make explicit, clarify, and manage the complexity of references, constraints, functional goals, and designer's ambition through a cognitively centered design process. Moreover, it helps in highlighting and coherently separating the key elements of the architectural creative process [116].

Types in architecture have been listed, explained and discussed over time in the attempt to introduce and specialize notions such as form and function. For example, in Palladio's [102] and Durand's [47] books we find listed some. They are about architectural objects and urban frame which are referred to buildings and infrastructure intended to accomplish determinate function (burial monuments and hospitals, city walls, city doors, streets, triumphal arches, bridges, public squares, public buildings, temples/churches, palaces, courthouses, colleges, libraries, museums, observatories, markets, lighthouses, customs trade fairs, theaters, and hospitals).

There are researches about how to manage via ontological analysis the intrinsic knowledge of architectural type as a first step to explore all the memories with the aim of supporting design processes in both the restoration and refurbishment of architectural heritage.

Our work of ontological modeling of the Saint Gall map will help to shed light on the *Monastery Type* and to capture significant logical, semantic, geometric, and functional knowledge about elements used when designing monastery. A monastery is an evolution of the nucleus of more ancient architectural artefacts that changed their function over time. Monasteries and later convents in their turn were converted in hospitals, schools, universities.

The SaintGall ontology has been designed in such a way that significant features and shapes about the Monastery type are adequately represented. Thus it can be a useful support for design tasks concerning buildings of similar kind.

6.3.4 Description of the ontology SaintGall

The ontology SaintGall describes buildings and green spaces depicted in the plan of Saint Gall considering their cardinal orientation, their position with respect to other entities inside the plan, and their architectonic, structural, and functional features.

The ontology exploits the following main classes: the class `Building` describes a generic building, `Garden` specifies a generic green space, `Element` describes architectural, natural, ornamental, and votive elements, furnitures and tools of common use illustrated in the plan. The ontology also provides classes and properties to describe the cardinal orientation, position, and shape of the structures of the plan, and the role of people living inside the monastery.

At first, we model the functional areas of the monastery classifying the buildings represented on the map according to their intended use. Specifically, we introduce as subclasses of `Building` the pairwise disjoint classes:

- `BuildingForEducation`,
- `BuildingForHospitality`,
- `BuildingForTheSickAndInfirm`,
- `FarmBuilding`,
- `PrincipalMonasticBuilding`.

The class `BuildingForEducation` includes, in particular, the class `School`, modeling a building intended for the education of scholars, and the class `NoviceCloister`, representing the novice cloister, dwelling of young people oriented to the monastic life. The class `BuildingForHospitality` contains among others the class `HospitiumDistingueshedGuests`, modeling the hospitium for the reception of eminent strangers, and the class `HospitiumPoorTravelersPilgrims`, representing the dwelling of poor travelers and pilgrims. The class `BuildingForTheSickAndInfirm` contains in particular the subclass `InfirmaryCloister`, representing the cloister where the sick brethren are lodged, and the class `DoctorHouse`, containing among others a private room for the physician and a room for very ill patients. The class `FarmBuilding` models the factory, the working house, and other buildings devoted to domestic cattle, poultry, and their keepers. The class `PrincipalMonasticBuilding` includes in particular the classes `AbbotHouse`, modeling the dwelling of the abbot, `TheCloister`, describing the cloister where monks live, and `TheChurch`, describing the abbey. The hierarchy of `Building` is shown in Figure 51.

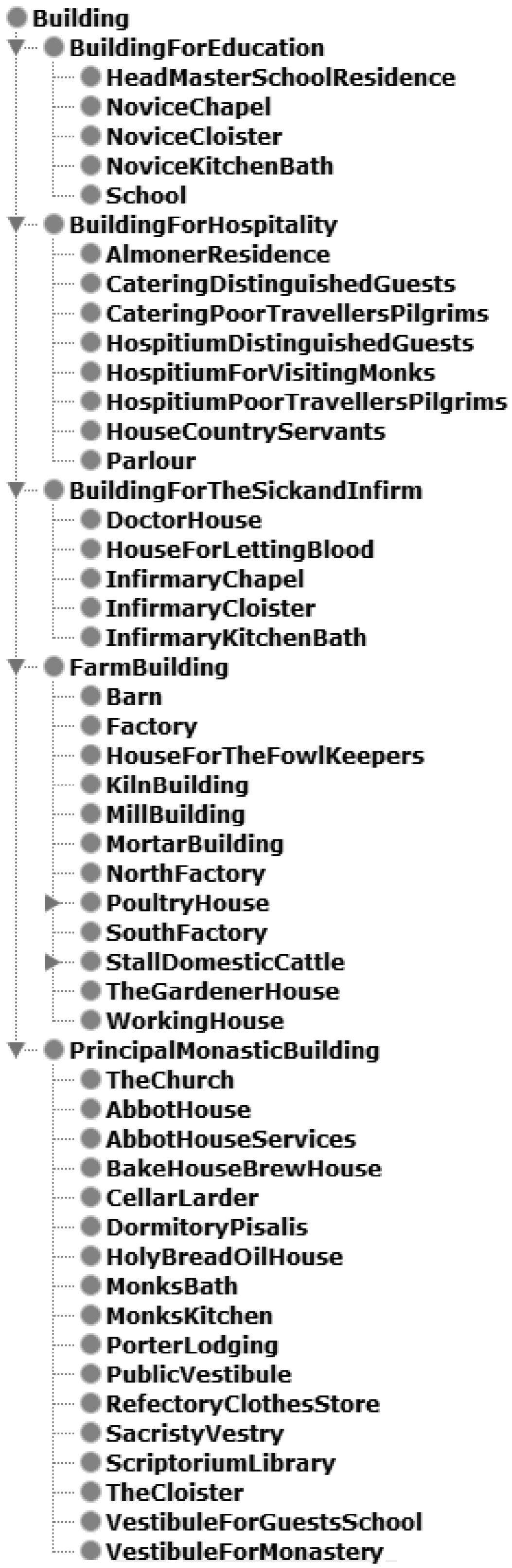

Figure 51: Subclass hierarchy of `Building`.

The green spaces inside the monastery are modeled by means of the class `Garden`, having the disjoint subclasses `Cemetery`, `KitchenGarden`, and `PhysicGarden`. Cardinal orientation of buildings and gardens on the map are modeled by the classes `CardinalDirection`, `CentralPosition`, and the object property `hasPosition`, having as range the union of `CardinalDirection` and `CentralPosition`. `CardinalDirection` is a finite enumeration of the values `East`, `North`, `NorthEast`, `NorthWest`, `South`, `SouthEast`, `SouthWest`, `West`. `CentralPosition` contains only the individual `Centre`. In addition, we introduced the defined classes `CentralArea`, `EastArea`, `NorthArea`, `NorthEastArea`, `NorthWestArea`, `WestArea`, `SouthEastArea`, `SouthWestArea`, `SouthArea`, whose subclasses, representing the buildings and gardens of the monastery, are deduced by inference. Figure 52 shows the description of the class `NorthArea`, whereas Figure 53 illustrates the inferred hierarchy of `NorthArea`.

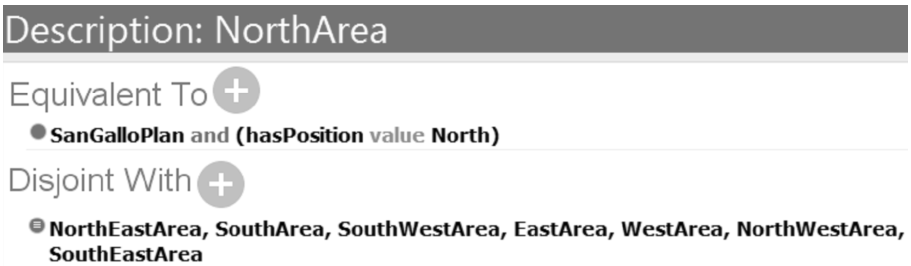

Figure 52: Description of `NorthArea`.

Figure 53: Inferred hierarchy of `NorthArea`.

In addition, we defined the position of buildings or gardens in the map with respect to other contiguous buildings or gardens by means of the object properties `onEastOf`, `onNorthEastOf`, `onNorthOf`, `onNorthWestOf`, `onSouthEastOf`, `onSouthOf`, `onSouthWestOf`, `onWestOf`, where `onEastOf` is the inverse of `onWestOf`, `onNorthOf` of `onSouthOf`, `onNorthWestOf` of `onSouthEastOf`, and `onNorthEastOf` of `onSouthWestOf`.

Next, we analyze the shape, the size, and the internal structure of buildings and gardens.[22] We define the class `Shape`, modeling the shape of structures such as cloisters (`RectangulaShape` or `SquareShape`), whose subclass hierarchy is shown in Figure 54, and the object property `hasShape`, having as range the class `Shape` and `owl:Thing` as domain.

The class `Size` and the object property `hasSize` model the size of buildings.

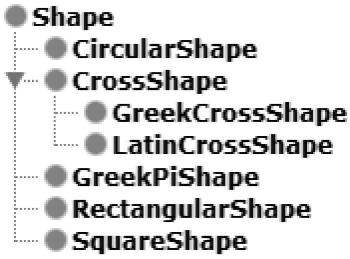

Figure 54: Subclass hierarchy of `Shape`.

The class `Element` has as subclasses the class `ArchitecturalElement`, describing general architectural elements inside the map, the class `Furniture`, modeling objects used in everyday life such as `Bedstand` and `Desk`, the class `Tool`, modeling tools of common use such as `Furnace` and `Boiler`, and classes describing rooms, clothes, food, votive and ornamental elements. Anything included in such classes can be used in other contexts outside the Saint Gall plan.

In addition, we provide the object properties `contains`, together with its subproperties `consistOf`, `containsAround`, and so on, and its inverse `isContainedIn`, together with its subproperties `isPartOf`, `isContainedAround`, and so on. The hierarchy of the subclasses of `Element` and of its related properties are illustrated in Figures 55 and 56.

In Figure 57 we show the picture of the abbot house in the Saint Gall scroll, and in Figure 58 we report the OWL 2 model of the abbot house, as represented by the tool *Protégé*.

This building, inhabited by the abbot, is surrounded by a fence. It consists of two stories of which the lower one has an open portico on the east and west sides. The inner space is split into two chambers: the abbot sleeping and sitting rooms. The upper story contains some small chambers and one large chamber. Details concerning the furniture of the abbot sleeping and sitting rooms are modeled by the classes `AbbotSleepingRoom` and `AbbotSittingRoom`, respectively, which are subclasses of `Chamber`.

[22]We take into account the information provided by the sources available to us, namely [128], [39], [42], [58], [80], and http://www.stgallplan.org/.

Figure 55: Subclass hierarchy of `Element`.

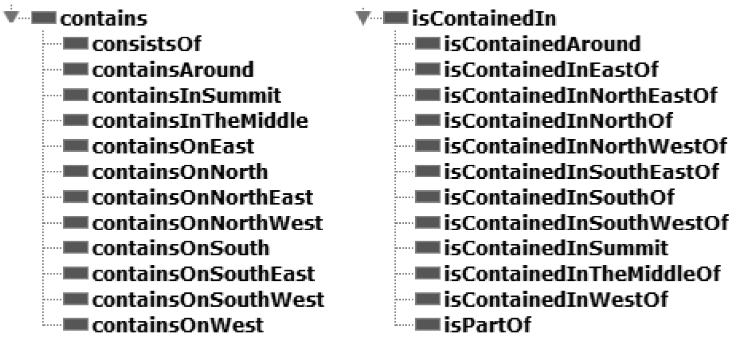

Figure 56: Object properties related to the class `Element`.

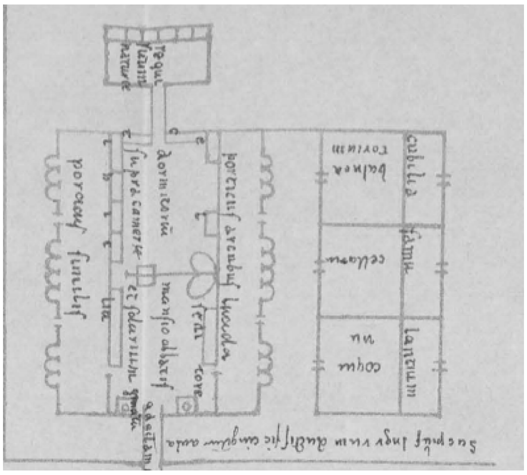

Figure 57: Picture of the abbot house.

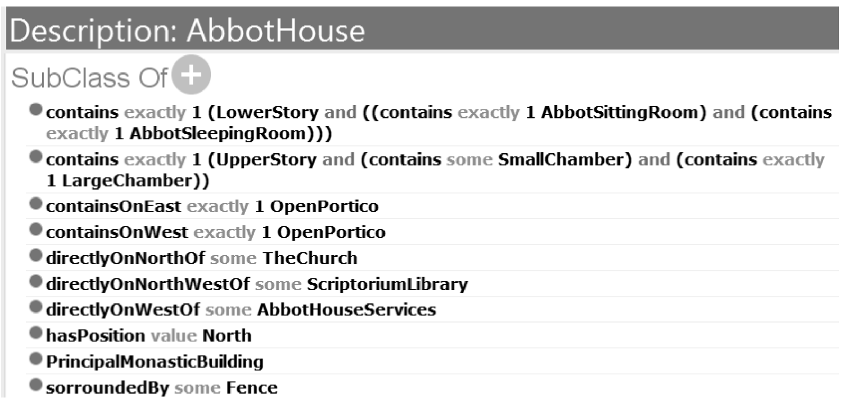

Figure 58: Description of `AbbotHouse`.

We also model people living in the monastery. As shown in the abbot example of Figure 59, they are classified according to the place in which they live and spend most of the day.

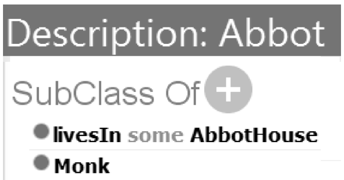

Figure 59: Description of `Abbot`.

As shown in Figure 60 for the specific case of the abbot house, our OWL 2 representation of the Saint Gall map was developed so as to be as faithful as possible to the text in natural language of [128] and to the map.

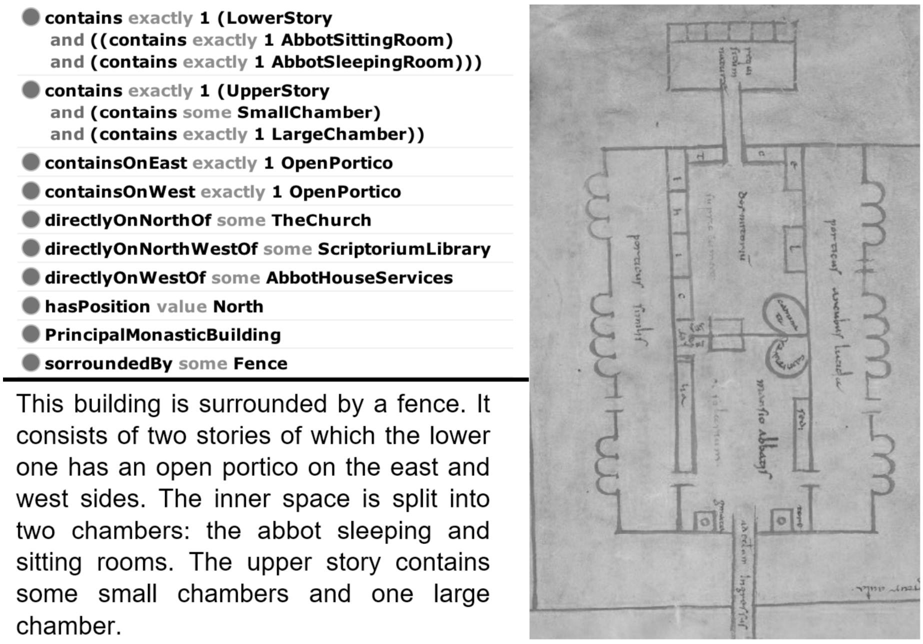

Figure 60: Example of the abbot house.

The illustration of the monk cloister in the Saint Gall scroll is reported in Figure 61.

The cloister court is located on the south wing of the church. It is constituted by a large area bounded on the north by the church and on the other sides by three important buildings of the monastery: the parlour, the dormitory, and the refectory. The parlour serves as a vestibule to the cloister and as a space where servants receive orders by superiors and visitors can meet the monks. The refectory is used as dining and amusement room. The cloister also contains a passage to a cellar with larder for storing food. It is surrounded by a covered walk, called *porticus*, and contains in the centre an open space with a fountain or water source, probably laid out with grass or shrubs. There is an open arch in the centre of each side, which gives access to the central space (cloister garth). The path from the side to the centre designs four quadrangular spaces that have the same organization. On either side of this central open arch there is a group of four arches, whose pillars rest on a low basement wall, being intended to admit light and air in the manner of open windows, but not to give access to the cloister garth. The cloister is characterized by the presence of a portico, rested on a low perimetric wall and admitting light inside the spaces annexed to the gallery. For each side of the cloister, there is a big central arch that gives access to the garden. The cloister, connecting the different areas of the cenoby, allowed monks to use the portico as a place where to rest and have shelter from the bad weather.

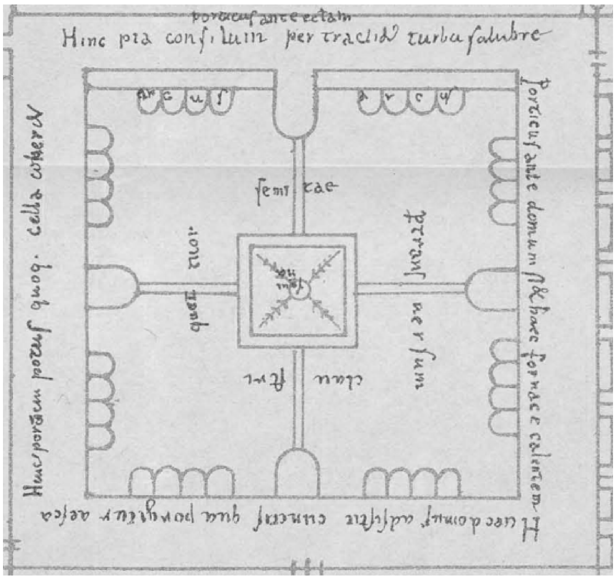

Figure 61: Picture of the monks cloister.

The OWL 2 model of the monk cloister, as represented by the tool *Protégé*, is illustrated in Figure 62.

Figure 62: Description of `TheCloister`.

7 Conclusions and Future Work

In this dissertation, we have used the decidable set-theoretic fragment of set theory $\mathsf{4LQS^R}$ to represent an expressive description logic called $\mathcal{DL}_{\mathbf{D}}^{4}$ and its extension, namely the DL $\mathcal{DL}_{\mathbf{D}}^{4,\times}$, and we have studied the main reasoning tasks for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ (and hence for $\mathcal{DL}_{\mathbf{D}}^{4}$). Specifically, we have proved that the *Conjunctive Query Answering* (*CQA*) problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ is decidable by reducing it to $\mathsf{4LQS^R}$ via a mapping θ which sends $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KBs to $\mathsf{4LQS^R}$-formulae. Likewise, we have proved also that a generalization of the *CQA* problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$, called *Higher-Order Conjunctive Query Answering* (*HOCQA*) problem, which instantiates the most important reasoning tasks for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ such as the consistency problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KBs, is decidable as well.

Moreover, we have provided a correct and terminating algorithm for the *HOCQA* problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$ which also serves as a decision procedure for the consistency problem for $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-KBs. The algorithm is based on the KE-tableau-system and turns out to be in **2-EXPTIME**.

We have also introduced an improvement to the proposed algorithm based on a variant of the KE-tableau-system, called KE$^{\gamma}$-tableau, which introduces a generalization of the elimination rule treating at the same time universally quantified formulae.

We have implemented in C++ both versions of the algorithm and have tested the efficiency of the two systems. It turns out that the implementation of KE$^{\gamma}$-tableau-based system is more efficient than the KE-tableau-based one and, incidentally, of the implementation of the FO-KE-system, as witnessed by suitable benchmark test sets.

Finally, we have introduced three OWL 2 ontologies for several problems in the context of *Human and Social Science*. The first ontology, called *OntoLocEstimation* and representable in $\mathcal{DL}_{\mathbf{D}}^{4,\times}$, has been introduced for the recognition of geographical places and as a knowledge representation and reasoning support for a rule-based location extraction algorithm built from simple grammar rules of the Italian language. The ontology also keeps track of the algorithm applied for detecting places and stores the degree of belief of each candidate location.

The second OWL 2 ontology, called *ArchivioMuseoFabbrica* and representable in $\mathcal{DL}_{\mathbf{D}}^{4,\times}$, has been introduced to study the history of the renovation of the Benedictine Monastery of San Nicolò L'Arena in Catania and its adaptation as a university campus by the architect Giancarlo De Carlo. In the context of characterizing a Benedictine monastery, we have also proposed an ontology called *SaintGall*, conceived to study the shape of the ideal Benedictine monastery and its peculiarities described in the Saint Gall map, one of the most ancient descriptions of a religious building. By means of the ontology *SaintGall*, scholars and researchers in Human Science can effectively compare several distinct monastic architectures, and from their differences and similarities make inferences not only in the architectonic and stylistic ambit but also in the interpretative and theological areas. The ontology is also a first step towards the definition of the architectonic type of monastery, and more in general, towards the understanding of types in architecture.

We plan to extend the set-theoretic fragment underpinning the reasoner so as to include a restricted version of the operator of relational composition, in order to be able to reason with description logics admitting full existential and universal quantification. Moreover, we also intend to extend the fragment $\mathsf{4LQS^R}$ with meta-modeling capabilities [93], so as to allow one to define concepts containing other concepts and roles (i.e., meta-concepts) and relationships between concepts or between roles (i.e., meta-roles). In addition, we intend to improve the reasoner in order to deal with the reasoning problem of ontology classification. We compare the resulting reasoner with existing well-known reasoners such as Hermit [57] and Pellet [112], providing also some benchmarking. In addition, we plan to allow data type reasoning by either integrating existing solvers for the Satisfiability Modulo Theories (SMT) problem or by designing new *ad hoc* solvers. Finally, as each branch of a KE^{γ}-tableau can be independently computed by a single processing unit, we plan to implement a parallel version of the system by using the Nvidia CUDA library.

Concerning the ontology *OntoLocEstimation*, we plan to introduce additional filters and rules in the algorithm so as to gain a larger coverage of the Italian grammar. In particular, we aim to make the algorithm sensitive to the contexts from which words are drawn, namely non-structured texts. Finally, we plan to abandon external datasets such as OpenStreetMap, in favor of internal built-in datasets that can be integrated by information provided by local governments or final users.

Additionally, since at present the formalization and the digitalization of the inventory of the *Archivio del Museo della Fabbrica* is not complete, the next goal is to extend the dataset of *ArchivoMuseoFabbrica* with any missing information as soon as it becomes available. In addition, we plan to analyze the historical context and its influence on the progress of the works by modeling the internal events and relating them with relevant coeval historical events. The latter task will be carried out by resorting to well-known ontologies, such as the LODE ontology. Subsequently, we intend to provide an on-line system together with a graphical interface to query and explore the dataset.

Finally, we are currently considering the integration of the ontology *SaintGall* with the *ArchivioMuseoFabbrica* and with other widespread ontologies for cultural heritage such as as *CIDOC-CRM*. Some generic classes from the ontology *SaintGall*, such as *Church* and *Cloister*, can be reused to design novel ontologies describing buildings outside the Benedectine context. Consider, for instance, the architectonic structure of closed garden (cloister or court), which can be also found in municipal buildings. A work concerning shapes and types in the Benedictine architecture is in progress.

References

[1] M. Acierno, S. Cursi, D. Simeone, and D. Fiorani. Architectural heritage knowledge modelling: An ontology-based framework for conservation process. In *Journal of Cultural Heritage*, volume 24, pages 124–133. Elsevier Masson France, 2017.

[2] E. Agichtein and L. Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital Libraries*, pages 85–94. ACM, 2000.

[3] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press, New York, NY, USA, 2003.

[4] F. Baader and P. Hanschke. A scheme for integrating concrete domains into concept languages. In John Mylopoulos and Raymond Reiter, editors, *Proceedings of the 12th International Joint Conference on Artificial Intelligence. Sydney, Australia, August 24-30, 1991*, pages 452–457. Morgan Kaufmann, 1991.

[5] F. Baader, I. Horrocks, and C. Lutz, editors. *Introduction to Description Logics.* Cambridge University Press, New York, NY, USA, 2017.

[6] F. Baader, I. Horrocks, and U. Sattler. Description logics as ontology languages for the semantic web. In *Festschrift in honor of Jörg Siekmann, Lecture Notes in Artificial Intelligence*, pages 228–248. Springer-Verlag, 2003.

[7] F. Baader, T. Horrocks, and U. Sattler. Description logics for the semantic web. *KI*, 16(4):57–59, 2002.

[8] A. Ballatore, D. C. Wilson, and M. Bertolotto. A survey of volunteered open geo-knowledge bases in the semantic web. In *Quality issues in the management of web information*, pages 93–120. Springer, 2013.

[9] J.I. Berman, H. H. Moore IV, and J.R. Wright. Prose stories: Enabling technologies for knowledge based systems. *AT&T Technical Journal*, January/February:69–78, 1994.

[10] A. Bernstein, D.R. Karger, T. Heath, F. Feigenbaum, D. Maynard, E. Motta, and K. Thirunarayan, editors. *The Semantic Web - ISWC*

2009, 8th International Semantic Web Conference, ISWC 2009, Chantilly, VA, USA, October 25-29, 2009. Proceedings, volume 5823 of *Lecture Notes in Computer Science*. Springer, 2009.

[11] J.M. Blanco, A. Illarramendi, and A. Goñi. Building a federated relational database system: An approach using a knowledge-based system. *International Journal of Cooperative Information Systems*, 3(4):415–456, 1994.

[12] D. Borri, D. Camarda, and M.R. Stufano Melone. Typological axioms for architecture. In *D'amato Guerrieri C. (ed.), Proceedings of the 1st Workshop on the State of the Art and Challenges of Research Efforts at Politecnico di Bari, Roma, ISBN 8849299796, 9788849299799*, pages 17–22, 2014.

[13] P. Bruscoli, G. Rossi, A. Dovier, E.G. Omodeo, and E. Pontelli. Extensional and intensional sets in CLP with intensional negation. In *ICLP Workshop on Logic Programming with Sets*, 1993.

[14] D. Calvanese, G. De Giacomo, and M. Lenzerini. Conjunctive query containment and answering under description logic constraints. *ACM Transactions on Computational Logic 9(3): 22:1-22:31*, 2008.

[15] C. Cantale. *Studiare il patrimonio culturale nella contemporaneità*. Ph.d. dissertation, University of Catania, 2018. In Italian.

[16] D. Cantone. Decision procedures for elementary sublanguages of set theory: X. Multilevel syllogistic extended by the singleton and powerset operators. In *Journal of Automated Reasoning*, volume 7, number 2, pages 193–230. Kluwer Academic Publishers, Hingham, MA, USA, 1991.

[17] D. Cantone and V. Cutello. A decidable fragment of the elementary theory of relations and some applications. In *ISSAC*, pages 24–29. ACM, 1990.

[18] D. Cantone and V. Cutello. Decision procedures for stratified set-theoretic syllogistics. In *ISSAC*, pages 105–110. ACM, 1993.

[19] D. Cantone, V. Cutello, and J.T. Schwartz. Decision problems for tarski and presburger arithmetics extended with sets. In *CSL*, volume 533 of *Lecture Notes in Computer Science*, pages 95–109. Springer, 1990.

[20] D. Cantone, A. Ferro, and E. G. Omodeo. *Computable set theory*. Number 6 in International Series of Monographs on Computer Science, Oxford Science Publications. Clarendon Press, Oxford, UK, 1989.

[21] D. Cantone and C. Longo. A decidable two-sorted quantified fragment of set theory with ordered pairs and some undecidable extensions. *Theoretical Computer Science*, 560:307–325, 2014.

[22] D. Cantone, C. Longo, and M. Nicolosi-Asmundo. A decision procedure for a two-sorted extension of multi-level syllogistic with the Cartesian product and some map constructs. In Wolfgang Faber and Nicola Leone, editors, *Proceedings of the 25th Italian Conference on Computational Logic (CILC 2010), Rende, Italy, July 7-9, 2010*, volume 598, pages 1–18 (paper 11). CEUR-WS, ISSN 1613-0073, June 2010.

[23] D. Cantone, C. Longo, and M. Nicolosi-Asmundo. A decidable quantified fragment of set theory involving ordered pairs with applications to description logics. In *Proceedings Computer Science Logic, 20th Annual Conf. of the EACSL, CSL 2011, September 12-15, 2011, Bergen, Norway*, pages 129–143, 2011.

[24] D. Cantone, C. Longo, and A. Pisasale. Comparing description logics with multi-level syllogistics: the description logic $\mathcal{DL}\langle\mathsf{MLSS}_{2,m}^{\times}\rangle$. In Paolo Traverso, editor, *6th Workshop on Semantic Web Applications and Perspectives (Bressanone, Italy, Sep. 21-22, 2010)*, pages 1–13, 2010.

[25] D. Cantone and M. Nicolosi-Asmundo. On the satisfiability problem for a 3-level quantified syllogistic. *CoRR*, abs/1304.2412, 2013.

[26] D. Cantone and M. Nicolosi-Asmundo. On the satisfiability problem for a 4-level quantified syllogistic and some applications to modal logic. *Fundamenta Informaticae*, 124(4):427–448, 2013.

[27] D. Cantone, M. Nicolosi-Asmundo, D. F. Santamaria, and F. Trapani. Ontoceramic: an OWL ontology for ceramics classification. In *Proceedings of CILC 2015, CEUR-WS, vol. 1459, pp. 122–127*, Genova, July 1-3, 2015.

[28] D. Cantone, E. G. Omodeo, and A. Policriti. *Set theory for computing: from decision procedures to declarative programming with sets.* Monographs in Computer Science. Springer-Verlag, New York, NY, USA, 2001.

[29] D. Cantone and P. Ursino. *An Introduction to the Technique of Formative Processes in Set Theory.* Springer International Publishing, 2018.

[30] Cantone, D. and Nicolosi-Asmundo, M. and Santamaria, D.F. A $\mathcal{DL}_{\mathbf{D}}^{4,\times}$-Reasoner. `https://github.com/dfsantamaria/DL4xD-Reasoner`.

[31] D. Caruso, R. Giunta, D. Messina, G. Pappalardo, and E. Tramontana. Rule-based location extraction from Italian unstructured text. In *Proceedings of XVI Workshop "From Object to Agents"(WOA)*, volume 1382, pages 46–52, July 17-19 2015.

[32] M. Cristiá and G. Rossi. A decision procedure for restricted intensional sets. In *CADE*, volume 10395 of *Lecture Notes in Computer Science*, pages 185–201. Springer, 2017.

[33] M. Cristiá and G. Rossi. Programming in java with restricted intensional sets. In *SETS@ABZ*, volume 2199 of *CEUR Workshop Proceedings*, pages 17–31. CEUR-WS.org, 2018.

[34] I.F. Cruz, S. Decker, J. Euzenat, and D.L. McGuinness, editors. *The Emerging Semantic Web, Selected papers from the first Semantic web working symposium, Stanford University, California, USA, July 30 - August 1, 2001*, volume 75 of *Frontiers in Artificial Intelligence and Applications*. IOS press, 2002.

[35] M. D'Agostino. Tableau methods for classical propositional logic. In D'Agostino M., Gabbay Dov M., Reiner H., and Posegga J., editors, *Handbook of Tableau Methods*, pages 45–123. Springer, 1999.

[36] M. D'Agostino and M. Mondadori. The taming of the cut. classical refutations with analytic cut. *Journal of Logic and Computation*, 4:285–319, 1994.

[37] E. Dantsin, T. Eiter, G. Gottlob, and A. Voronkov. Complexity and expressive power of logic programming. *ACM Comput. Surv.*, 33(3):374–425, 2001.

[38] G. De Carlo. *Un progetto per Catania. Il recupero del Monastero di San Nicoló lArena per l' Universitá, (a cura di Daniele Brancolino)*. Sagep, Genova, 1988.

[39] P. De Vecchi and E. Cerchiari. Il medioevo. *Arte nel Tempo, Vol. I, Tomo II, Milano*, 1995.

[40] N. Dershowitz and J. Jouannaud. Rewrite systems. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science (Vol. B)*, pages 243–320. MIT Press, Cambridge, MA, USA, 1990.

[41] M. Doerr. The CIDOC CRM An Ontological Approach to Semantic Interoperability of Metadata. In *AI Magazine*, volume 24, Number 3, pages 75–92, 2003.

[42] M. D'Onofrio. Il panorama dell'architettura religiosa. *I Normanni popolo d'Europa 1030-1200, Roma Palazzo Venezia, Venice*, January 28 - May 30 1994.

[43] A. Dovier, E.G. Omodeo, E. Pontelli, and G. Rossi. {log}: A logic programming language with finite sets. In *ICLP*, pages 111–124. MIT Press, 1991.

[44] A. Dovier, E.G. Omodeo, E. Pontelli, and G. Rossi. Embedding finite sets in a logic programming language. In *ELP*, volume 660 of *Lecture Notes in Computer Science*, pages 150–167. Springer, 1992.

[45] A. Dovier, E.G. Omodeo, E. Pontelli, and G. Rossi. A language for programming in logic with finite sets. *J. Log. Program.*, 28(1):1–44, 1996.

[46] A. Dovier and G. Rossi. Embedding extensional finite sets in CLP. In *ILPS*, pages 540–556. MIT Press, 1993.

[47] J.N.L. Durand. Précis des lecons d'architecture, gillé. Paris, 1799.

[48] Earthquake Commission (EQC), GNS Science, and Land Information New Zealand (LINZ). GeoNet Geological hazard information for New Zealand. `https://www.geonet.org.nz`, 2001.

[49] D. Edelstein, P. Findlen, G. Ceserani, C. Winterer, and Coleman N. Historical research in a digital age: Reflections from the *Mapping the Republic of Letters Project. American Historical Review, Volume 122 Issue 2, April 2017, pp 400-424*, 2017.

[50] R. Ferrario and R. Oltramari, editors. *Formal Ontologies Meet Industry, Proceedings of the 4th Workshop FOMI 2009, September 2, 2009, Vicenza, Italy, in association with the 10th European Conference on Knowledge Management*, volume 198 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2009.

[51] A. Ferro and Omodeo E.G. An efficient validity test for formulae in extensional two-level syllogistic. In *Le Matematiche, 33:130137*, 1978.

[52] A. Ferro, E.G. Omodeo, and T.J. Schwartz. Decision procedures for some fragments of set theory. In *CADE*, volume 87 of *Lecture Notes in Computer Science*, pages 88–96. Springer, 1980.

[53] A. Formisano, E. G. Omodeo, and A. Policriti. Three-variable statements of set-pairing. *Theoretical Computer Science*, 322(1):147–173, 2004.

[54] A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, and L. Schneider. Sweetening ontologies with DOLCE. In *EKAW*, volume 2473 of *Lecture Notes in Computer Science*, pages 166–181. Springer, 2002.

[55] L. Giordano and A. Policriti. Power (set) ALC (extended abstract). In *Proceedings of the 33rd Italian Conference on Computational Logic, Bolzano, Italy, September 20-22, 2018.*, pages 17–21, 2018.

[56] B. Glimm, M. Horridge, B. Parsia, and P.F. Patel-Schneider. A syntax for rules in OWL 2. In *Proceedings of the 5th International Workshop on OWL: Experiences and Directions (OWLED 2009), Chantilly, VA, United States, October 23-24, 2009*, 2009.

[57] B. Glimm, I. Horrocks, B. Motik, G. Stoilos, and Z. Wang. HermiT: An OWL 2 Reasoner. *Journal of Automated Reasoning*, 53(3):245–269, 2014.

[58] E. Gombrich. La storia dell'arte. *Phaidon, London*, 1950.

[59] B.N. Grosof, I. Horrocks, R. Volz, and S. Decker. Description logic programs: combining logic programs with description logic. In *Proceedings of the Twelfth International World Wide Web Conference, WWW 2003, Budapest, Hungary, May 20-24, 2003*, pages 48–57, 2003.

[60] N. Guarino and M.R. Stufano Melone. On the ontological status of design objects. In *Proceedings of the 1st Workshop on Artificial Intelligence and Design, A workshop of the XIV International Conference of the Italian Association for Artificial Intelligence (AI*IA 2015), Ferrara, Italy, September 22, 2015.*, pages 27–32, 2015.

[61] V. Haarslev, R. Möller, and M. Wessel. The Description Logic ALCNHR+ Extended with Concrete Domains: A Practically Motivated Approach. In *Proceedings of International Joint Conference on Automated Reasoning, IJCAR 2001*, pages 29–44. Springer, 2001.

[62] D. Harel and A. Pnueli. On the development of reactive systems. In Krzysztof R., editor, *Logics and Models of Concurrent Systems*, pages 477–498, Berlin, Heidelberg, 1985. Springer Berlin Heidelberg.

[63] G. Hencsey, B. White, Y.R. Chen, L. Kovács, and S. Lawrence, editors. *Proceedings of the Twelfth International World Wide Web Conference, WWW 2003, Budapest, Hungary, May 20-24, 2003.* ACM, 2003.

[64] G.G. Hillebrand, P.C. Kanellakis, H.G. Mairson, and M.Y. Vardi. Undecidable boundedness problems for datalog programs. *Journal of Logic Program*, 25(2):163–190, 1995.

[65] T. Hofweber. *Logic and Ontology.* The Stanford Encyclopedia of Philosophy (Summer 2018 Edition). Edward N. Zalta (ed.), 2018.

[66] J. Hois, M. Bhatt, and O. Kutz. Modular ontologies for architectural design. In *Formal Ontologies Meet Industry, Proceedings of the 4th Workshop FOMI 2009, September 2, 2009, Vicenza, Italy, in association with the 10th European Conference on Knowledge Management*, pages 66–77, 2009.

[67] M. Horridge and S. Bechhofer. The OWL API: A Java API for OWL ontologies. *Semantic Web*, 2(1):11–21, 2011.

[68] I. Horrocks. OWL: A description logic based ontology language. In Peter van Beek, editor, *Principles and Practice of Constraint Programming - CP 2005*, pages 5–8, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[69] I. Horrocks, O. Kutz, and U. Sattler. The even more irresistible SROIQ. In *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning, in Doherty, P. and Mylopoulos, J. and Welty, C. A., eds.*, pages 57–67. AAAI Press, 2006.

[70] I. Horrocks and P. F. Patel-Schneider. A proposal for an owl rules language. In *Proceedings of the 13th International Conference on World Wide Web*, WWW 2004, pages 723–731, New York, NY, USA, 2004. ACM.

[71] I. Horrocks, P.F. Patel-Schneider, S. Bechhofer, and D. Tsarkov. OWL rules: A proposal and prototype implementation. *Journal of Web Semantic*, 3(1):23–40, 2005.

[72] I. Horrocks and U. Sattler. Ontology reasoning in the SHOQ(D) description logic. In *Proceedings of IJCAI 2001*, pages 199–204, 2001.

[73] I. Horrocks and S. Tessaris. A conjunctive query language for description logic aboxes. In *Proceedings of the 17th Nat. Conf. on Artificial Intelligence (AAAI-2000)*, pages 399–404, 2000.

[74] Istituto centrale per gli Archivi (ICAR)- Ministero dei beni e delle attività culturali e del turismo (MIBACT). SAN LOD - Linked Open Data del Sistema Archivistico Nazionale. `http://www.san.beniculturali.it/`, 2011.

[75] Y. Kazakov. RIQ and SROIQ are harder than SHOIQ. In Gerhard Brewka and Jérôme Lang, editors, *Proceedings of the 11th International Conference, KR 2008, Sydney, Australia, Sept. 16-19, 2008*, pages 274–284, 2008.

[76] J. Kleb and R. Volz. Ontology based entity disambiguation with natural language patterns. In *Fourth International Conference on Digital Information Management, 2009. ICDIM 2009.*, pages 1–8. IEEE, 2009.

[77] A. Krisnadhi, F. Maier, and P. Hitzler. OWL and rules. In *Reasoning Web. Semantic Technologies for the Web of Data - 7th International Summer School 2011, Galway, Ireland, August 23-27, 2011, Tutorial Lectures*, pages 382–415, 2011.

[78] M. Krötzsch. OWL 2 Profiles: An Introduction to Lightweight Ontology Languages. In *Reasoning Web. Semantic Technologies for Advanced Query Answering: 8th International Summer School 2012, Vienna, Austria, September 3-8, 2012. Proceedings*, pages 112–183. Springer Berlin Heidelberg, 2012.

[79] M. Krötzsch, S. Rudolph, and P. Hitzler. Description logic rules. In Malik Ghallab, Constantine D. Spyropoulos, Nikos Fakotakis, and Nikos Avouris, editors, *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI 2008)*, pages 80–84. IOS Press, 2008.

[80] H. E. Kubach. Architettura romanica. *Storia Universale dell'Architettura, Milano 1972, consulted edition, Mi*, 2001.

[81] R.E. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM Journal of Computing 6: 467-480*, 1977.

[82] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML)*, pages 282–289, 2001.

[83] Leipzig University and University of Mannheim and OpenLink Software. DBPedia. `https://wiki.dbpedia.org`, 2007.

[84] A. Leonardi and C. (a cura di) Cantale. *La gentilezza e la rabbia. 105 lettere di Giancarlo De Carlo sul recupero del Monastero di San Nicoló a Catania.* Editoriale Agorá, Catania, 2017.

[85] A. Y. Levy and M. C. . Rousset. Combining horn rules and description logics in {CARIN}. *Artificial Intelligence*, 104(1–2):165 – 209, 1998.

[86] C. Lutz. *The Complexity of Reasoning with Concrete Domains.* Ph.d. dissertation, Teaching and Research Area for Theoretical Computer Science, RWTH Aachen, Germany, 2002.

[87] C. Lutz, C. Areces, I. Horrocks, and U. Sattler. Keys, Nominals, and Concrete Domains. *J. Artif. Intell. Res. (JAIR)*, 23:667–726, 2005.

[88] F. Maietti, R. Di Giulio, M. Balzani, E. Piaia, M. Medici, and F. Ferrari. Digital Memory and Integrated Data Capturing: Innovations for an Inclusive Cultural Heritage in Europe Through 3D Semantic Modelling. In Papagiannakis G. (eds) Ioannides M., Magnenat-Thalmann N., editor, *Mixed Reality and Gamification for Cultural Heritage*. Springer, Cham, 2007.

[89] F. Mannino (a cura di). *Breve storia del Monastero dei Benedettini di Catania*. Giuseppe Maimone editore, Catania, 2015.

[90] D. C. Martínez and P. Hitzler. Extending description logic rules. In *The Semantic Web: Research and Applications - 9th Extended Semantic Web Conference, ESWC 2012, Heraklion, Crete, Greece, May 27-31, 2012. Proceedings*, pages 345–359, 2012.

[91] C. Martís Arís. *Le variazioni dell'identità*. CittáStudi, 1990.

[92] G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3:235–244, 1990.

[93] B. Motik. On the properties of metamodeling in owl. In *In 4th International Semantic Web Conference (ISWC 2005)*, pages 548–562, 2005.

[94] B. Motik and I. Horrocks. OWL datatypes: Design and implementation. In *Proceedings of the 7th International Semantic Web Conference (ISWC 2008)*, volume 5318 of *LNCS*, pages 307–322. Springer, October 26–30 2008.

[95] B. Motik and R Rosati. Reconciling description logics and rules. *Journal of the ACM*, 57(5):30:1–30:62, June 2008.

[96] B. Motik, B. Shearer, and I. Horrocks. Hypertableau reasoning for description logics. *Jounal of Artificial Intelligence Research*, 36:165–228, 2009.

[97] S. Muratori. *Saggi di critica e di metodo nello studio dell'architettura*. Edited by G.Marinucci, Roma, Centro Studi di Storia Urbanistica, 1960.

[98] K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *IJCAI-99 Workshop on Machine Learning for Information Filtering*, volume 1, pages 61–67, 1999.

[99] D. Oberle, N. Guarino, and S. Staab. *What is an ontology?* Handbook on Ontologies, 2nd edition. Springer, 2009.

[100] The Open Huniversity. Digital humanities, 2018.

[101] M. Ortiz, R. Sebastian, and M. Šimkus. Query answering in the Horn fragments of the description logics $\mathcal{SHOIQ}$ and $\mathcal{SROIQ}$. In *Proceedings of the 22th International Joint Conference on Artificial Intelligence - Vol. Two*, IJCAI'11, pages 1039–1044. AAAI Press, 2011.

[102] A. Palladio. I quattro libri dell'architettura. Dominico de Franceschi, Venezia, 1570.

[103] J. Z. Pan and I. Horrocks. Web ontology reasoning with datatype groups. In *Proceedings of the 2nd International Semantic Web Conference (ISWC 2003)*, pages 47–63. Springer, 2003.

[104] J. Z. Pan and I. Horrocks. OWL-Eu: adding customised datatypes into OWL. *Journal of Web Semantics*, 4(1):29–39, 2006.

[105] B. Parsia, E. Sirin, B. Cuenca Grau, E. Ruckhaus, and D. Hewlett. *Cautiously Approaching SWRL*. Elsevier Science, 2005.

[106] A. Robinson. Automatic deduction with hyper-resolution. *International Journal of Computer Mathematics*, 1:227–234, 1965.

[107] R. Rosati. On conjunctive query answering in EL. In *Proceedings of the 2007 International Workshop on Description Logic (DL 2007)*, pages 1–8. CEUR-WS, 2007.

[108] D. F. Santamaria. *A Set-Theoretical Representation for OWL 2 Profiles*. LAP Lambert Academic Publishing, ISBN 978-3-659-68797-6, 2015.

[109] S. Sarawagi. Information extraction. *Foundations and trends in databases*, 1(3):261–377, 2008.

[110] S. Schreibman, R. Siemens, and J. eds. Unsworth. *A Companion to Digital Humanities*. Wiley Publishing, 2004.

[111] J. T. Schwartz, D. Cantone, and E. G. Omodeo. *Computational Logic and Set Theory: Applying Formalized Logic to Analysis*. Texts in Computer Science. Springer-Verlag New York, Inc., 2011.

[112] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz. Pellet: a practical OWL-DL reasoner. *Web Semantics: science, services and agents on the World Wide Web*, 5(2):51–53, 2007.

[113] R. M. Smullyan. *First-order Logic*. Dover books on advanced Math. Dover, 1995.

[114] C. Stadler, J. Lehmann, K. Höffner, and S. Auer. LinkedGeoData: A Core for a Web of Spatial Open Data. Semantic Web Journal, 3 (4), 333-354, 2012.

[115] M.R. Stufano Melone. *Il ruolo della creatività nel processo progettuale in architettura: una approccio cognitivista*. Ph.d. dissertation, University of Pisa, 2011. In Italian.

[116] M.R. Stufano Melone and S. Borgo. Towards an understanding of shapes and types in architecture. *Proceedings of the 1st Workshop on Artificial Intelligence and Design A workshop of the XIV International Conference of the Italian Association for Artificial Intelligence, CEUR-ws.org/Vol-1616 IISSN: 1613-0073*, 2016.

[117] The Apache Software Foundation. Apache stanbol. `https://stanbol.apache.org`, 2013.

[118] The Apache Software Foundation. Jena Ontology API. `https://jena.apache.org`, 2014.

[119] The OpenStreetMap Foundation. OpenStreetMap. `http://www.openstreetmap.org`, 2004.

[120] The Stanford Center for Biomedical Informatics Research. Protégé. `http://webprotege.stanford.edu`, 1999.

[121] The U.S. Board on Geographic Names (BGN). The Geographic Names Information System (GNIS). `https://geonames.usgs.gov/domestic/`, 2014.

[122] Y.M. Vardi. The complexity of relational query languages (extended abstract). In *STOC*, pages 137–146. ACM, 1982.

[123] R. Volz, S. Decker, and I.F. Cruz, editors. *PSSS1 - Practical and Scalable Semantic Systems, Proceedings of the First International Workshop on Practical and Scalable Semantic Systems, Sanibel Island, Florida, USA, October 20, 2003*, volume 89 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2003.

[124] G. Wagner, G. Antoniou, S. Tabet, and H. Boley. The abstract syntax of RuleML - towards a general web rule language framework. *IEEE/WIC/ACM International Conference on Web Intelligence (WI'04)*, pages 628–631, 2004.

[125] R. Wagner-Pacifici, J.W. Mohr, and R.L. Breiger. Ontologies, methodologies, and new uses of big data in the social and cultural sciences. *Big Data & Society*, 2(2):2053951715613810, 2015.

[126] S. Weibel. Metadata: The foundations of resource description. Technical report, Corporation for National Research Initiatives, 1995.

[127] Wikimedia Foundation, Inc. WikiData. `https://wiki.dbpedia.org`, 2012.

[128] R. Willis. Description of the ancient of the monastery of St. Gall in the ninth century. *The Archaelogical Journal*, June 1848.

[129] World Wide Web Consortium. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. `http://www.w3.org/Submission/SWRL/`, 2004.

[130] World Wide Web Consortium. SPARQL Query Language for RDF. `https://www.w3.org/TR/rdf-sparql-query/`, 2008.

[131] World Wide Web Consortium. Ontology Web Language. `http://www.w3.org/2001/sw/wiki/OWL`, 2009.

[132] World Wide Web Consortium. OWL 2 Web Ontology Language Profiles. `https://www.w3.org/TR/owl2-profiles/`, 2012.

[133] World Wide Web Consortium. OWL 2 Web ontology language structural specification and functional-style syntax (second edition). `https://www.w3.org/TR/owl-syntax/`, 2012.

[134] World Wide Web Consortium. OWL Web Ontology Language Semantics and Abstract Syntax. `https://www.w3.org/TR/owl-semantics/syntax.html`, 2012.

[135] World Wide Web Consortium. W3C XML Schema Definition Language (XSD). `https://www.w3.org/TR/xmlschema11-1/`, 2012.

[136] World Wide Web Consortium. Resource Description Framework. `https://www.w3.org/RDF/`, 2014.

[137] Q. Yang, J. You, and Z. Feng. Integrating rules and description logics by circumscription. In Wolfram Burgard and Dan Roth, editors, *AAAI*. AAAI Press, 2011.