Reasoning with Contexts in Description Logics

Szymon Klarman

SIKS Dissertation Series No. 2013-03

The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Graduate School for Information and Knowledge Systems.

Cover design by Nina Gierasimczuk

VRIJE UNIVERSITEIT

Reasoning with Contexts in Description Logics

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad Doctor aan
de Vrije Universiteit Amsterdam,
op gezag van de rector magnificus
prof.dr. L.M. Bouter,
in het openbaar te verdedigen
ten overstaan van de promotiecommissie
van de Faculteit der Exacte Wetenschappen
op woensdag 27 februari 2013 om 15.45 uur
in de aula van de universiteit,
De Boelelaan 1105

door

Szymon Klarman

geboren te Lodz, Polen

promotor:	prof.dr. F.A.H. van Harmelen
copromotor:	dr. K.S. Schlobach

promotiecommissie: prof.dr. Wan Fokkink
dr. Chiara Ghidini
prof.dr. Carsten Lutz
prof.dr. Heiner Stuckenschmidt
prof.dr. Yde Venema

CONTENTS

ACKNOWLEDGMENTS

This thesis marks a successful end to the challenging and curious period of my academic life — the period of being a PhD student in Amsterdam. There are many people whom I owe my gratitude for supporting me over that time in different ways.

First of all, I wish to thank my supervisors, Stefan Schlobach and Frank van Harmelen, for offering me a PhD position to start with, securing it in the difficult times of the economic crisis, and for providing me with all the freedom and resources I needed to pursue my own academic interests. I am particularly grateful to Stefan for being a committed coach with a great sense of empathy and an intuition for giving me the right motivational pushes exactly when needed. Frank has been a true role model as the head of our group. I feel very lucky to have been able to watch him working and to learn what it takes to nourish a good research environment. To both of them I also owe words of appreciation for having patiently trained me in the intricate art of saying precisely what one wants to say and nothing different.

I would like to thank the members of our Knowledge Representation and Reasoning group for generating a cheerful atmosphere during the regular working hours and... a charmingly radical one long after them. I thank Annette ten Teije, alongside my supervisors, for guiding me through the excellent didactic experience we shared while teaching the Automated Reasoning course over the four consecutive years of my PhD studies. I am very grateful to Paul Groth for paying a good deal of attention to my research and getting me interested in his own, regardless of how remote those two areas had seemed to be at the first sight. This was an important lesson for me. My special thanks are also due to Elly Lammers, who was incredibly helpful whenever I needed to navigate

through the labyrinths of the university administration.

I am greatly indebted to Carsten Lutz and Luciano Serafini, along with their colleagues from the TAI group at the University of Bremen and the DKM group at the FBK Trento, for inviting me and hosting during my short research visits over the last few years. All those occasions were extremely enjoyable and inspiring, and had a decisive impact on the direction of my work.

Above all, I wish to thank my fellow PhD student from the University of Bremen, Víctor Gutiérrez-Basulto, who has been my longtime co-author and a fantastic sparring partner in our numerous discussions about logic, complexity and research. If I were to point out a single person without whom completing this thesis would not have been possible, it would have to be him. Thanks, Víctor, and good luck with finishing your own *opus magnum*!

As far as this thesis is concerned, I thank the members of my doctoral committee for reading it and contributing their valuable comments, which greatly helped me improve the presented material. Naturally, I take the full blame for any errors or omissions in case such can be still found here. I also bow to Nina Gierasimczuk for designing the splendid cover for this book, to Laurens Rietveld for translating the English summary of the thesis to Dutch, and to my fabulous paranymphs, Kathrin Dentler and Krystyna Milian, for their assistance in organizing the defence ceremony.

One of the peculiarities of PhD life is that it notoriously resists a proper demarcation of its professional and personal aspects. As this has been surely true in my case I would like to express here my final and deepest gratitude to my family and friends — those in the Netherlands, in Poland, and in a few other random locations. Following this long route without them delivering constant support and regular distractions would be much more difficult or — honestly — just not worth the effort. To all of them I have simply this to say: *Wasze zdrowie*! *Cheers*!

Amsterdam,
November, 2012

Szymon Klarman

CHAPTER
ONE

INTRODUCTION

1.1 Motivation

The Semantic Web is an initiative of turning the Web into a global, distributed repository of machine-understandable information. The *machine-understandability*, or equivalently, the *semantic* aspect of this project is achieved by the use of collectively agreed upon, well-defined knowledge representation formalisms for publishing information on the Web, in particular: Resource Description Framework (RDF), RDF Schema (RDFS), and the family of Web Ontology Languages (OWL). In the popular illustration of the Semantic Web architecture, known as the Semantic Web layer cake (see Figure 1.1), these three formalisms are laid on top of each other, and further topped with the layer simply referred to as *logic*. The positioning of this latter layer signals two messages: a descriptive and a normative one. The first one says that RDF(S) and, especially, OWL are equipped with sufficiently rich formal semantics and precisely defined syntaxes, so that managing knowledge expressed in those formalisms can be seen as a task of a genuinely logical nature, in the strict, traditional sense of mathematical logic. Indeed, OWL languages and considerable parts of RDF(S) are simply notational variants of specific Description Logics (DLs), thereby close relatives of modal logics, and consequently, fragments of first-order logic. For a logician this is quite a fortunate circumstance, as he is justified to abstract from the often verbose notation in which pieces of knowledge are actually exposed on the Web, and instead, consider them as logic formulas. Moreover, he can directly apply his home perspective and expertise to many problems pertinent to

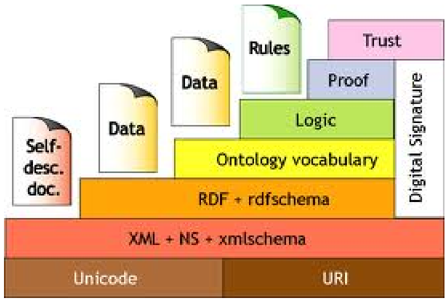

Figure 1.1: Semantic Web layer cake.

information management on the Semantic Web. However, the firm logic foundations shared by RDF(S) and OWL are not themselves enough to regulate the semantic and algorithmic standards for all knowledge-intensive tasks to which the Semantic Web is and might be employed. Different application scenarios reveal diverse knowledge-related phenomena and problems which cannot be resolved within the formal limits of the underlying representation languages. The normative message, implied by the layer cake, states therefore that in such cases, an additional layer of logic must be imposed on the Semantic Web architecture. In this context, *logic* should be understood in a broader sense, as the requirement of adhering to certain rational principles, possibly articulated as a coherent formal system.

The *contextuality* of knowledge is a well-known epistemological phenomenon, of a notorious reputation in the field of symbolic AI. In short, it is the problem of *the general inability of determining the meaning of a piece of information, or verifying its validity, without assuming the context in which this information has been stated, and thereby, in which it should be interpreted*. Regardless of the particular field of interest, this constant observation has been motivating diverse formal mechanisms for explicit handling of contexts in knowledge-based systems. As it turns out, the phenomenon lies also at the heart of numerous knowledge representation problems encountered in the practice and theory of the Semantic Web, and as such it is the core subject of the presented thesis. It is addressed here from a logician's viewpoint, or more specifically, from the perspective of the DL-based knowledge representation paradigm. Let us illustrate the problem with a simple example. Consider the following statements expressed in the DL or OWL/RDF(S) syntax:

$$\begin{array}{rl} PhDstudent \sqsubseteq Student & \langle PhDstudent \texttt{ rdfs:subClassOf } Student\rangle \\ Student \sqcap Staff \sqsubseteq \bot & \langle Student \texttt{ owl:disjointWith } Staff\rangle \\ PhDstudent(john) & \langle john \texttt{ rdf:type } PhDstudent\rangle \end{array}$$

These state formally that all PhD students are students, that no one can be a student and a staff member at the same time, and that John is in fact a PhD student. Suppose now that this knowledge is exposed on, and subsequently collected from the Web, and let us ask: is it justified to infer that John is not a staff member? The answer to this question obviously cannot be decided without taking into account the respective contexts in which the expressions are stated. In certain countries or particular universities PhD students are indeed considered regular students, but in others, they are full-time employees, and therefore staff members. If the first two axioms assume such two different contexts, then the inference should fail. Also, John might have been a PhD student at some earlier time or at a different institution, while at the time of querying he is a legitimate staff member. If any of such circumstances apply, the inference again does not hold. Only when all the statements refer to the same institution, at the same place and time, is the inference justified.

An orthodox logician could rebut the issue by resorting to the standard DL nomenclature and claiming that the inference holds whenever the two axioms are in the same knowledge base (resp. OWL ontology, RDF graph). Whether or not this is the case is a question going beyond logic. Such a response, however, dismisses the problem rather than solving it. On the one hand, including the axioms in the same knowledge base could be already taken as a sign that the same context for them is implicitly assumed by the author. One should therefore rightfully argue, that involving an explicit, declarative contextualization mechanism would not alter the formal or conceptual character of the representation but merely render it more transparent and easier to manage. On the other hand, the classical notion of knowledge base, originating from the early era of stand-alone DL systems, hardly lives up to the standards of the knowledge representation practice on the Semantic Web. Information on the Web is spread across many physical locations, and is generated, revised, combined and interlinked in a non-centralized manner by many independent parties. Very often, the fact that two axioms are published within the same or different datasets might be largely coincidental, and should not be treated as a decisive factor for whether such axioms can be employed within the same inference process or not.

Driven by this motivation, our goal is to develop a generic logic-based framework, compatible with the paradigm of DLs, for explicit representation and reasoning with contexts and contextual information in the Semantic Web environment. In other words, we extend the logic layer of the Semantic Web architecture with a mechanism for dealing with different problems related to the phenomenon of contextuality of knowledge. In doing so we set ourselves and propose answers to the following research questions:

- What theory of contexts is adequate for integration with the DL-based knowledge representation paradigm?
- How should such theory be technically reinterpreted and implemented on the grounds of DL semantics, syntax, and the general philosophy and methodology of DL-based knowledge representation?
- What are the formal properties of the resulting framework (the expressiveness, relationships to other known formalisms, computational complexity of reasoning tasks, etc.)?
- How and to what extent can such a framework be applied in and adapted to different application scenarios, motivated by use-cases and problems observed in the practice of the Semantic Web?

1.2 Contributions and content

The key concept which we technically explicate and utilize in this thesis is that of *context*. What is a context? In the full generality, *a context is the whole of relevant information about the situation in which a certain piece of knowledge is true, necessary for a correct and complete interpretation of this knowledge*. Unfortunately, most attempts of making this formulation more specific fall victim of implicit application biases, as 'context' is likely one of the most context-dependent terms used in science, and particularly in computer science and AI. In order to avoid such arbitrariness, and keep our approach maximally generic, we embrace an ingenious proposal of John McCarthy, which is the cornerstone of his theory of formalizing contexts in AI. A context, according to McCarthy, is simply a formal object which can be used in the place of the constant $\boldsymbol{c}$ in assertions of the form:

$$ist(\boldsymbol{c}, PhDstudent \sqsubseteq Student)$$
$$ist(\boldsymbol{c}, Student \sqcap Staff \sqsubseteq \bot)$$
$$ist(\boldsymbol{c}, PhDstudent(john))$$

where the predicate *ist* reads *is-true-in*. This description is seemingly vacuous, but in fact highly non-trivial. What is crucial about it is that it does not adjudicate what essentially context is, as a deeply involved philosophical concept, but instead directly suggests how to operationalize it as a first-class citizen in a formal system. In the above case, we learn that all axioms presented earlier hold in the same context $\boldsymbol{c}$. Further, we can describe this context on the meta-level, e.g. by stating that:

$$\boldsymbol{University}(\boldsymbol{c})$$
$$\boldsymbol{locatedIn}(\boldsymbol{c}, \boldsymbol{poland})$$

Such a conceptual leap — introduction of a primitive term — is something that mathematicians and other formal scientists must usually do to avoid the trap of descending into an infinite definition chain. For the Semantic Web community this is also a familiar manoeuvre. For those doubting, we propose an exercise of defining terms 'web resource' or 'OWL individual'. There are obviously numerous common-sense intuitions associated with the notion of context. Although these must be treated with high reserve, as more likely leading to confusion than clarity, we should nevertheless try to provide at least a rough guideline to our suggested understanding of context. Let us do it by analogy to the notion of DL/OWL individual. Individuals are first-order objects said to be representing things in the real world. Their descriptions in the ontology language are then meant to represent the real-world properties of those things. But what are those things? This naturally depends on the application at hand and the intentions of the ontology modeler. Sometimes they are physical objects, other times, abstract entities, yet in different cases, just handy, conventional reifications of some complex occurrences in the modeled domain. What eventually matters is only that a set of assertions about a designated individual, being the formal abstraction of some fraction or aspect of the real world, leads to a sufficiently intelligent behavior of the system reasoning over it. In our take, a context is an alike formal, instrumental entity, whose meaning manifests itself only in the practical function it plays in a particular knowledge representation system. Intuitively, it is an abstraction of a situation in which some domain knowledge is said to hold. Contextual information is then a description of the properties of this situation, relevant for proper interpretation of the knowledge. What counts as a relevant situation description is again a purely application-driven choice, made by the knowledge engineer. In different use-cases, which we study in this thesis, we identify contexts and contextual information, respectively, with different data sources and meta-level descriptions of their content or origin (Chapter 4), different stages of data-oriented computations and descriptions of their history (Chapter 5), or time points and their descriptions in temporal vocabularies (e.g. calendars) (Chapter 6). There might be naturally many others. The AI and Semantic Web literature mentions also as possible types of relevant contextual information: propositional attitude, point of view, geographic location, culture, topic, granularity, justification, and others [Len98, GM03, GMF04].

The main conceptual contribution of the presented thesis is a reinterpretation of the theory of contexts, outlined above, in terms of two-dimensional, two-sorted combinations of DL-based representations. Consider a DL language $\mathcal{L}_O$ in which domain knowledge is expressed, such as captured by axioms $PhDstudent \sqsubseteq Student$ or $PhDstudent(john)$. The standard model-theoretic semantics of $\mathcal{L}_O$ is defined through interpretations of the form $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a domain of individuals and $\cdot^{\mathcal{I}}$ is an interpretation function, which fixes the meaning of the vocabulary of $\mathcal{L}_O$ by mapping all atoms (concepts, roles, individual names) on the domain $\Delta^{\mathcal{I}}$. Such interpretations are obviously the basis for deciding any reasoning problems over the formulas of $\mathcal{L}_O$. We advocate a shift from this perspective towards two-dimensional possible world semantics given via interpretations of the form $\mathfrak{M} = (\mathfrak{C}, \{\mathcal{I}(i)\}_{i \in \mathfrak{C}})$, where $\mathfrak{C}$ is a domain of contexts, and for every $i \in \mathfrak{C}$, $\mathcal{I}(i)$ is a usual DL interpretation of $\mathcal{L}_O$ in the context i. Consequently, $\mathfrak{C}$ is considered to be the second dimension in the semantics, with respect to which the meaning of $\mathcal{L}_O$, and thus all reasoning problems over $\mathcal{L}_O$ are defined. For instance, in our running example, it will logically follow that $\neg Staff(john)$ holds in $\boldsymbol{c}$. Additionally, we introduce another language $\mathcal{L}_C$ for describing contexts *per se*, as in $\boldsymbol{University}(\boldsymbol{c})$ or $\boldsymbol{locatedIn}(\boldsymbol{c}, \boldsymbol{poland})$, whose meaning is fixed by means of a separate interpretation function, mapping the context vocabulary onto $\mathfrak{C}$. All through this thesis we refer to $\mathcal{L}_O$ as the object or domain language, and to $\mathcal{L}_C$ as the context or metalanguage, modulo the differences in definitions of $\mathcal{L}_O$ and $\mathcal{L}_C$ in the specific scenarios considered. Analogically, from the ontology modeling perspective, we talk about the object- or domain-level and the context- or meta-level of the representation, to denote the sets of axioms expressed in the respective languages.

We argue that this general setup brings a unifying and highly explicatory perspective on a number of diverse problems related to contextuality of knowledge expressed in DL-based formalisms. It is generic and conceptually very natural — in fact, not rarely implemented in practical Semantic Web applications,[1] although without proper understanding of its formal aspects. This thesis advances the state of the art precisely on this frontier. Through our investigations, we lay down broad logical foundations for representing knowledge and reasoning in two-level, object-context DL-based systems.

The content of this thesis is divided into five major chapters, whose particular contributions are highlighted in the following paragraphs.

Chapter 2: We introduce the necessary technical background, covering DLs and relevant reasoning problems over DL-based representations. Next, we lay down the conceptual foundations for our proposed context framework. We carefully outline how this framework is derived from McCarthy's theory of formalizing contexts in AI, and further, we ground it in two-dimensional semantics by identifying context entities with possible worlds comprising the second dimension included in DL interpretations. Finally, we give a brief overview of the state of the art in the research areas most related to the subject of our thesis.

Chapter 3: We define a novel family of two-dimensional, two-sorted DLs incorporating our proposed context framework. We evaluate the expressiveness of the formalisms by establishing precise relationships to better known product-like combinations of DLs with modal logics, in particular $(\mathbf{K}_n)_{\mathcal{L}}$ and $\mathbf{S5}_{\mathcal{L}}$. We demonstrate potential applicability of the proposed logics to such diverse problems as modeling inherently contextualized knowledge or expressing interoperability constraints between DL ontologies. Finally, we study the basic decision problems and deliver corresponding complexity results, ranging from ExpTime-completeness, in certain fragments of smaller expressiveness, up to 2ExpTime-completeness, in the most expressive formalisms in the family.

Chapter 4: We adapt our framework to address the well-known problem of ontology integration, and we introduce a novel task of metaknowledge-driven selection and querying of data. In this scenario, the context entities are associated with ontology names, while the context language is used for describing arbitrary meta-information about those ontologies, as witnessed in several real-life use-cases. This meta-information is used to identify relevant portions of data to be queried over with standard ontology query languages. As in the remaining chapters of the thesis, the logical interaction between the object- and

[1] See e.g., NCBO BioPortal project `http://bioportal.bioontology.org/`, Jena MultiModel `http://www.swed.org.uk/`, applications of named RDF graphs `http://www.w3.org/2004/03/trix/#apps`, annotated RDF data architecture `http://ardfsql.blogspot.nl/`.

meta-level representations is considerably restricted here to guarantee good computational behavior of the resulting systems. We highlight the ease of the addressed tasks under the proposed approach, and report on a case study of aligning different versions of Wordnet ontologies.

Chapter 5: Data provenance is the history of derivation of a data artifact from its original sources. A provenance record can be seen as a directed graph with nodes representing datasets involved in the derivation and edges with particular derivation steps. In recent years, the use and management of such records have become important and challenging tasks for the Semantic Web community. In this chapter, we apply the context framework to model this problem. We interpret the nodes of derivation graphs as contexts, express datasets in the object language, and encode the corresponding provenance information in the context language. Further, we define provenance specification logic, based on a combination of Propositional Dynamic Logic with ontology query languages, and use it for verifying and querying thus represented provenance records. We validate our proposal against the test queries of The First Provenance Challenge, and analyze the computational properties of the logic.

Chapter 6: In this chapter, we apply our framework in a more remote scenario of representing and querying temporal data in DLs. Here, contexts correspond to time instants on a time line. Unlike in usual temporal databases, we advocate the use of a metalanguage for describing a variety of practical temporal concepts over the time domain, which can be utilized on the level of queries and data annotations. This, as we claim, brings the approach closer to the Semantic Web practices of dealing with temporal information, involving popular time ontologies. We introduce a generic mechanism of defining corresponding temporal query languages, based on combinations of linear temporal logics and ontology query languages. We elaborate on the practicality of our approach by proposing special restrictions that, as we demonstrate, render temporal querying computationally cheap and relatively straightforward to implement.

Notably, the four major chapters, 3–6, differ in the adopted research methodology. Chapter 3 is largely theory-driven, and contains technically most challenging results. Its aim is to define a monolithic logic with desired properties, in the traditional fashion of constructing a logic-based knowledge representation formalism. Chapters 4–6, on the other hand, are more practical and follow the spirit of logic engineering. There, we adapt our context framework to some real-life problems encountered in the practice of the Semantic Web and then develop lightweight, possibly hybrid, logic tools and techniques, having a reasonably low implementation overhead, which can be efficiently applied in addressing those problems. To ease the reading of the thesis, each of those four chapters are preceded with a short summary, highlighting basic connections between the framework and the contributions proposed in the chapter. Appendix A contains the proofs of the complexity results provided in Chapter 3. Finally, in Chapter 7 we conclude the work with a discussion and an outlook on open problems.

1.3 Sources of chapters

This thesis includes the work presented in the following of the author's publications and technical reports:

Section 2.3 and Chapter 3:

- Szymon Klarman and Víctor Gutiérrez-Basulto, *$\mathcal{ALC}_{\mathcal{ALC}}$: A Context Description Logic*, Proceedings of the 12th European Conference on Logics in Artificial Intelligence (JELIA-10), LNCS 6341, Springer Berlin Heidelberg, 2010.
- Szymon Klarman and Víctor Gutiérrez-Basulto, *Two-Dimensional Description Logics for Context-Based Semantic Interoperability*, Proceedings of the 25th Conference on Artificial Intelligence (AAAI-11), AAAI Press, 2011.
- Szymon Klarman and Víctor Gutiérrez-Basulto, *Two-Dimensional Description Logics of Context*, Proceedings of the 24th International Workshop on Description Logics (DL-11), CEUR Workshop Proceedings 745, 2011.
- Szymon Klarman and Víctor Gutiérrez-Basulto, *Description Logics of Context*, Journal of Logic and Computation, accepted for publication.

Chapter 4:

- Szymon Klarman and Víctor Gutiérrez-Basulto, *Two-Dimensional Description Logics for Context-Based Semantic Interoperability*, Proceedings of the 25th Conference on Artificial Intelligence (AAAI-11), AAAI Press, 2011.

- Paul Groth, Szymon Klarman, Stefan Schlobach and Jacco van Ossenbruggen, *Metadata-Driven Selection and Integration of Object-Level Knowledge*, Technical Report, VU University Amsterdam, 2011.

Chapter 5:

- Szymon Klarman, Stefan Schlobach and Luciano Serafini, *Formal Verification of Data Provenance Records*, Proceedings of the 11th International Semantic Web Conference (ISWC-12), LNCS 7649, Springer Berlin Heidelberg, 2012.

Chapter 6:

- Víctor Gutiérrez-Basulto and Szymon Klarman, *Towards a Unifying Approach to Representing and Querying Temporal Data in Description Logics*, Proceedings of the 6th International Conference on Web Reasoning and Rule Systems (RR-12), LNCS 7497, Springer Berlin Heidelberg, 2012.

CHAPTER

TWO

PRELIMINARIES

2.1 Description Logics

Description Logics (DLs) are a popular family of knowledge representation formalisms, intended for modeling and reasoning with terminological and assertional (factual) knowledge about an application domain [BCM+03]. In the previous section, we used as examples some well-formed DL axioms:

$$PhDstudent \sqsubseteq Student$$
$$Student \sqcap Staff \sqsubseteq \bot$$
$$PhDstudent(john)$$

where the first two express terminological constraints, and the last one, assertional information about some object in the domain. All through this thesis we use the standard nomenclature and notation for the syntax and semantics of DLs. A DL language $\mathcal{L}$ is defined over a vocabulary $\Sigma = (N_C, N_R, N_I)$, where $N_C = \{A, B, \ldots\}$ is a countably infinite set of *concept names*, $N_R = \{r, s, \ldots\}$ a set of *role names*, and $N_I = \{a, b, \ldots\}$ a set of *individual names*. The semantics of $\mathcal{L}$ is given through *interpretations* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, where $\Delta^\mathcal{I}$ is a non-empty domain of individuals and $\cdot^\mathcal{I}$ is an interpretation function, which fixes the meaning of the vocabulary by mapping $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$, for every $A \in N_C$, $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$, for every $r \in N_R$, and $a^\mathcal{I} \in \Delta^\mathcal{I}$, for every $a \in N_I$. The grammar of $\mathcal{L}$ is defined relative to the given DL. Particular logics in the DL family allow different collections of *constructors* for expressing complex concepts, roles and axioms. The interpretation function is inductively extended over those complex expressions

Concept and role constructors			
(1)	$\top$	$\Delta^{\mathcal{I}}$	top
(2)	$\bot$	$\emptyset$	bottom
(3)	$C \sqcap D$	$\{x \in \Delta^{\mathcal{I}} \mid x \in C^{\mathcal{I}} \cap D^{\mathcal{I}}\}$	intersection
(4)	$C \sqcup D$	$\{x \in \Delta^{\mathcal{I}} \mid x \in C^{\mathcal{I}} \cup D^{\mathcal{I}}\}$	union
(5)	$\exists r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y : (x, y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$	exist. restriction
(6)	$\forall r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \forall y : (x, y) \in r^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$	univ. restriction
(7)	$\neg C$	$\{x \in \Delta^{\mathcal{I}} \mid x \notin C^{\mathcal{I}}\}$	complement
(8)	$\{a\}$	$\{a^{\mathcal{I}}\}$	nominal
(9)	r^-	$\{(x, y) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (y, x) \in r^{\mathcal{I}}\}$	role inverse
Axioms			
(10)	$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$	concept assertion
(11)	$r(a, b)$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$	role assertion
(12)	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$	concept inclusion
(13)	$r \sqsubseteq s$	$r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$	role inclusion
(14)	$\text{dom}(r) \sqsubseteq C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y : (x, y) \in r^{\mathcal{I}}\} \subseteq C^{\mathcal{I}}$	domain restriction
(15)	$\text{ran}(r) \sqsubseteq C$	$\{y \in \Delta^{\mathcal{I}} \mid \exists x : (x, y) \in r^{\mathcal{I}}\} \subseteq C^{\mathcal{I}}$	range restriction

Table 2.1: DL constructors and their semantics.

according to the fixed conditions associated with each constructor. Table 2.1 presents an overview of the syntax and semantics of the constructors referred to in the technical discussions and examples used in this thesis. In the table, C, D are (possibly complex) concepts, $r, s \in N_R$, and $a, b \in N_I$. Different DLs are named with conventional acronyms, indicating the sets of constructors allowed in the logic. In this thesis, we address in more detail the following DLs:

- $\mathcal{ALC}$, defined by constructors (1)-(7), (10)-(12),
- $\mathcal{ALCO}$, defined by constructors of $\mathcal{ALC}$ and (8),
- $\mathcal{ALCI}$, defined by constructors of $\mathcal{ALC}$ and (9),
- $\mathcal{SHI}$, defined by constructors of $\mathcal{ALCI}$, (13) and transitive roles, i.e. a distinguished subset of role names $N_{R+} \subseteq N_R$, such that every $r \in N_{R+}$ must be interpreted as a transitive relation,
- $\mathcal{SHIO}$, defined by constructors of $\mathcal{SHI}$ and (8),

- $\mathcal{EL}^{++}$, defined by constructors (1)-(3), (5), (8), (10)-(15), role inclusions of type $r_1 \circ \ldots \circ r_n \sqsubseteq s$, where $(r_1 \circ \ldots \circ r_n)^{\mathcal{I}} = r_1^{\mathcal{I}} \circ \ldots \circ r_n^{\mathcal{I}}$, and additional extras[1] and syntactic restrictions [BBL08].

Additionally, we sometimes refer to $\mathcal{EL}$, defined by (1), (3), (5), (10)-(12), $\mathcal{SHIQ}$, based on $\mathcal{SHI}$ with qualified role restrictions ($\mathcal{Q}$), and $\mathcal{SROIQ}$, which is one of the most expressive DLs in popular use, allowing for all constructors introduced above and more [HKS06]. We also mention the DL-*Lite* family of lightweight DLs, whose design is deeply motivated by database applications [ACKZ09] (see below). This selection of languages is motivated mainly by two reasons: their different logical and computational properties, which lead to different characterizations of the resulting context framework, and their specific, typical applications. Given two DLs, $\mathcal{L}_1$ and $\mathcal{L}_2$, we write $\mathcal{L}_1 \preceq \mathcal{L}_2$ to mark that $\mathcal{L}_2$ is at least as expressive as $\mathcal{L}_1$. In many cases, we refer generically to an "arbitrary DL language", meaning that the claims made in that context apply equally to all DLs introduced here, and possibly others not explicitly addressed in the thesis.

A DL *knowledge base* $\mathcal{K}$ (sometimes also marked as $\mathcal{O}$) is a finite set of DL axioms, of the syntax permitted in a given DL language. Knowledge bases are conventionally split into two components: the (terminological) TBox, which formalizes the relationships between roles and concepts, and the (assertional) ABox, which includes descriptions of domain individuals. For a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, the *TBox* $\mathcal{T}$ is a finite set of concept and role inclusions (12)-(13). Whenever both $C \sqsubseteq D$ and $D \sqsubseteq C$ (resp. $r \sqsubseteq s$ and $s \sqsubseteq r$) belong to $\mathcal{T}$ one can abbreviate them into a single equivalence axiom $C \equiv D$ (resp. $r \equiv s$). The *ABox* $\mathcal{A}$ is a finite set of assertions of types $C(a)$ and $r(a, b)$.

The term *ontology* is most of times used synonymously to knowledge base. However, in some scenarios regarding ontology-based data access, which we touch upon in Chapter 6, it has a narrower meaning. The *ontology-based data access* (OBDA) is a paradigm of managing data in presence of background knowledge, represented as a formal ontology, enabling convenient query answering over *incomplete* data. In recent years, special attention has been given to ontologies based on DL languages. In this context, as it is commonly used, the term ontology denotes just the TBox of a knowledge base, whereas the ABox is considered the data, accessed through this ontology. Moreover, without loss of generality, it is assumed that ABox consists exclusively of assertions of types $A(a)$ and $r(a, b)$, where $A \in N_C$. A considerable amount of research has been

[1] For the brevity of presentation these are omitted here but are accounted for in the proofs of the relevant results.

devoted to the problem of query answering in DLs, focusing predominantly on conjunctive queries, which we discuss in the next section. This has led to establishing a clear picture of the computational complexity of query answering, and to the development of algorithmic approaches. The study has been focused on two major lines: 1) utilization of classical DLs with high expressive power, where the complexity of query answering typically turns out too high for practical applications [GHLS08]; 2) development of DLs allowing efficient query answering over large amounts of data. Calvanese et.al. [ACKZ09] introduced the DL-*Lite* family of DLs, for which efficient OBDA can be achieved by reduction to query answering in relational database management systems (RDBMSs).

Another prominent application of DLs, which is of central interest to this work, is their use as the logical underpinning of the Semantic Web knowledge representation formalisms. Notably, the OWL languages are defined entirely as notational variants of designated DLs [HPsH03]. In particular, OWL 2 DL corresponds to the DL $\mathcal{SROIQ}$, OWL 2 EL to $\mathcal{EL}^{++}$ and OWL 2 QL to the DL-*Lite* family [MGH^{+}09]. Further, considerable fragments of RDF(S) have been also found out to be rewritable into DLs [DBH07]. Table 2.2 gives a rough overview of the correspondence between the syntaxes of DLs and the usual OWL/RDF(S) notation. Based on this relationship, most claims regarding DLs made in this thesis, naturally carry over to OWL/RDF(S). In this sense, contributions presented in this thesis apply to the Semantic Web environment.

2.2 Reasoning problems

The methodology of DL-based knowledge representation comes with a number of well-defined reasoning problems, underlying practical reasoning services most commonly employed in managing DL knowledge bases. All these problems are grounded on the classical entailment relation.

A DL axiom α is *satisfied* in an interpretation $\mathcal{I}$, written $\mathcal{I} \models \alpha$, whenever the semantic condition associated with the axiom is satisfied in $\mathcal{I}$ (see Table 2.1). An interpretation $\mathcal{I}$ is a model of a knowledge base $\mathcal{K}$ (resp. TBox $\mathcal{T}$, ABox $\mathcal{A}$), written $\mathcal{I} \models \mathcal{K}$ (resp. $\mathcal{I} \models \mathcal{T}$, $\mathcal{I} \models \mathcal{A}$) *iff* it satisfies all axioms from $\mathcal{K}$ (resp. $\mathcal{T}$, $\mathcal{A}$). An axiom α is *entailed* by a knowledge base $\mathcal{K}$ *iff* α is satisfied in every model of $\mathcal{K}$.

The basic reasoning problem for most DLs is *satisfiability* checking, i.e. deciding whether a knowledge base has a model. Whenever this is the case, the knowledge base is called satisfiable, and unsatisfiable otherwise. In less

DL Syntax	OWL/RDF(S) syntax
Concept/Class constructors	
$\top$	`owl:Thing`
$\neg$	`owl:complementOf`
$\sqcap$	`owl:intersectionOf`
$\sqcup$	`owl:unionOf`
$\exists$	`owl:someValuesFrom`
$\forall$	`owl:allValuesFrom`
Axioms	
$C(a)$	$\langle a$ `rdf:type` $C\rangle$
$r(a,b)$	$\langle a\ r\ b\rangle$
$C \sqsubseteq D$	$\langle C$ `rdfs:subClassOf` $D\rangle$
$C \equiv D$	$\langle C$ `owl:equivalentClass` $D\rangle$
$r \sqsubseteq s$	$\langle r$ `rdfs:subPropertyOf` $s\rangle$
$r \equiv s$	$\langle r$ `owl:equivalentProperty` $s\rangle$
$\{a\} \equiv \{b\}$	$\langle a$ `owl:sameAs` $b\rangle$
$\text{dom}(r) \sqsubseteq C$	$\langle r$ `rdfs:domain` $C\rangle$
$\text{ran}(r) \sqsubseteq C$	$\langle r$ `rdfs:range` $C\rangle$

Table 2.2: DL vs. OWL/RDF(S) syntax.

expressive languages, such as $\mathcal{EL}$, where due to the lack of negation satisfiability becomes trivial, or at least not interesting from the knowledge engineering perspective, the main reasoning task is (concept) *subsumption*, i.e. deciding whether for two concepts C, D it is the case that the knowledge base entails the inclusion $C \sqsubseteq D$. Obviously, in more expressive languages, this problem is directly reducible to satisfiability. The computational complexity of the two problems depends on the expressiveness of the given DL, and ranges (among others) over the following classes:

- ExpTime-complete for satisfiability/subsumption in $\mathcal{L}$, for $\mathcal{ALC} \preceq \mathcal{L} \preceq \mathcal{SHIO}$ [BCM+03, HM04],
- N2ExpTime-complete for satisfiability/subsumption in $\mathcal{SROIQ}$ [Kaz08],
- PTime-complete for subsumption in $\mathcal{EL}, \mathcal{EL}^{++}$ [BBL05, BBL08].

Whenever we generically refer to the complexity of reasoning in a given DL language, we implicitly assume the reasoning problem considered central in this DL, as indicated above. All the standard reasoning tasks in dedicated DLs are performed by off-the-shelf DL reasoners, such as Pellet [SPG+07], FaCT++ [TH06], KAON2 [Mot06], ELK [KKS12], and others.[2]

Another type of reasoning services of great practical importance is querying. The basic form of this reasoning, known as *instance retrieval*, is defined as follows: given a concept C and an individual name a (resp. a role r and individuals a, b) decide whether $\mathcal{K} \models C(a)$ (resp. $\mathcal{K} \models r(a, b)$). In this form, the complexity of the problem remains the same as for other basic reasoning tasks in the respective DLs. For many applications, however, instance retrieval is too restrictive to offer sufficiently flexible access to ABox information, hence more expressive query languages are often considered. One of the most popular classes of first-order queries studied in the context of DLs are conjunctive queries [Ros, GHLS08], which we extensively use in Chapters 4–6.

Let N_V be a countably infinite set of variables. A *conjunctive query* (CQ) over a DL vocabulary Σ is a first-order formula $\exists \vec{y}.\varphi(\vec{x}, \vec{y})$, where $\vec{x}, \vec{y}$ are sequences of variables. The sequence $\vec{x}$ denotes the free, *answer variables* in the query, while $\vec{y}$ denotes the quantified ones. The formula φ is a conjunction of atoms over Σ of the form $A(u), r(u, v)$, where $u, v \in N_V \cup N_I$ are called terms. By default, we assume that CQs are always expressed over the same DL vocabulary as the knowledge bases over which they are used. The following is an example of a CQ retrieving all people having the same parent as individual *john*:

$$q(x) ::= \exists y.(\mathit{Person}(x) \land \mathit{hasParent}(x, y) \land \mathit{hasParent}(\mathit{john}, y))$$

Let $\mathcal{I}$ be an interpretation and $q(\vec{x})$ a CQ with the answer variables $\vec{x} = x_1, \ldots, x_k$. By $\mathrm{term}(q)$ we denote the set of all terms occurring in q. For a sequence $\vec{a} = a_1, \ldots, a_k \in N_I$, an $\vec{a}$-match to a query $q(\vec{x})$ in $\mathcal{I}$ is a mapping $\mu : \mathrm{term}(q) \mapsto \Delta^{\mathcal{I}}$, such that $\mu(x_i) = {a_i}^{\mathcal{I}}$, for every $1 \leq i \leq k$, $\mu(a) = a^{\mathcal{I}}$, for every $a \in N_I \cap \mathrm{term}(q)$, for every $A(u)$ in q it is the case that $\mu(u) \in A^{\mathcal{I}}$ and for every $r(u, v)$ in q it is the case that $(\mu(u), \mu(v)) \in r^{\mathcal{I}}$. We write $\mathcal{I} \models q[\vec{a}]$ whenever there exists an $\vec{a}$-match to q in $\mathcal{I}$, and $\mathcal{T}, \mathcal{A} \models q[\vec{a}]$ whenever there exists an $\vec{a}$-match to q in every model of $\mathcal{T}$ and $\mathcal{A}$. In the latter case $\vec{a}$ is called a *certain answer* to q w.r.t. $\mathcal{T}$ and $\mathcal{A}$.

The problem of finding all certain answers is known as *query answering*, whereas deciding whether some $\vec{a}$ is such an answer as *query entailment*. Clearly, the former problem is reducible to the latter. The complexity of query entailment is usually measured relative to two parameters: the combined com-

[2] See http://www.cs.man.ac.uk/~sattler/reasoners.html.

plexity and the data complexity. The combined complexity is the complexity with respect to the total size of the knowledge base together with the query. The data complexity reflects the complexity of the problem only with respect to the size of the ABox (data). In this thesis, we make several references to the combined complexity of CQ entailment in DLs. Although particular complexity classes are not of primary importance here, we recall the following results:

- EXPTIME-complete for CQ entailment in $\mathcal{L}$, for $\mathcal{ALC} \preceq \mathcal{L} \preceq \mathcal{SHQ}$ [Lut08],
- 2EXPTIME-complete for CQ entailment in $\mathcal{L}$, for $\mathcal{ALCI} \preceq \mathcal{L} \preceq \mathcal{SHIQ}$ [GHLS08, Lut08].

One of the major reasons for the interest behind the class of CQs is their reasonable balance regarding the complexity-expressiveness trade-off. It has been noted that more expressive languages become easily undecidable, for instance, when negation over atoms of CQs is allowed [Ros]. One of the largest classes of queries that has been shown decidable over DLs, are the unions of conjunctive queries (UCQs) defined as formulas of the form:

$$q \ ::= \ q_1 \vee \ldots \vee q_n$$

where every q_i is a CQ. UCQs and CQs are supported by practical query engines for DL-*Lite* family, e.g.: Presto [RA10], Mastro [CDGL$^+$11] and others. Query answering in more expressive DLs is only partially supported by existing tools, including DL reasoners, e.g. Pellet, KAON2, and SPARQL-based query engines, currently under development [KGH11].

2.3 Context framework

Over two decades ago John McCarthy introduced the AI community to a new paradigm of formalizing contexts in logic-based knowledge systems. This idea, presented in his Turing Award Lecture [McC87], was quickly picked up by others and by now has led to a significant body of work studying different implementations of the approach in a variety of formal frameworks and applications [BM93, BBM95, Buv96, McC93, Guh91, Nos03]. The great appeal of McCarthy's paradigm stems from the simplicity and intuitiveness of the three major postulates it is based on:

1. Contexts are formal objects. More precisely, a context is anything that can be denoted by a first-order term and used meaningfully in a statement of the form $ist(\boldsymbol{c}, \varphi)$, saying that formula φ *is true* (*ist*) in context ***c***, e.g., $ist(Hamlet,$ 'Hamlet is a prince.') [McC87, McC93, Guh91, BM93]. By adopting a strictly formal view on contexts, one can bypass unproductive debates on what they really are and instead take them as primitives underlying practical models of contextual reasoning.

2. Contexts have properties and can be described. As first-order objects, contexts can be in a natural way described in a first-order language [Buv96, Guh91]. This allows for addressing them generically through quantified formulas such as $\forall x(\boldsymbol{C}(x) \rightarrow ist(x, \varphi))$, expressing that φ is true in every context of type ***C***, e.g., $\forall x(\mathit{barbershop}(x) \rightarrow ist(x,$ 'Main service is a haircut.')).

3. Contexts are organized in relational structures. In commonsense reasoning, contextual assumptions are dynamically and directionally altered [Nos03], [BM93]. Contexts are entered and then exited, accessed from other contexts or transcended to broader ones. Formally, this can be captured by allowing nestings of the form $ist(\boldsymbol{c}, ist(\boldsymbol{d}, \varphi))$, e.g., $ist(France, ist(capital,$ 'The city river is Seine.')).

In this thesis, we study a number of formalisms and application scenarios dealing with contexts within the DL-based knowledge representation paradigm, which conceptually adhere to the theory of contexts postulated by McCarthy. Let us now formally reconstruct those conceptual foundations, in a way which is in our belief most faithful to McCarthy's position and allows for a convenient operationalization of the theory.

The key step to importing McCarthy's theory into the DL framework is to faithfully reinterpret his three postulates on the model-theoretic grounds of DLs. Essentially, what needs to be settled is what kind of formal structures we generally intend to speak about. DLs, just like other modal logics, are tailored for expressing constraints about one-dimensional relational structures, whose nodes represent domain objects and edges the relations between them. Our intuitive reading of McCarthy's postulates compels us to extend this picture with another dimension, leading to structures of the type illustrated in Figure 2.1.

The figure presents a model of an application domain supporting multiple contexts of representation, where each context sustains its own viewpoint and focus on the represented world. This model (the upper part of the figure) has two visible levels. The *context-level* (the outer graph, bold line/font) consists of context entities (postulate 1), which are possibly described in a language containing constants and unary predicates (postulate 2), and linked to each other via certain relations (postulate 3). For instance, context ***c*** is of type ***D***

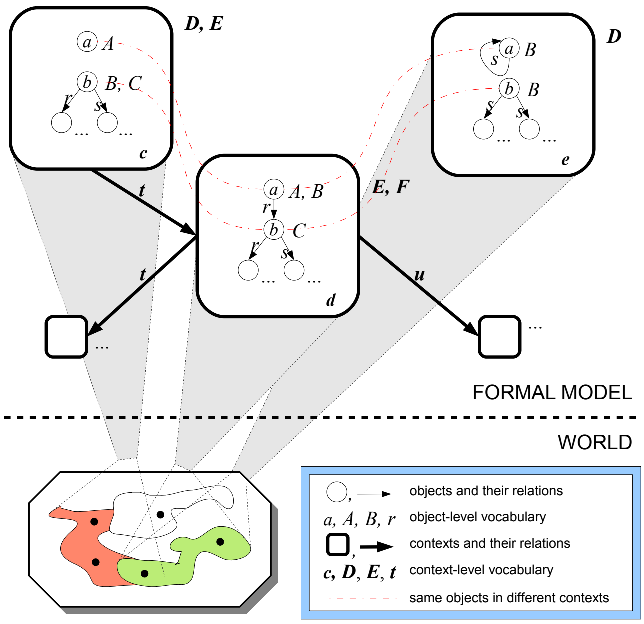

Figure 2.1: A formal context-object structure representing an application domain, based on McCarthy's postulates.

and is related to $\boldsymbol{d}$ through a relation of type $\boldsymbol{t}$. Intuitively, each context in the model carries a piece of the *object-level* representation (the inner graphs, thin line/font). Thus, instead of a unique, global model of the object domain, we associate with every context a single local model of the object domain. Hence, rather than with standard DL interpretations of type $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, we want to work with models of type $\mathfrak{M} = (\mathfrak{C}, \{\mathcal{I}(i)\}_{i \in \mathfrak{C}})$, where $\mathfrak{C}$ is a domain of contexts, and for every $i \in \mathfrak{C}$, $\mathcal{I}(i)$ is a usual DL interpretation over the domain of objects. The object models might obviously differ from each other, reflecting specific viewpoints on the represented world. Moreover, they might not necessarily cover the same fragment or aspect of the application domain and not necessarily use the same fragment of the object language for describing it. For instance, objects a and b occur at the same time in contexts $\boldsymbol{c}, \boldsymbol{d}, \boldsymbol{e}$, but in each of them they are described differently and remain in different relations to other objects. The central insight emerging under such a perspective is that the semantic structures addressed in the projected context framework are inherently two-dimensional.

Once the intended structures are identified, the next step is to find convenient languages for speaking about them. It is crucial here to distinguish between languages playing two very different functions in our investigations: representation languages, used for expressing knowledge about a particular application domain, including its object- and context-level descriptions, and languages required for executing certain reasoning tasks over the resulting representations, for instance query formalisms. In the scope of this work, we consider DLs as suitable representation languages for describing the object-level knowledge, and DL-oriented query languages, particularly CQs and UCQs, as the languages for accessing the DL-based knowledge models. The key challenge is then to support this kind of formalisms with additional expressive means that would facilitate representing and accessing the context-level information. A first crucial observation in this direction is that the context-level structures, as pictured in Figure 2.1, can be seen as Kripke frames, with possible worlds representing context entities and accessibility relations capturing relations between contexts. Interestingly, such a perspective offers a very clearcut formal reading of the notoriously elusive notion of context. Namely:

CONTEXT = POSSIBLE WORLD

This interpretation of contexts resonates very well with the philosophically neutral and application-agnostic notion of *context-as-formal-object* lying at the heart of McCarthy's theory. At the same time it is technically non-trivial, as it immediately encourages the use of the rich machinery of modal logics for capturing and studying different aspects of contextualization. In particular, various contextualization and lifting operations, i.e. context-sensitive transfers of knowledge between different contexts [McC87], can be naturally modeled by means of modal operators $\Diamond, \Box$. Based on these insights, we advocate that the context-level representation should be modeled and managed using a separate modal language, with its own vocabulary for describing properties of contexts. Consequently, we find ourselves in the realm of *two-dimensional*, *two-sorted* modal representations and formalisms, where the respective dimensions and sorts are suitably crafted and combined on the semantic and syntactic level, depending on the considered application scenario. In the following chapters, we study a number of such constructions, each one shortly summarized at the beginning of the respective chapter. In a nutshell:

- In Chapter 3, we develop a full-fledged representation language combining two DL languages, and consider the standard satisfiability problem for the proposed contextualized knowledge bases.

- In Chapter 4, we consider systems consisting of only finite sets of context, where each context is uniquely associated with a DL ontology and described on the meta-level in another DL language. The object language is augmented with restricted modal operators enabling cross-referencing between the ontologies. Thus we obtain a practical mechanism for ontology integration. Further, we propose a new type of two-sorted queries, which allow for metaknowledge-driven querying over preselected subsets of ontologies.

- In Chapter 5, we assume an analogical, two-level representation, where the object ontologies are arbitrary datasets contextualized with provenance information. On top of it, we develop a new query language for executing provenance-driven verification of data, which combines Propositional Dynamic Logic [Lan06], for traversing the context dimension, with object-level queries, for accessing data in the object ontologies.

- In Chapter 6, in a similar fashion as in the previous chapter, we combine Linear Temporal Logics with object queries, in order to query data contextualized with temporal information. Here, the context-level representation is expressed in Linear Temporal Logics with integer periodicity constraints [Dem06], which allow for compact representation of a number of interesting temporal constraints, such as calendars.

2.4 State of the art and related work

The problem of formalizing and operationalizing contexts for use in AI applications has a relatively long history and has been addressed by numerous researchers in a plethora of works. For a comprehensive overview we refer the reader to [AS96]. Next to McCarthy's *contexts-as-formal-objects*, discussed in detail in the previous section, another dominant tradition in this area is the paradigm of *multi-context logics* (MCLs), originally proposed and developed by Giunchiglia, Serafini et al. [Giu93, GS94, GG01]. This proposal is largely motivated by observations on human ability of reasoning with local representations, while adequately migrating between them depending on the problem-solving task at hand. The main accent in this approach is thus placed on formalizing contextual reasoning, understood as "translating" information between different contexts. A context is understood here metaphorically as a "box" containing certain local theory, which is related to the theories contained in other contexts via so-called bridge rules — i.e. rules whose bodies and heads are interpreted in different contexts. This conceptual perspective is arguably quite different from the one adopted in McCarthy's approach. There the focus is mostly on the representation of object knowledge with respect to a complex context structure, and the formalisms are augmented with a number of specific operations used for traversing this structure, such as entering, exiting context, lifting knowledge between contexts, etc. [BGGB03, SB04]. Regardless of those differences, the convenience of modal logics has been exploited in the design of formalisms based both: on McCarthy's theory [BM93, Buv96, Nos03] as well as MCLs [Giu93, GSS92, Ghi99]. Although both types of logics adopt also some limited two-dimensional perspective on the problem, in none of them are contexts so straightforwardly embedded in the possible world semantics as proposed in our thesis, nor they are supported with a dedicated, expressive enough context language to facilitate simple, ontological-style modeling of context structures. The power of our approach lies, as we believe, exactly in that we strongly refrain from focusing on specific intuitions regarding mechanisms of contextual reasoning per se or context-oriented operations and from hard-coding them in the formalism and its semantics, as is commonly done in the existing context logics. Instead, we simply offer as rich languages as possible for expressing knowledge about contexts and knowledge about domain objects with respect to those contexts, and further, we advocate a clear-cut separation of this representation layer from additional application-specific mechanisms added in a second stage, for solving particular context-driven problems. As a consequence we gain a substantial generality for our approach, lacked by different competing formalisms, and the flexibility needed for our framework to be applicable in a variety of use-cases.

The need for explicit treatment of context on the Semantic Web by extending the current knowledge representation paradigm and existing formalisms has been by now broadly acknowledged by the Semantic Web community [BGvH+03, GMF04, BSS05, GWW07, BTMS10, BHS12a, HS12]. Remarkably, the common motives, persistently recurring in this related work, include all three McCarthy's postulates, even if those are stated in diverse disguises. Regardless of those conceptual commonalities and largely shared motivation, however, the proposed contributions do not offer a complete nor a generalizable solution to the problem. From the perspective of the DL-based knowledge representation paradigm, adopted in our thesis, especially noticeable is the virtual lack of more generic contributions, offering deeper insights into logic foundations of the problem, which could foster better understanding of the subject and approaching it in a principled and a theoretically rigorous manner. This is exactly the gap that this thesis aims to fill.

A detailed comparison of our contributions with work by other authors is given alongside the individual chapters of the thesis. Here we want to briefly summarize three main research themes, addressed in the literature, which besides McCarthy's theory of contexts relate to and influence our work to the largest extent in various aspects.

The first such theme, from which we derive the semantic foundations for our framework, are two-dimensional DLs [WZ99, BL95]. Two-dimensional DLs are a family of logics constructed by extending the syntax of standard DL languages with pairs of modal operators $\Diamond_i, \Box_i$ applicable to chosen types of DL expressions: concepts, roles or axioms. The resulting language is interpreted over extended, two-dimensional DL interpretations whose core is a pair $(\mathfrak{W}, \{\mathcal{I}(w)\}_{w \in \mathfrak{W}})$, where $\mathfrak{W}$ is a set of possible worlds, and $\mathcal{I}(w)$ is a usual DL interpretation associated with world w. Intuitively, concept $\Diamond_i C$ describes the set of all domain individuals which are instances of C in some possible world reachable from the current one through the accessibility relation associated with $\Diamond_i$. The operators, quantifying over possible worlds of the second dimension, enable explicit modeling of a variety of intensional aspects of knowledge, e.g. temporal [LWZ08], evolutionary [ALT07], probabilistic [LS10], epistemic [DLN$^+$92] or dynamic [WZ00]. The research on two-dimensional DLs, carried on for over two decades now, has brought a spectrum of technical results regarding mathematical foundations and computational properties of the logics. In particular, it has been observed that two-dimensional DLs are a special type of product-like combinations of modal logics [MV97, KWZG03]. What is essential in terms of our research goals, is that two-dimensional DLs constitute a concrete paradigm of constructing logics which are intended for capturing different types of dependence of DL-expressed knowledge on implicit states present in the semantics. This capacity is fully exploited in the design of DLs of Context, in Chapter 3, but is also pivotal in shaping the semantics for the languages studied in all remaining chapters of this thesis. On the syntactic level our formalisms differ from the standard two-dimensional DLs in a number of specific aspects, predominantly, in that they involve a second sort of language, as explained in the previous section. To our knowledge, extension of this form to the framework of two-dimensional DLs has not been so far studied in the literature.

The second research area which partially overlaps with our work focuses on logic-based ontology integration. The central problem in this field is formulated as follows: how does one logically combine knowledge contained in several DL ontologies, without spoiling the local, context-specific scope of their axioms? As such the problem is clearly a reminiscence of the view on contextuality inherent to the framework of MCLs, mentioned in the beginning of this section. Most widely known solutions are Package-based DLs [BVSH09], Distributed DLs [BS03, BGvH+03], and $\mathcal{E}$-Connections [KLWZ03, CGPS09]. For a detailed survey, we refer the reader to [CK07], while here we only sketch the technical intuition behind the proposals. Essentially, each of the frameworks offers a formal mechanism of relating vocabularies between different sources, while preserving the independent semantics for every source. Let ***c*** and ***d*** be the identifiers for two knowledge bases, and $\boldsymbol{c}{:}A$, $\boldsymbol{d}{:}A$ be two concept names from the languages in which the respective sources are expressed. Package DLs allow for a direct use of concept $\boldsymbol{c}{:}A$ in the source ***d***, ensuring that its interpretation in both sources remains identical. In Distributed DLs one can impose an external bridge rule $\boldsymbol{c}{:}A \rightsquigarrow \boldsymbol{d}{:}A$, which states that for every object of type $\boldsymbol{c}{:}A$ in ***c*** there must be a "corresponding" object of type $\boldsymbol{d}{:}A$ in ***d***, where this correspondence is interpreted in terms of special mapping relations between the models of the ontologies. Finally, in the framework of $\mathcal{E}$-Connections one can express a concept $\exists con.\boldsymbol{d}{:}A$ in the ontology ***c***, which describes the set of all objects which are "$\mathcal{E}$-connected" to some objects of type $\boldsymbol{d}{:}A$ in ***d***. The $\mathcal{E}$-connection relation (here con) is again interpreted via designated semantic mappings. The differences in the proposed mechanisms obviously result in different properties of the integrated systems, e.g. in terms of strength of integration (more or less inferences possible) or robustness to inconsistency, which tend to satisfy certain intuitive expectations to different degrees, under different circumstances. Hence, the merits of each approach cannot be ultimately judged without a reference to some particular use-cases. The problem of ontology integration has motivated also similar frameworks in technically less demanding setting of RDF [BSS05, SBPR07]. As the idea of integrating multiple, local knowledge sources lies at the heart of our subject of study, the context framework proposed in this thesis also provides a substantial support for such scenarios, and it achieves it strictly on the grounds of two-dimensional semantics. This problem is addressed in detail in Chapter 4 and is also touched upon in Chapter 3.

Finally, we are strongly interested in representing and reasoning with meta-level descriptions of knowledge. One existing proposal in this area involves higher-order DLs, introduced in [GLR11], which allow for domain meta-modeling by quantifying and expressing constraints over the domain vocabulary. Although this approach is theoretically well-elaborate, it is based on mechanisms typical to higher-order logics rather than two-dimensional formalisms, addressed here. Arguably, logics of the latter type are conceptually and technically easier to manage, hence, not surprisingly, their features are more commonly employed on the Semantic Web. For instance, some data providers have been delivering meta-level descriptions over RDF using so-called Named Graphs [CBHS05], i.e. RDF graphs accompanied by their unique identifiers which can be treated as objects in other RDF graphs — a characteristic mechanism of introducing two-dimensionality. In other use-cases, the `owl:Annotation` property is sometimes used to attach additional qualifying information to particular axioms in OWL ontologies. A more systematic research on those representation methodologies has led to a couple of proposals on how to express and utilize such metadata in certain applications. In the pure RDF paradigm one such framework, called RDF$^+$ [DSSS09], supports queries over domain data, qualified in terms of the data's provenance, uncertainty level or other meta-description. Another approach, based on axiom annotations, allows for selection of sets of OWL axioms, whose annotations match given specifications [THM$^+$08]. Another similar framework, proposed in [ZLPS12], considers arbitrary Semantic Web data described with annotations belonging to certain well-behaved annotation languages, e.g. temporal or fuzzy, and supports some basic forms of annotation-driven inference over such data. For instance, whenever annotation t_1 is more specific than t_2, then the knowledge holding with respect to t_2 must also hold in the scope of t_1. The main shortcoming of those and similar contributions is, in our view, a quite limited treatment of the meta-level representation, which is often expressed in restricted, non-logical languages strongly impeding the semantic transparency and reasoning capabilities of the proposed systems. Nonetheless, the very idea of describing knowledge on the meta-level has a great impact on our work and is prevalent all through this thesis. In particular, Chapters 4-6 focus on how such knowledge can be systematically represented and utilized on the query level.

As our last general mention, we acknowledge the substantial work by Homola et al. [HS12, BHS12b], in which the framework of Contextualized Knowledge Repositories (CKRs) is defined. Notably, this proposal integrates to a certain extent all key aspects discussed so far: DL-based representation of object knowledge, contexts as formal objects, two-dimensional semantics, a mechanism of knowledge integration, and meta-level descriptions of contexts. However, apart from the unrestricted use of DL languages for expressing object knowledge, all the remaining features are considerably restricted. For instance, the number of contexts involved is always finite, integration is always performed on one-to-one basis, and the metalanguage is pruned down to a fixed set of contextual properties (dimensions), e.g. time, location, topic, along with their pre-defined values, and the coverage relation for organizing contexts in a generality-specificity hierarchy. Our work, although very closely aligned with this one in terms of motivation, conceptual foundations, and basic formal insights, could be seen as a result of pushing the envelope of the CKR proposal far beyond the limits drawn by its authors, in almost all possible respects. Consequently, we are able to demonstrate that the key principles of handling contexts, underlying the work by Homola and ours, indeed generalize beyond the specific setting of CKRs or other particular systems developed in the chapters of this thesis.

CHAPTER
THREE

DESCRIPTION LOGICS OF CONTEXT

In this chapter, we define a novel family of two-dimensional, two-sorted DLs of context, similar to product-like combinations of DLs with modal logics. We present results regarding their expressiveness, relationships to other known formalisms, and computational complexity of the basic decision problems, ranging from EXPTIME- *to* 2EXPTIME-*completeness. We consider the following setting:*

contexts $\doteq$ *abstract entities of the second semantic dimension*

context representation language $\doteq$ *DL*

contextual information $\doteq$ *descriptions in the context language*

object representation language $\doteq$ *DL with modal-like context operators on concepts and annotations on axioms*

reasoning task $\doteq$ *satisfiability checking*

3.1 Introduction

Description Logics have been applied successfully in a number of fields as logic tools for constructing and managing ontologies, i.e. formal models of terminologies and instance data, representative of particular domains of interest. The Semantic Web is one of the outstanding environments where such ontologies, expressed in the DL-based OWL languages, play a key architectural role: facilitating publication of knowledge on the Web in a machine-understandable way [BHS03]. Through the close ties to OWL, DLs effectively provide the Semantic Web with its mathematical foundations and determine the methodology of knowledge modeling and the reasoning regime observed by the ontology-based Web applications. Along with the benefits gained from this relationship, come also significant limitations inherent to the DL paradigm. One such shortcoming, which we focus on here, is the lack of well-defined, generic theory of dealing with contextual aspects of knowledge in DLs.

Under the standard Kripkean semantics, a DL ontology imposes a unique, global and uniform view on the represented domain. Put technically, the axioms of an ontology are interpreted as unconditionally and universally true in all models of that ontology, e.g. $Heart \sqsubseteq HumanOrgan \in \mathcal{O}$ enforces all domain individuals of type $Heart$ to be of type $HumanOrgan$ in all possible models of $\mathcal{O}$. Such a representation philosophy is well-suited as long as everyone shares the same conceptual perspective on the domain and there is no need for considering alternative viewpoints. Alas, this is hardly ever the case as most of times a domain should be modeled differently depending on the context in which it is considered, where the context might depend on a spatio-temporal coordinate, the thematic focus, a subjective perspective of the modeler, the adopted level of granularity of the representation, an intended application of the ontology, etc. For instance, axiom $Heart \sqsubseteq HumanOrgan$, valid about the domain of human anatomy, loses its generality once it is considered from a broader perspective of mammal anatomy. This intrinsic inability of accounting for contexts in DLs poses two kinds of fundamental problems. 1) In principle, it is impossible to create ontologies that would be at the same time broad enough as to capture all relevant information about the domain and yet sufficiently detailed as to cover all context-specific peculiarities in the formal representation of that knowledge. This challenge is commonly faced by the creators of huge knowledge bases, aiming at maximum coverage of the representation, such as SNOMED [Spa08] or CYC [Len98], and typically leads to development of ad hoc, application-driven mechanisms of contextualization. 2) The second problem concerns the reuse of knowledge from multiple existing sources — such as the numerous DL-based ontologies already published on the Web — in new applications. Naturally, portions of such knowledge retrieved from different ontologies are likely to pertain to different, heterogenous contexts, which are implicitly assumed on the creation of the sources. Consequently, a faithful reuse of such data cannot be achieved without special semantic mechanisms which are capable of acknowledging and respecting its local, context-specific character [GMF04, BTMS10].

Interestingly, variants of these two problems are well-recognized in the field of knowledge representation and used as a motivation for two basic views on the use of contexts in knowledge representation systems. Bouquet et al. denote these views as *divide-and-conquer* and *compose-and-conquer* [BGGB03] and describe them as follows:

> [...] According to the first view, which we call *divide-and-conquer*, there is something like a global theory of the world. This global theory has an internal structure, and this structure is articulated into a collection of contexts. According to the second view, which we call *compose-and-conquer*, there is not such a thing as a global theory of the world, but only many local theories. Each local theory represents a viewpoint on the world. Also, there may exist relations between local theories that allow a reasoner to (partially) compose them into a more comprehensive view. [...]

According to Bouquet et al., these two theories of context are relevant for problems of two very different types and hence they naturally give rise to two diverse sorts of formal solutions. Not surprisingly, the ongoing research efforts on incorporating contexts into the DL framework exhibit this exact dichotomy as well. On the one side, there have been specific attempts of extending the DL languages with operators for modeling the dependence of knowledge on implicit contextual states, such as levels of abstraction over an ontology [GWW07, Gro07] or states of some fixed modal dimension — most typically a temporal one [LWZ08] [AKL+07]. On the other side, there have been several formalisms proposed for supporting the task of integrating local ontologies, including Distributed DLs [BS03, BGvH+03], Package-based DLs [BVSH09] or $\mathcal{E}$-connections [KLWZ04, GPS06], with no real consensus on the most natural or generic approach.

As the existing solutions are notoriously specialized in their scope, the problem of formulating a broad and well-grounded theory of contexts within the DL paradigm remains open. In this chapter, we systematically develop an extension of classical DLs called *Description Logics of Context* (DLCs), which aims at filling this gap, and arguably, bridges the two theories of context under one unifying formal approach. Our proposal is inspired by J. McCarthy's theory of formalizing contexts [McC87], whose gist is to replace logical formulas φ, as the basic knowledge carriers, with assertions of the form $ist(\boldsymbol{c}, \varphi)$. Such assertions state that φ *is true* in $\boldsymbol{c}$, where $\boldsymbol{c}$ denotes an abstract first-order entity called a *context*. Further, contexts can be on their own described in a first-order language. For instance, the formula:

$$ist(\boldsymbol{c}, Heart(a)) \wedge \boldsymbol{HumanAnatomy}(\boldsymbol{c})$$

states that the object a is a heart in a certain context $\boldsymbol{c}$ of type human anatomy. Formally, we interpret McCarthy's theory in terms of two-dimensional possible world semantics, characteristic of product-like combinations of DLs with modal logics [WZ99, KWZG03], or even more generally, of products of modal logics [MV97, KWZG03]. In DLCs, one semantic dimension represents a usual object domain and the other a (possibly infinite) domain of contexts. Thus, the notion of context is identified with that of Kripkean *possible world*, which provides the former with a philosophically neutral, yet technically substantial reading, presupposed at the core of McCarthy's theory. Unlike conventional two-dimensional DLs, the DLCs are equipped with two interacting DL languages — the object and the context language — interpreted over the respective domains. These languages allow for explicit modeling of both: the (contextualized) object-level knowledge and the meta-level knowledge, i.e. descriptions of contexts as first-class citizens. Consequently, we define a whole family of *two-sorted*, *two-dimensional* combinations of pairs of DLs, comprising the DLC framework.

Problem: This chapter addresses and suggests new answers to the following research questions:

1. How to extend DLs to support the representation of inherently contextualized knowledge?
2. How to use knowledge from coexisting classical DL ontologies while respecting its context-specific scope?
3. Is it possible to capture these two perspectives on contextualization within one unifying formal framework?

Contributions: In this chapter, we propose a novel family of logics for modeling and studying mechanisms of contextualization in the DL paradigm. The framework is derived from two roots: *conceptually* — from McCarthy's theory of formalizing contexts, grounding our approach in a longstanding research tradition in AI; *formally* — from two-dimensional DLs, ensuring strong and well-understood mathematical foundations. We demonstrate the applicability of DLCs to a range of representation problems dealing with contexts. We prove 2ExpTime-completeness of the satisfiability problem in the maximally expressive fragment of the framework studied here, with the object and the context language based on the DL $\mathcal{SHIO}$ [HM04]. As a corollary, we show that the same result holds also for several underlying two-dimensional DLs with global TBoxes and local interpretation of roles, including the prominent $(\mathbf{K}_n)_{\mathcal{ALC}}$. Further, we deliver tight complexity bounds for several less expressive fragments of DLCs, ranging between EXPTIME- and 2EXPTIME-completeness.

Content: The chapter is organized as follows. We start in the next section with discussing the formal motivation behind the proposed design of DLCs. In Section 3.3 we present the syntax and semantics for DLs of Context. Further, in Section 3.4, we provide an overview of possible application scenarios, emphasizing their relation to the *divide-and-conquer* and *compose-and-conquer* theories of contexts. Finally, in Section 3.5 we address the formal relationships of the framework to the well-known two-dimensional DLs $(\mathbf{K}_n)_{\mathcal{L}}$ and $\mathbf{S5}_{\mathcal{L}}$ and elaborate on the computational properties of a number of discussed logics. The chapter is concluded with a discussion in Section 6.7. Some technical proofs of the presented results are deferred to Appendix A.

3.2 Overview

The logics proposed in this chapter originate as an attempt of constructing an extension of DLs which faithfully accommodates the context framework inspired by McCarthy's theory of formalizing contexts in AI. As is usually the case, the extent of such accommodation can be purposely limited in its scope in order to obtain formalisms of different expressiveness and, consequently, of different computational properties. Thus, rather than a single logic, we develop a family of formalisms generically referred to as Description Logics of Context (DLCs).

In Sections 2.3, we have observed that the semantic structures implied by McCarthy's postulates have two apparent levels, where each level can be seen as a Kripke frame over different set of objects and relations. It is known, that such frames can be formally combined in a product-like fashion, giving rise to two-dimensional modal logics [MV97, KWZG03], and in our case, to two-dimensional DLs [WZ99, BL95], shortly introduced in Section 2.4 and further presented in more detail in Section 3.5. As context-dependency bears many apparent similarities to other dynamic aspects of knowledge, commonly addressed on the grounds of two-dimensional DLs, building DLCs on top of the architecture of such logics seems indeed very appealing. In particular, such an approach offers potential benefits of easing the transfer of known results and proof techniques. However, there is one serious caveat which re-

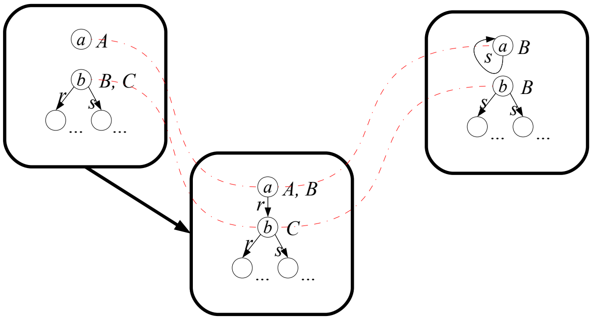

Figure 3.1: A model of a two-dimensional DL.

quires revisiting this strategy. Namely, typical two-dimensional DLs do not come with means of describing properties of the possible worlds, and so the formal structures captured by the semantics of those logics clearly do not include rich meta-level information (see Figure 3.1). From our perspective, this means that we can easily augment a DL language with modal '*contextualization*' operators for traversing the context dimension of the models and quantifying over the context entities, but it is not possible to explicitly assert properties of the accessed contexts *per se*, for instance to express global contextual dependencies, such as 'In every context of type human anatomy, it holds that: $Heart \sqsubseteq HumanOrgan$'. Intuitively, such functionality seems essential for obtaining a fine-grained contextualization mechanism, at least if one follows closely the provisions declared by McCarthy. The solution which we propose here is to employ a second DL language for describing the context dimension. As a consequence, we obtain a *two-sorted*, *two-dimensional* logic, where each sort of the language is interpreted over the respective dimension in the semantics. The two languages are suitably integrated on the syntactic and semantic level, so that their models can be eventually combined as presented in Figure 3.2. The style of combination is fully compatible with the underlying two-dimensional DLs. In principle, the two-dimensional models of the object language are embedded in the models of the context language, where possible worlds are mapped on selected (context) individuals and accessibility relations are mapped on selected (context) roles. Thus, we are able to show that, depending on the choice of the context operators, our logics are proper extensions of the well-known two-dimensional logics $(\mathbf{K}_n)_\mathcal{L}$ or $\mathbf{S5}_\mathcal{L}$ [WZ99].

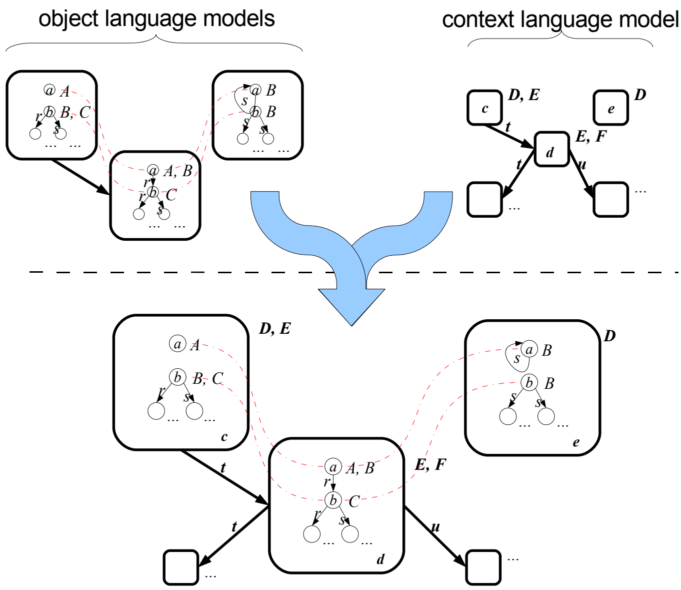

Figure 3.2: Combining models of two DLs.

3.3 Syntax and semantics

In this section we define the syntax and semantics of DLCs. A *Description Logic of Context* $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$ consists of the DL context language $\mathcal{L}_C$, supporting context descriptions, and of the object language $\mathcal{L}_O$ equipped with context operators for representing object knowledge relative to contexts.

Definition 1 ($\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$-context language)**.** *The* context language *of* $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$ *is a DL language* $\mathcal{L}_C$ *over vocabulary* $\Gamma = (M_C, M_R, M_I)$*, with a designated subset* $M_I^\star \subseteq M_I$ *of* context names.

Distinguishing a subset of context names $M_I^\star$ from the set of all individual names belonging to the context language reflects a broader intuition, that not all elements of the context domain are to be considered as contexts *per se*. Certain elements of this domain might instead serve only as individuals used for describing contexts (cf. Figure 2.1), without any object knowledge associated with them. For instance, this is often the case in applications concerned with provenance of knowledge [BTMS10]. A context c, associated with a single knowledge source, might be described there with an axiom $\boldsymbol{hasAuthor}(\boldsymbol{c}, \boldsymbol{henry})$, where $\boldsymbol{henry}$ is an individual related to $\boldsymbol{c}$, but not a context itself.

Definition 2 ($\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$-object language)**.** *Let $\mathcal{L}_O$ be a DL language over vocabulary $\Sigma = (N_C, N_R, N_I)$. The* object language *of $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$ is the smallest language containing $\mathcal{L}_O$ and closed under the constructors of $\mathcal{L}_O$, and additionally closed under at least one of the two types —* $\mathfrak{F}_1$ resp. $\mathfrak{F}_2$ *— of concept-forming operators:*

$$\langle \boldsymbol{r}.\boldsymbol{C} \rangle\, D \ \mid\ [\boldsymbol{r}.\boldsymbol{C}]\, D \tag{$\mathfrak{F}_1$}$$

$$\langle \boldsymbol{C} \rangle D \ \mid\ [\boldsymbol{C}] D \tag{$\mathfrak{F}_2$}$$

where $\boldsymbol{C}$ and $\boldsymbol{r}$ are a concept and a role of the context language and D is a concept of the object language.

Intuitively, the concept $\langle \boldsymbol{r}.\boldsymbol{C} \rangle D$ denotes all objects which are D in *some* context of type $\boldsymbol{C}$ accessible from the current one through $\boldsymbol{r}$. Analogically, $[\boldsymbol{r}.\boldsymbol{C}]D$ denotes all objects which are D in *every* such context. In the case of $\mathfrak{F}_2$ operators, the concept $\langle \boldsymbol{C} \rangle D$ denotes all objects which are D in *some* context of type $\boldsymbol{C}$, whereas $[\boldsymbol{C}]D$ all objects which are D in *every* such context. For example, $\langle \boldsymbol{neighbor}.\boldsymbol{Country} \rangle Citizen$, refers to the concept $Citizen$ in some context of type $\boldsymbol{Country}$ accessible through the $\boldsymbol{neighbor}$ relation from the current context. Analogically, $\langle \boldsymbol{HumanAnatomy} \rangle Heart$ refers to the concept $Heart$ in some context of $\boldsymbol{HumanAnatomy}$.

Clearly, the operators $\mathfrak{F}_1$ and $\mathfrak{F}_2$ behave very similar to the usual modalities of $\mathbf{K}_n$ and **S5**, respectively. In particular, for any $\boldsymbol{r}$ and $\boldsymbol{C}$ the expected dualities hold: $\langle \boldsymbol{r}.\boldsymbol{C} \rangle = \neg[\boldsymbol{r}.\boldsymbol{C}]\neg$ and $\langle \boldsymbol{C} \rangle = \neg[\boldsymbol{C}]\neg$. In fact, the only difference is that the contexts (possible worlds) accessed by means of $\mathfrak{F}_1$ or $\mathfrak{F}_2$ are additionally qualified with a certain concept of the context language. We formally elaborate on this relationship in Section 3.5.

Definition 3 ($\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$-knowledge base)**.** *A* $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$-knowledge base (CKB) *is a pair $\mathcal{K} = (\mathcal{C}, \mathcal{O})$, where $\mathcal{C}$ is a set of axioms over the context language (in the syntax allowed by $\mathcal{L}_C$), and $\mathcal{O}$ is a set of formulas of the form:*

$$\boldsymbol{c} : \varphi \mid \boldsymbol{C} : \varphi$$

where φ is an axiom over the object language (in the syntax allowed by $\mathcal{L}_O$), $\boldsymbol{c} \in M_I^\star$ and $\boldsymbol{C}$ is a concept of the context language.

A formula $\boldsymbol{c} : \varphi$ states that axiom φ holds in the context denoted by the context name $\boldsymbol{c}$. Note, that this corresponds directly to McCarthy's $ist(\boldsymbol{c}, \varphi)$. Axioms of the form $\boldsymbol{C} : \varphi$ assert the truth of φ in all contexts of type $\boldsymbol{C}$. For example, the formula:

$$\boldsymbol{Country} : \langle \boldsymbol{neighbor}.\boldsymbol{Country} \rangle\, Citizen \sqsubseteq No\,VisaRequirement$$

states that in every country, the citizens of its neighbor countries do not require visas. Observe, that the style of contextualization involved here is very generic. The constraint applies to every pair of contexts in the system consisting of a country and its respective neighbor, depending purely on the descriptions of those contexts supplied in the context language. To compare with, typical multi-context logics, e.g. DDLs (see Section 2.4) would allow for expressing similar constraints only between two explicitly indicated contexts (boxes), without a possibility of addressing them by their abstract types, for instance:

$$\boldsymbol{germany} : Citizen \overset{\sqsubseteq}{\rightarrow} \boldsymbol{france} : No\,VisaRequirement$$

The semantics is given through $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$*-interpretations* and $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$*-models*, which combine the interpretations of $\mathcal{L}_C$ with those of $\mathcal{L}_O$. As explained before, the (possibly infinite) domain of contexts $\mathfrak{C}$ is subsumed by the interpretation domain of the context language Θ. For technical reasons, we assume a constant object domain Δ for all contexts. This assumption, though often unnatural in practical scenarios, grants greater generality to the complexity results and can be most of times relaxed to the varying domain case without consequences for the hardness of reasoning [KWZG03].

Definition 4 ($\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-interpretations)**.** *A* $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$*-interpretation is a tuple* $\mathfrak{M} = (\Theta, \mathfrak{C}, \cdot^{\mathcal{J}}, \Delta, \{\cdot^{\mathcal{I}(i)}\}_{i \in \mathfrak{C}})$*, such that:*

1. $(\Theta, \cdot^{\mathcal{J}})$ *is an interpretation of the context language, where* Θ *is a non-empty domain of individuals and* $\cdot^{\mathcal{J}}$ *an interpretation function, where:*
   - $\mathfrak{C} \subseteq \Theta$ *is a non-empty domain of contexts,*
   - $\boldsymbol{c}^{\mathcal{J}} \in \mathfrak{C}$*, for every* $\boldsymbol{c} \in M_I^{\star}$*,*
2. Δ *is a non-empty domain of individuals,*
3. $(\Delta, \cdot^{\mathcal{I}(i)})$*, for every* $i \in \mathfrak{C}$*, is an interpretation of the object language, where* $\cdot^{\mathcal{I}(i)}$ *an interpretation function, such that:*

($\mathfrak{F}_1$) *for every* $\langle \boldsymbol{r}.\boldsymbol{C}\rangle D$ *and* $[\boldsymbol{r}.\boldsymbol{C}]D$*:*

– $(\langle \boldsymbol{r}.\boldsymbol{C}\rangle\, D)^{\mathcal{I}(i)} = \{x \in \Delta \mid \exists j \in \mathfrak{C} : (i,j) \in \boldsymbol{r}^{\mathcal{J}} \wedge j \in \boldsymbol{C}^{\mathcal{J}} \wedge x \in D^{\mathcal{I}(j)}\}$,

– $([\boldsymbol{r}.\boldsymbol{C}]\, D)^{\mathcal{I}(i)} = \{x \in \Delta \mid \forall j \in \mathfrak{C} : (i,j) \in \boldsymbol{r}^{\mathcal{J}} \wedge j \in \boldsymbol{C}^{\mathcal{J}} \rightarrow x \in D^{\mathcal{I}(j)}\}$.

($\mathfrak{F}_2$) *for every* $\langle \boldsymbol{C}\rangle D$ *and* $[\boldsymbol{C}]D$*:*

– $(\langle \boldsymbol{C}\rangle D)^{\mathcal{I}(i)} = \{x \in \Delta \mid \exists j \in \mathfrak{C} : j \in \boldsymbol{C}^{\mathcal{J}} \wedge x \in D^{\mathcal{I}(j)}\}$,

– $([\boldsymbol{C}]D)^{\mathcal{I}(i)} = \{x \in \Delta \mid \forall j \in \mathfrak{C} : j \in \boldsymbol{C}^{\mathcal{J}} \rightarrow x \in D^{\mathcal{I}(j)}\}$.

Clearly, the difference between the context operators of type $\mathfrak{F}_1$ and $\mathfrak{F}_2$ lies in the choice of the relational structures observed when quantifying over the context domain. $\mathfrak{F}_1$-operators bind contexts only along the roles of the context language (as **K**-modalities), while $\mathfrak{F}_2$-operators follow the universal relation over $\mathfrak{C}$ (as **S5**-modalities). This leads to some clear consequences in the scope and the character of the distribution of the object knowledge over contexts in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-models. For instance, in Figure 2.1, the concept $\langle \boldsymbol{t}.\boldsymbol{F}\rangle B$ is satisfied by object a only in context $\boldsymbol{c}$, while $\langle \boldsymbol{F}\rangle B$ is satisfied by a in all contexts in the model. From the perspective of McCarthy's theory, employing operators $\mathfrak{F}_2$, rather than $\mathfrak{F}_1$, is equivalent to sacrificing postulate (3). This means that every two contexts in the model become in principle accessible to each other. The focus on **K**-like and **S5**-like modalities is quite arbitrary here and driven merely by the formal simplicity of the two types of operators and easiness of their integration with the DL semantics. In principle, however, nothing prevents from constructing logics containing contextualization operators which mimic other common modalities.

Finally, we define the notion of $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-model.

Definition 5 ($\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-models)**.** *A* $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$*-interpretation* $\mathfrak{M} = (\Theta, \mathfrak{C}, \cdot^{\mathcal{J}}, \Delta, \{\cdot^{\mathcal{I}(i)}\}_{i\in\mathfrak{C}})$ *is a* model *of a CKB* $\mathcal{K} = (\mathcal{C}, \mathcal{O})$ iff*:*

- *for every* $\varphi \in \mathcal{C}$, $(\Theta, \cdot^{\mathcal{J}})$ *satisfies* φ,
- *for every* $\boldsymbol{c} : \varphi \in \mathcal{O}$, $(\Delta, \cdot^{\mathcal{I}(\boldsymbol{c}^{\mathcal{J}})})$ *satisfies* φ,
- *for every* $\boldsymbol{C} : \varphi \in \mathcal{O}$ *and* $i \in \mathfrak{C}$*, if* $i \in \boldsymbol{C}^{\mathcal{J}}$ *then* $(\Delta, \cdot^{\mathcal{I}(i)})$ *satisfies* φ.

Analogically to the standard DLs, we say that a $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-knowledge base is *satisfiable iff* it has a $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-model. Likewise, the central reasoning problem in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ is deciding knowledge base satisfiability.

3.4 Application scenarios

As reflected in the dichotomy of theories of context [BGGB03] (see Section 3.1), context-based applications can be in a natural way classified into two categories: divide-and-conquer and compose-and-conquer. These two types are commonly considered as incompatible with each other, hence the formal solutions that are developed are typically specialized in tackling exclusively either of the two types of applications. As pointed out before, DLCs are proposed as a general framework capable to give solutions to both kinds of context-based applications, within the DL representation paradigm. In this section, we roughly indicate how this can be achieved.

Divide-and-conquer

The divide-and-conquer philosophy is characterized by the assumption of the existence of a unique and universal view of the knowledge. Such global view is then assumed to be partitioned in a set of interrelated fragments, where every fragment specializes the global view into a specific local view. The pieces of representation belonging to these fragments are *lifted* [McC87] from one local view to another via structural relations between the different local views. Formally, each local view is regarded as a separate context. The main application scenario under the divide-and-conquer approach is then to represent and reason about knowledge of an inherently contextual nature, in such a way that the underlying contextual structure assumed in the knowledge is faithfully accounted for. We present two examples showing that DLCs can be used as native representation languages dedicated to such applications. Here, the local views are naturally represented as the elements of the context domain in $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$-models, enforced by individual context names and context operators.

$\mathcal{C}$:	$\boldsymbol{Country}(\boldsymbol{germany})$	(1)
	$\boldsymbol{neighbor}(\boldsymbol{france}, \boldsymbol{germany})$	(2)
	$\boldsymbol{hasLanguage}(\boldsymbol{france}, \boldsymbol{french})$	(3)
$\mathcal{O}$:	$\boldsymbol{germany} : \exists hasParent.Citizen(john)$	(4)
	$\boldsymbol{Country} : \exists hasParent.Citizen \sqsubseteq Citizen$	(5)
	$\boldsymbol{france} : \langle \boldsymbol{neighbor}.\boldsymbol{Country} \rangle Citizen \sqsubseteq NoVisaRequirement$	(6)

Table 3.1: A sample knowledge base in $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$ with $\mathfrak{F}_1$-operators.

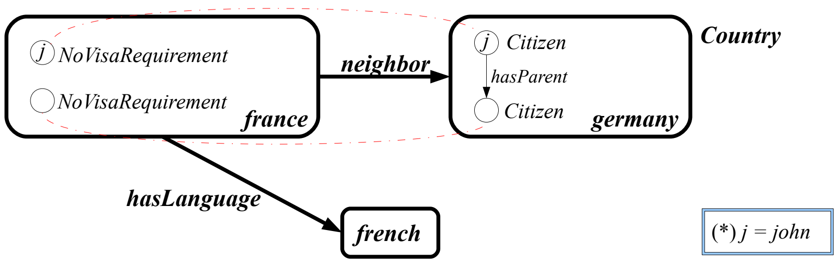

Figure 3.3: A possible model of the CKB in Table 3.1.

A contextualized knowledge base with $\mathfrak{F}_1$-operators Consider a simple representation of knowledge about the legal status of people, contextualized with respect to geographic locations. In Table 3.1, we define a CKB $\mathcal{K} = (\mathcal{C}, \mathcal{O})$ with $\mathfrak{F}_1$-operators, consisting of the context (geographic) ontology $\mathcal{C}$ and the object (people) ontology $\mathcal{O}$. Countries, in particular ***france*** and ***germany***, play here the role of contexts, described in the context language by axioms (1)-(3). Notably, the individual ***french*** serves in the description of the context ***france*** without carrying any specific object knowledge on its own. One would thus assume that ***france***, ***germany*** $\in M_I^\star$, while ***french*** $\in M_I \setminus M_I^\star$, and consequently ***france***$^\mathcal{J}$, ***germany***$^\mathcal{J}$ $\in \mathfrak{C}$ while ***french***$^\mathcal{J}$ $\in \Theta \setminus \mathfrak{C}$. In the context of ***germany***, it is known that $john$ has a parent who is a citizen (4). Since in every ***Country*** context — thus including ***germany*** — the concept $\exists hasParent.Citizen$ is subsumed by $Citizen$ (5), therefore it must be true that $john$ is an instance of $Citizen$ in ***germany***. Finally, since ***germany*** is related to ***france*** via the role ***neighbor***, it follows that $john$ (assuming rigid interpretation of this name across contexts) has to be an instance of $NoVisaRequirement$ in the context of ***france*** (6). A sample $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-model of $\mathcal{K}$ is depicted in Figure 3.3.

$\mathcal{C}$:	$\boldsymbol{Geometry} \sqsubseteq \boldsymbol{Math}$	(1)
$\mathcal{O}$:	$\boldsymbol{disambiguation} : Ring \sqsubseteq \langle \boldsymbol{Math} \rangle Ring \sqcup \langle \boldsymbol{People} \rangle Ring$	(2)
	$\boldsymbol{Math} : Ring \sqsubseteq AlgebStruct \sqcup \langle \boldsymbol{Geometry} \rangle Annulus$	(3)
	$\boldsymbol{People} : Ring \sqsubseteq \{nickRing\}$	(4)

Table 3.2: A sample knowledge base in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ with $\mathfrak{F}_2$-operators.

A contextualized knowledge base with $\mathfrak{F}_2$-operators In Table 3.2, we model

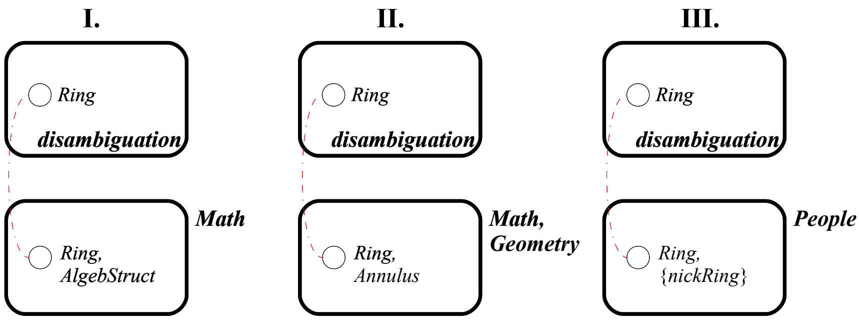

Figure 3.4: Possible models of the CKB in Table 3.2.

a piece of information presented on the disambiguation website of Wikipedia on querying for the term $Ring$. In particular, $Ring$ is contextualized according to whether it is defined as a mathematical object or as person.[1] Observe, that the named context ***disambiguation*** provides basic distinction on $Ring$ in some ***Math*** context and in some ***People*** context (2). This is further enhanced, by the distinction defined on the level of all ***Math*** contexts. There, $Ring$ denotes either $AlgebStruct$ or $Annulus$ in some further ***Geometry*** context (3), where ***Geometry*** contexts are known to be a subset of ***Math*** contexts. In case of ***People*** context, $Ring$ actually denotes an individual $nickRing$ (4). Some possible $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-models of this representation are depicted in Figure 3.4.

These two examples illustrate that DLCs allow in a natural way to contextualize knowledge. In particular, the contextual language allows to refine the representation at different levels of detail in each local view. Furthermore, DLCs permit to lift the knowledge from one context to another by means of the contextual operators, e.g., $\langle \boldsymbol{C} \rangle$ ($\langle \boldsymbol{r}.\boldsymbol{C} \rangle$) lifts the knowledge from any context $\boldsymbol{C}$ ($\boldsymbol{C}$ accessible via $\boldsymbol{r}$) to the current context.

Compose-and-conquer

In contrast to divide-and-conquer, the compose-and-conquer approach assumes neither the existence of a global view of the knowledge nor of a predefined relational structure among the different local views comprising the global view. Instead, compose-and-conquer assumes the existence of many

[1]See http://en.wikipedia.org/wiki/Ring.

independent local theories, each of them representing a particular viewpoint on the world. Naturally, each local view, can be focused on a particular part of the world, or can be described on a different granularity level. To create a composed view of the world from the local views, one needs to use a sort of *bridge rules* [Giu93, GS94, BS03], i.e. constraints stating the relationships between the knowledge in two different views. In peer-to-peer environments, such constraints allow peers to locally derive new information given the relationships of their own knowledge to the knowledge of other peers [BGK+02, BGGB03]. Analogical problems emerge in scenarios concerned with ontology alignment, reuse and versioning, where each ontology is treated as a separate local theory in some way or another related to other ontologies. In general, under the compose-and-conquer philosophy each autonomous local view can be seen as a knowledge source, a peer or a context. Essentially, what we are interested in is establishing formal interdependencies over both object- and metaknowledge descriptions, in order to facilitate the semantic interoperability and coordination of the local views [BGK+02, AKK+03].

Below we show how such constraints can be expressed in the DLC framework. It is important to note that in compose-and-conquer applications, very often rather than imposing constraints on the information flow in a possibly unbounded space of local views, we might need to integrate only a finite set of views, where each view is syntactically represented as a single DL ontology. For instance, at query time one is usually interested only in the knowledge pertaining to the views currently participating in the system [AKK+03]. Observe, that in DLCs a collection of DL ontologies $\mathcal{O}_1, \ldots, \mathcal{O}_n$ in some language $\mathcal{L}_O$ can be seen as a set of formulas $\mathcal{O} = \{\boldsymbol{c}_i : \varphi \mid \varphi \in \mathcal{O}_i,\ i \in (1, n)\}$ in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$, where every ontology corresponds to a unique context name. Further, whenever required, one can straightforwardly finitize the context domain by using the nominals of the context language representing exactly the ontologies to be integrated. This is achieved by imposing the axiom $\top \sqsubseteq \{\boldsymbol{a}\} \sqcup \{\boldsymbol{b}\} \sqcup \ldots$ for all relevant context names $\boldsymbol{a}, \boldsymbol{b}, \ldots$. In this way, the context operators are forced to implicitly quantify over a finite set of named ontologies. Similar assumptions regarding the finiteness of the set of contexts are employed in the scenarios studied in Chapters 4 and 5.

Interoperability constraints for ontology alignment and reuse. Consider an architecture such as the NCBO BioPortal project[2], which gathers numerous published biohealth ontologies, and categorizes them via thematic tags, e.g.: ***Cell***, ***Health***, ***Anatomy***, etc., organized in a meta-ontology. The intention of the pro-

[2]See http://bioportal.bioontology.org/.

ject is to facilitate the reuse of the collected resources in new applications. Note, that the division between the context and the object language is already present in the architecture of the BioPortal, this is naturally reflected in the example of Table 3.3 where (2) maps the concept $Heart$ from any ***HumanAnatomy*** ontology to the concept $HumanHeart$ in every ***Anatomy*** ontology; (3) imposes the axiom $Heart \sqsubseteq Organ$ of an upper anatomy ontology over all ***Anatomy*** ontologies, which due to axiom (1) carries over to all ***HumanAnatomy*** ontologies.

$\mathcal{C}$:	$\boldsymbol{HumanAnatomy} \sqsubseteq \boldsymbol{Anatomy}$	(1)
$\mathcal{O}$:	$\top : \langle \boldsymbol{HumanAnatomy} \rangle Heart \sqsubseteq [\boldsymbol{Anatomy}] HumanHeart$	(2)
	$\boldsymbol{Anatomy} : Heart \sqsubseteq Organ$	(3)

Table 3.3: A set of interoperability constraints expressed as a knowledge base in $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$ with $\mathfrak{F}_2$-operators.

In general, $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$ provides logic-based explications of some interesting notions, relevant to the problem of semantic interoperability of ontologies, such as:

concept alignment: $\top : \langle \boldsymbol{A} \rangle C \sqsubseteq [\boldsymbol{B}] D$
every instance of C in any ontology of type $\boldsymbol{A}$ is D in every ontology of type $\boldsymbol{B}$

semantic importing: $\boldsymbol{c} : \langle \boldsymbol{A} \rangle C \sqsubseteq D$
every instance of C in any ontology of type $\boldsymbol{A}$ is D in ontology $\boldsymbol{c}$

upper ontology axiom: $\boldsymbol{A} : C \sqsubseteq D$
axiom $C \sqsubseteq D$ holds in every ontology of type $\boldsymbol{A}$

Interoperability constraints for ontology versioning management and change analysis. The context operators can be also interpreted as change operators, in the style of DL of Change [ALT07], for instance, for representing and studying dynamic aspects of ontology versioning, especially when evolutionary constraints apply to a whole collection of semantically interoperable ontologies. Some central issues arising in this setup are integrity (constraining the scope of changes allowed due to versioning), evolvability (ability of coordinating the evolution of ontologies) and formal analysis of differences between the versions [HS05]. In the examples below, we assume that contexts represent possible versions, each metalanguage concept refers to all versions of the same ontology and ***updatedBy*** denotes the relation of being an immediate updated version.

version-invariant concepts: $\top : \langle \boldsymbol{A} \rangle C \equiv [\boldsymbol{A}]C$
C is a version-invariant concept within the scope of versions of type $\boldsymbol{A}$,

dynamic analysis: $\top : \langle \boldsymbol{A} \rangle (C \sqcap \langle \boldsymbol{updatedBy}.\top \rangle \neg C) \sqsubseteq C^{\star}$
$C^{\star}$ retrieves all instances which are C in some version of type $\boldsymbol{A}$ and evolve into $\neg C$ in some immediate updated version,

evolvability constraints: $\boldsymbol{A} : C \sqsubseteq \langle \boldsymbol{updatedBy}.\boldsymbol{B} \rangle D$
in any version of type $\boldsymbol{A}$, every instance of C has to evolve into D in some immediate updated version of type $\boldsymbol{B}$.

Examples like the ones above demonstrate how DLCs can be used to establish the semantic relationships of different local views which are given in terms of standard DLs representations. Note that, as in the divide-and-conquer scenario, the context operators are used to control the information flow from one context to another, i.e. they are the main mechanism involved in the bridge rules. Recall that one of the key feature of the compose-and-conquer philosophy is that the local views are autonomous. Hence, the ability to describe particular properties of the local views using the metalanguage plays a central role in the successful establishment of their semantic interoperability and coordination. In particular, it has been argued that the description of each source or view requires not only tags naming the source but also complex properties [BTMS10, Haa06], e.g. for expressing 1) provenance (authorship, date, place), 2) description of the source (keywords, topic, name), 3) relationships of the source to other sources, etc.

3.5 Formal properties

In this section we touch upon two basic formal properties of DLCs: expressiveness and complexity of reasoning. In addressing these issues we rely heavily on the fact that the DLC framework is grounded in the well-known two-dimensional DLs [WZ99]. Having such properly established mathematical foundations provides us with two kinds of benefits. Firstly, it allows for a rough demarcation of expressive limits of the DLCs by direct comparisons to related formalisms, which have already been investigated in the literature. Secondly, it enables the adoption of some known proof techniques for studying computational properties of the framework. The results which we deliver here are not exhaustive, but nevertheless, they offer a good limiting characterization of the proposed logics. We show that the expressive power of the full DLC framework properly subsumes the expressiveness of the two-dimensional DLs $(\mathbf{K}_n)_\mathcal{L}$ and $\mathbf{S5}_\mathcal{L}$ and that the problem of satisfiability of $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-knowledge bases, for $\mathcal{L}_C$ and $\mathcal{L}_O$ up to the DL $\mathcal{SHIO}$, is decidable — in fact, 2EXPTIME-complete. Finally, we show that for certain fragments of DLCs, involving only the operators $\mathfrak{F}_2$, the complexity of the satisfiability problem can be lowered to NEXPTIME- and even EXPTIME-complete.

3.5.1 Relationships to other logics

We consider two-dimensional DL languages with modal operators applied only to concepts.

Definition 6 (Two-dimensional DL). *Let $\mathcal{L}$ be a DL language and $\Diamond_i$, $\Box_i$, for $i \in (1, n)$, be a set of n pairs of modalities of a modal logic L. Then a two-dimensional DL concept language $L_\mathcal{L}$ is the smallest set of concepts closed under constructors of $\mathcal{L}$ and two concept constructors:*

$$\Diamond_i C \quad | \quad \Box_i C$$

for any concept $C \in L_\mathcal{L}$.

An interpretation of $L_\mathcal{L}$ is a tuple $(\mathfrak{W}, \{R_i\}_{i\in(1,n)}, \Delta, \{\cdot^{\mathcal{I}(w)}\}_{w\in\mathfrak{W}})$, where:

- *$\mathfrak{W}$ is a non-empty set of possible worlds,*
- *each R_i, for $i \in (1, n)$, is an accessibility relation over $\mathfrak{W}$ associated with the operators $\Diamond_i, \Box_i$,*
- *Δ is a non-empty domain of individuals,*
- *for every $w \in \mathfrak{W}$, $\cdot^{\mathcal{I}(w)}$ is a DL interpretation over Δ in the world w, such that:*
  - $(\Diamond_i C)^{\mathcal{I}(w)} = \{x \in \Delta \mid \exists v : wR_iv \wedge x \in C^{\mathcal{I}(v)}\}$,
  - $(\Box_i C)^{\mathcal{I}(w)} = \{x \in \Delta \mid \forall v : wR_iv \to x \in C^{\mathcal{I}(v)}\}$.

The logic $(\mathbf{K}_n)_\mathcal{L}$ is defined as an extension of a DL language $\mathcal{L}$ with n pairs of $\mathbf{K}$-modalities, i.e. operators associated with arbitrary relations $R_i \subseteq \mathfrak{W} \times \mathfrak{W}$. Analogically, the logic $\mathbf{S5}_\mathcal{L}$ augments $\mathcal{L}$ with a single pair of $\mathbf{S5}$-modalities associated with the universal relation over $\mathfrak{W}$. We consider the problem of *concept satisfiability w.r.t. global TBoxes*, i.e. the problem of deciding whether given a concept C and a TBox $\mathcal{T}$ in $L_\mathcal{L}$ there exists an interpretation of $L_\mathcal{L}$ such that every axiom in $\mathcal{T}$ is satisfied in every possible world in the interpretation and there exists at least one possible world $w \in \mathfrak{W}$ and an individual $x \in \Delta$ such that $x \in C^{\mathcal{I}(w)}$.

Definitions 4 and 6 reflect the formal relationship between the interpretations of two-dimensional DLs and those of DLCs. Essentially, the latter structures are built strictly on top of the former in the following sense. For every two-dimensional DL interpretation $(\mathfrak{W}, \{R_i\}_{i\in(1,n)}, \Delta, \{\cdot^{\mathcal{I}(w)}\}_{w\in\mathfrak{W}})$ there exist infinitely many different $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-interpretations $(\Theta, \mathfrak{C}, \cdot^{\mathcal{J}}, \Delta, \{\cdot^{\mathcal{I}(i)}\}_{i\in\mathfrak{C}})$, such that $\mathfrak{C} = \mathfrak{W}$ and $(\Theta, \cdot^{\mathcal{J}})$ is an interpretation of the context language, such that for every non-universal accessibility relation R_i, there exists a context role $\boldsymbol{r}_i$ with $R_i = \boldsymbol{r}_i^{\mathcal{J}} \cap \mathfrak{C} \times \mathfrak{C}$. Conversely, for every $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-interpretation there exists a unique two-dimensional DL interpretation contained in it with $\mathfrak{C} = \mathfrak{W}$. The comparison of the expressive power of DLCs, which follows in this chapter, hinges exactly on this observation. It aims at demonstrating that since semantic structures associated with DLCs are in fact just "enriched" two-dimensional DL interpretations, the DLC languages, which extend two-dimensional DLs with additional syntactic constructs, can naturally express more properties over such structures.

In the following theorem we show that for $(\mathbf{K}_n)_{\mathcal{L}}$ the problem of concept satisfiability w.r.t. global TBoxes can be immediately restated as the problem of knowledge base satisfiability in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ with $\mathfrak{F}_1$ operators. Notably, this correspondence holds regardless of whether object roles are interpreted rigidly in both types of logics, i.e. $r^{\mathcal{I}(w)} = r^{\mathcal{I}(v)}$, for every $r \in N_R$ and $w, v \in \mathfrak{W}$, as in proper products of modal logics, or only locally, as defined in the semantics of $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ presented in this chapter.

Theorem 1 (($(\mathbf{K}_n)_{\mathcal{L}}$ vs. $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$)**.** *Deciding concept satisfiability w.r.t. a global TBox in $(\mathbf{K}_n)_{\mathcal{L}}$ is linearly reducible to knowledge base satisfiability in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$, for $\mathcal{L}_O = \mathcal{L}$, with the context operators of type $\mathfrak{F}_1$ only, regardless whether object roles are interpreted rigidly or locally.*

Proof. Let $(C, \mathcal{T})$ be a problem instance in $(\mathbf{K}_n)_{\mathcal{L}}$. Define the corresponding knowledge base $\mathcal{K} = (\mathcal{C}, \mathcal{O})$ in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ as follows. First, set $\mathcal{C} = \emptyset$ and $\mathcal{O} = \{\top : B \sqsubseteq D \mid B \sqsubseteq D \in \mathcal{T}\} \cup \{\top : (\langle \boldsymbol{s}.\top\rangle C)(a)\}$, for a context role $\boldsymbol{s}$ and some fresh individual object name a. Then, with every pair of **K**-modalities $\Diamond_i, \Box_i$ in $(\mathbf{K}_n)_{\mathcal{L}}$ associate a distinct context role name $\boldsymbol{r}_i$ and replace every occurrence of $\Diamond_i$ in $\mathcal{O}$ with $\langle \boldsymbol{r}_i.\top\rangle$ and every occurrence of $\Box_i$ with $[\boldsymbol{r}_i.\top]$. Then, C is satisfiable w.r.t. $\mathcal{T}$ in $(\mathbf{K}_n)_{\mathcal{L}}$ *iff* the resulting knowledge base $\mathcal{K}$ is satisfiable in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$. This conclusion follows immediately by observing the direct correspondence between the semantics of both languages, in particular the semantics of the $\mathbf{K}$-modalities and global TBox axioms in $(\mathbf{K}_n)_{\mathcal{L}}$ and of the corresponding $\mathfrak{F}_1$ operators and formulas $\top : \varphi$ in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$. Based on this observation, it is easy to see that $(\mathfrak{W}, \{R_i\}_{i\in(1,n)}, \Delta, \{\cdot^{\mathcal{I}(w)}\}_{w\in\mathfrak{W}})$ is a model of $\mathcal{T}$ *iff* $(\Theta, \mathfrak{W}, \cdot^{\mathcal{J}}, \Delta, \{\cdot^{\mathcal{I}(w)}\}_{w\in\mathfrak{W}})$ is a model of $\mathcal{K}$, where $R_i = (\boldsymbol{r}_i)^{\mathcal{J}}$, for every $i \in (1, n)$, and the concept C is satisfied in some $w \in \mathfrak{W}$ by the individual $a^{\mathcal{I}(w)} \in \Delta$. ❑

In the same manner, we devise a reduction from $\mathbf{S5}_{\mathcal{L}}$ to $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ with $\mathfrak{F}_2$ operators.

Theorem 2 **($\mathbf{S5}_{\mathcal{L}}$ vs. $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$).** *Deciding concept satisfiability w.r.t. a global TBox in* $\mathbf{S5}_{\mathcal{L}}$ *is linearly reducible to knowledge base satisfiability in* $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$*, for* $\mathcal{L}_O = \mathcal{L}$*, with the context operators of type* $\mathfrak{F}_2$ *only, regardless whether object roles are interpreted rigidly or locally.*

Proof. Let $(C, \mathcal{T})$ be a problem instance in $\mathbf{S5}_{\mathcal{L}}$. Again, define the knowledge base $\mathcal{K} = (\mathcal{C}, \mathcal{O})$ in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ by setting $\mathcal{C} = \emptyset$ and $\mathcal{O} = \{\top : B \sqsubseteq D \mid B \sqsubseteq D \in \mathcal{T}\} \cup \{\top : (\langle\top\rangle C)(a)\}$, for some fresh individual name a. Then, replace every occurrence of $\Diamond$ in $\mathcal{O}$ with $\langle\top\rangle$ and every occurrence of $\Box$ with $[\top]$. Consequently, C is satisfiable w.r.t. $\mathcal{T}$ in $\mathbf{S5}_{\mathcal{L}}$ *iff* the resulting knowledge base $\mathcal{K}$ is satisfiable in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$. Analogically to the previous case, observe that the semantics of $\mathbf{S5}$-modalities coincides with that of $\mathfrak{F}_2$ operators and so $(\mathfrak{W}, R, \Delta, \{\cdot^{\mathcal{I}(w)}\}_{w\in\mathfrak{W}})$ is a model of $\mathcal{T}$ *iff* $(\Theta, \mathfrak{W}, \cdot^{\mathcal{J}}, \Delta, \{\cdot^{\mathcal{I}(w)}\}_{w\in\mathfrak{W}})$ is a model of $\mathcal{K}$, where R is the universal relation over $\mathfrak{W}$ and the concept C is satisfied in some $w \in \mathfrak{W}$ by the individual $a^{\mathcal{I}(w)} \in \Delta$. ❑

Observe that for the reductions we use only a residual context language. In the former case we merely require the top concept and a set of context role names, while in the latter only the top concept is used. Clearly, there is also no need for employing axioms of the context language. This suggests that the expressive power of DLCs might be in general even greater and strictly subsume that of the union of $(\mathbf{K}_n)_{\mathcal{L}}$ and $\mathbf{S5}_{\mathcal{L}}$. Indeed, it is not difficult to instantiate this intuition with concrete examples of properties which are expressible in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ but cannot be captured by any of the underlying two-dimensional languages. For instance, context names allow for introducing certain forms of functional modalities pointing at uniquely identifiable possible worlds, as done e.g., in axioms of type $\boldsymbol{c} : \varphi$, where the constraint φ is placed exactly over the world named $\boldsymbol{c}$. By allowing nominals in the context language, one can further exploit this expressive capability, for instance, to impose cardinality constraints over the possible worlds domain:[3]

$$\top \sqsubseteq \{\boldsymbol{c}\} \sqcup \{\boldsymbol{d}\}, \quad \{\boldsymbol{c}\} \sqcap \{\boldsymbol{d}\} \sqsubseteq \bot,$$

The context language supports also construction of other complex modalities, as e.g., in the concept:

$$\langle \boldsymbol{A} \rangle C \sqcup [\boldsymbol{A} \sqcap \neg \boldsymbol{B}]C,$$

which describes the set of objects which are C in any context of type $\boldsymbol{A}$ or in all contexts of type $\boldsymbol{A}$ and $\neg\boldsymbol{B}$. Obviously neither $(\mathbf{K}_n)_{\mathcal{L}}$ or $\mathbf{S5}_{\mathcal{L}}$, nor any of the standard two-dimensional DLs, allows for expressing such properties, as they require a more fine-grained mechanism of quantifying over possible worlds, offered by the context language in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$.

Our next result suggests, that such a behavior of $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ can be to some extent simulated in two-dimensional DLs extended with global concepts, i.e. concepts C such that for every $w \in \mathfrak{W}$ it either holds that $\alpha^{\mathcal{I}(w)} = \Delta$ or $\alpha^{\mathcal{I}(w)} = \emptyset$. Technically, rigid concepts can be seen as context concepts. However, even if a complete reduction of $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ to two-dimensional DLs with global concepts was possible, this approach would be still conceptually inadequate to our motivation, as the semantics of global expressions would be defined purely in terms of the object domain and not the domain of contexts. Moreover, the interaction between the two levels of representation would be highly obscured, making it hard to define fragments of $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ in a modular fashion — simply by selecting DLs of desired expressiveness for $\mathcal{L}_C$ and $\mathcal{L}_O$. Nevertheless, to give a final insight into the expressiveness of DLCs, we show that $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ with operators of type $\mathfrak{F}_1$ and rigid interpretation of roles is equally expressive to the full $\mathcal{ALC}$ language over the union of two vocabularies interpreted in product models, where one sort of concepts is interpreted globally. This shows, similar to Theorems 1 and 2, that $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ (in its concept component) does not seriously deviate, at least in the technical sense, from the usual products of modal logics. In principle, the only feature distinguishing it from $(\mathbf{K}_n)_{\mathcal{ALC}}$ (both with and without rigid roles) is the condition (†) imposed on the interpretations of selected concepts, which in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ we happen to call context concepts. What also follows

[3]One could therefore argue that context names introduce some characteristic features of hybrid logic over the possible world dimension in the same way as individual names introduce them over the object dimension [AdR01]. From this perspective, the DLC languages come close to product-like combinations of DLs with hybrid logics, or even more — to combinations of pairs of hybrid logics [San10].

from this result is that the syntactic constraints of $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$, mainly regarding the adopted shape of context operators, which make the logic more intuitive and well-behaved, by no means lead to loss of expressiveness.

Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be two $\mathcal{ALC}$ concept languages over disjoint vocabularies $\Gamma = (M_C, M_R, \emptyset)$ and $\Sigma = (N_C, N_R, \emptyset)$, respectively. Now, let $\mathcal{L}_{1\times 2}$ be the $\mathcal{ALC}$ concept language over vocabulary $\Theta = (M_C \cup N_C, M_R \cup N_R, \emptyset)$. The semantics for $\mathcal{L}_{1\times 2}$ is given through *product interpretations* $\mathcal{P} = (\mathfrak{C} \times \Delta, \cdot^{\mathcal{P}})$, which align every $r \in N_R$ along the 'vertical' dimension and every $\boldsymbol{p} \in M_R$ along the 'horizontal' one. Thus, $r^{\mathcal{P}}, \boldsymbol{p}^{\mathcal{P}} \subseteq (\mathfrak{C} \times \Delta) \times (\mathfrak{C} \times \Delta)$ and for every $u, v, w \in \mathfrak{C}$ and $x, y, z \in \Delta$:

$$\langle (u,x),(v,y)\rangle \in r^{\mathcal{P}} \rightarrow u = v \;\&\; \langle (w,x),(w,y)\rangle \in r^{\mathcal{P}},$$
$$\langle (u,x),(v,y)\rangle \in \boldsymbol{p}^{\mathcal{P}} \rightarrow x = y \;\&\; \langle (u,z),(v,z)\rangle \in \boldsymbol{p}^{\mathcal{P}}.$$

All concepts are interpreted as subsets of $\mathfrak{C} \times \Delta$. Additionally, we force every $\boldsymbol{A} \in M_C$ to be interpreted rigidly across the 'vertical' dimension, i.e., for every $v \in \mathfrak{C}$ and $x, y \in \Delta$ we assume:

$$(v,x) \in \boldsymbol{A}^{\mathcal{I}} \rightarrow (v,y) \in \boldsymbol{A}^{\mathcal{I}} \qquad (\dagger)$$

Finally, $\cdot^{\mathcal{P}}$ is extended inductively as usual. A concept $C \in \mathcal{L}_{1\times 2}$ is satisfiable *iff* for some product model $\mathcal{P} = (\mathfrak{C} \times \Delta, \cdot^{\mathcal{P}})$ it is the case that $C^{\mathcal{P}} \neq \emptyset$. On the contrary to the others, the condition (†) is rather uncommon in the realm of products of modal logics. Nevertheless, it captures precisely the difference between the semantics of the two sorts of concepts. Without it the sorts collapse into one, while the whole logic turns into a notational variant of $(\mathbf{K}_n)_{\mathcal{ALC}}$. It turns out that the following claim holds:

Theorem 3 ($\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$ vs. modal products)**.** *The language $\mathcal{L}_{1\times 2}$ interpreted in product models is exactly as expressive as the concept language of $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$ with operators of type $\mathfrak{F}_1$ interpreted in models with rigid interpretation of object roles.*

Proof. To prove the claim we will show that (1) for every $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$ concept D there is a concept $C \in \mathcal{L}_{1\times 2}$ and, conversely, (2) for every concept $C \in \mathcal{L}_{1\times 2}$ there is an $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$ concept D, such that C is satisfied in some product model *iff* D is satisfied in some $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$ model in which object roles are interpreted rigidly. Note that we consider C and D to be arbitrary syntactically well-formed concepts. In case of $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$ this includes both context and object concepts.

We start by defining a mapping between two kinds of interpretations w.r.t. the vocabulary in C and D. We say that a product interpretation $\mathcal{P}$ and an $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$

interpretation $\mathfrak{M}$ in which object roles are interpreted rigidly are *matching iff* $\mathcal{P} = (\mathfrak{C} \times \Delta, \cdot^{\mathcal{P}})$, $\mathfrak{M} = (\mathfrak{C}, \cdot^{\mathcal{J}}, \Delta, \{\cdot^{\mathcal{I}(i)}\}_{i \in \mathfrak{C}})$, and the functions $\cdot^{\mathcal{P}}$, $\cdot^{\mathcal{J}}$ and $\cdot^{\mathcal{I}}$ are related as follows:

- for every $\boldsymbol{p}$: $\langle v, w \rangle \in \boldsymbol{p}^{\mathcal{J}}$ *iff* $\langle (v, x), (w, y) \rangle \in \boldsymbol{p}^{\mathcal{P}}$, for any $x, y \in \Delta$,
- for every $\boldsymbol{A}$: $v \in \boldsymbol{A}^{\mathcal{J}}$ *iff* $(v, x) \in \boldsymbol{A}^{\mathcal{P}}$, for any $x \in \Delta$,
- for every r and $i \in \mathfrak{C}$: $\langle x, y \rangle \in r^{\mathcal{I}(i)}$ *iff* $\langle (i, x), (i, y) \rangle \in r^{\mathcal{P}}$,
- for every A and $i \in \mathfrak{C}$: $x \in A^{\mathcal{I}(i)}$ *iff* $(i, x) \in A^{\mathcal{P}}$.

Obviously every product interpretation has a matching $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$ interpretation and vice versa.

Case (1) is straightforward. Let D be an $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$ concept. Apply the following rules to all subconcepts D' of D:

- if $D' = \langle \boldsymbol{B} \rangle_{\boldsymbol{p}} C$ then replace it with $\exists \boldsymbol{p}.(\boldsymbol{B} \sqcap C)$,
- if $D' = [\boldsymbol{B}]_{\boldsymbol{p}} C$ then replace it with $\forall \boldsymbol{p}.(\neg \boldsymbol{B} \sqcup C)$.

Let C be the result of the transformation. Clearly, C is a well-formed $\mathcal{L}_{1 \times 2}$. By structural induction on the concepts it is easy to see that if D is satisfied in some $\mathfrak{M}$ then C is satisfied in the matching product interpretation, and if C is satisfied in some $\mathcal{P}$ then D is satisfied in the matching $\mathcal{ALC}_{\mathcal{ALC}}$ interpretation. In particular, if $x \in (\langle \boldsymbol{B} \rangle_{\boldsymbol{p}} C)^{\mathcal{I}(c)}$ then for some j we have: $j \in \boldsymbol{B}^{\mathcal{J}}$, $\langle i, j \rangle \in \boldsymbol{p}^{\mathcal{J}}$ and $x \in C)^{\mathcal{I}(j)}$. This to the matching product model, where $(j, y) \in \boldsymbol{B}^{\mathcal{P}}$, $\langle (c, y), (c', y) \rangle \in \boldsymbol{p}^{\mathcal{P}}$, for all $y \in \Delta$, and $(j, x) \in C^{\mathcal{P}}$. Similarly in the opposite direction.

Case (2) is a bit more tedious. Basically, we need to first transform an $\mathcal{L}_{1 \times 2}$ concept into a form in which concepts of M_C occur right after the restrictions on roles M_R. Then we can smoothly translate them into $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$ following the opposite transformation to the one used in case (1). Let $C \in \mathcal{L}_{1 \times 2}$. W.l.o.g. we can assume that $C = \exists \boldsymbol{s}.C'$ for some role name $\boldsymbol{s} \in M_R$ and C' in NNF. We say that a concept B is:

1. in Conjunctive Normal Form (CNF) *iff* $B = \bigsqcap_i \bigsqcup_j B_{ij}$,
2. in Disjunctive Normal Form (DNF) *iff* $B = \bigsqcup_i \bigsqcap_j B_{ij}$,
3. in clausal form *iff* $B = \bigsqcup_i \boldsymbol{B}_i \sqcup \bigsqcup_j B_j$,

4. in conjunctive form *iff* $B = \bigsqcap_i \boldsymbol{B}_i \sqcap \bigsqcap_j B_j$,

where every B_{ij} is a role restriction or a literal, i.e., a concept name, its negation, $\bot$ or $\top$; every B_j is a Σ-literal (including $\bot$ and $\top$), or a role restriction (on any role from $N_R \cup M_R$); every $\boldsymbol{B}_i$ is a Γ-literal (excluding $\bot$ and $\top$). First, we perform a number of equivalence preserving transformations on C. We follow the structure of nestings of role restrictions, starting from the innermost restrictions and proceeding inside-out. On the way we exhaustively apply the π rule:

1. for $\exists r.B$:
   (a) if B is in conjunctive form and $B = \bigsqcap_i \boldsymbol{B}_i \sqcap \bigsqcap_j B_j$, then: $\pi(\exists r.B) = \bigsqcap_i \boldsymbol{B}_i \sqcap \exists r. \bigsqcap_j B_j$,
   (b) if B is in DNF and $B = \bigsqcup_i \bigsqcap_j B_{ij}$ then: $\pi(\exists r.B) = \bigsqcup_i \exists r. \bigsqcap_j B_{ij}$,
   (c) else transform B to DNF and repeat.
2. for $\forall r.B$:
   (a) if B is in clausal form and $B = \bigsqcup_i \boldsymbol{B}_i \sqcup \bigsqcup_j B_j$, then: $\pi(\forall r.B) = \bigsqcup_i \boldsymbol{B}_i \sqcup \exists r. \bigsqcup_j B_j$,
   (b) if B is in CNF and $B = \bigsqcap_i \bigsqcup_j B_{ij}$ then: $\pi(\forall r.B) = \bigsqcap_i \forall r. \bigsqcup_j B_{ij}$,
   (c) else transform B to CNF and repeat.
3. for $\exists \boldsymbol{p}.B$:
   (a) if B is in DNF and $B = \bigsqcup_i \bigsqcap_j B_{ij}$ then: $\pi(\exists \boldsymbol{p}.B) = \bigsqcup_i \exists \boldsymbol{p}. \bigsqcap_j B_{ij}$,
   (b) else transform B to DNF and repeat.
4. for $\forall \boldsymbol{p}.B$:
   (a) if B is in CNF and $B = \bigsqcap_i \bigsqcup_j B_{ij}$ then: $\pi(\forall \boldsymbol{p}.B) = \bigsqcap_i \forall \boldsymbol{p}. \bigsqcup_j B_{ij}$,
   (b) else transform B to CNF and repeat.

Note, that given the difference in the semantics of concept names from N_C and M_C, steps 1a and 2a also preserve equivalence. As a result we obtain a concept in which all concept names of M_C occur only on the first depth inside restrictions on roles M_R. Moreover, in all concepts of the form $\exists \boldsymbol{p}.B$, B is in conjunctive form and in all concepts of the form $\forall \boldsymbol{p}.B$, B is in clausal form. Using these observations we can translate the outcome of the transformation into the syntax of $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ by applying the following rules:

1. for $\exists \boldsymbol{p}.B$ where $B = \bigsqcap_i \boldsymbol{B}_i \sqcap \bigsqcap_j B_j$:
   (a) if $i \neq 0$ and $j \neq 0$ then $\pi(\exists \boldsymbol{p}.B) = \langle \bigsqcap_i \boldsymbol{B}_i \rangle_{\boldsymbol{p}} \bigsqcap_j B_j$,
   (b) if $i = 0$ then $\pi(\exists \boldsymbol{p}.B) = \langle \top \rangle_{\boldsymbol{p}} \bigsqcap_j B_j$,
   (c) if $j = 0$ then $\pi(\exists \boldsymbol{p}.B) = \langle \bigsqcap_i \boldsymbol{B}_i \rangle_{\boldsymbol{p}} \top$.
2. for $\forall \boldsymbol{p}.B$ where $B = \bigsqcup_i \boldsymbol{B}_i \sqcap \bigsqcup_j B_j$:
   (a) if $i \neq 0$ and $j \neq 0$ then $\pi(\forall \boldsymbol{p}.B) = [\neg(\bigsqcup_i \boldsymbol{B}_i)]_{\boldsymbol{p}} \bigsqcup_j B_j$,
   (b) if $i = 0$ then $\pi(\forall \boldsymbol{p}.B) = [\top]_{\boldsymbol{p}} \bigsqcup_j B_j$,
   (c) if $j = 0$ then $\pi(\exists \boldsymbol{p}.B) = [\neg(\bigsqcup_i \boldsymbol{B}_i)]_{\boldsymbol{p}} \bot$.

Let D be the result of the translation. Clearly, D is an $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$ concept. Again it is not difficult to find out by structural induction on the concepts that if D is satisfied in some $\mathfrak{M}$ then C is satisfied in the matching product interpretation, and if C is satisfied in some $\mathcal{P}$ then D is satisfied in the matching $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$ interpretation. ❑

3.5.2 Complexity

What follows immediately from Theorems 1 and 3 is that deciding concept satisfiability w.r.t. global TBoxes in $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$ with operators of type $\mathfrak{F}_1$ interpreted in models with rigid interpretation of object roles, must belong to the same complexity class as in the case of proper products of DLs with $\mathbf{K}_n$. This class, however, turns out to be an undecidable one in general. We demonstrate this by a straightforward reduction of the $\mathbb{N} \times \mathbb{N}$-tiling problem [KWZG03]. For greater generality we focus on the logic $(\mathbf{DAlt}_n)_{\mathcal{L}}$, which given its specific frame conditions allows to quickly transfer obtained results also to several other combinations of DLs with modal logics. We first show that $(\mathbf{DAlt}_n)_{\mathcal{L}}$ can be reduced to $(\mathbf{K}_n)_{\mathcal{L}}$, which in turn, as shown in Theorem 1, can be embedded in $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$, for $\mathcal{L}_O = \mathcal{L}$. The logic $(\mathbf{DAlt}_n)_{\mathcal{L}}$ extends a DL $\mathcal{L}$ with a set of functional modalities $\bigcirc_i$, i.e. operators associated with accessibility relations R_i satisfying the properties of seriality (**D**) and quasi-functionality (**Alt**):

(seriality) $\forall w \in \mathfrak{W}\, \exists v \in \mathfrak{W}\, (wR_iv)$,

(quasi-functionality) $\forall w, v, u \in \mathfrak{W}\, (wR_iv \wedge wR_iu \rightarrow v = u)$.

It is easy to show that there exists a chain of straightforward reductions relating the logics between $(\mathbf{DAlt}_n)_{\mathcal{L}}$ and $(\mathbf{K}_n)_{\mathcal{L}}$, including also $(\mathbf{D}_n)_{\mathcal{L}}$, based on serial frames, and $(\mathbf{Alt}_n)_{\mathcal{L}}$, based on quasi-functional frames.

Proposition 1. *Concept satisfiability w.r.t. global TBoxes is polynomially reducible between the following logics (where $\mapsto$ means* reduces to*):*

$$(\mathbf{DAlt}_n)_{\mathcal{L}} \mapsto \{(\mathbf{D}_n)_{\mathcal{L}}, (\mathbf{Alt}_n)_{\mathcal{L}}\} \mapsto (\mathbf{K}_n)_{\mathcal{L}}.$$

Proof. To see that the reductions indeed hold, it is sufficient to notice that the properties of seriality and quasi-functionality can be axiomatized (or at least emulated) in the languages of the considered logics. Hence, if $(C, \mathcal{T})$ is an instance of the concept satisfiability problem w.r.t. a global TBox in some lefthandside logic, then one can decide it in a righthandside logic by applying simple transformations of C and $\mathcal{T}$ which encode the missing conditions and thus allow for enforcing only models which are bisimilar to those of the original logic:

(seriality) Let $\mathcal{T}' = \mathcal{T} \cup \{\top \sqsubseteq \Diamond_i \top \mid i \in (1, n)\}$, where n is the number of all modalities occurring in $\mathcal{T}$ and C. Then, $(C, \mathcal{T})$ is satisfiable on a serial frame *iff* $(C, \mathcal{T}')$ is satisfiable.

(quasi-functionality) W.l.o.g. assume that $C = \text{NNF}(C)$, where NNF stands for Negation Normal Form, and $\mathcal{T} = \{\top \sqsubseteq C_{\mathcal{T}}\}$, for some $C_{\mathcal{T}} = \text{NNF}(C_{\mathcal{T}})$. Let C' and $C'_{\mathcal{T}}$ be the result of replacing every subconcept $\Diamond_i B$ occurring in C and $C_{\mathcal{T}}$, respectively, with $(\Diamond_i \top) \sqcap (\Box_i B)$. Then, $(C, \mathcal{T})$ is satisfiable on a quasi-functional frame *iff* $(C', \{\top \sqsubseteq C'_{\mathcal{T}}\})$ is satisfiable. ❑

Theorem 4 (Undecidability of $\mathbf{DAlt}_{\mathcal{ALC}}$)**.** *Concept satisfiability in $\mathbf{DAlt}_{\mathcal{ALC}}$ w.r.t. global TBoxes and with a single rigid role is undecidable.*

First, we observe the following correspondence:

Proposition 2. *A concept C is satisfiable w.r.t. a global TBox $\mathcal{T}$ in $\mathbf{DAlt}_{\mathcal{ALC}}$* iff *it is satisfied w.r.t. $\mathcal{T}$ in some model $\mathfrak{M} = (\mathbb{N}, <, \Delta, \{\cdot^{\mathcal{I}(i)}\}_{i \in \mathbb{N}})$, where $\langle \mathbb{N}, < \rangle$ is a linear order over natural numbers and $<$ is the accessibility relation of $\bigcirc$.*

Consequently, we can consider only such linear $\mathbf{DAlt}_{\mathcal{ALC}}$-models. This shows that $\mathbf{DAlt}_{\mathcal{ALC}}$ can be in fact seen as the subset of $\text{LTL}_{\mathcal{ALC}}$ consisting of the $\mathcal{ALC}$ component and the *next-time* operator. This turns out to be enough to encode the undecidable $\mathbb{N} \times \mathbb{N}$ *tiling problem*, in the same way as in [LWZ08, Theorem 4]. An instance of the problem is defined as follows: given a finite set $S = \{t_0, \ldots, t_n\}$ of tile types, where each t_i is a 4-tuple of colors $\langle \mathit{left}(t_i), \mathit{right}(t_i), \mathit{up}(t_i), \mathit{down}(t_i) \rangle$, decide whether it is possible to cover $\mathbb{N} \times \mathbb{N}$-grid with tiles of these types. Moreover, it has to be ensured that only types of matching colors can be horizontal (vertical) neighbors in the tiling, i.e., ones for which $right(t_i) = left(t_j)$ ($up(t_i) = down(t_j)$). Let $A_0, \ldots, A_n$ be concept names representing the tile types from S and r be a rigid role. The following TBox $\mathcal{T}$ encodes the constraints of the tiling problem:

$$\top \sqsubseteq (\bigsqcup_{i \leq n} A_i) \sqcap (\bigsqcap_{i \neq j \leq n} \neg(A_i \sqcap A_j)) \tag{3.1}$$

$$\top \sqsubseteq \exists r.\top \tag{3.2}$$

$$A_i \sqsubseteq \forall r. \bigsqcup_{up(t_i)=down(t_j)} A_j, \text{ for every} \quad i \leq n \tag{3.3}$$

$$A_i \sqsubseteq \bigcirc \bigsqcup_{right(t_i)=left(t_j)} A_j, \text{ for every} \quad i \leq n \tag{3.4}$$

Now we can prove the target claim:

Lemma 1. *The concept $\top$ is satisfiable w.r.t. $\mathcal{T}$* iff *there exists a tiling $\tau : \mathbb{N} \times \mathbb{N} \mapsto S$.*

Proof. ($\Rightarrow$) Let $\mathfrak{M} = (\mathbb{N}, <, \Delta, \{\cdot^{\mathcal{I}(i)}\}_{i \in \mathbb{N}})$ be a model of $\mathcal{T}$. For an arbitrary individual $d \in \Delta$ we first fix the vertical axis ρ of the $\mathbb{N} \times \mathbb{N}$-grid:

- $\rho(0) = d$;
- $\rho(n+1) = e$, for any e such that $\langle \rho(n), e \rangle \in r^{\mathcal{I}(0)}$.

By the axiom (2) of $\mathcal{T}$, every individual in the domain has an r-successor, hence, it is easy to see that the infinite chain ρ can be extracted from the model. Moreover by (1) it follows that every individual satisfy exactly one of the concepts representing tile types.

- $\tau(n, m) = t_i$ iff $\rho(m) \in A_i^{\mathcal{I}(n)}$

Finally, since r is rigid the conditions (3) and (4) of the encoding sufficiently guarantee proper coloring of the neighbors.

($\Leftarrow$) For tiling τ define an interpretation $\mathfrak{M} = (\mathbb{N}, <, \Delta, \{\cdot^{\mathcal{I}(i)}\}_{i \in \mathbb{N}})$, where $\Delta = \{d_i \mid i \in \mathbb{N}\}$, and:

- $d_m \in A_i^{\mathcal{I}(n)}$ iff $\tau(n, m)$;

- $\langle d_n, d_{n+1} \rangle \in r^{\mathcal{I}(m)}$, for $n, m \in \mathbb{N}$.

Clearly $\top$ and all axioms from $\mathcal{T}$ are satisfied by $\mathcal{I}$ so we obtain a desired $\mathbf{DAlt}_{\mathcal{ALC}}$ model. ❑

Theorems 4 and 1, together with the reductions established in Proposition 1, immediately entail the target undecidability result.

Theorem 5 (Undecidability of $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ with rigid roles)**.** *Satisfiability of a knowledge base in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$, with at least context operators of type $\mathfrak{F}_1$ and $\mathcal{ALC} \preceq \mathcal{L}_O$ is undecidable for rigid interpretation of roles.*

This result reveals an obvious limitation of the formalism, but a limitation one has to live with, considering that combinations of rigid roles with global TBoxes are rarely decidable in two-dimensional DLs, unless the expressive power of the modal or the DL component is significantly reduced [KWZG03, LWZ08]. This also explains the general impossibility of applying context operators over (local) roles, as then their rigidity can be enforced by the use of the modal box operator. Although we have not proven it, we conjecture that the problem remains decidable, in fact 2EXPTIME-complete, when only operators of type $\mathfrak{F}_2$ are present. This result is likely to follow from the 2EXPTIME-completeness result for the satisfiability problem in $\mathbf{S5}_{\mathcal{ALC}}$ with rigid roles, obtained in [ALT07].

To regain decidability we must restrict our attention to the case of $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ with local interpretation of roles only. With this assumption we are able to demonstrate decidability of the knowledge base satisfiability problem in the most expressive fragment $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ studied in this thesis, including both type of context operators and $\mathcal{L}_O = \mathcal{L}_C = \mathcal{SHIO}$. For the proof we devise a *quasistate elimination algorithm*, similar to [KWZG03, Theorem 6.61], which extends the standard Pratt-style type elimination technique, commonly used in demonstrating upper bounds for modal logics. Essentially, instead of looking directly for a model of a knowledge base, we abstract from the possibly infinite domains $\mathfrak{C}$ and Δ and consider only a finite number of quasistates which represent possible types of contexts, inhabited by a finite number of possible types of objects. Further, all object types and all quasistates which do not satisfy certain criteria are iteratively eliminated. If at the end of the elimination process there are some non-empty quasistates left, it is guaranteed that a model exists. In the opposite case, the knowledge base is unsatisfiable. As the proof is quite involved we only sketch its key steps below, while full details are presented in Section A.1 of Appendix.

Theorem 6 (Upper bound). *Satisfiability of a knowledge base in* $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$*, for* $\mathcal{L}_O = \mathcal{L}_C = \mathcal{SHIO}$*, any combination of context operators* $\mathfrak{F}_1/\mathfrak{F}_2$ *and for local interpretation of object roles, is decidable in* 2EXPTIME.

Proof sketch. Let $\mathcal{K} = (\mathcal{C}, \mathcal{O})$ be a $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$-knowledge base whose satisfiability we want to decide. For simplicity we consider simplified semantics based only on models $\mathfrak{M} = (\Theta, \mathfrak{C}, \cdot^{\mathcal{J}}, \Delta, \{\cdot^{\mathcal{I}(i)}\}_{i\in\mathfrak{C}})$ with $\Theta = \mathfrak{C}$, abbreviated to $\mathfrak{M}^\star = (\Theta, \cdot^{\mathcal{J}}, \Delta, \{\cdot^{\mathcal{I}(i)}\}_{i\in\Theta})$. In fact, there exists a straightforward, linear reduction $\cdot^\star$ such that $\mathcal{K}$ is satisfiable with respect to the standard semantics *iff* $\mathcal{K}^\star$ is satisfiable with respect to the simplified semantics. We assume that $\mathcal{K}$ is given in such a reduced form and, moreover, that several satisfiability preserving transformations have been applied to $\mathcal{K}$, in order to reduce the number of syntactic cases to by addressed in the proof. Among others, we assume that all axioms $\boldsymbol{a} : \varphi \in \mathcal{O}$ are replaced with their equivalents $\{a\} : \varphi$. We use the following notation to mark the sets of symbols of particular types occurring in $\mathcal{K}$:

- $con_c(\mathcal{K})$: all context concepts, closed under negation,
- $con_o(\mathcal{K})$: all object concepts, closed under negation,
- $sub_o(\mathcal{K})$: all axioms in the set $\{\varphi \mid \boldsymbol{C} : \varphi \in \mathcal{O} \text{ for any } \boldsymbol{C}\}$.

Next, we introduce three central notions: context types, object types and quastistates.

A *context type* for $\mathcal{K}$ is a subset $c \subseteq con_c(\mathcal{K})$, where:

- $\boldsymbol{C} \in c$ iff $\neg\boldsymbol{C} \notin c$, for all $\boldsymbol{C} \in con_c(\mathcal{K})$,
- $\boldsymbol{C} \sqcap \boldsymbol{D} \in c$ iff $\{\boldsymbol{C}, \boldsymbol{D}\} \subseteq c$, for all $\boldsymbol{C} \sqcap \boldsymbol{D} \in con_c(\mathcal{K})$.

An *object type* for $\mathcal{K}$ is a subset $t \subseteq con_o(\mathcal{K})$, where:

- $C \in t$ iff $\neg C \notin t$, for all $C \in con_o(\mathcal{K})$,
- $C \sqcap D \in t$ iff $\{C, D\} \subseteq t$, for all $C \sqcap D \in con_o(\mathcal{K})$.

A *quasistate* for $\mathcal{K}$ is a tuple $q = \langle c_q, f_q, O_q \rangle$, where c_q is a context type for $\mathcal{K}$, $f_q \subseteq sub_o(\mathcal{K})$ and O_q is a non-empty set of object types for $\mathcal{K}$.

Intuitively, a context type represents a possible element of the context domain in a $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$-model. The precise identity of this element is irrelevant. What matters is only the set of concepts of the context language which could completely describe this element in a model of $\mathcal{K}$, where the context language is restricted only to the concepts (and their negations) explicitly occurring in $\mathcal{K}$. Analogically, an object type represents a full description of a possible element of the object domain. Finally, a quasistate captures a "slice" of a model representing one possible context inhabited by a set of possible objects.

Eventually we define the notion of quasimodel, which corresponds to a finitized abstraction of a $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-model. A *quasimodel* for $\mathcal{K}$ is a set $\mathfrak{N}$ of quasistates for $\mathcal{K}$ satisfying a number of specific "integrity" conditions. Most importantly, it has to be guaranteed that all axioms of $\mathcal{K}$ are satisfied by the appropriate types (object and context) in the appropriate quasistates. Also, it has to be ensured that for all types containing concepts based on some forms of existential restrictions ($\exists r.\cdot$, $\exists \boldsymbol{r}.\cdot$, $\langle \boldsymbol{r}.\cdot \rangle$, $\langle \cdot \rangle$) there exist suitable types that could possibly represent their matching successors in a model. Under these constraints we are then able to prove the key quasimodel lemma.

Lemma 2. *There is a quasimodel for* $\mathcal{K}$ iff *there is an* $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$*-model of* $\mathcal{K}$*.*

The basic, brute-force algorithm deciding whether a quasimodel for $\mathcal{K}$ exists starts by enumerating the set $\mathfrak{N}$ of all possible quasistates and then iteratively eliminates all those which violate any of the constraints mentioned above. If the elimination terminates returning a non-empty set of quasistates each containing at least one object type, then this set is guaranteed to be a quasimodel and the search is finished with the answer "$\mathcal{K}$ is satisfiable". Else, no quasimodel exists and the algorithm returns "$\mathcal{K}$ is unsatisfiable".

As the size of a quasimodel is at most double exponential in the size of $\mathcal{K}$, therefore the elimination procedure must terminate in at most double exponential time in the size of $\mathcal{K}$. Hence deciding satisfiability of a $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-knowledge base is in 2EXPTIME. ❑

It turns out that the 2EXPTIME upper bound obtained above is optimal in the assumed setting, at least whenever context operators $\mathfrak{F}_1$ are involved and $\mathcal{ALC} \preceq \mathcal{L}_O$. To demonstrate this we reduce the 2EXPTIME-hard *word problem* for exponentially space-bounded *Alternating Turing Machine* [CKS81] to the concept satisfiability problem in $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$. The full proof is presented in Section A.2 of Appendix.

Theorem 7. *Deciding concept satisfiability in* $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$ *w.r.t. global TBoxes and only with local roles is* 2EXPTIME*-hard.*

Proof sketch. Let $\mathcal{M} = (Q, \Sigma, \Gamma, q_0, \delta)$ be an Alternating Turing Machine (ATM), where:

- Q is a set of *states*, including existential, universal, halting, accepting and rejecting states,
- Σ is an *input alphabet* and Γ a *working alphabet*, where $\Sigma \subseteq \Gamma$,
- q_0 is the *initial state*,
- δ is a *transition relation*, which to every pair $(q, a) \in Q \times \Gamma$ assigns at least one triple $(q', b, m) \in Q \times \Gamma \times \{l, n, r\}$ describing the transition, where $m = l$ / $m = r$ indicates a shift of the head to the left/right, whereas $m = n$ indicates no shift.

A *configuration* of an ATM is a sequence $\omega q \omega'$, where $\omega\omega'$ is a word based on Σ, q is a state of the machine and the head of the machine is on the leftmost symbol of ω'. A succeeding configuration is defined by transitions δ. An ATM *computation tree* is a finite tree whose nodes are labeled with configurations, where:

- the root contains the *initial configuration* $q_0\omega$, where ω is of length n,
- every configuration $\omega q \omega'$ on the tree, where $\omega\omega'$ is of length at most 2^n, is succeed by:
  - at least one successor configuration, whenever q is an existential state,
  - all successor configurations, whenever q is a universal state,
- all leaves are labeled with halting configurations.

A tree is *accepting iff* all the leaves are labeled with accepting configurations and *rejecting* otherwise. An ATM *accepts* an input ω *iff* there exists an accepting ATM tree with $q_0\omega$ as its initial configuration.

To reduce the word problem, for a word ω over Σ, we formulate a global TBox $\mathcal{T}_\mathcal{M}$ and a concept $C_{\mathcal{M},\omega}$ in $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$, such that $\mathcal{M}$ accepts ω *iff* $C_{\mathcal{M},\omega}$ is satisfiable w.r.t. $\mathcal{T}_\mathcal{M}$. The size of the resulting problem instance $(C_{\mathcal{M},\omega}, \mathcal{T}_\mathcal{M})$ is at most polynomial in the size of $\mathcal{M}$ and ω. The reduction is quite involved and essentially relies on an extensive use of $\mathbf{DAlt}$-modalities. We define two separate sets of such operators:

alphabet modalities: $\bigcirc_a$, for every $a \in \Gamma$,

transition modalities: $\bigcirc_{q,a,m}$, for every $(q, a, m) \in \Theta$, where $\Theta = \{(q, a, m) \mid (q', b, q, a, m) \in \delta$ for any $b \in \Gamma$ and $q' \in Q\}$,

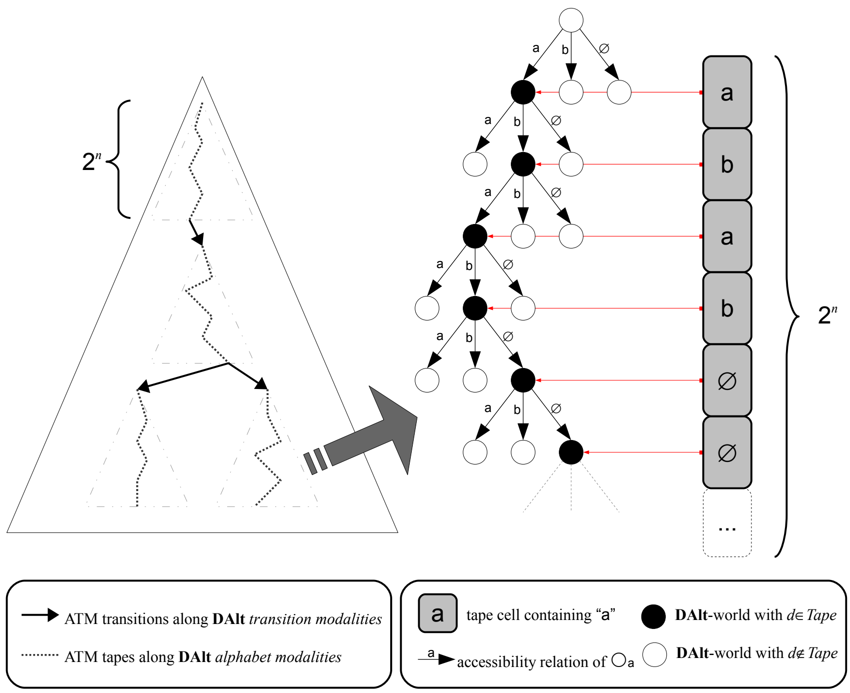

Figure 3.5: Embedding of ATM computation trees (left) and ATM tapes (right) in $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$-tree-models.

By a suitable use of these operators we are able to encode the complete syntactic structure of an ATM computation tree in the specific fragments of $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$-tree-models, as illustrated in Figure 3.5. In particular, a selected object domain individual $d \in \Delta$ is forced to instantiate the designated concept $Tape$ exactly in those **DAlt**-worlds which represent the cells of the ATM tape in the subsequent configurations. The accessibility relations connecting those worlds encode the content of the cells and the transitions between the configurations. Further, specific concepts are used to represent the corresponding positions of the head and the states of the machine. Finally, using special counting concepts, which enable traversing the ATM tree structure downwards and upwards, we align the succeeding configurations semantically, ensuring they satisfy the constraints of the respective transitions. ❑

This result grants immediately a lower complexity bound for $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$.

Theorem 8 (Lower bound)**.** *Deciding satisfiability of a knowledge base in* $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$*, for* $\mathcal{L}_O = \mathcal{ALC}$ *and arbitrary* $\mathcal{L}_C$*, with context operators* $\mathfrak{F}_1$ *and for local interpretation of object roles, is* 2EXPTIME*-hard.*

Proof. Immediate by Proposition 1 and Theorems 1 and A.2. ❑

As an interesting corollary, we also obtain a lower bound for the problem of concept satisfiability w.r.t. global TBoxes in several two-dimensional DLs considered above, most prominently in $(\mathbf{K}_n)_{\mathcal{ALC}}$.

Corollary 1. *For any* $L \in \{\mathbf{DAlt}_n, \mathbf{D}_n, \mathbf{Alt}_n, \mathbf{K}_n\}$*, deciding concept satisfiability in* $L_{\mathcal{ALC}}$ *w.r.t. global TBoxes and only with local roles is* 2EXPTIME*-hard.*

Proof. Immediate by Proposition 1 and Theorem A.2. ❑

This increase in the complexity by one exponential, as compared to $\mathcal{ALC}$ alone (for which the problem is EXPTIME-complete [BCM+03]), is notable and quite surprising. It could be expected that without rigid roles the satisfiability problem in two-dimensional DLs can be straightforwardly reduced to satisfiability in fusion models of conventional DLs. This in turn should yield EXPTIME upper bound by means of the standard techniques. However, as the following counterexample for $(\mathbf{K}_n)_{\mathcal{ALC}}$ shows, this strategy fails.

$$(\dagger)\ \Diamond_i C \sqcap \exists r.\Box_i \bot \qquad (\ddagger)\ \exists succ_i.C \sqcap \exists r.\forall succ_i.\bot$$

Although (†) clearly does not have a model, its reduction (‡) to a fusion language, where context operators are translated to restrictions on fresh $\mathcal{ALC}$ roles, is satisfiable. The reason is that while in the former case the information about the structure of the $\mathbf{K}$-frame is global for all individuals, in the latter it becomes local. The r-successor in (‡) is simply not 'aware' that it should actually have a $succ_i$-successor. This effect, amplified by presence of multiple modalities and global TBoxes (which can enforce infinite $\mathbf{K}$-trees), makes the reasoning harder. The result is quite robust under changes of domain assumptions and holds already in the case of expanding/varying domains in $(\mathbf{Alt}_n)_{\mathcal{ALC}}$. The only exception applies to $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$ and $(\mathbf{D}_n)_{\mathcal{ALC}}$ with expanding/varying domains, where reduction to $\mathcal{ALC}$ is still possible.

Both complexity bounds established above warrant the following consequences.

Theorem 9. *Deciding satisfiability of a knowledge base in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$, for $\mathcal{L}_C \preceq \mathcal{SHIO}$, $\mathcal{ALC} \preceq \mathcal{L}_O \preceq \mathcal{SHIO}$, with at least context operators $\mathfrak{F}_1$ (and possibly also $\mathfrak{F}_2$) and for local interpretation of object roles, is* 2EXPTIME*-complete.*

Proof. Immediate by Theorem 6 and Theorem 8. ❑

Corollary 2. *For any $L \in \{\mathbf{DAlt}_n, \mathbf{D}_n, \mathbf{Alt}_n, \mathbf{K}_n\}$ and $\mathcal{ALC} \preceq \mathcal{L} \preceq \mathcal{SHIO}$ deciding concept satisfiability in $L_{\mathcal{L}}$ w.r.t. global TBoxes and only with local roles is* 2EXPTIME*-complete.*

Proof. Immediate by Proposition 1, Corollary 1 and Theorems 6 and 1. ❑

As the increase in the complexity can be observed already in two-dimensional DLs, the only way of obtaining better behaved DLCs is to reduce the expressiveness of those underlying formalisms, by restricting the use of context operators. It turns out that when only operators of type $\mathfrak{F}_2$ are allowed, the complexity of the satisfiability problem in DLCs can be taken down to NEXPTIME- and even EXPTIME-complete, depending on the configurations of context and object languages. In the following two theorems we establish precise conditions warranting the respective results.

Theorem 10. *Deciding satisfiability of a knowledge base in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$, for $\mathcal{ALC} \preceq \mathcal{L}_C \preceq \mathcal{SHIO}$, $\mathcal{ALC} \preceq \mathcal{L}_O \preceq \mathcal{SHIO}$, and for $\mathcal{L}_C \preceq \{\mathcal{SHIO}, \mathcal{EL}^{++}\}$, $\mathcal{ALCO} \preceq \mathcal{L}_O \preceq \mathcal{SHIO}$, with context operators $\mathfrak{F}_2$ only and for local interpretation of object roles, is* NEXPTIME*-complete.*

Proof. Immediate by Theorems 16, 17, 22 (see Sections A.3 and A.4) and Theorem 2. ❑

Theorem 11. *Deciding satisfiability of a knowledge base in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$, for $\mathcal{L}_C \preceq \mathcal{EL}^{++}$, $\mathcal{ALC} \preceq \mathcal{L}_O \preceq \mathcal{SHI}$, with context operators $\mathfrak{F}_2$ only and for local interpretation of object roles, is* EXPTIME*-complete.*

Proof. Immediate by Theorem 23 (see Section A.4) and the known complexity results for DLs [BCM+03, Tob01]. ❑

The jump from EXPTIME to NEXPTIME-completeness, captured in Theorem 10, is triggered in two cases: by employing at least $\mathcal{ALC}$ as the context language, or $\mathcal{ALCO}$ as the object language. The non-determinism involved in the first case can be interpreted by the need of guessing the interpretation of the context language first, before finding the model of the object component of the combination. In particular, the lower bound is obtained by an encoding of the $2^n \times 2^n$ tiling problem, known to be NEXPTIME-complete [KWZG03]. In the case of $\mathcal{L}_\mathcal{O} = \mathcal{ALCO}$ this jump can be explained by the interaction of nominals and the context operators, which enables encoding the $2^n \times 2^n$ tiling problem. For the upper bounds we devise another variant of the type elimination algorithm. As shown in Theorem 11, if the nominals are avoided in the object language, while the context language is restricted to the tractable $\mathcal{EL}^{++}$, the satisfiability problem remains no harder than in the object language, at least up to the DL $\mathcal{SHI}$. These results transfer further to $\mathbf{S5}_\mathcal{L}$.

Corollary 3. *Deciding concept satisfiability in* $\mathbf{S5}_\mathcal{L}$ *w.r.t. global TBoxes and only with local roles is* EXPTIME*-complete for* $\mathcal{ALC} \preceq \mathcal{L} \preceq \mathcal{SHI}$ *and* NEXPTIME*-complete for* $\mathcal{ALCO} \preceq \mathcal{L} \preceq \mathcal{SHIO}$.

Proof. Immediate by Theorems 10 and 11.

3.6 Conclusion

The problems of representing inherently contextualized knowledge within the paradigm of DLs and reasoning with multiple heterogenous, but semantically interoperating DL ontologies, are both interesting and important issues, motivated by numerous application scenarios. It is our strong belief that these two challenges are in fact two sides of the same coin and, consequently, they should be approached within the same, unifying formal framework. In this chapter, we have proposed a novel family of two-dimensional, two-sorted *Description Logics of Context*. We have argued, that these formalisms achieve this objective to a large extent, by providing sufficient syntactic and semantic means to support both functionalities, seamlessly integrated on the grounds of one formal theory. The pivotal premise of this theory is that contexts should be interpreted as possible worlds in the second modal dimension added to the standard semantics of DLs. In this way the formalistic, application-agnostic spirit of McCarthy's theory of contexts can be successfully combined with the machinery of modal logics.

The richness and diversity of the expressive means offered by the presented logics afford a considerable space for choice of specific fragments with desirable properties, suitable for particular applications. Naturally, the largest fragment studied here, $\mathfrak{C}^{\mathcal{SHIO}}_{\mathcal{SHIO}}$ with operators $\mathfrak{F}_1$ and $\mathfrak{F}_2$, is computationally quite expensive, and admittedly, not the most straightforward in practical use. However, as we have also demonstrated, by properly balancing the expressiveness of the object language along with the power of the involved contextualization mechanism, as in the case of $\mathfrak{C}^{\mathcal{EL}^{++}}_{\mathcal{SHI}}$ with operators $\mathfrak{F}_2$, one can obtain convenient, contextualized DLs without paying a significant price in the computational complexity, which remains the same as in the underlying (object) DLs. Furthermore, the reasoning task which we have focused on in this chapter, is the standard satisfiability checking. Nevertheless, one of the most interesting application lines for DLCs might in fact be centered around variants of model checking problems, which are in general less computationally challenging. We share the view that the Semantic Web architecture is currently saturated with logic-based knowledge representation formalism, and the cost of a broad adoption of yet more expressive extension of the existing W3C ontology languages can hardly be afforded by the community in the near future. Hence, the divide-and-conquer scenarios, although very exciting from the purely knowledge representation perspective, are not likely to be soon picked up as realistic use-cases for DLCs. On the contrary, the practice of the Semantic Web shows that different compose-and-conquer methodologies are very much required, in order to deal with the distributed character of the Web. In this respect, DLCs, or their suitable modifications, can indeed be useful in modeling high-level interdependency constraints over the contents and meta-level descriptions of different datasets, to be verified (thus model-checked) against structures composed of those datasets. One type of such structures is discussed further in Chapter 4, while Chapters 5 and 6 study some reasoning problems that can be formally related to model-checking over certain two-level representations.

CHAPTER
FOUR

INTEGRATION AND SELECTION OF KNOWLEDGE

In this chapter, we apply the context framework to the problem of ontology integration, and introduce a novel task of metaknowledge-driven selection and querying of data. We demonstrate the ease of the tasks under the proposed approach and report on a case study of aligning different versions of Wordnet ontologies. We consider the following setting:

contexts $\doteq$ *logical entities denoted by ontology URIs/names*

context representation language $\doteq$ *DL*

contextual information $\doteq$ *arbitrary information about the ontologies associated with the URIs*

object representation language $\doteq$ *DL over prefixed vocabulary, where prefixes correspond to the URIs*

reasoning task $\doteq$ *integration, selection, querying*

task-specific language $\doteq$ *two-part context-object queries*

4.1 Introduction

As the adoption of Semantic Web approaches has grown so has the availability of large amounts of overlapping knowledge sources pertaining to the same domain. For example, in the Web of Data, we see macro clusters of knowledge in diverse areas from government and research to music and biomedicine. At a micro level, we see ontologies being progressively updated and multiple versions of the same ontology being used simultaneously. Additionally, knowledge is often being generated by a variety of different mechanisms from automated mapping techniques to expert entry. For instance, the sig.ma search engine [TCC+10], at the time of writing, returns twenty different knowledge sources used to describe the concept of "heart disease", ranging from Wikipedia to Examiner.com, a local news site, to slide sets from anonymous users and the Pew Internet Trust.

Problem: In this environment, applications developers are faced with a challenge, how does one select and integrate the right set of *object-level knowledge* while not statically encoding which knowledge to use. Applications, for example, may want to focus on up-to-date knowledge, knowledge generated by particular software mechanisms, or knowledge provided by a particular organization. This *metaknowledge* is key to being able to select the right set of domain data to be used within the application. In practice, applications often encode the decisions about which object-level knowledge to use either in an off-line selection process or in every query they issue to an integrated knowledge base. Thus, developers are faced with either less flexible approaches or increased query complexity. Furthermore, these approaches provide no formal grounding about the consequences of reasoning when integrating knowledge. Specifically, we formulate the problem as follows: *How does one systematically, rigorously and simply deploy metaknowledge in order to facilitate selective reasoning over object-level knowledge?*

Contributions: To address this problem, we introduce a framework for the selection and integration of object-level knowledge based on formally modeled metaknowledge. The framework provides three crucial benefits:

1. it has a clear formal grounding ensuring guarantees that reasoning complexity does not exceed that of the underlying languages used,
2. it builds upon widely deployed Semantic Web representations and tools,
3. it is timely, as many semantic datasets come already with formal annotations such as OPMV[1] and VOiD[2], which are ready for use in the framework.

[1] See `http://open-biomed.sourceforge.net/opmv/ns.html`.
[2] See `http://semanticweb.org/wiki/VoID`.

Our framework thus strikes a balance between theoretical rigor and ease of implementation. To emphasize this ease of use we have built the selection component of the framework using an existing Semantic Web development platform, the Large Knowledge Collider [FvHA+08], and explain its potential with a use-case study from the automated alignment of the Wordnet vocabulary for the cultural heritage domain.

In summary, the contributions of this chapter are as follows:

1. a generic formal framework combining two key features: representation and reasoning with metaknowledge and integration of multiple, context-specific object knowledge representations,

2. the first such framework expressed purely in terms of compositions of standard Semantic Web representations (DL/OWL/RDF(S) ontologies),

3. formal results showing that the complexity of reasoning in the framework does not exceed that of the underlying languages,

4. an implementation of the approach showing that the framework can be easily deployed using an existing Semantic Web development platform.

Content: The rest of the chapter is organized as follows. We begin by presenting an informal overview of the framework in Section 4.2. In Section 4.3 we introduce a detailed formalization of the framework, study its formal properties and the considered reasoning tasks. Further, we discuss our implementation in Section 4.4 and describe the case study in Section 4.5. Then, in Section 4.6, we provide an overview of the related work, particularly emphasizing relationships to alternative formal approaches and conclude the chapter in Section 4.7.

4.2 The ISM framework: overview

The proposed framework supports integration of multiple representation systems containing possibly fragmentary and heterogenous object-level knowledge, with a parallel representation of the meta-level knowledge over those systems, concerning their content, provenance and any other relevant types of contextual information. Reasoning over the framework intertwines inference over these two levels. Importantly, the framework is reducible to existing formalisms and reasoning problems, which ensures strong and well-understood formal foundations and straightforward implementations. Moreover, the formulation of our approach is sufficiently generic to permit most current Semantic Web languages for modeling object and meta-ontologies.

The knowledge models supported by the presented framework shall be denoted as *Interoperability Systems with Metaknowledge* (ISM). The central components of an ISM, as illustrated in Figure 4.1, are:

***object ontologies*:**	formal representations of different portions of object-level knowledge about an application domain,
***meta-ontology*:**	formal representation of meta-level knowledge about the object ontologies.

The object ontologies are standard DL ontologies which can be metaphorically depicted as "boxes" [BBG00, BBG08]. Each box is equipped with its own vocabulary and associated with a unique formal entity called a *context*. A box contains a portion of domain knowledge specific to its context. Boxes can be integrated by sharing their local vocabularies. An interpretation of a shared term is always restricted by its original box. This approach is motivated by the typical solutions found in the Web environment, where contexts correspond to the URIs (names) of ontologies and allow one to involve pieces of vocabulary from different sources. The meta-ontology is another DL ontology, in which the contexts are represented as individuals. A box is thus given a two-fold representation in an ISM: on the meta-level it is treated as an atomic individual described in the metalanguage — on the object level it is associated with a unique ontology.

Interestingly, constructions similar to this one can be commonly witnessed in the actual publishing practice on the Web, typically based on named RDF(S) graphs [CBHS05] or OWL ontologies. For instance, in the BioPortal project, all collected biomedical ontologies are described as instances in the BioPortal's meta-ontology [3]. Hence, rather than a novel representation standard, our framework is proposed as a simple, yet motivated and rigorous way of systematizing the logical foundations behind that practice.

The semantics of the framework is grounded directly in the standard model-theoretic semantics of the languages used on the object and the meta-level of representation, by adopting a simple, compositional approach. A model of an ISM is a *composition of* (standard) *models* of the ontologies included in the ISM, which must satisfy certain compatibility criteria. The formal characteristics of the framework are determined largely by the following two properties, whose

[3] See http://bioportal.bioontology.org/.

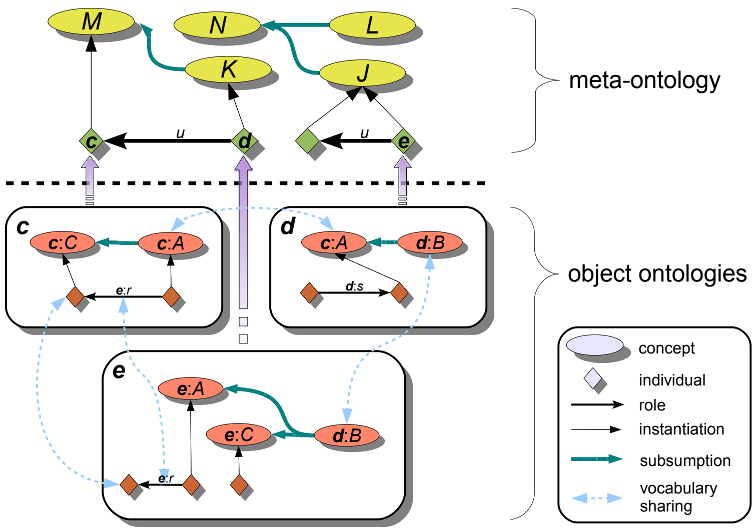

Figure 4.1: An Interoperability System with Metaknowledge.

formal explications are provided in the next section:

P1 The semantic interoperability mechanism used for relating the contents of the object ontologies is of a purely *extensional character*, in the sense that two ontologies can be semantically related only by aligning the interpretations (extensions) of some parts of their vocabularies.

P2 The semantic relationship between the object and the meta-level of the representation is *largely conventional*, i.e. it involves no genuine formal interaction between the semantics of both levels.

4.3 Formalization

In this section we formally introduce components of the ISM framework. We start with discussing the object-level representation and the corresponding problem of ontology integration, and further extend it with the meta-level features used for knowledge selection. In terms of DLCs, studied in the previous chapter, ISM can be seen as a fragment involving always only a finite number of contexts, and the object language with a strongly restricted use of context operators, of type $\langle\{\boldsymbol{c}\}\rangle$, for any context name $\boldsymbol{c}$. The context names are intuitively interpreted as the names of object ontologies (context boxes) containing local object knowledge.

4.3.1 Object-level knowledge integration

The definition of the object language involves the context names as unique prefixes for distinguishing between box-specific vocabularies.

Definition 7 (Object language)**.** *Let $M_I^\star$ be a countably infinite set of context names, and let $\Sigma = (N_C, N_R, N_I)$ be a DL vocabulary. Then an* object language $\mathcal{L}_\mathcal{O}$ *over $M_I^\star$ and Σ is a DL language over the vocabulary $\Sigma^\star = (N_C^\star, N_R^\star, N_I^\star)$, where:*

- $N_C^\star = \{\boldsymbol{c}{:}C \mid \boldsymbol{c} \in M_I^\star,\ C \in N_C\}$,
- $N_R^\star = \{\boldsymbol{c}{:}r \mid \boldsymbol{c} \in M_I^\star,\ r \in N_R\}$,
- $N_I^\star = \{\boldsymbol{c}{:}a \mid \boldsymbol{c} \in M_I^\star,\ a \in N_I\}$.

The elements of the sets $N_C^\star$, $N_R^\star$, $N_I^\star$ are concept names, role names and individual names of $\mathcal{L}_\mathcal{O}$, respectively.

The design of the object language aims at capturing the following intuition: a vocabulary Σ, interpreted over the object domain, might be used differently in different contexts. To avoid ambiguities, instead of referring to a plain atom $\alpha \in \Sigma$, one should rather use it in combination with a prefix $\boldsymbol{c} \in M_I^\star$, explicitly indicating the intended context of interpretation. Effectively, the object vocabulary can be restated as the set of all prefixed atoms $\Sigma^\star = \{\boldsymbol{c}{:}\alpha \mid \boldsymbol{c} \in M_I^\star,\ \alpha \in \Sigma\}$, where interpretation of atoms with the same prefix $\boldsymbol{c} \in M_I^\star$ is effectively restricted by the designated DL interpretation $\mathcal{I}_{\boldsymbol{c}} = (\Delta^{\mathcal{I}_c}, \cdot^{\mathcal{I}_c})$, as specified in Definition 8. Note, that the prefix $\boldsymbol{c}$: is in fact a notational variant of the (functional) operator $\langle\{\boldsymbol{c}\}\rangle$ of DLCs, with the same semantics as implied by Definition 4. Complex expressions, containing atoms with possibly different prefixes, are given their meaning straightforwardly by combining the respective interpretations.

Definition 8 (Object language semantics)**.** *An interpretation of $\mathcal{L}_\mathcal{O}$ is a pair $\mathfrak{M} = (\Delta, \{\mathcal{I}_{\boldsymbol{c}}\}_{\boldsymbol{c} \in M_I^\star})$, where Δ is a non-empty (global) domain of individuals and for every $\boldsymbol{c} \in M_I^\star$, $\mathcal{I}_{\boldsymbol{c}} = (\Delta^{\mathcal{I}_c}, \cdot^{\mathcal{I}_c})$ is an interpretation of Σ and $\Sigma^{\mathcal{L}c}$ such that $\Delta^{\mathcal{I}_c} \subseteq \Delta$ and:*

- $\top^{\mathcal{I}_c} = \Delta^{\mathcal{I}_c}$,

- $A^{\mathcal{I}_c} \subseteq \Delta^{\mathcal{I}_c}$, $r^{\mathcal{I}_c} \subseteq \Delta^{\mathcal{I}_c} \times \Delta^{\mathcal{I}_c}$, $a^{\mathcal{I}_c} \in \Delta^{\mathcal{I}_c}$, *for every* $A \in N_C$, $r \in N_R$, $a \in N_I$,
- $(\boldsymbol{d}{:}A)^{\mathcal{I}_c} = \Delta^{\mathcal{I}_c} \cap A^{\mathcal{I}_d}$, *for every* $\boldsymbol{d}{:}A \in N_C^\star$,
- $(\boldsymbol{d}{:}r)^{\mathcal{I}_c} = \Delta^{\mathcal{I}_c} \times \Delta^{\mathcal{I}_c} \cap r^{\mathcal{I}_d}$, *for every* $\boldsymbol{d}{:}r \in N_R^\star$,
- $(\boldsymbol{d}{:}a)^{\mathcal{I}_c} = a^{\mathcal{I}_d}$ *whenever* $a^{\mathcal{I}_d} \in \Delta^{\mathcal{I}_c}$, *or* $(\boldsymbol{d}{:}a)^{\mathcal{I}_c}$ *is undefined otherwise, for every* $\boldsymbol{d}{:}a \in N_I^\star$,
- $\cdot^{\mathcal{I}_c}$ *is further inductively extended in the usual way over all complex expressions of* $\mathcal{L}_\mathcal{O}$.

Consequently, the language supports interoperability of DL ontologies, in the sense of the property **P1**. Interoperability is understood here as the ability of a system to interpret expressions in different ontologies via shared extensions, according to specified constraints. A collection of ontologies in the object language forms an *interoperability system* (IS), as stated in the following definition.

Definition 9 (Interoperability system)**.** *An object ontology $\mathcal{O}$ is a set of DL axioms over $\mathcal{L}_\mathcal{O}$. An* interoperability system *over $\mathcal{L}_O$ is a finite set of object ontologies $\{\mathcal{O}_{\boldsymbol{c}}\}_{\boldsymbol{c}\in\Omega}$ over $\mathcal{L}_\mathcal{O}$, for $\Omega \subseteq M_I^\star$. An interpretation $\mathfrak{M} = (\Delta, \{\mathcal{I}_{\boldsymbol{c}}\}_{\boldsymbol{c}\in M_I^\star})$ is a model of $\{\mathcal{O}_{\boldsymbol{c}}\}_{\boldsymbol{c}\in\Omega}$* iff *$\mathcal{I}_{\boldsymbol{c}} \models \mathcal{O}_{\boldsymbol{c}}$ for every $\boldsymbol{c} \in \Omega$. An axiom φ over $\mathcal{L}_O$ is entailed by $\{\mathcal{O}_{\boldsymbol{c}}\}_{\boldsymbol{c}\in\Omega}$ in $\boldsymbol{c} \in \Omega$, written $\{\mathcal{O}_{\boldsymbol{c}}\}_{\boldsymbol{c}\in\Omega}, \boldsymbol{c} \models \varphi$,* iff *for every model $\mathfrak{M} = (\Delta, \{\mathcal{I}_{\boldsymbol{c}}\}_{\boldsymbol{c}\in M_I^\star})$ of $\{\mathcal{O}_{\boldsymbol{c}}\}_{\boldsymbol{c}\in\Omega}$, it is the case that $\mathcal{I}_{\boldsymbol{c}} \models \varphi$.*

Observe, that whenever a term $\boldsymbol{c}{:}\alpha$ is used in an object ontology $\mathcal{O}_{\boldsymbol{d}}$ it is ensured that its extension is always restricted by the interpretation $\mathcal{I}_{\boldsymbol{c}}$ of its "source" context. The form of this restriction depends essentially on the assumption regarding the scope of the local object domain in $\boldsymbol{d}$. As an example let us consider an atomic concept $A \in N_C$, as illustrated in Figure 4.2. Under the default, varying domain assumption, it is known that:

$$(\boldsymbol{c}{:}A)^{\mathcal{I}_d} \subseteq (\boldsymbol{c}{:}A)^{\mathcal{I}_c} = A^{\mathcal{I}_c}$$

where $\subseteq$ might be in principle the proper subset relation. This is because the interpretation of $\boldsymbol{c}{:}A$ in $\boldsymbol{d}$ is always restricted to the local domain $\Delta^{\mathcal{I}_d}$, as imposed by Definition 8, via the condition:

$$(\boldsymbol{c}{:}A)^{\mathcal{I}_d} = \Delta^{\mathcal{I}_d} \cap A^{\mathcal{I}_c}$$

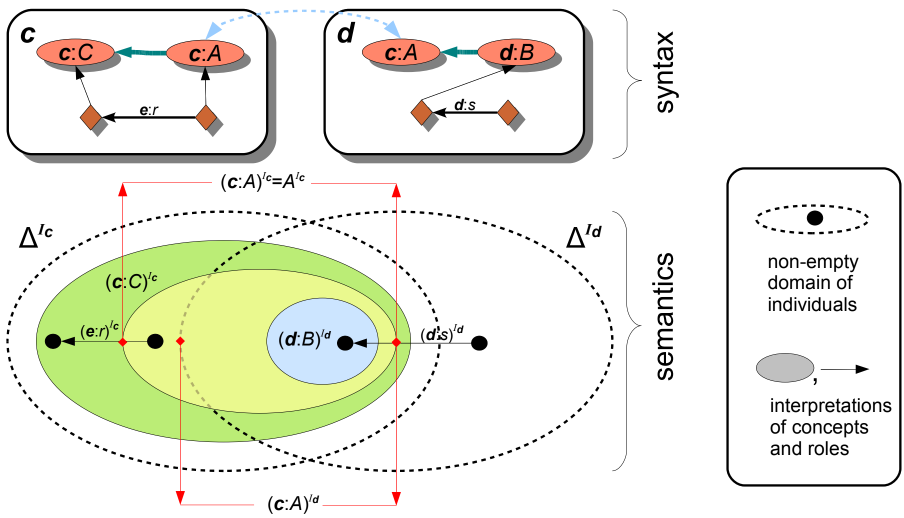

Figure 4.2: A sample model of an interoperability system.

This basic and natural setup captures the intuition that every context covers a certain fragment of the domain, and the knowledge expressed in this context applies only to that given fragment. In some scenarios, however, it might be useful to ensure that in fact $A^{\mathcal{I}_c} \subseteq \Delta^{\mathcal{I}_d}$ — for instance by imposing $\Delta^{\mathcal{I}_d} = \Delta$ i.e. that the local object domain of $\boldsymbol{d}$ covers the whole global domain. Such an assumption would automatically entail that:

$$(\boldsymbol{c}{:}A)^{\mathcal{I}_d} = \Delta^{\mathcal{I}_d} \cap A^{\mathcal{I}_c} = A^{\mathcal{I}_c}$$

The choice of appropriate domain assumptions for particular contexts depends on the intended application, and leaves a crucial degree of freedom in parameterizing the interoperability mechanism. Below we present two examples, in which we make use of different type of assumptions in order to adequately structure the desired interoperability mechanism.

Example 1 (local importing)**.** *Consider an IS $\{\mathcal{O}_{\boldsymbol{c}}\}_{\boldsymbol{c}\in\Omega}$, where $\Omega = \{\boldsymbol{c}, \boldsymbol{d}, \boldsymbol{e}\}$. Suppose context $\boldsymbol{c}$ is associated with a generic, upper ontology for international academic institutions, whereas $\boldsymbol{d}$ and $\boldsymbol{e}$ are contexts of two local universities: one in the Netherlands, the other in the US.*

$\mathcal{O}_{\boldsymbol{c}}$:	$\boldsymbol{c}{:}PhDstudent \sqsubseteq \exists \boldsymbol{c}{:}enrolled.\boldsymbol{c}{:}PhDprogram$
	$\boldsymbol{c}{:}Employee \sqsubseteq \exists \boldsymbol{c}{:}receives.\boldsymbol{c}{:}Salary$
$\mathcal{O}_{\boldsymbol{d}}$:	$\boldsymbol{d}{:}AiO \equiv \boldsymbol{c}{:}PhDstudent$
	$\boldsymbol{d}{:}Medewerker \equiv \boldsymbol{c}{:}Employee$
	$\boldsymbol{d}{:}AiO(\boldsymbol{d}{:}jSmith)$
	$\boldsymbol{d}{:}Medewerker(\boldsymbol{d}{:}jSmith)$
$\mathcal{O}_{\boldsymbol{e}}$:	$\boldsymbol{e}{:}GradStudent \equiv \boldsymbol{c}{:}PhDstudent$
	$\boldsymbol{e}{:}Staff \equiv \boldsymbol{c}{:}Employee$
	$\boldsymbol{e}{:}Staff \sqcap \boldsymbol{e}{:}GradStudent \sqsubseteq \bot$

Here, we leave the default, varying domain assumption for all contexts. Observe, that contexts $\boldsymbol{d}$ *and* $\boldsymbol{e}$ *"import" certain vocabulary from the context* $\boldsymbol{c}$*. For instance, within context* $\boldsymbol{d}$*, concept* $\boldsymbol{d}{:}AiO$ *is said to be equivalent to* $\boldsymbol{c}{:}PhDstudent$*. Similarly, in context* $\boldsymbol{e}$*,* $\boldsymbol{e}{:}GradStudent$ *is equivalent to* $\boldsymbol{c}{:}PhDstudent$*. Consequently, it is possible to infer that* $\boldsymbol{d}{:}AiO \sqsubseteq \exists \boldsymbol{c}{:}enrolled.\boldsymbol{c}{:}PhDprogram$ *holds in* $\boldsymbol{d}$*, while* $\boldsymbol{e}{:}GradStudent \sqsubseteq \exists \boldsymbol{c}{:}enrolled.\boldsymbol{c}{:}PhDprogram$*. However, it does not follow in any context that* $\boldsymbol{d}{:}AiO \equiv \boldsymbol{e}{:}GradStudent$*, as due to the varying domain assumptions, asserted equivalences apply only to the fragments of the global domain covered by the particular contexts.*

Example 2 (global integration)**.** *Consider an* $\{\mathcal{O}_c\}_{c\in\Omega}$*, with* $\Omega = \{\boldsymbol{c}, \boldsymbol{d}, \boldsymbol{e}, \boldsymbol{f}, \boldsymbol{g}\}$*, where contexts* $\boldsymbol{c}, \boldsymbol{d}, \boldsymbol{e}$ *are associated with three knowledge sources to be integrated, while* $\boldsymbol{f}$ *and* $\boldsymbol{g}$ *correspond to sets of ontology mappings to be globally imposed over the sources.*

$\mathcal{O}_{\boldsymbol{c}}$:	$\boldsymbol{c}{:}Employee \sqsubseteq \exists \boldsymbol{c}{:}receives.\boldsymbol{c}{:}Salary$
	$\boldsymbol{c}{:}Employee(\boldsymbol{c}{:}jSmith)$
$\mathcal{O}_{\boldsymbol{d}}$:	$\boldsymbol{d}{:}Staff \sqsubseteq \exists \boldsymbol{d}{:}hasInsurance.\boldsymbol{d}{:}CollectivePolicy$
	$\boldsymbol{d}{:}Staff(\boldsymbol{d}{:}mBrown)$
$\mathcal{O}_{\boldsymbol{e}}$:	$\top \sqsubseteq \boldsymbol{e}{:}Person$
	$\top(\boldsymbol{e}{:}johnSmith)$
$\mathcal{O}_{\boldsymbol{f}}$:	$\boldsymbol{c}{:}Employee \sqsubseteq \boldsymbol{d}{:}Staff$
	$\boldsymbol{d}{:}Staff \sqsubseteq \boldsymbol{e}{:}Person$
$\mathcal{O}_{\boldsymbol{g}}$:	$\{\boldsymbol{c}{:}jSmith\} \equiv \{\boldsymbol{e}{:}johnSmith\}$

In this scenario, we set the varying domain assumption for the knowledge sources, while for contexts $\boldsymbol{f}$ *and* $\boldsymbol{g}$ *we require that* $\Delta^{\mathcal{I}_f} = \Delta^{\mathcal{I}_g} = \Delta$*. Clearly, statements in the ontologies associated with the two latter contexts are intended as global constraints over the entire domain, and hence must be interpreted over the global domain. As a result, we obtain a number of desired inferences, for instance* $\boldsymbol{c}{:}Employee \sqsubseteq \exists \boldsymbol{d}{:}hasInsurance.\boldsymbol{d}{:}CollectivePolicy$*, which holds in every context.*

Observe, that in both examples taking the union of the ontologies instead of interpreting them in the context-based semantics introduced here, violates the local character of axioms, leading to unintended consequences. In the first case, the union is clearly unsatisfiable. In the latter, we would obtain some undesired inferences, such as $\boldsymbol{c}{:}Salary \sqsubseteq \boldsymbol{e}{:}Person$ or $\boldsymbol{d}{:}CollectivePolicy \sqsubseteq \boldsymbol{e}{:}Person$.

The "contextualized" entailment relation, introduced in Definition 9 and involved in the examples above, can be in a natural way extended towards the problem of query entailment, and further, to the problem of query answering. We consider arbitrary first-order queries, for instance the standard CQs (see Section 2.1).

Definition 10 (IS query answering)**.** *Let $q(\vec{x})$ be a query over $\mathcal{L}_O$, with answer variables $\vec{x} = x_1, \ldots, x_k$. A sequence of names $\vec{\boldsymbol{c}{:}a} = (\boldsymbol{c}{:}a)_1, \ldots, (\boldsymbol{c}{:}a)_k \in N_I^\star$ is a certain answer to $q(\vec{x})$ w.r.t. an interoperability system $\{\mathcal{O}_{\boldsymbol{c}}\}_{\boldsymbol{c}\in\Omega}$, in $\boldsymbol{c} \in M_I^\star$, written $\{\mathcal{O}_{\boldsymbol{c}}\}_{\boldsymbol{c}\in\Omega}, \boldsymbol{c} \models q[\vec{\boldsymbol{c}{:}a}]$* iff *for every model $\mathfrak{M} = (\Delta, \{\mathcal{I}_{\boldsymbol{c}}\}_{\boldsymbol{c}\in M_I^\star})$ of $\{\mathcal{O}_{\boldsymbol{c}}\}_{\boldsymbol{c}\in\Omega}$, it is the case that $\mathcal{I}_{\boldsymbol{c}} \models q[\vec{\boldsymbol{c}{:}a}]$.*

As an example, consider the query:

$$q(x) \ ::= \ \boldsymbol{d}{:}Staff(x) \wedge \boldsymbol{e}{:}Person(x)$$

The answers to this query w.r.t. the IS used in Example 2 are:

- $\boldsymbol{c}{:}jSmith$, $\boldsymbol{d}{:}mBrown$, $\boldsymbol{e}{:}johnSmith$ in $\boldsymbol{d}, \boldsymbol{e}, \boldsymbol{f}$,
- $\boldsymbol{c}{:}jSmith$, $\boldsymbol{e}{:}johnSmith$ in $\boldsymbol{c}$.

Given the standard DL foundations of interoperability systems, both the satisfiability problem and query answering are solvable using existing reasoning tools. To this end we employ a reduction of any arbitrary IS to an equisatisfiable DL ontology. The reduction, presented in Table 4.1 and shown correct in the subsequent proposition, is straightforward and can be accomplished in linear time in the size of the input. It applies directly to all non-trivial DLs.

Proposition 3 (Correctness)**.** *Let $\{\mathcal{O}_{\boldsymbol{c}}\}_{\boldsymbol{c}\in\Omega}$ be an interoperability system in $\mathcal{L}_O$ and $\mathcal{O}$ the ontology resulting from applying the reduction procedure (Table 4.1). Then $\{\mathcal{O}_{\boldsymbol{c}}\}_{\boldsymbol{c}\in\Omega}$ is satisfiable* iff *$\mathcal{O}$ is satisfiable.*

Proof. Suppose $\{\mathcal{O}_{\boldsymbol{c}}\}_{\boldsymbol{c}\in\Omega}$ is satisfiable and let $\mathfrak{M} = (\Delta, \{\mathcal{I}_{\boldsymbol{c}}\}_{\boldsymbol{c}\in M_I^\star})$ be its model, with $\mathcal{I}_{\boldsymbol{c}} = (\Delta^{\mathcal{I}(\boldsymbol{c})}, \cdot^{\mathcal{I}(\boldsymbol{c})})$, for every $\boldsymbol{c} \in M_I^\star$. The proof is by construction of a model $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of $\mathcal{O}$. We fix the following:

- $\Delta^{\mathcal{I}} := \Delta$,

INPUT: An interoperability system $\{\mathcal{O}_{\boldsymbol{c}}\}_{\boldsymbol{c}\in\Omega}$ in $\mathcal{L}_O$

1. For every $\boldsymbol{c} \in \Omega$, replace every occurrence of $\top$ in $\mathcal{O}_{\boldsymbol{c}}$ with a fresh concept $\boldsymbol{c}{:}\top$.
2. Set $\mathcal{O}^\star := \emptyset$. For every $\boldsymbol{d} \in M_I^\star$ and every concept name $\boldsymbol{d}{:}A$, individual name $\boldsymbol{d}{:}a$ and role name $\boldsymbol{d}{:}r$ occurring in $\{\mathcal{O}_{\boldsymbol{c}}\}_{\boldsymbol{c}\in\Omega}$, extend $\mathcal{O}^\star$ with the following axioms:
$$\boldsymbol{d}{:}A \sqsubseteq \boldsymbol{d}{:}\top, \qquad \boldsymbol{d}{:}\top(\boldsymbol{d}{:}a), \qquad \mathrm{dom}(\boldsymbol{d}{:}r) \sqsubseteq \boldsymbol{d}{:}\top, \qquad \mathrm{ran}(\boldsymbol{d}{:}r) \sqsubseteq \boldsymbol{d}{:}\top.$$
3. For every $\boldsymbol{c} \in M_I^\star$, whenever $\Delta^{\mathcal{I}_{\boldsymbol{c}}} = \Delta$ is imposed, extend $\mathcal{O}^\star$ with $\top \sqsubseteq \boldsymbol{c}{:}\top$.
4. For every $\boldsymbol{c} \in \Omega$, $\boldsymbol{d} \in M_I^\star$ and every individual name $\boldsymbol{d}{:}a$ occurring in $\mathcal{O}_{\boldsymbol{c}}$, extend $\mathcal{O}_{\boldsymbol{c}}$ with axiom $\boldsymbol{c}{:}\top(\boldsymbol{d}{:}a)$.
5. Replace every occurrence of $C \equiv D$ in $\{\mathcal{O}_{\boldsymbol{c}}\}_{\boldsymbol{c}\in\Omega}$ with $C \sqsubseteq D, D \sqsubseteq C$. Further, for every $\boldsymbol{c} \in \Omega$, replace every $C \sqsubseteq D \in \mathcal{O}_{\boldsymbol{c}}$ with $\boldsymbol{c}{:}\top \sqcap C \sqsubseteq D$.
6. If $\mathcal{L}_O$ does not support role inclusions, then for every $\boldsymbol{c} \in M_I^\star$, $r \in N_R$, concept C, $\boldsymbol{d} \in \Omega$, replace every occurrence of $\exists \boldsymbol{c}{:}r.C$ (resp. $\forall \boldsymbol{c}{:}r.C$) in $\mathcal{O}_{\boldsymbol{d}}$ with $\exists \boldsymbol{c}{:}r.(\boldsymbol{d}{:}\top \sqcap C)$ (resp. $\forall \boldsymbol{c}{:}r.(\neg \boldsymbol{d}{:}\top \sqcup C)$).
7. If $\mathcal{L}_O$ supports role inclusions, then for every $\boldsymbol{c} \in M_I^\star$, $r \in N_R$, $\boldsymbol{d} \in \Omega$, replace every occurrence of $\boldsymbol{c}{:}r$ in $\mathcal{O}_{\boldsymbol{d}}$ with a fresh name $\boldsymbol{c}{:}r^{\boldsymbol{d}}$ and extend $\mathcal{O}^\star$ with the following axioms:
$$\mathrm{dom}(\boldsymbol{c}{:}r^{\boldsymbol{d}}) \sqsubseteq \boldsymbol{d}{:}\top, \qquad \mathrm{ran}(\boldsymbol{c}{:}r^{\boldsymbol{d}}) \sqsubseteq \boldsymbol{d}{:}\top, \qquad \boldsymbol{c}{:}r^{\boldsymbol{d}} \sqsubseteq \boldsymbol{c}{:}r.$$
8. Set $\mathcal{O} := \bigcup_{\boldsymbol{c}\in\Omega} \mathcal{O}_{\boldsymbol{c}} \cup \mathcal{O}^\star$.

OUTPUT: DL ontology $\mathcal{O}$

Table 4.1: Reduction of an interoperability system to an equisatisfiable DL ontology.

- $(\boldsymbol{c}{:}\top)^{\mathcal{I}} := \Delta^{\mathcal{I}(\boldsymbol{c})}$, for every $\boldsymbol{c} \in M_I^\star$,
- $(\boldsymbol{c}{:}\alpha)^{\mathcal{I}} := \alpha^{\mathcal{I}(\boldsymbol{c})}$, for every $\boldsymbol{c} \in M_I^\star$ and $\alpha \in N_C \cup N_R \cup N_I$,
- if $\mathcal{L}_O$ supports role inclusions, then $(\boldsymbol{c}{:}r^{\boldsymbol{d}})^{\mathcal{I}} = (\boldsymbol{c}{:}r)^{\mathcal{I}(\boldsymbol{d})}$, for every $\boldsymbol{c} \in M_I^\star$, $r \in N_R$ and $\boldsymbol{d} \in \Omega$.

Further, $\cdot^{\mathcal{I}}$ is extended inductively in the usual manner, according to the standard semantics of the constructors of $\mathcal{L}_O$. By the construction of $\mathcal{I}$, it follows that all axioms in $\mathcal{O}$ must be satisfied in $\mathcal{I}$. In particular:

- satisfaction of axioms added in step 2, 3, 4, 7 follows immediately by the construction and the semantics of the object language,

- satisfaction of ABox assertions follows immediately by the construction,

- for satisfaction of concept inclusions, observe that for every possibly complex concept C in $\mathcal{L}_O$ and $\boldsymbol{c} \in \Omega$, $C^{\mathcal{I}_c} = (\boldsymbol{c}{:}\top \sqcap \pi(C))^{\mathcal{I}}$, where $\pi(C)$ denotes the result of applying steps 1 and 6 (resp. 7) to C. This can be shown by structural induction over the syntax of C, of which the only step requiring a comment concerns role restrictions. Suppose $C = \exists \boldsymbol{d}{:}r.D$, for an arbitrary $\boldsymbol{d} \in M_I^\star$ and a possibly complex concept D. Then $\pi(C) = \exists \boldsymbol{d}{:}r.(\boldsymbol{c}{:}\top \sqcap D)$ (resp. $\pi(C) = \exists \boldsymbol{d}{:}r^{\boldsymbol{c}}.D$). By the semantics $(\exists \boldsymbol{d}{:}r.D)^{\mathcal{I}_c} = \{x \in \Delta^{\mathcal{I}_c} \mid \exists y.(x,y) \in (\boldsymbol{d}{:}r)^{\mathcal{I}_c} \wedge y \in D^{\mathcal{I}_c}\}$, while $(\boldsymbol{c}{:}\top \sqcap \pi(C))^{\mathcal{I}} = \Delta^{\mathcal{I}_c} \cap \{x \in \Delta \mid \exists y.(x,y) \in r^{\mathcal{I}_d} \wedge y \in (\Delta^{\mathcal{I}_c} \cap D^{\mathcal{I}_c})\}$ (resp. $(\boldsymbol{c}{:}\top \sqcap \pi(C))^{\mathcal{I}} = \Delta^{\mathcal{I}_c} \cap \{x \in \Delta \mid \exists y.(x,y) \in (\boldsymbol{d}{:}r)^{\mathcal{I}(\boldsymbol{c})} \wedge y \in D^{\mathcal{I}_c}\}$), which clearly must coincide. The case of $C = \forall \boldsymbol{d}{:}r.D$ follows analogically.

- satisfaction of role inclusions follows by the construction of $\mathcal{I}$ and step 7.

By an analogical, converse construction of a model $\mathfrak{M} = (\Delta, \{\mathcal{I}_{\boldsymbol{c}}\}_{\boldsymbol{c} \in M_I^\star})$ of $\{\mathcal{O}_{\boldsymbol{c}}\}_{\boldsymbol{c} \in M_I^\star}$ from a model $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of $\mathcal{O}$, the opposite direction of the argument can be made. In that case, for all $\boldsymbol{c} \in M_I$ not occurring in $\mathcal{O}$, we arbitrarily pick some $\boldsymbol{d} \in M_I^\star$ and fix $\mathcal{I}_{\boldsymbol{c}} := \mathcal{I}_{\boldsymbol{d}}$. ❑

Proposition 4 (IS query answering reduction)**.** *Let $\{\mathcal{O}_{\boldsymbol{c}}\}_{\boldsymbol{c} \in \Omega}$ be an interoperability system, $\mathcal{O}$ the ontology resulting from applying the reduction procedure (Table 4.1), and $q = \exists \vec{y}.\varphi(\vec{x}, \vec{y})$ a query over $\mathcal{L}_O$ with free variables $\vec{x} = x_1, \ldots, x_k$ and bounded variables $\vec{y} = y_1, \ldots, y_n$. For any sequence of names $\vec{\boldsymbol{c}{:}a} = (\boldsymbol{c}{:}a)_1, \ldots, (\boldsymbol{c}{:}a)_k \in N_I^\star$ and $\boldsymbol{d} \in M_I^\star$, it is the case that $\vec{\boldsymbol{c}{:}a}$ is a certain answer w.r.t. $\{\mathcal{O}_{\boldsymbol{c}}\}_{\boldsymbol{c} \in \Omega}$ in $\boldsymbol{d}$* iff *it is a certain answer to the query $q^{\boldsymbol{d}} = \exists \vec{y}. \bigwedge_{1 \le i \le k} \boldsymbol{d}{:}\top(x_i) \wedge \bigwedge_{1 \le i \le n} \boldsymbol{d}{:}\top(y_i) \wedge \varphi(\vec{x}, \vec{y})$ w.r.t. $\mathcal{O}$.*

Proof. Observe that all individuals satisfying the query $q^{\boldsymbol{d}}$ are forced to satisfy $\boldsymbol{d}{:}\top$, while q is evaluated in $\boldsymbol{d}$ over $\Delta^{\mathcal{I}(\boldsymbol{d})}$. Hence, by Definition 10 and the correspondence between the models of $\{\mathcal{O}_{\boldsymbol{c}}\}_{\boldsymbol{c} \in \Omega}$ and $\mathcal{O}$ established in Proposition 3, it follows that the sets of certain answers for both query problems must coincide. ❑

4.3.2 Meta-level knowledge selection

Notably, the context names play a twofold role in the framework. As vocabulary prefixes, they determine the *logical space* of contexts relevant for interpreting a collection of object knowledge statements. As unique identifiers for object ontologies, they allow for enumerating and retrieving *portions of data* which actually convey those statements. Such ambiguity is inherent to (and hence justified by) the real-life use of URIs on the Web. By acknowledging it, we also emphasize that the problem of reconciling logically partitioned knowledge is orthogonal to the problem of integrating physically partitioned data, studied extensively in the field of distributed databases. In this section, we elaborate on the ability of selecting and retrieving the suitable portions of data that are relevant for a particular query answering problem. To this end, we extend interoperability systems with the meta-level representation and utilize it for guiding the selection process.

The metalanguage $\mathcal{L}_\mathcal{C}$ is defined exactly as in the previous chapter, in Definition 1. It is then a DL with the standard syntax and semantics over a vocabulary $\Gamma = (M_C, M_R, M_I)$, where $M_I^\star \subseteq M_I$ is the set of context names, used on the object-level as the vocabulary prefixes. To avoid ambiguities in the context naming, we assume the metalanguage adheres to the Unique Name Assumption. A meta-ontology is a standard DL ontology over $\mathcal{L}_\mathcal{C}$. With this addition, we can now define the notion of *interoperability system with metaknowledge* (ISM).

Definition 11 (Interoperability system with metaknowledge)**.** *Let $\mathcal{L}_\mathcal{C}$ be a metalanguage and $\mathcal{L}_\mathcal{O}$ an object language. Then a tuple $\mathcal{S} = \langle \mathcal{C}, \{\mathcal{O}_c\}_{c\in\Omega}\rangle$ is a* interoperability system with metaknowledge *(ISM), where $\mathcal{C}$ is a meta-ontology in $\mathcal{L}_\mathcal{C}$, and $\{\mathcal{O}_c\}_{c\in\Omega}$ is an interoperability system in $\mathcal{L}_\mathcal{O}$, with $\Omega \subseteq M_I^\star$.*

The semantics of an ISM is defined straightforwardly by combining the semantics of both levels of representation.

Definition 12 (Semantics)**.** *A pair $\langle \mathcal{J}, \mathfrak{M}\rangle$ is a* model *of an* ISM $\mathcal{S} = \langle \mathcal{C}, \{\mathcal{O}_c\}_{c\in\Omega}\rangle$ iff $\mathcal{J} = (\Delta^\mathcal{J}, \cdot^\mathcal{J})$ *is a model of $\mathcal{C}$, and $\mathfrak{M} = (\Delta, \{\mathcal{I}_c\}_{c\in M_I^\star})$ is a model of $\{\mathcal{O}_c\}_{c\in\Omega}$.*

What follows from the above definition is that deciding satisfiability of an ISM is equivalent to deciding two independent problems: satisfiability of the meta-ontology and satisfiability of the interoperability system. This characteristic is the formal essence of the property **P2**, highlighted in Section 4.2. The only relationship between the two levels of representation is that some of the individuals appearing in the model of the meta-ontology are *conventionally associated* with the corresponding object ontologies. Such separation guarantees good computational properties of the framework, while at the same time, provides still enough expressive power to support interesting forms of querying.

Example 3 (ISM)**.** *Consider an ISM* $\mathcal{S} = \langle \mathcal{C}, \{\mathcal{O}_{\boldsymbol{c}}\}_{c\in\Omega}\rangle$*, which extends the IS* $\{\mathcal{O}_{\boldsymbol{c}}\}_{c\in\Omega}$ *from Example 2 with the meta-level representation. We recall* $\{\mathcal{O}_{\boldsymbol{c}}\}_{c\in\Omega}$ *below and define* $\mathcal{C}$ *as follows:*

$\mathcal{C}$:	$\boldsymbol{FinancialDep} \sqcup \boldsymbol{HRDep} \sqsubseteq \boldsymbol{CompanyDep}$ $\boldsymbol{ConceptMap} \sqcup \boldsymbol{InstanceMap} \sqsubseteq \boldsymbol{Mappings}$ $\boldsymbol{FinancialDep}(\boldsymbol{c})$ $\boldsymbol{HRDep}(\boldsymbol{d})$ $\boldsymbol{Census}(\boldsymbol{e})$ $\boldsymbol{ConceptMap}(\boldsymbol{f})$ $\boldsymbol{InstanceMap}(\boldsymbol{g})$
$\mathcal{O}_{\boldsymbol{c}}$:	$\boldsymbol{c}{:}Employee \sqsubseteq \exists \boldsymbol{c}{:}receives.\boldsymbol{c}{:}Salary$ $\boldsymbol{c}{:}Employee(\boldsymbol{c}{:}jSmith)$
$\mathcal{O}_{\boldsymbol{d}}$:	$\boldsymbol{d}{:}Staff \sqsubseteq \exists \boldsymbol{d}{:}hasInsurance.\boldsymbol{d}{:}CollectivePolicy$ $\boldsymbol{d}{:}Staff(\boldsymbol{d}{:}mBrown)$
$\mathcal{O}_{\boldsymbol{e}}$:	$\top \sqsubseteq \boldsymbol{e}{:}Person$ $\top(\boldsymbol{e}{:}johnSmith)$
$\mathcal{O}_{\boldsymbol{f}}$:	$\boldsymbol{c}{:}Employee \sqsubseteq \boldsymbol{d}{:}Staff$ $\boldsymbol{d}{:}Staff \sqsubseteq \boldsymbol{e}{:}Person$
$\mathcal{O}_{\boldsymbol{g}}$:	$\{\boldsymbol{c}{:}jSmith\} \equiv \{\boldsymbol{e}{:}johnSmith\}$

The meta-ontology $\mathcal{C}$ *above, represents the metaknowledge over contexts integrated in the ISM. For instance,* ***c*** *is stated to be a context of a financial department, and thus it is an instance of a company department due to the axiom* $\boldsymbol{FinancialDep} \sqcup \boldsymbol{HRDep} \sqsubseteq \boldsymbol{CompanyDep}$*. Contexts* ***f*** *and* ***g*** *are both instances of* ***Mappings****, where* ***f*** *belongs to the subclass of concept mappings and* ***g*** *to instance mappings.*

Admittedly, satisfiability of $\mathcal{C}$ is logically independent from the satisfiability of the accompanying IS and vice versa. Regardless of this semantic separation of the two levels of representation, the framework supports a simple, yet practically useful form of queries which allow for metaknowledge-driven selection of the object-level knowledge to be queried over. ISM queries, of the form $\lceil m(y) \rceil q(\vec{x})$, comprise a meta-level query $m(y)$ and an object-level query $q(\vec{x})$. The meta-level component selects the object ontologies which satisfy certain meta-level descriptions. The object query is then applied over the integrated fragments of knowledge contained in those ontologies. Again, in both cases we consider arbitrary first-order queries. In the example to follow, we use UCQs for the meta-level query and a CQ for the object component. The reasoning problem is formally defined as follows:

Definition 13 (ISM query). *An* ISM query *over an* ISM $\mathcal{S} = \langle \mathcal{C}, \{\mathcal{O}_{\boldsymbol{c}}\}_{c \in \Omega} \rangle$ *is an expression* $\lceil m(y) \rceil q(\vec{x})$*, where:*

- $m(y)$ *is a query over* $\mathcal{L}_{\mathcal{C}}$*, with a single answer variable* y*,*
- $q(\vec{x})$ *is a query over* $\mathcal{L}_{\mathcal{O}}$*, with free variables* $\vec{x} = x_1, \ldots, x_k$*.*

A sequence of names $\vec{\boldsymbol{c}{:}a} = (\boldsymbol{c}{:}a)_1, \ldots, (\boldsymbol{c}{:}a)_k \in N_I^\star$ *is a certain answer to* $\lceil m(y) \rceil q(\vec{x})$ *w.r.t.* $\mathcal{S}$*, in* $\boldsymbol{c} \in M_I^\star$*, written* $\mathcal{S}, \boldsymbol{c} \models q[\vec{\boldsymbol{c}{:}a}]$ iff $\{\mathcal{O}_{\boldsymbol{c}}\}_{c \in \Omega'}, \boldsymbol{c} \models q[\vec{\boldsymbol{c}{:}a}]$*, where* $\Omega' = \{\boldsymbol{d} \mid \boldsymbol{d} \in \Omega,\ \mathcal{C} \models m[\boldsymbol{d}]\}$*.*

As a notable consequence of the loose semantic coupling of the two levels, the combined complexity of answering ISM queries carries over directly from the complexity of answering queries in the two component languages. More specifically, if $(q, \mathcal{L})$ is the problem of answering queries of type q in the language $\mathcal{L}$, then the combined complexity of answering an ISM query $\lceil m(y) \rceil q(\vec{x})$ is equivalent to the higher of the complexities of the two problems: $(m(y), \mathcal{L}_{\mathcal{C}})$ and $(q(\vec{x}), \mathcal{L}_{\mathcal{O}})$. For instance, if the meta-ontology and the object ontologies of a ISM are expressed in the DL $\mathcal{ALC}$ and $\mathcal{SHIQ}$, respectively, then answering ISM queries whose both components are CQs is 2ExpTime-complete, as this is the complexity of CQ answering in $\mathcal{SHIQ}$ (which is higher than the 2ExpTime-complete complexity of solving the same problem in $\mathcal{ALC}$. See Section 2.1). Regardless of this worst-case complexity analysis, the actual effort of answering the object query can be further dramatically reduced, as the meta-query can significantly restrict the amount of data to be queried over.

Finally, we illustrate the querying mechanism using the ISM from Example 3. Let $\lceil m_i(y) \rceil q(x)$ be an ISM query over $\mathcal{S}$, where the object-query $q(x)$ is defined as:

$$q(x) \ ::= \ \boldsymbol{d}{:}\mathit{Staff}(x) \land \boldsymbol{d}{:}\mathit{Person}(x)$$

and further, the meta-query $m_i(y)$ varies across the following alternatives:

- $m_1(y) \ ::= \ \boldsymbol{\mathit{CompanyDep}}(y)$,
- $m_2(y) \ ::= \ \boldsymbol{\mathit{CompanyDep}}(y) \lor \boldsymbol{\mathit{ConceptMap}}(y)$,
- $m_3(y) \ ::= \ \boldsymbol{\mathit{CompanyDep}}(y) \lor \boldsymbol{\mathit{Mappings}}(y)$.

Observe that different meta-queries return different sets of context names. Consequently, different sets of object ontologies are selected as the basis for answering the object query. This in turn leads to obtaining different answers, as presented below:

	selected ontologies	object-query answers
m_1	$\mathcal{O}_{\boldsymbol{c}}, \mathcal{O}_{\boldsymbol{d}}$	$\emptyset$
m_2	$\mathcal{O}_{\boldsymbol{c}}, \mathcal{O}_{\boldsymbol{d}}, \mathcal{O}_{\boldsymbol{f}}$	$\boldsymbol{c}$:$jSmith$, $\boldsymbol{d}$:$mBrown$, in $\boldsymbol{d}, \boldsymbol{e}, \boldsymbol{f}, \boldsymbol{g}$
		$\boldsymbol{c}$:$jSmith$, in $\boldsymbol{c}$
m_3	$\mathcal{O}_{\boldsymbol{c}}, \mathcal{O}_{\boldsymbol{d}}, \mathcal{O}_{\boldsymbol{f}}, \mathcal{O}_{\boldsymbol{g}}$	$\boldsymbol{c}$:$jSmith$, $\boldsymbol{d}$:$mBrown$, $\boldsymbol{e}$:$johnSmith$, in $\boldsymbol{d}, \boldsymbol{e}, \boldsymbol{f}, \boldsymbol{g}$
		$\boldsymbol{c}$:$jSmith$, $\boldsymbol{e}$:$johnSmith$, in $\boldsymbol{c}$

4.4 Implementation

Figure 4.4 presents a schematic workflow for answering ISM queries in practical implementations, following directly as an operationalization of the notions involved in Definition 13.

We implemented a substantial part of this workflow using the Large Knowledge Collider (LarKC) [FvHA+08] — a platform for the creation and execution of Semantic Web reasoning workflows.[4] Each LarKC workflow consists of a number of plugins, each of which performs some reasoning service over a given set of RDF statements. The platform ships with a number of pre-built plugins for various kinds of reasoning, and also facilitates the development of new ones. Plugins can take advantage of a number of services available in the platform including execution on cluster machines and RDF data management. Our implemented workflow consists of the following steps:

1. The meta-ontology is loaded into LarKC.

2. A SPARQL query representing the meta-ontology query is performed to select a series of files (i.e. object ontologies) to be loaded. Note that the underlying triple store (OWLIM)[5] is configured to perform $pD*$ reasoning (also called OWL-Horst) [tH05] at this stage.

3. The selected files are loaded into LarKC.

[4] We acknowledge Paul Groth as the author of the described implementation.

[5] See http://www.ontotext.com/owlim.

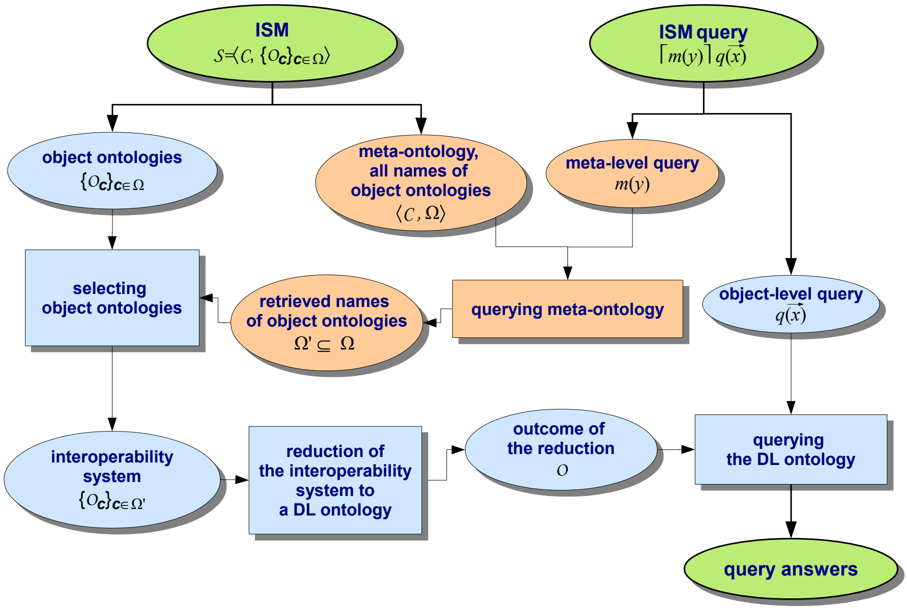

Figure 4.3: The workflow for answering ISM queries.

4. A SPARQL query representing the object ontology query is performed and results are returned. Again, results are returned under OWL-HORST reasoning.

In our implementation we have omitted the integration step, focusing only on the selection aspect, which is dominant in the use-case studied in the next section. Nevertheless, due to the limited expressiveness of the OWL-Horst fragment (predominantly lack of negation), the integration step is in fact hardly necessary, as even when the local character of OWL-Horst axioms is ignored it is unlikely to observe new, unintended inferences which could have been otherwise avoided by applying the integration step. The workflow required only lightweight implementation of two LarKC plugins and the definition of the overall workflow. Importantly, only reasoning services that were already available in LarKC were required for the implementation of the framework. The workflow and associated plugins are accessible on-line at http://www.few.vu.nl/~pgroth/sismd/.

4.5 Case study: Wordnet alignment

We now describe the application of the ISM framework and its LarCK-based implementation to the problem of reasoning over alignments between two versions of Wordnet — a large lexical database categorizing English words in linguistic categories.

4.5.1 Use-case

This use-case stems from a cultural heritage portal serving documents that have been semantically annotated using Wordnet [SAA+08]. As the portal integrates documents from different collections, part of the collection has been annotated using W3C's RDF representation of Wordnet 2.0, while another part uses 3.0. Obviously, one would like to be able to ignore the version differences when these are not relevant. For example, for a given query, all relevant documents annotated using either version need to be found.

To achieve this, an alignment needs to be created that describes, for as many concepts as possible, which concept in one version corresponds to the same or at least a very similar concept in the other version. Creating such an alignment is, however, not an exact science. In an idealized world, two concepts are either equivalent or they are not. In practice, similarity levels vary on a continuous scale, and what level is "sufficiently similar" may depend on the application. Additionally, similarity levels can vary on multiple dimensions. For example, a concept A can be very similar to B along one dimension, but more similar to C along another. A good weighing scheme that takes this into account is typically also application- or context-dependent. Finally, for large vocabularies such as Wordnet (both versions have over 100k concepts), the number of potential mappings (e.g. the Cartesian product of both sets) is very large, and automatic tools are needed to either fully automate the alignment process, or at least to help human experts in creating alignments interactively. As some correspondences are much harder to find than others, the resulting set of all correspondences produced by alignment tools tend to vary in nature and quality. An application might prefer to use only parts of the results.

Typically, creating a good alignment is a task that is too complex to be done at query time. On the other hand, alignments are too context-dependent to create a single alignment *a priori*. One solution is to create multiple sets of correspondences and annotate each set describing its properties. Applications can then query on a meta-level which sets are available and what their properties are. Based on this information, context-specific reasoning can be applied to decide which correspondences the system should use when answering future object-level queries. For example, a retrieval application might opt for high recall performance and include all mappings. An application that uses the mappings to upgrade a corpus that has been manually annotated would prefer high precisions alignments.

The study of the use-case shows that the ISM framework, supported by a lightweight implementation presented in the previous section, is very well suited for this and similar scenarios. It provides such applications with the minimum formal underpinning that is necessary for guaranteeing the reliability of the involved information management processes.

4.5.2 Alignment selection

The described use-case is a typical example for an application with a large variety of related, but different sets of object-knowledge, together with a rich metadata ontology. We have applied the framework to Wordnet alignments produced using the Amalgame system [JvOdB11]. To demonstrate its usage, we discuss a small example. A user is interested in how verbs can be aligned between Wordnet 3.0 and Wordnet 2.0. However, in one case the application is interested in mappings produced with the best numeric score as returned by the mapping algorithms. In the second case, the user is interested in mappings that were returned by multiple different mapping algorithms. Analogically to Example 3, we thus consider a collection of object ontologies consisting of Wordnet 3.0, Wordnet 2.0, and a number of ontologies containing only mappings between the instances of the two Wordnet ontologies (expressed in terms of `owl:sameAs` statements) — all of them described in the additional meta-level ontology. Here, we show how modifying the meta-level query over the provenance changes the results of the same object-level query. For readability, we constrain the query to look at the word "catch".

Formally, we define two ISM queries: $\lceil m1(y) \rceil q(e1, e2)$ and $\lceil m2(y) \rceil q(e1, e2)$, with the same object query, requesting pairs of alignment entities where one of them is of type VerbSynset and has a label "catch":

$$q(e1, e2) \ ::= \ \exists x.(align{:}entity1(x, e1) \wedge align{:}entity2(x, e2) \wedge \\ wn20schema{:}VerbSynset(e1) \wedge label(e1, "catch"))$$

Further, we consider two variants of the meta-query:

- $m1(y) \ ::= \exists x.((\boldsymbol{wasGeneratedBy}(y, x) \wedge \boldsymbol{BestNumeric}(x)) \vee \boldsymbol{WordNet}(y))$

- $m2(y) ::= \exists x.((\textbf{\textit{wasGeneratedBy}}(y,x) \wedge \textbf{\textit{MostMethods}}(x)) \vee \textbf{\textit{WordNet}}(y))$

The meta-queries request the relevant object data sources, including those of type WordNetItem (effectively, the Wordnet ontologies) and all sets of mappings generated with the BestNumeric and MostMethods approaches, respectively. For this application, the object-level query $q(e1, e2)$ is formulated in SPARQL as shown in Figure 4.4. For all SPARQL queries, the following prefixes are defined:

- `rdfs` ::= `http://www.w3.org/2000/01/rdf-schema#`
- `rdf` ::= `http://www.w3.org/1999/02/22-rdf-syntax-ns#`
- `ag` ::= `http://purl.org/vocabularies/amalgame#`
- `opmv` ::= `http://purl.org/net/opmv/ns#`
- `wn20sch` ::= `http://www.w3.org/2006/03/wn/wn20/schema/#`
- `align` ::= `http://knowledgeweb.semanticweb.org/heterogeneity/alignment#`

```
select ?e1 ?e2 where {
   ?map align:entity1 ?e1.
   ?map align:entity2 ?e2.
   ?e1 rdf:type wn20sch:VerbSynset.
   ?e1 rdfs:label "catch"@en-us. }
```

Figure 4.4: An example object-level query $q(e1, e2)$ formulated in SPARQL.

```
select ?file where {
   {?file opmv:wasGeneratedBy ?alg.
    ?alg rdf:type ag:BestNumeric. }
  UNION
   {?file rdf:type ag:WordNet.}}
```

Figure 4.5: An example meta-level query $m1(x)$ formulated in SPARQL.

```
select ?file where {
   {?file opmv:wasGeneratedBy ?s.
    ?s rdf:type ag:MostMethods.}
  UNION
   {?file rdf:type ag:WordNet.}}
```

Figure 4.6: An example meta-level query $m2(x)$ formulated in SPARQL.

The meta-level query $m1(x)$ for best-numeric mappings algorithms is shown in Figure 4.5. Applying, the object-level query over the results of $m1(x)$ (446 377 triples), produces two result bindings mapping to the same synset. Note, that since the integration step is omitted, we effectively query over the union of the selected sources, thus obtaining the same answers for all contexts involved. The bindings 1 and 2 are as follows:

```
?e1: http://purl.org/vocabularies/princeton/wn30/
     synset-catch-verb-18

?e2: http://www.w3.org/2006/03/wn/wn20/instances/
     synset-catch-verb-18
```

The meta-level query $m2(x)$ for mappings from different mapping algorithms is shown in Figure 4.6. With this query, 353 303 triples are used and no results are returned. Note, that here the mappings are only 167 triples of the total number of triples as compared to 93 241 triples for the prior set of mappings. While the above queries are simple, they do require reasoning. They show how by changing the view over provenance (or meta information) we can achieve different results. Most importantly, the case study emphasizes the simplicity of the framework and the ease with which it can be implemented and applied.

4.6 Related work

Firstly, let us summarize the relation of the ISM framework to the DLCs presented in the previous chapter of the thesis. Essentially, an ISM $\mathcal{S} = \langle \mathcal{C}, \{\mathcal{O}_{\boldsymbol{c}}\}_{\boldsymbol{c}\in\Omega} \rangle$ can be restated as a fragment of DLCs with context operators only of type $\langle\{\boldsymbol{c}\}\rangle$ and object axioms only of type $\boldsymbol{c} : \varphi$ (with $\boldsymbol{c} : \varphi$ replacing $\varphi \in \mathcal{O}_{\boldsymbol{c}}$), where the class of relevant $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-interpretations includes precisely those whose context domain $\mathfrak{C}$ is finite and consists of the elements denoted by the context names occurring explicitly in $\mathcal{S}$ (cf. Section 3.4). Because of this finiteness of the context domain, ISMs can safely support the use of $\langle\{c\}\rangle$ operators also over role atoms and individual names, which is not possible in the general DLC setting.

The ISM framework closely coincides with the architecture of Contextualized Knowledge Repositories, proposed in [HS12] (see Section 2.4). The notable difference is that the metalanguage in CKRs is highly restricted, whereas in ISM it is an arbitrary, full DL language, whose vocabulary and expressiveness are left entirely as an application-driven choice. In the pure RDF paradigm, another framework similar to ours and CKRs, called RDF^+ is discussed in [DSSS09] and based on the use of Named Graphs [CBHS05] for representing both levels of knowledge. Given the expressive limitations of RDF, the scope of metalanguage in RDF^+ is again restricted to a set of relational properties. Moreover, unlike in our case, the notions of selection and integration of the object-level knowledge are not considered. A framework that supports meta-level selection of object-level knowledge was proposed in [THM^+08]. It provides a mechanism for selecting a subset of a single ontology based on axiom annotations. The framework, however, does not support the context-sensitive integration, in the sense discussed here, as it is assumed that the entire object-level knowledge is given in one ontology.

The conceptual foundation of contextual reasoning employed here, based on the use of multiple logic theories suitably aligned on the semantic level, was succinctly spelled out by Ghidini et al. in [GS98, GG01] as the proviso: *contextual reasoning = locality + compatibility*. This form of contextuality is quintessentially involved in the problem of logic-based ontology integration. In particular, our semantic interoperability mechanism, based on vocabularies shared among multiple ontologies, is characteristic also to Package-based DLs (P-DLs) [BVSH09]. The main technical difference is that the semantics of P-DLs captures the notion of *importing* the full extension of a shared term to a given context, rather than restricting it to the local domain as here. In yet different ontology integration formalisms, such as e.g. Distributed DLs [BS03] or $\mathcal{E}$-Connections [CGPS09], interoperability is achieved by use of special sorts of link relations between the elements of the domains of different ontologies, which grants a weaker style of integration (less inferences possible) but a more robust one with respect to possible inconsistencies arising due to heterogeneity of integrated knowledge. For a formal survey, we refer the reader to [CK07].

4.7 Conclusion

We presented a framework that allows for an adaptive selection and integration of object-level knowledge based on meta-level knowledge. To the best of our knowledge, this is the first formal framework that deals with the interrelationship between meta-level knowledge and object-level knowledge purely in terms of standard Semantic Web knowledge representations (e.g. DLs, OWL and RDF(S)). Importantly, we demonstrated that the framework can be realized using an existing Semantic Web development framework (LarKC) and applied to an existing use-case — the alignment of vocabularies in a cultural heritage setting. Going forward, we aim to study the application of the framework in more dynamic or streaming settings. Additionally, we aim to apply the approach to large sets of biomedical concept mappings provided by a range of providers.

In the next chapter, we study a representation setting which effectively coincides with ISM structures, without the prefixes over the vocabulary. Thus, we in fact consider collections of ontologies whose URIs are described in a meta-language. Such a very basic construction turns out to go a long way, as it allows for a relatively straightforward introduction of additional, task-oriented languages, which can conveniently utilize these two-dimensional, two-sorted representations in order to solve application-specific problems.

CHAPTER

FIVE

VERIFICATION OF DATA PROVENANCE RECORDS

In this chapter, we apply the context framework to the problem of formal verification of data provenance records, and propose a novel provenance specification logic, based on a combination of Propositional Dynamic Logic with ontology query languages. Our proposal is validated against the test queries of The First Provenance Challenge, and supported with an analysis of its computational properties. We consider the following setting:

contexts $\doteq$ *states of recorded data-oriented computations*
context representation language $\doteq$ *DL*
contextual information $\doteq$ *provenance metadata*
object representation language $\doteq$ *DL*
reasoning task $\doteq$ *formal verification, querying*
task-specific language $\doteq$ *provenance specification logic*

5.1 Introduction

Data provenance is the history of derivation of a data artifact from its original sources [SPG05, MCF+]. A provenance record stores all the steps and contextual aspects of the entire derivation process, including the precise sequence of operations executed, their inputs, outputs, parameters, the supplementary data involved, etc., so that third parties can unambiguously interpret the final data product in its proper context. It has been broadly acknowledged that provenance information is crucial for facilitating reuse, management and reproducibility of published data [SSH08, SPG05]. For instance, the ability of verifying whether past experiments conformed to some formal criteria is a key in the process of validation of eScientific results [MWF+07].

Problem: As provenance records can cover thousands of data items and derivation steps, one of the pressing challenges becomes the development of formal frameworks and methods to automate verification. Such a logic back-end for practical reasoning tools could, e.g. be useful for provenance-driven data querying, or for validating conformance of provenance records to formal specifications. Let us consider a concrete example taken from The First Provenance Challenge, a community effort aimed at understanding the capabilities of available provenance systems [M+08]. 17 teams competed in answering 9 queries over provenance records obtained from executing a real-life scientific workflow (see Figure 5.1) for creating population-based "brain atlases" of high resolution anatomical data. One representative task was to:

Q6. *Find all output averaged images of softmean (average) procedures, where the warped images taken as input were align warp'ed using a twelfth order nonlinear 1365 parameter model, i.e. where softmean was preceded in the workflow, directly or indirectly, by an align warp procedure with argument -m 12.*

A distinctive feature of this sort of queries is their inherent two-dimensionality: the domain data (here: image identifiers) is queried relative to its meta-level provenance description. To date all existing approaches to support such queries are based on ad hoc combinations of techniques and formalisms, dependent on the internal representation structures, and are procedural in nature. Given the semantic character of the task, and in light of the soon to be expected standardization of the Provenance vocabularies by the W3C,[1] a principled, logic-based language for querying and verifying provenance graphs, which could significantly improve reusability and generalizability, is critically missing. The work presented in this chapter aims at closing this gap.

Contributions: We introduce *provenance specification logic* (PSL^M) which, to the best of our knowledge, offers the first systematic view on the logical foundations of formal verification of data provenance records. Our focus is on data

[1] See http://www.w3.org/2011/prov/wiki/Main_Page.

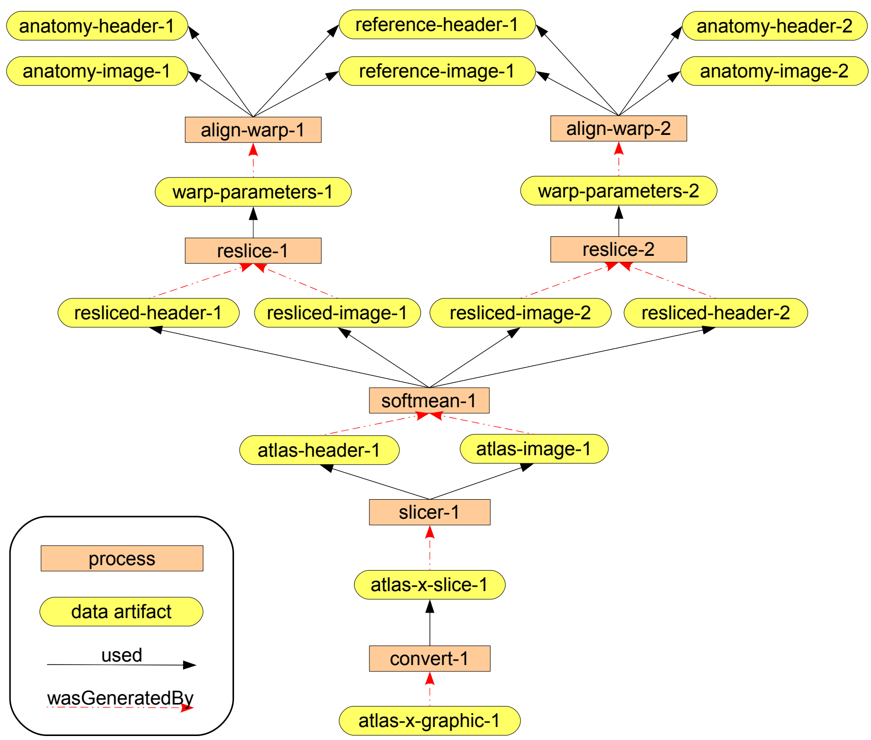

Figure 5.1: A data provenance record describing a run of The First Provenance Challenge workflow [M+08].

expressed in the Semantic Web ontology languages, such as OWL and RDF(S), whose formal core is essentially captured by Description Logics, underpinning the Semantic Web architecture.

The basic idea is very intuitive. A data provenance record is represented as a directed provenance graph (effectively a finite transition system), with certain nodes being treated as identifiers for datasets, containing the data involved in the respective stages of the computation. We construct the basic variant of our logic, called PSL, by substituting atoms of Propositional Dynamic Logic (PDL) with queries belonging to a selected query language. The dynamic component, thus inherited from PDL, enables expressing complex provenance patterns, while the embedded queries support access to data artifacts. In the second step, we lift this approach to cater for scenarios in which provenance graphs are themselves described in dedicated provenance ontologies. This way, we obtain the target formalism PSL^M, which, on top of the functionalities offered by PSL, also facilitates the use of a rich metalanguage.

This mechanism is highly independent from the employed representation formalisms, and can be reused in a plug-and-play fashion for a number of combinations of ontology/query languages. Moreover, we demonstrate that PSL^M is computationally well-behaved. By separating the DL-level reasoning tasks from the pure model checking of provenance graphs, we obtain a constant PTIME overhead over the complexity of standard DL-based reasoning tasks, invariant to the particular choice of the employed ontology/query languages, which is carried over from model checking in PDL.

In summary, in this work we deliver three main contributions:

1. We introduce PSL^M, a declarative language for expressing complex constraints over data provenance records.

2. By systematically studying the collection of test queries from The First Provenance Challenge, mentioned above, we show that PSL^M offers desired modeling capabilities.

3. Finally, we provide a computational analysis of the approach, and report on some satisfying results.

Content: In the remainder of this chapter, we first give a short overview of the related work in Section 5.2 and preliminary notions in Section 5.3. Next, in Sections 5.4 and 5.5, we incrementally introduce PSL^M and validate it against the test queries in Section 5.6. Finally, in Section 5.7 we study the computational aspects of our framework.

5.2 Related work

In the recent decade, provenance has been a subject of intensive studies in the field of database technologies, resulting in a body of foundational work [GKT07, CCT09]. Nowadays, provenance is also recognized as one of the critical problems to be addressed by the Semantic Web community, attracting increasing interest, e.g. [GG11]. Existing Semantic Web-based approaches to the problem of verification and querying, such as [GH08] are persistently technology-driven, and employ combinations of web services, ontologies, triple stores, SPARQL queries, etc. and fail to lay down systematic perspectives on the formal foundations of the problem. Noteworthy exceptions are [Mor11] and [BHPS11] which provide, respectively: reproducibility semantics, which are executional in nature, and logic programming-based framework for reasoning with provenance-annotated linked data, where both annotations and data language are specifically restricted. Our contribution goes beyond those proposals by providing a cohesive declarative semantic framework based on standard logic and ontology languages, and rich metamodels.

On the formal level, the problem of provenance verification bears a strong resemblance to the traditionally studied verification of transition systems, which in principle encourages the use of similar logic-based techniques [CGP00]. This analogy, however, must be treated with caution. While in usual transition systems states represent complete, propositional abstractions of system's configurations, in the data provenance context states are effectively datasets, reflecting the knowledge of the system in a certain configuration. This creates a need for more expressive verification formalisms, extending the basic program logics, such as PDL [Lan06]. Even Dynamic DLs [WZ00], which are capable of modeling transition systems with states corresponding to DL knowledge bases, are not flexible enough to express rich constraints on the data level. Some other verification formalisms, of a more suitable, data-oriented flavor, have been proposed for verification of data-driven systems [Via09], knowledge base programs [CDGLR11], or workflow schemas [KGMW00]. However, the central motivation behind their design is to enable representation of all permissable data-altering operations over a fixed data language, with the aim of studying general properties of programs composed of such operations. Consequently, the considered representation languages are strongly restricted in order to ensure decidability of those properties. Such general problems, however, are not of primary interest in our case, since a provenance record describes by definition a single, completed computation process, which one wants to study ex-post. Hence, rather than abstracting from the richness of a given system and focusing on its possible behaviors, we must enable reasoning machinery which can maximally utilize the available information about the system.

5.3 Preliminaries

For clarity of exposition, in this chapter we consider data represented and managed within the framework of DLs, following the paradigm of Ontology-Based Data Access, as discussed in Chapter 2. As usual, all claims made in this context extend naturally to arbitrary fragments of OWL/RDF(S) languages.

The definition of a provenance record that we adopt here, and further refine in Section 5.5, is the simplest abstraction of the proposals currently discussed in the course of a W3C standardization effort. Those proposals, building largely on the specification of the Open Provenance Model [MCF+], consider a provenance record to be a basic graph structure (such as presented in Figure 5.1) representing the whole documented history of interactions between processes and data artifacts during a certain computation, where data artifacts are in fact datasets (knowledge bases) expressed in DLs. The choice of the OPM foundations for our approach is motivated largely by the fact that OPM is suggested as the intended formalism for representing provenance in the expected W3C recommendation. In principle, however, the level of abstraction which we endorse here goes beyond particular, concrete encodings of provenance information, and builds only on generic provenance notions present also in other formalisms used for recording provenance, such as Proof Markup Language [dSMF06, MDSC07]. Crucially, our approach generalizes over any (transition) graph-based representation of data provenance.

A *directed graph* is a pair (V, E), where V is a non-empty set of nodes and E is a set of ordered pairs from $V \times V$, called edges. A *bipartite graph* is a graph $(V \cup W, E)$, where $V \cup W$ is a set of nodes and E a set of edges such that $E \subseteq V \times W \cup W \times V$. An *edge-labeled graph* is a triple (V, E, l), such that (V, E) is a graph and $l : E \mapsto R$ assigns a relation name from a set R to every edge in E. A graph (V, E) is called *acyclic iff* for every node $v \in V$, there exists no sequence $w_1, \ldots, w_n \in V$, such that $(v, w_1), \ldots, (w_{n-1}, w_n), (w_n, v) \in E$.

Definition 14 (Provenance graph). *Let $\mathcal{L}$ be a DL language and let $\mathbf{K}(\mathcal{L})$ denote the set of all knowledge bases over $\mathcal{L}$. An $\mathcal{L}$-*provenance graph *is a tuple $G = (P, D, E, l, k)$, where $(P \cup D, E, l)$ is a bipartite, directed, acyclic, edge-labeled graph, and k is a function $k : D \mapsto \mathbf{K}(\mathcal{L})$. The nodes in P are called* processes *and in D* data artifacts.

By convention, we identify process nodes with unique process invocations that occurred during the recorded computation, and data artifact nodes with the corresponding DL knowledge bases $\{k(d) \mid d \in D\}$ that were involved. Note, that we do not presume any specific causal relationships between the represented entities. We are only interested in the formal properties of the graphs.

5.4 Provenance specification logic

Formal verification is the task of checking whether a certain formal structure satisfies the property described by a given formula of a dedicated specification language. The properties of data provenance records which we aim to capture here are essentially complex relationships between the structural patterns occurring in the provenance graphs and the contents of data artifacts. Three typical constraints, representative of most reasoning tasks requested from practical provenance systems [SSH08, MWF+07, M+08], are e.g.:

1. $r(a, b)$ holds in data artifact d_1, where d_1 is reachable via edge $accessed$ from processes p_1 and p_2,

2. a data artifact in which $D(a)$ does not hold is reachable via a finite sequence of two-step edge compositions $wasGeneratedBy$-$used$ from a data artifact in which $D(a)$ holds,

3. if $D(a)$ holds in any data artifact related to process p_1 via either $input_1$ or $input_2$, then p_1 must be related via $output$ to some data artifact in which $r(a, y)$ holds, for some arbitrary y.

These informally stated properties are clearly satisfied by the respective provenance graphs, illustrated in Figure 5.2, where nodes p_1, p_2 represent process nodes, and d_1, d_2, d_3 data artifacts, whose contents are listed inside the nodes.

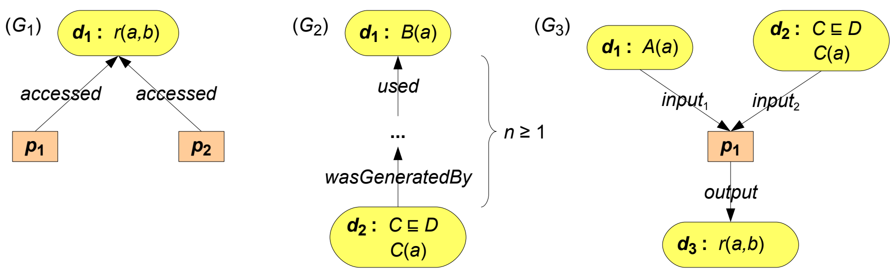

Figure 5.2: Sample provenance graphs.

The ability of expressing constraints of this flavor is the key feature of a big family of program verification formalisms based on dynamic logics, in particular the prominent Propositional Dynamic Logic (PDL) [Lan06]. The *provenance specification logic* (PSL), which we introduce below, is a data-oriented extension of PDL. Essentially, we substitute propositional letters of PDL formulas with queries belonging to a certain query language. The dynamic component of PSL enables explicit modeling of requested provenance patterns, while the queries allow for accessing the contents of data artifacts. The choice of an adequate query language is in principle an application-driven decision, depending strongly on the underlying data language. For instance, if data artifacts use RDF(S) representation, a natural candidate is SPARQL [PS08]. As our focus is on the general DL setup, we consider the class of conjunctive queries, as introduced in Section 5.3.

Definition 15 (PSL: syntax)**.** *Let $G = (P, D, E, l, k)$ be an $\mathcal{L}$-provenance graph and R the set of relation names used in G. Then the* provenance specification language *over G is the smallest language induced by the following grammar:*

***Object queries**:*
$q(\vec{x})$:= *CQs over $\mathcal{L}$*

***Path expressions**:*
$\pi := r \mid \pi;\pi \mid \pi \cup \pi \mid \pi^- \mid \pi^* \mid v? \mid \alpha?$
where $r \in R$ and $v \in P \cup D$,

***Provenance formulas**:*
$\alpha := \{q(\vec{x})\} \mid \top \mid \langle\pi\rangle\alpha \mid \alpha \wedge \alpha \mid \neg\alpha$

Whenever convenient we use the usual abbreviations $\bot = \neg\top$, $[\pi] = \neg\langle\pi\rangle\neg$, $\alpha \vee \beta = \neg(\neg\alpha \wedge \neg\beta)$ and $\alpha \rightarrow \beta = \neg\alpha \vee \beta$.

An answer to a provenance formula is a sequence of individual names from N_I, which substituted for the respective answer variables in the embedded CQs must satisfy the formula. Note, that different CQs are allowed to share same answer variables. This way one can capture interesting data dependencies between the contents of data artifacts. To formally introduce the semantics of PSL, we first fix useful notation for handling subsequences of CQ answers. Let $\vec{x} = x_1, \ldots, x_k$ be a sequence of answer variables and $\vec{a} = a_1, \ldots, a_k$ a corresponding sequence of individual names. For an arbitrary subsequence $\vec{x}' \subseteq \vec{x}$, i.e. a subset of $\vec{x}$ preserving the ordering, we write $\vec{a}|_{\vec{x}'}$ to denote the subsequence of $\vec{a}$ such that for every $1 \leq i \leq k$, a_i occurs in $\vec{a}|_{\vec{x}'}$ *iff* x_i occurs in $\vec{x}'$.

Definition 16 (PSL: semantics). *Let $\vec{x} = x_1, \ldots, x_k$ be the answer variables of a PSL provenance formula α and $\vec{a} = a_1, \ldots, a_k \in N_I$ a sequence of individual names. We call $\vec{a}$ a* certain answer *to α in $\mathcal{L}$-provenance graph $G = (P, D, E, l, k)$ in a node $v \in P \cup D$* iff *$G, v \Vdash_{\vec{a}} \alpha$, where the* satisfaction relation *$\Vdash_{\vec{a}}$ is given by a simultaneous induction over the structure of provenance formulas and path expressions. For every $v, w \in P \cup D$:*

Provenance formulas:

$G, v \Vdash_{\vec{a}} \{q(\vec{x})\}$	iff	*$v \in D$ and $k(v) \models q[\vec{a}\|_{\vec{x}}]$,*
$G, v \Vdash_{\vec{a}} \top$,		
$G, v \Vdash_{\vec{a}} \langle\pi\rangle\alpha$	iff	*there exists $w \in P \cup D$, s.t. $G \Vdash_{\vec{a}} v \xrightarrow{\pi} w$ and $G, w \Vdash_{\vec{a}} \alpha$,*
$G, v \Vdash_{\vec{a}} \alpha \wedge \beta$	iff	*$G, v \Vdash_{\vec{a}} \alpha$ and $G, v \Vdash_{\vec{a}} \beta$,*
$G, v \Vdash_{\vec{a}} \neg\alpha$	iff	*$G, v \nVdash_{\vec{a}} \alpha$,*

Path expressions:

$G \Vdash_{\vec{a}} v \xrightarrow{r} w$	iff	*$(v, w) \in E$ and $l(v, w) = r$,*
$G \Vdash_{\vec{a}} v \xrightarrow{\pi;\sigma} w$	iff	*there is $u \in P \cup D$ s.t. $G \Vdash_{\vec{a}} v \xrightarrow{\pi} u$ and $G \Vdash_{\vec{a}} u \xrightarrow{\sigma} w$,*
$G \Vdash_{\vec{a}} v \xrightarrow{\pi\cup\sigma} w$	iff	*$G \Vdash_{\vec{a}} v \xrightarrow{\pi} w$ or $G \Vdash_{\vec{a}} v \xrightarrow{\sigma} w$,*
$G \Vdash_{\vec{a}} v \xrightarrow{\pi^-} w$	iff	*$G \Vdash_{\vec{a}} w \xrightarrow{\pi} v$,*
$G \Vdash_{\vec{a}} v \xrightarrow{\pi^*} w$	iff	*$v(\xrightarrow{\pi})^* w$, where $(\xrightarrow{\pi})^*$ is the transitive reflexive closure of $\xrightarrow{\pi}$ on G,*
$G \Vdash_{\vec{a}} v \xrightarrow{v?} v$,		
$G \Vdash_{\vec{a}} v \xrightarrow{\alpha?} v$	iff	*$G, v \Vdash_{\vec{a}} \alpha$.*

Observe, that unlike in typical transition systems, only selected nodes in provenance graphs — exactly the data artifacts in D — represent the states over which object queries can be evaluated. Irrespective of this deviation, the model checking problem, underlying formal verification tasks, is defined as usual.

Model Checking 1 (PSL formulas). *Given an $\mathcal{L}$-provenance graph $G = (P, D, E, l, k)$, a node $v \in P \cup D$, a PSL provenance formula α and a sequence $\vec{a} = a_1, \ldots, a_k \in N_I$, decide whether $\vec{a}$ is a certain answer to α in G, v.*

It is easy to check that the following PSL formulas express precisely the properties from the three examples presented in the opening of this section, and are satisfied by the specified graphs, nodes and answers (Figure 5.2):

1. $\alpha := \langle p_1?; accessed; d_1?\rangle(\{r(x,y)\} \wedge \langle accessed^-; p_2?\rangle\top)$, where $G_1, p_1 \Vdash_{\vec{a}} \alpha$ for $\vec{a} = a, b$.

2. $\alpha := \{D(x)\} \wedge \langle(wasGeneratedBy; used)^*\rangle\neg\{D(x)\}$, where $G_2, d_2 \Vdash_{\vec{a}} \alpha$ for $\vec{a} = a$.

3. $\alpha := \langle p_1?\rangle(\langle(input_1 \cup input_2)^-\rangle\{D(x)\} \rightarrow \langle output\rangle\{\exists y.r(x,y)\})$, where $G_3, p_1 \Vdash_{\vec{a}} \alpha$ for $\vec{a} = a$.

For a more practical illustration, we model two use-cases from the eScience domain. The first one illustrates a typical problem of provenance-based validation of an eScience experiment, reported in [MWF+07].

Example 4 (eScience experiment validation)**.** *A bioinformatician, B, downloads a file containing sequence data from a remote database. B then processes the sequence using an* analysis service*. Later, a reviewer, R, suspects that the sequence may have been a* nucleotide *sequence but processed by a service that can only analyze meaningfully* amino acid *sequences. R determines whether this was the case.*

$$\alpha := \{\exists y.Sequence(x) \wedge analysis\text{-}result(x,y)\} \rightarrow$$
$$[output; analysis\text{-}service?; input](\{Amino\text{-}acid(x)\} \wedge \neg\{Nucleotide(x)\})$$

Solution: *The requested property is satisfied by a graph G if $G, v \Vdash_{\vec{a}} \alpha$ for every $v \in P \cup D$ and $\vec{a} = a$, where a is the individual name associated with the nucleotide sequence in question. Naturally, we implicitly assume a certain underlying representation model, where e.g. analysis-service is the name of the cited service, the result of the analysis is given via an axiom of type analysis-result(a, y), etc.*

As the second example, we formalize one of the queries from The First Provenance Challenge [M+08].

Example 5 (FPC Q6)**.** *See Q6 in Section 5.1* (cf. *Figure 5.1).*

$$\alpha := \{Image(x)\} \wedge \langle wasGeneretedBy; softmean_{1\ldots n}; used\rangle(\{\exists y.Image(y)\} \wedge$$
$$\langle(wasGeneratedBy; used)^*; wasGeneratedBy; align\text{-}warp_{1\ldots m}\rangle\top)$$

where $softmean_{1\ldots n} := softmean_1? \cup \ldots \cup softmean_n?$ includes all invocations of softmean process in the graph, while $align\text{-}warp_{1\ldots m} := align\text{-}warp_1? \cup \ldots \cup align\text{-}warp_m?$ all invocations of align warp with the specified parameter value.

Solution: *For every $v \in P \cup D$ and sequence $\vec{a}$, if $G, v \Vdash_{\vec{a}} \alpha$, then $\vec{a}$ is the requested resource.*

$Artifact \sqsubseteq \neg Process$	$Softmean \sqsubseteq Process$
$Artifact \sqsubseteq \forall wasGeneratedBy.Process$	$Align\text{-}warp \sqsubseteq Process$
$Process \sqsubseteq \forall used.Artifact$	$Align\text{-}warp \sqsubseteq \exists hasArgValue.\mathtt{String}$
$Softmean(softmean_i)$	- for every node $softmean_i$
$Align\text{-}warp(align\text{-}warp_i)$	- for every node $align\text{-}warp_i$
$hasArgValue(align\text{-}warp_i,$ "-m 12"$)$	- for every node $align\text{-}warp_i$ corresponding to an invocation of *align warp* with argument "-m 12"

Table 5.1: A DL knowledge base encoding (part of) a provenance graph.

Observe, that in the latter example not all information requested in the query can be expressed is the PSL formula in a direct, declarative manner. Namely, the selection of *softmean* and *align warp* invocations has to be encoded by an exhaustive enumeration of all the nodes satisfying the specified description. This shortcoming, which affects the high-level modeling capabilities of our formalism, is exactly what motivates the extension introduced in the next section.

5.5 Provenance metalanguage

In practice, the relevant provenance information can be much richer than reflected in our abstract notion of provenance graphs. Typically, provenance records account also for the execution context of all processes, including their parametrization, time, responsible actors, etc. [SPG05, MCF+, SSH08]. Moreover, they use complex taxonomies for classifying all these resources. Consequently, the structure of a provenance graph, along the accompanying contextual information, is commonly expressed by means of another DL-based language, used orthogonally to that representing the contents of data artifacts [BCC+11, SSH08]. For instance, the provenance graph G implicitly referred to in Example 2, would be likely represented as a knowledge base containing, among others, the axioms listed in Table 5.1.

Such a meta-level representation of provenance graphs is defined as follows.

Definition 17 (Metalanguage, metaknowledge base)**.** *Let* $G = (P, D, E, l, k)$ *be* *an $\mathcal{L}$-provenance graph and R the set of relation names used in G. Let $\mathcal{L}_G$ be a DL language with the vocabulary $\Gamma = (M_C, M_R, M_I)$, such that $R \subseteq M_R$ and $P \cup D \subseteq M_I$, and $\mathcal{K}_G$ a knowledge base over $\mathcal{L}_G$. Then, $\mathcal{L}_G$ is called the* metalanguage *and $\mathfrak{K}_G = (\mathcal{K}_G, k)$ the* metaknowledge base *over G* iff *the following conditions are satisfied:*

1. *$D = \{v \in M_I \mid \mathcal{K}_G \models Artifact(v)\}$, for a designated concept Artifact $\in M_C$,*
2. *for every $r \in R$ and $v, w \in M_I$, it holds that $(v, w) \in E$ and $l(v, w) = r$* iff *$\mathcal{K}_G \models r(v, w)$.*

Assuming the names in M_I are interpreted uniquely, it is easy to see that the structure of G is isomorphically encoded in the set of role assertions, over role names in R, entailed by $\mathcal{K}_G$. As the positions of data artifacts remain unaltered, one can immediately rephrase the definition of the satisfaction relation $\Vdash_{\vec{a}}$, to show that for any PSL formula α, node $v \in P \cup D$, and a sequence of names $\vec{a}$ it is the case that $G, v \Vdash_{\vec{a}} \alpha$ *iff* $\mathfrak{K}_G, v \Vdash_{\vec{a}} \alpha$. More interestingly, however, we can instead slightly extend the provenance specification language to make a vital use of the newly included meta-level information.

Definition 18 (PSL^M: syntax)**.** *Let $G = (P, D, E, l, k)$ be an $\mathcal{L}$-provenance graph and $\mathcal{L}_G$, $\mathfrak{K}_G$ the metalanguage and the metaknowledge base over G, respectively. Then the* provenance specification language (with metalanguage) *over $\mathfrak{K}_G$ is the smallest language induced by the grammar of PSL over G (Definition 15), modulo the revision of path expressions:*

***Path expressions**:*

$\pi := r \mid \pi;\pi \mid \pi \cup \pi \mid \pi^- \mid \pi^* \mid v? \mid C? \mid \alpha?$
where $r \in M_R$, $v \in M_I$ and C is a concept in $\mathcal{L}_G$,

The semantics is revised by relativizing the satisfaction relation w.r.t. the metaknowledge base instead of the provenance graph.

Definition 19 (PSL^M: semantics)**.** *Let $\mathcal{L}_G$ be the metalanguage with vocabulary $\Gamma = (M_C, M_R, M_I)$ and $\mathfrak{K}_G = (\mathcal{K}_G, k)$ the metaknowledge base over an $\mathcal{L}$-provenance graph G. Further, let $\vec{x} = x_1, \ldots, x_k$ be the answer variables of a PSL^M provenance formula α and $\vec{a} = a_1, \ldots, a_k \in N_I$ a sequence of individual names. We call $\vec{a}$ a* certain answer *to α in $\mathfrak{K}_G$ in a node $v \in M_I$* iff *$\mathfrak{K}_G, v \Vdash_{\vec{a}} \alpha$, where the* satisfaction relation *$\Vdash_{\vec{a}}$ is given by a simultaneous induction over the structure of provenance formulas and path expressions. For all individual names $v, w \in M_I$:*

Provenance formulas:

$$
\begin{array}{lll}
\mathfrak{K}_G, v \Vdash_{\vec{a}} \{q(\vec{x})\} & \text{iff} & \mathcal{K}_G \models \mathit{Artifact}(v) \text{ and } k(v) \models q[\vec{a}|_{\vec{x}}], \\
\mathfrak{K}_G, v \Vdash_{\vec{a}} \langle \pi \rangle \alpha & \text{iff} & \text{there exists } w \in M_I \text{, s.t. } \mathfrak{K}_G \Vdash_{\vec{a}} v \xrightarrow{\pi} w \text{ and } \\
& & \mathfrak{K}_G, w \Vdash_{\vec{a}} \alpha,
\end{array}
$$

Path expressions:

$$
\begin{array}{lll}
\mathfrak{K}_G \Vdash_{\vec{a}} v \xrightarrow{r} w & \text{iff} & \mathcal{K}_G \models r(v, w), \\
\mathfrak{K}_G \Vdash_{\vec{a}} v \xrightarrow{\pi;\sigma} w & \text{iff} & \text{there is } u \in M_I \text{ s.t. } \mathfrak{K}_G \Vdash_{\vec{a}} v \xrightarrow{\pi} u \text{ and } \mathfrak{K}_G \Vdash_{\vec{a}} \\
& & u \xrightarrow{\sigma} w, \\
\mathfrak{K}_G \Vdash_{\vec{a}} v \xrightarrow{C?} v & \text{iff} & \mathcal{K}_G \models C(v),
\end{array}
$$

where the remaining conditions are exactly as in Definition 16 (modulo the replacement $G/\mathfrak{K}_G$).

The model checking problem is rephrased accordingly.

Model Checking 2 (PSL^M formulas)**.** *Given a metaknowledge base $\mathfrak{K}_G$ over an $\mathcal{L}$-provenance graph G, an instance $v \in M_I$, a PSL^M provenance formula α, and a sequence $\vec{a} = a_1, \ldots, a_k \in N_I$, decide whether $\vec{a}$ is a certain answer to α in $\mathfrak{K}_G, v$.*

The usefulness of the presented extension, in particular of the test operator $C?$, which allows for referring to graph nodes generically by their types, inferred from the metaknowledge base, can be observed in the following example.

Example 6 (FPC Q6 cont.)**.** *See Q6 in Section 5.1 and Example 5. We restate the formula α as:*

$$
\begin{array}{c}
\alpha := \{\mathit{Image}(x)\} \wedge \langle \mathit{wasGeneretedBy}; \mathit{Softmean}?; \mathit{used}\rangle \\
(\{\exists y.\mathit{Image}(y)\} \wedge \langle(\mathit{wasGeneratedBy}; \mathit{used})^*; \mathit{wasGeneratedBy}; \\
(\mathit{Align\text{-}warp} \sqcap \exists \mathit{hasArgValue}.\{\text{“-m 12”}\})?\rangle \top)
\end{array}
$$

where $\mathcal{K}_G$, in the metaknowledge base $\mathfrak{K}_G = (\mathcal{K}_G, k)$, contains (among others) the axioms from Table 5.1.

Solution: *For every $v \in M_I$ and a sequence $\vec{a}$, if $\mathfrak{K}_G, v \Vdash_{\vec{a}} \alpha$, then $\vec{a}$ is a requested resource.*

Compared to its PSL variant from Example 5, the PSL^M formula used in Example 6 is much more succinct and explicitly represents all requested information. More importantly, thanks to the use of a generic vocabulary for classifying nodes (here: concepts $\mathit{Softmean}$, $\mathit{Align\text{-}warp} \sqcap \exists \mathit{hasArgValue}.\{\text{“-m 12”}\}$), instead of their enumerations, as done before, the formula is also more input-independent, in the sense that it can be directly reused to verify/query alternative provenance records obtained from different executions of the same workflows.

5.6 Evaluation

In order to validate our approach in practical scenarios, we have analyzed the complete list of test queries from The First Provenance Challenge, which to our knowledge constitutes a so far unique 'golden standard' for the provenance community. Below we model possible solutions using the logic PSL^M and elaborate on our findings.

Q1. *Find the process that led to Atlas X Graphic / everything that caused Atlas X Graphic to be as it is. This should tell us the new brain images from which the averaged atlas was generated, the warping performed etc.*

$$\alpha_1 := (\top \vee \{\top(x)\}) \wedge \langle (\mathit{used}^- \cup \mathit{wasGeneratedBy}^-)^*; \textit{Atlas-X-Graphic}?\rangle \top$$

Solution: Every $v \in M_I$ and $\vec{a}$ such that $\mathfrak{K}, v \Vdash_{\vec{a}} \alpha_1$ are requested resources.

Q2. *Find the process that led to Atlas X Graphic, excluding everything prior to the averaging of images with softmean.*

$$\alpha_2 := \alpha_1 \wedge [(\mathit{used}^- \cup \mathit{wasGeneratedBy}^-)^*]\neg\langle \mathit{wasGeneratedBy}^-; \mathit{Softmean}?\rangle \top$$

Solution: Every $v \in M_I$ and $\vec{a}$ such that $\mathfrak{K}, v \Vdash_{\vec{a}} \alpha_2$ are requested resources.

Q3. *Find the Stage 3, 4 and 5 details of the process that led to Atlas X Graphic.*

Comment: This is a complex search/verification task, whose reasoning parts can be accomplished by a mix of formulas used in Q1, Q2. Essentially, one must decide what the relevant details are and retrieve them by applying appropriate provenance formulas over the provenance graphs.

Q4. *Find all invocations of procedure align warp using a twelfth order nonlinear 1365 parameter model, i.e. align warp procedure with argument -m 12, that ran on a Monday.*

$$\alpha_4 := \langle (\textit{Align-warp} \sqcap \exists \mathit{hasArgValue}.\{\text{“-m 12”}\} \sqcap \exists \mathit{executedOn}.\mathit{Monday})?\rangle \top$$

Solution: Every $v \in M_I$ such that $\mathfrak{K}, v \Vdash \alpha_4$ is a requested resource.

Q5. *Find all Atlas Graphic images outputted from workflows where at least one of the input Anatomy Headers had an entry global maximum=4095. The contents of a header file can be extracted as text using the scanheader AIR utility.*

$$\alpha_5 := \langle \mathit{AtlasGraphic}? \rangle (\{\text{Image}(x)\} \wedge \langle (\mathit{wasGeneratedBy}; \mathit{used})^* \rangle \\ \langle \mathit{AnatomyHeader}? \rangle \{\text{hasValue}(\text{global-maximum}, \text{“4095”})\})$$

Solution: For every $v \in M_I$ and $\vec{a}$ such that $\mathfrak{K}, v \Vdash_{\vec{a}} \alpha_5$, $\vec{a}$ is a requested resource.

Q6. Example 6, discussed in the previous section.

Q7. *A user has run the workflow twice, in the second instance replacing each procedure (convert) in the final stage with two procedures: pgmtoppm, then pnmtojpeg. Find the differences between the two workflow runs. The exact level of detail in the difference that is detected by a system is up to each participant.*

Comment: This is a complex search/verification task, whose reasoning parts can be accomplished by posing a number of model checking problems. Essentially, for each relevant provenance formula one must verify it over both graphs and compare the obtained answers.

Q8. *A user has annotated some anatomy images with a key-value pair center=UChicago. Find the outputs of align warp where the inputs are annotated with center=UChicago.*

$$\alpha_8 := \{\top(x)\} \wedge \langle \mathit{wasGeneratedBy}; \mathit{Align\text{-}warp}?; \mathit{used}; \mathit{AnatomyImage}? \rangle \\ \{\exists y.(\text{Image}(y) \wedge \text{center}(y, \text{UChicago}))\}$$

Solution: For every $v \in M_I$ and $\vec{a}$ such that $\mathfrak{K}, v \Vdash_{\vec{a}} \alpha_8$, $\vec{a}$ is a requested resource.

Q9. *A user has annotated some atlas graphics with key-value pair where the key is studyModality. Find all the graphical atlas sets that have metadata annotation studyModality with values speech, visual or audio, and return all other annotations to these files.*

$$\alpha_9 := \langle \mathit{AtlasGraphic}?; \exists \mathit{studyModality}.\{\mathit{speech}\}? \cup \exists \mathit{studyModality}.\{\mathit{visual}\}? \cup \\ \exists \mathit{studyModality}.\{\mathit{radio}\}? \rangle \top$$

Solution: Every $v \in M_I$ such that $\mathfrak{K}, v \Vdash \alpha_9$ is a requested resource. Finding other annotations can be accomplished by posing a number of model checking problems w.r.t. the identified resources.

The above analysis shows that typical reasoning tasks over data provenance records consist of two components: search and logical verification. As far as verification is concerned, the logic PSL^M proves well suited for modeling requested properties and queries. In particular, out of the 9 considered problems, at least 5 — Q1, Q2, Q5, Q6, Q8 — can be solved directly, using a combination of all distinctive features of PSL^M, namely: PDL-like path expressions, embedded CQs and the metalanguage. Queries Q4, Q9 can be answered without the use of embedded CQs. Problems Q3, Q7 and partially Q9 are in fact descriptions of complex search/verification tasks, which can be decomposed into a number of individual verification problems. Those, in turn, can be addressed using PSL^M in the same fashion as in the remaining cases.

5.7 Reasoning and complexity

The close relationship of PSL^M to PDL can be conveniently exploited on the computational level. Crucially, PSL^M model checking can be decoupled into two separate problems:

1. construction of a finite-state transition system and a PDL formula (involving polynomially many CQ answering / DL entailment problems),

2. PDL model checking.

This technically unsurprising result has some significant theoretical and practical implications. From the theoretical perspective, it allows for identifying a complexity bound, invariant to the cost of reasoning with the particular DL languages used in the representation. From the practical viewpoint, it opens up a possibility of building simple, yet well-grounded and efficient reasoning architectures based on existing, highly optimized DL reasoners, query engines (e.g. Pellet, Mastro), and PDL model checkers (e.g. MCPDL[2]).

In the remainder of this section we formally outline the reduction procedure.

[2]See http://www2.tcs.ifi.lmu.de/~axelsson/veri_non_reg/pdlig_mc.html.

Definition 20 (Transition system). *Let $\mathcal{P} = \{p, q, \ldots\}$ be a set of propositional letters and $\mathcal{A} = \{r, s, \ldots\}$ a set of atomic program names. Then a* finite-state transition system *is a tuple $\mathcal{S} = (W, \{\xrightarrow{r} \mid r \in \mathcal{A}\}, \mathcal{I}\})$, where:*

- *W is a finite, non-empty set of elements called* states*,*
- *$\xrightarrow{r} \subseteq W \times W$ is a* transition relation *corresponding to program r,*
- *$\mathcal{I} : W \mapsto 2^{\mathcal{P}}$ is a mapping assigning a propositional* valuation *to every state from W.*

Let α be a PSL^M provenance formula, $\vec{a} = a_1, \ldots, a_k \in N_I$ a sequence of individual names, $\mathcal{L}_G$ a metalanguage with vocabulary $\Gamma = (M_C, M_R, M_I)$, and $\mathfrak{K}_G = (\mathcal{K}_G, k)$ a metaknowledge base over an $\mathcal{L}$-provenance graph G. By $ind(\mathcal{K}_G)$ we denote the set of individual names occurring in $\mathcal{K}_G$, by $ind(\alpha)$ the set of individual names from M_I occurring in α (test operators $v?$), by $CQ(\alpha)$ the set of CQs occurring in α, by $con(\alpha)$ the set of concepts in $\mathcal{L}_G$ occurring in α (test operators $C?$), by $rol(\mathcal{K}_G)$ the set of role names occurring in $\mathcal{K}_G$. We define a finite-state transition system $\mathcal{S}(\mathfrak{K}_G, \vec{a}, \alpha) = (W, \{\xrightarrow{r} \mid r \in \mathcal{A}\}, \mathcal{I}\})$ as follows:

1. $W := ind(\mathcal{K}_G)$,
2. $\xrightarrow{r} := \{(v, w) \mid v, w \in W; \mathcal{K}_G \models r(v, w)\}$, for every $r \in rol(\mathcal{K}_G)$,
3. for every $v \in W$, $\mathcal{I}(v)$ is the smallest set containing propositions:
   (a) $p_v \in \mathcal{I}(v)$,
   (b) $p_{artifact} \in \mathcal{I}(v)$ *iff* $\mathcal{K}_G \models Artifact(v)$,
   (c) $p_C \in \mathcal{I}(v)$ *iff* $\mathcal{K}_G \models C(v)$, for every $C \in con(\alpha)$,
   (d) $p_{q[\vec{a}|_{\vec{x}}]} \in \mathcal{I}(v)$ *iff* $k(v) \models q[\vec{a}|_{\vec{x}}]$, for every $q(\vec{x}) \in CQ(\alpha)$.

Next, we transform the formula α by consistently applying the following substitutions, where $p_v, p_C, p_{artifact}, p_{q[\vec{a}|_{\vec{x}}]} \in \mathcal{P}$:

- p_v for every occurrence of $v \in ind(\alpha)$ in α,
- p_C for every occurrence of $C \in con(\alpha)$ in α,
- $(p_{artifact} \wedge p_{q[\vec{a}|_{\vec{x}}]})$ for every occurrence of $q(\vec{x}) \in CQ(\alpha)$,

The resulting expression α^{PDL} is clearly a well-formed PDL formula. Thanks to such "propositionalization" of the input we obtain the following reduction result, where $\mathcal{S}, v \models \varphi$ denotes the model checking problem in PDL, i.e. the problem of deciding whether a PDL formula φ is satisfied in state v of the transition system $\mathcal{S}$.

Theorem 12 (PSL^M vs. PDL)**.** $\mathfrak{K}_G, v \Vdash_{\vec{a}} \alpha$ iff $\mathcal{S}(\mathfrak{K}_G, \vec{a}, \alpha), v \models \alpha^{\mathrm{PDL}}$.

Proof. By the construction procedure, $\mathcal{S}(\mathfrak{K}_G, \vec{a}, \alpha)$ is isomorphic to the graph of role assertions $r(v, w)$ entailed by $\mathcal{K}_G$. Clearly, the isomorphism preserves edge labeling and marking of the data artifact states (by means of the proposition $p_{artifact}$). By the structural induction over α and α^{PDL} one can see that the satisfaction relation is also preserved. We demonstrate three key cases. Suppose $\mathfrak{K}_G, v \Vdash_{\vec{a}} \alpha$ and $\alpha = q(\vec{x})$ for some CQ $q(\vec{x})$. But then it must be the case that $p_{q[\vec{a}|_{\vec{x}}]} \in \mathcal{I}(v)$, while $\alpha^{\mathrm{PDL}} = p_{q[\vec{a}|_{\vec{x}}]}$. Hence $\mathcal{S}(\mathfrak{K}_G, \vec{a}, \alpha), v \models \alpha^{\mathrm{PDL}}$. Suppose now $\alpha = \langle v? \rangle \beta$, so that $\mathfrak{K}_G, v \Vdash_{\vec{a}} \beta$. By construction we have $p_v \in \mathcal{I}(v)$ and $\alpha^{\mathrm{PDL}} = \langle p_v? \rangle \mu(\beta)^{\mathrm{PDL}}$. Hence, $\mathcal{S}(\mathfrak{K}_G, \vec{a}, \alpha), v \models \beta^{\mathrm{PDL}}$. Finally, let $\alpha = \langle C? \rangle \beta$. Then it must be the case that $\mathcal{K}_G \models C(v)$ and so $p_C \in \mathcal{I}(v)$. By construction $\alpha^{\mathrm{PDL}} = \langle p_C? \rangle \beta^{\mathrm{PDL}}$, and therefore $\mathcal{S}(\mathfrak{K}_G, \vec{a}, \alpha), v \models \beta^{\mathrm{PDL}}$. The opposite direction follows analogically. ❑

It is known that the complexity of model checking in PDL is PTIME-complete [Lan06]. Moreover, it is easy to see that the size of the transition system $\mathcal{S}(\mathfrak{K}_G, \vec{a}, \alpha)$ and of the formula α^{PDL} is polynomial in $\ell(\mathfrak{K}_G, \alpha, \vec{a})$, where $\ell(\mathfrak{K}_G, \alpha, \vec{a})$ is the total size of $\mathfrak{K}_G$, α and $\vec{a}$ measured in the number of symbols used. This means, that by disregarding the cost of DL reasoning involved in the construction of $\mathcal{S}(\mathfrak{K}_G, \vec{a}, \alpha)$, we obtain the following upper time bound.

Theorem 13 (PSL^M model checking: complexity)**.** *Let $\mathfrak{K}_G$ be a metaknowledge base, expressed in $\mathcal{L}_G$, over an $\mathcal{L}$-provenance graph G. Model checking PSL^M formulas over $\mathfrak{K}_G$ is in* PTIME^{DL}*, where DL is an oracle answering CQs in $\mathcal{L}$ and deciding DL entailment in $\mathcal{L}_G$.*

Proof. The PTIME^{DL} upper bound follows by Theorem 12 and by observing that constructing $\mathcal{S}(\mathfrak{K}_G, \vec{a}, \alpha)$, for given $\mathfrak{K}_G, \alpha, \vec{a}$, requires time polynomial in $\ell(\mathfrak{K}_G, \alpha, \vec{a})$, plus at most polynomially many calls to the oracle DL, involved in steps (2) and (3b-d) of the construction procedure. The formula α^{PDL} can be constructed in time linear in $\ell(\vec{a}, \alpha)$. ❑

Finally, we observe that for a given problem $\mathfrak{K}_G, v \Vdash_{\vec{a}} \alpha$ there are at most $2^{\ell(\mathfrak{K}_G, \alpha)}$ different sequences $\vec{a}$, and thus, maximum $2^{\ell(\mathfrak{K}_G, \alpha)}$ different possible pairs $\mathcal{S}(\mathfrak{K}_G, \vec{a}, \alpha), \alpha^{\mathrm{PDL}}$ to be considered. In practice this number can be dramatically reduced by using smart heuristics to guess only potentially "promising" sequences $\vec{a}$. Analogically, the described procedure of constructing the transition systems leaves a considerable space for practical optimizations.

5.8 Conclusion

In this chapter we have introduced the provenance specification logic PSL^M— a dynamic logic-based formalism for verification of data provenance records. The validation, which we have conducted using the test queries of The First Provenance Challenge, shows that a typically requested reasoning task over a data provenance record consists of two components: search and logical verification. As far as the search aspect goes beyond the scope of this work and remains an interesting problem in its own right, requiring smart retrieval and heuristic techniques, we have demonstrated that the logical reasoning part can be successfully captured using the logic and the framework developed here. Moreover, we have shown that the computational cost of performing such tasks is very moderate, and depends mostly on the expressiveness of the languages used for representing the data and the provenance record.

With this contribution, we hope to make a novel and promising link between the traditional field of formal verification and the newly emerging area of reasoning with provenance on the Semantic Web. We have also demonstrated that the meta-level provenance information, which is commonly recorded along the domain data in many existing eScience applications, can be effectively utilized for context-aware reasoning with such data.

CHAPTER

SIX

REPRESENTING AND QUERYING TEMPORAL DATA

In this chapter, we apply the context framework to the problem of reasoning with temporal data, and define a generic mechanism for constructing corresponding temporal query languages, based on combinations of linear temporal logics with ontology query languages. We elaborate on the practicality of our approach by enriching the query language and data annotations with additional temporal terms, and by proposing special restrictions that render temporal querying computationally cheap and relatively straightforward to implement. We consider the following setting:

contexts $\doteq$ *time points*

context representation language $\doteq$ *variants of linear temporal logics*

contextual information $\doteq$ *calendric descriptions*

object representation language $\doteq$ *DL with temporal annotations over ABox axioms*

reasoning task $\doteq$ *querying*

task-specific language $\doteq$ *temporal query languages*

6.1 Introduction

The use of ontologies for describing and interpreting data is acknowledged by now as a self-standing paradigm of data management in different areas of computer science — prominently also on the Semantic Web. One big and yet unresolved challenge in this context, called for by numerous applications, is to formally incorporate and operationalize the notion of data's *validity time*, i.e. the explicitly declared time span within which the data is known to be true.

Problem: In this chapter, we study the problem of managing temporal data in the framework of DLs. Our goal is to make first steps towards establishing a unifying approach to representing and querying such data under DL ontologies. Given the multifaceted nature of the problem and the scope of expected applications, one of main challenges faced lies in reconciling a number of valuable contributions developed within diverse research areas. In particular:

- *temporal databases*: for ensuring commensurability with the commonly adopted temporal data models for representing validity time and with standard query languages based on temporal first-order logic,
- *query answering* in DLs: for enabling transfer of known query answering techniques, complexity results, and facilitating reuse of existing tools,
- SW *temporal vocabularies*: for supporting typical SW practices involving OWL-based time ontologies, which provide rich temporal vocabularies employed on the level of queries and data annotations,
- *temporal DLs*: for enabling the possibility of managing temporal data under DL ontologies which capture temporal constraints on the intensional level.

Clearly, under such a variety of influences, it is critical to carefully balance theoretical foundations of a proposed approach with good prospects for reusing existing techniques, tools and methodologies.

Contributions: We introduce a basic framework for representing temporal data in arbitrary DLs, where the data takes the form of time-stamped ABox assertions $[t_1, t_2] : \alpha$, stating validity of the assertion α during the interval $[t_1, t_2]$. Then we propose a general mechanism of defining corresponding temporal query languages, based on combinations of linear temporal logics with classes of first-order queries — specifically, with well-known conjunctive queries, in a similar manner as advocated in Chapter 5, where we combined Propositional Dynamic Logic with CQs. In particular:

- we systematically motivate the proposed mechanism, present the syntax and certain answer semantics for the query languages that the mechanism generates, and the relationship of those languages to temporal first-order logic.

- we advocate a controlled use of epistemic semantics in order to warrant practical query answering in the defined setting. Under this restriction, we obtain a $\text{PSPACE}^{QA(\mathcal{L})}$ upper bound for the combined complexity of answering temporal queries in an arbitrary DL $\mathcal{L}$, where $QA(\mathcal{L})$ is an oracle answering conjunctive queries in $\mathcal{L}$. We highlight some essential theoretical and practical implications of this result.

- further we extend the entire framework with meta-level modeling features which allow for defining abstract temporal vocabularies to be used on the level of queries and data annotations. We show that given this extension, query answering remains in $\text{PSPACE}^{QA(\mathcal{L})}$.

- we discuss the possibility of pushing the approach further towards integration with temporal DLs.

Content: In Section 6.2, we provide a comprehensive overview of the background research and identify the core requirements for the proposed approach. Next, we introduce the temporal data model in Section 6.3. In Section 6.4, we present and study the proposed mechanism of defining temporal query languages. Further, in Section 6.5 we extend the framework with a rich temporal metalanguage. In Section 6.6, we discuss similarities to existing approaches and outline some future research directions. We conclude the chapter in Section 6.7.

6.2 Overview and background

Extending information systems with capabilities for managing temporal information has been deeply studied and advocated in many areas of computer science, particularly, in those concerned with relational databases and knowledge representation. Surprisingly, despite the successful use of the ontology-based data access (OBDA) paradigm as an application of DL technologies in databases, the development of mechanisms for extending the OBDA approach towards accessing temporal data has not been yet investigated, with very few limited exceptions discussed in Section 6.6. A proposed mechanism should naturally take into account the already well-founded research lines on representing and querying temporal information, as well as valuable contributions in related areas, which we outline in the following paragraphs.

Ontology-Based Data Access The *ontology-based data access* is a paradigm of managing data in presence of background knowledge, as introduced in Chapter 2. One of the key motivations behind the design of the temporal query languages presented in this chapter is to enable easy, modular reuse of the known techniques and results on query answering in DLs in the context of temporal data querying.

Temporal Databases During the 90s, the database community conducted an extensive study on temporal extensions of the standard relational data models, supporting management of temporal information. The common way of constructing *temporal relational databases* (TDBs) is to enrich traditional data models with *time-stamps* representing data's validity time, i.e. the time span within which the data is known to be true. As one of the crucial requirements for our approach we pose formal compatibility with the TDB paradigm of representing temporal data. Inspired by the notion of *concrete temporal database* [CT05], we construct a *temporal ABox* by time-stamping every ABox assertion with a weak-interval of the form $[t_1, t_2]$, compactly representing a set of time points in which the assertion is valid. The semantics of a temporal ABox, by analogy to the TDBs case [CT05], is given by mapping each time point in the underlying *time domain* to the non-temporal (standard) ABox — a so-called *snapshot* — containing exactly the assertions valid in that point. Eventually, the OBDA paradigm is applied within the scope of respective snapshots.

Regarding the choice of the time domain, the TDB literature reports on a number of possible representations, each one having far-reaching philosophical, logical and computational consequences [MC09]. The available degrees of freedom concern, among others: the nature of the atomic time entities (points *vs.* intervals), their ordering relationships (linear *vs.* branching *vs.* partial orders), the density (discrete *vs.* continuous), the boundaries (finite *vs.* infinite). Although strict commitment to any representation is always arbitrary to some extent, arguably one of the most natural and commonly used setups in TDBs, which we also adopt here, is the one capturing the intuition of a point-based time line [MC09].

A temporal data model is complemented by an adequate *temporal query* *language* for querying temporal data. In this aspect, we ground our proposal in two well-known research lines. 1) Following the research on TDBs, we consider languages based on fragments of *temporal first-order logic*, which has been advocated as a suitable high-level formalism for querying TDBs [CT05]. It has been shown, that queries expressed in temporal first-order logic can be translated directly to TSQL2 [BCST96] — a temporal extension of the standard database query language SQL — and thus efficiently handled using existing TDB systems. 2) Given the known landscape of complexity results and developed techniques for query answering in DLs, we pay special attention to the expressiveness of the first-order component within the intended fragments of temporal first-order logic. As explained in detail in Section 6.4, our motivation is to provide a mechanism for defining such fragments in a controlled, modular manner, by selecting particular sets of temporal operators and particular classes of first-order queries to be combined. By specifying those two parameters one should effectively obtain a ready query language of a well-characterized computational behavior. To this end we make use of the methodology of temporalizing logic systems [FG92].

Semantic Web In recent years, the problem of managing time-varying knowledge has gained a lot of interest also in the Semantic Web research community. Particularly, the need for describing temporal information on the Web gave rise to various *time ontologies* [HP04], which formalize common temporal notions, such as temporal instants, temporal intervals and calendar terms, and offer standardized formats for representing different types of temporal information. Although such ontologies succeed in facilitating exchange of time-oriented data among Web agents, they are not accompanied by any formally grounded methodologies of processing such information. Specifically, they offer no inference mechanisms to support genuinely temporal reasoning. This lack of rigorous logical foundations, is in practical scenarios partially remedied by the use of programming tools and ad-hoc hybrid architectures [BSP11, OD11]. Following this demand of formally supporting rich temporal vocabularies, we show that such an extension can be quite smoothly integrated with our proposed framework by suitably axiomatizing the structure of the time domain and employing additional predicates in queries and data annotations.

Temporal DLs A somewhat orthogonal research effort has gone into designing a family of *temporal Description Logics* (TDLs) [LWZ08] tailored for representing and reasoning with inherently temporal terminologies. As proper combinations of temporal logics with DLs, TDLs count with a well-defined temporal semantics, which makes them very appealing from the theoretical perspective. Nevertheless, most of the contributions in this area focus on traditional reasoning tasks such as satisfiability and subsumption, related mostly to conceptual modeling rather than querying temporal data, with very few, limited exceptions [AFWZ02]. In general, a potential transfer of the known query answering techniques for DLs to the TDL setting seems highly non-trivial. Although in this thesis we do not address the problem of querying temporal data in presence of TDL ontologies, we do acknowledge it as a worthwhile challenge for future research, and we briefly reconsider it in Section 6.6.

6.3 Temporal data model

A temporal data model is formally specified by two basic characteristics: the choice of the underlying time domain and the syntax and semantics of temporal annotations linking data to the time domain. As outlined in Section 6.2, a time domain permitted in our scenario is a structure defined as a linear ordering of a set of time instants [MC09].

Definition 21 (Time domain)**.** *A* time domain *is a tuple* $(T, <)$*, where* T *is a nonempty set of elements called* time instants *and* $<$ *is an irreflexive, linear ordering on* T*.*

Some popularly considered structures satisfying this definition are based on sets of numbers, e.g. naturals $(\mathbb{N}, <)$, integers $(\mathbb{Z}, <)$, reals $(\mathbb{R}, <)$, with the ordering $<$ being interpreted as the usual *smaller-than* relation. By convention, we write $\leq$ to denote the reflexive closure of $<$.

Temporal annotations are based on the weak-interval time-stamping mechanism. Intuitively, a time-stamped ABox assertion $[t_1, t_2] : \alpha$ states that the ABox assertion α is valid in all time instants within the interval $[t_1, t_2]$. Additionally, we allow special symbols $-\infty$ and $+\infty$ to represent possibly unbounded intervals.

Definition 22 (Temporal ABox)**.** *Let* $(T, <)$ *be a time domain. A* temporal assertion *is an expression in one of the following forms:*

$$[t_1, t_2] : \alpha \;\mid\; [-\infty, t_1] : \alpha \;\mid\; [t_1, +\infty] : \alpha \;\mid\; [-\infty, +\infty] : \alpha$$

where α *is an ABox assertion and* $t_1, t_2 \in T$*. A* temporal ABox $\mathcal{A}_T$ *is a finite set of temporal ABox assertions. A* t-snapshot *of* $\mathcal{A}_T$*, for* $t \in T$*, is the smallest ABox* $\mathcal{A}_T(t)$ *containing all assertions* α*, for which any of the following conditions hold:*

$$
\begin{aligned}
[t_1, t_2] &: \alpha \in \mathcal{A} \ \textit{and} \ \ t_1 \leq t \leq t_2, \\
[-\infty, t_1] &: \alpha \in \mathcal{A} \ \textit{and} \ \ t \leq t_1, \\
[t_1, +\infty] &: \alpha \in \mathcal{A} \ \textit{and} \ \ t_1 \leq t, \\
[-\infty, +\infty] &: \alpha \in \mathcal{A}.
\end{aligned}
$$

The standard DL semantics is extended in a natural way by adding the temporal dimension and assigning a single DL interpretation to every time instant.

Definition 23 (Snapshot semantics)**.** *Let* $(T, <)$ *be a time domain,* $\mathcal{T}$ *a TBox and* $\mathcal{A}_T$ *a temporal ABox. A* temporal interpretation *of* $\mathcal{T}$ *and* $\mathcal{A}_T$ *is a tuple* $\mathfrak{M} = (T, <, \mathcal{I})$*, where* $\mathcal{I}$ *is a function assigning to every* $t \in T$ *a DL interpretation* $\mathcal{I}(t) = (\Delta(t), \cdot^{\mathcal{I}(t)})$*. We say that* $\mathfrak{M}$ *is a* model *of* $\mathcal{T}$ *and* $\mathcal{A}_T$*, whenever* $\mathcal{I}(t)$ *is a model of* $\mathcal{T}$ *and* $\mathcal{A}_T(t)$*, for every* $t \in T$*.*

6.4 Temporal query language

In this section, we define and study a novel temporal query language $\mathcal{TQL}$, or strictly speaking, a family of such languages for querying temporal ABoxes *w.r.t.* standard TBoxes. At its core, our contribution should be seen as a general mechanism for constructing practical query formalisms, based on combinations of temporal logics with certain classes of first-order queries over DLs. This mechanism can be shortly described as follows. Consider a temporal logic $\mathcal{TL}$ and a class of queries $\mathcal{Q}$. We aim at identifying a fragment of *temporal first-order logic*, based on the operators of $\mathcal{TL}$, whose first-order component coincides with the class $\mathcal{Q}$. To this end, we follow the well-studied methodology of *temporalization of logic systems*, introduced by Finger and Gabbay [FG92]. Essentially, $\mathcal{TQL}$ is defined as the set of all $\mathcal{TL}$-formulas whose atomic subformulas are substituted with $\mathcal{Q}$-queries. The central motivation behind such a construction is to enable decoupling the problem of answering embedded $\mathcal{Q}$-queries from reasoning in $\mathcal{TL}$, which can be both efficiently addressed by existing, specialized tools. As it turns out, some potential interactions between the Boolean operators of $\mathcal{TL}$-formulas with those of $\mathcal{Q}$-queries make such decoupling still impossible in general. Hence, as a solution, we advocate a controlled use of epistemic semantics for interpreting the embedded $\mathcal{Q}$-queries, along the lines proposed by Calvanese et al. [CGL+07]. This, as we argue in Section 6.4.2, leads to a desirable theoretical and practical characterization of $\mathcal{TQL}$.

In our scenario, we focus on the class of conjunctive queries, as the most popular type of queries studied in the context of DLs. As the baseline temporal language we consider *first-order monadic logic of orders* (FOMLO), which is known to be expressively complete w.r.t. all linear orders [Rey10], and thus subsumes a number of most popular linear temporal logics, including the prominent Propositional Linear Temporal Logic (PLTL). The syntax of FOMLO is based on a countably infinite set T_V of time instant variables, such that $T_V \cap N_V = \emptyset$, one binary predicate $<$ and a countably infinite set P_V of monadic predicate variables.

Definition 24 (FOMLO: syntax)**.** *Let* $(T, <)$ *be a time domain. A FOMLO-formula is an expression constructed according to the grammar:*

$$u < v \mid \neg\varphi \mid \psi \wedge \varphi \mid \forall x.\varphi \mid X(u)$$

where $u, v \in T \cup T_V$*,* $x \in T_V$ *and* $X \in P_V$*.*

The satisfaction relation is defined in terms of the standard first-order semantics, *modulo* an extra condition warranting the satisfaction of constructs substituted for the predicate variables.

Definition 25 (T-substitution)**.** *For a time domain* $(T, <)$*, a* T-substitution *is a mapping* $\pi : T \cup T_V \mapsto T$ *such that* $\pi(t) = t$ *for every* $t \in T$*.*

Definition 26 (FOMLO: satisfaction relation)**.** *For a time domain* $\mathfrak{M} = (T, <)$ *and a* T*-substitution* π*, the satisfaction relation for FOMLO-formulas is defined inductively as follows:*

- $\mathfrak{M}, \pi \models u < v$ iff $\pi(u) < \pi(v)$*,*
- $\mathfrak{M}, \pi \models \neg\varphi$ iff $\mathfrak{M}, \pi \not\models \neg\varphi$*,*
- $\mathfrak{M}, \pi \models \varphi \wedge \psi$ iff $\mathfrak{M}, \pi \models \varphi$ *and* $\mathfrak{M}, \pi \models \psi$*,*
- $\mathfrak{M}, \pi \models \forall x.\varphi$ iff *for every* $t \in T$ *it is the case that* $\mathfrak{M}, \pi[x \mapsto t] \models \varphi$*,*
- $\mathfrak{M}, \pi \models X(u)$ iff *the formula substituted for* X *is satisfied in* $\mathfrak{M}, \pi(u)$*,*

where $\pi[x \mapsto t]$ *denotes a* T*-substitution exactly as* π *except for that we fix* $\pi(x) = t$*.*

Apart from the constructs defined above, we also use some common abbreviations $\exists, \vee, \rightarrow, \leftrightarrow, \leq, =$, as well as compositions $x_1 < x_2 < x_3$, defined in the usual manner. In the remainder of this chapter, we extend FOMLO and its semantics with additional components in order to obtain several variants of query and representation languages. By a slight abuse of notation, we keep using the same symbol $\mathfrak{M}$ do denote different extensions of temporal interpretations. The semantics of the proposed language extensions is backward compatible with that of FOMLO. For instance, in the next section we extend $\mathfrak{M} = (T, <)$ with DL interpretations, obtaining interpretations of the form $\mathfrak{M} = (T, <, \mathcal{I})$ and state that $\mathfrak{M}, \pi \models \varphi$ is defined as in the case of FOMLO, including additionally some new conditions associated with new constructs in the language.

6.4.1 Syntax and semantics

To keep the design of $\mathcal{TQL}$ possibly modular, and yet maximally generic, we first introduce a mechanism of abbreviating the temporal components of the queries into customary *temporal connectives*. Those connectives, defined analogical to Chomicki and Toman [CT98], are used as templates to be instantiated with particular CQs and further combined by means of Boolean operators in the target temporal query language.

Definition 27 (Temporal connectives)**.** *An n-ary* temporal connective *is a FOMLO-formula containing $k \geq 0$ free variables $x_1, \dots, x_k \in T_V$, called the* temporal answer variables, *and $n \geq 0$ predicate variables $X_1, \dots, X_n \in P_V$. We define Ω to be a finite set of temporal connectives, where each connective $\omega \in \Omega$ is given via a definition consisting of a name $\omega(\vec{x})(\vec{X})$, with $\vec{x} = x_1 \dots, x_k$ and $\vec{X} = X_1, \dots, X_n$, and a (definitional) ϕ-formula ω^*.*

Intuitively, the predicate variables are place holders for CQs, which we add in the next step. The temporal answer variables range over time instants, which are explicitly represented in the answers to temporal queries.[1] A small sample of possible temporal connectives is given in Table 6.1.

Further, we define the syntax and semantics of the temporal query language. Observe that the operators $\neg$ and $\wedge$ included here are technically speaking redundant, given they are also present in the underlying FOMLO. This two-level design of the temporal query language, advocated also by Chomicki and Toman [CT98], is driven mostly by pragmatic motives, as it is assumed that the user should only have access to the query-level language in which the

[1] In practice, the range of answer variables might need to be further restricted in order to finitize the number of possible answers. In the context of temporal databases, it is common to consider only time instants that are explicitly mentioned in the data (in our case: temporal ABox). This, however, might require certain normalization of the used time-stamps — a problem which we do not address here.

$$
\begin{array}{rcl}
\textbf{always}(X_1) & \triangleq & \forall x_1.X_1(x_1) \\
\textbf{sometime}(X_1) & \triangleq & \exists x_1.X_1(x_1) \\
\textbf{in}(x_1)(X_1) & \triangleq & X_1(x_1) \\
\textbf{after}(x_1, x_2) & \triangleq & x_2 < x_1 \\
\textbf{during-interval}(x_1, x_2)(X_1) & \triangleq & x_1 \leq x_2 \wedge \forall x_3.(x_1 \leq x_3 \leq x_2 \rightarrow X_1(x_3)) \\
\textbf{in-since}(x_1)(X_1, X_2) & \triangleq & \exists x_2.(x_2 < x_1 \wedge X_2(x_2) \wedge \forall x_3.(x_2 < x_3 \leq \\
& & x_1 \rightarrow X_1(x_3))) \\
\textbf{in-until}(x_1)(X_1, X_2) & \triangleq & \exists x_2.(x_1 < x_2 \wedge X_2(x_2) \wedge \forall x_3.(x_1 \leq x_3 < \\
& & x_2 \rightarrow X_1(x_3)))
\end{array}
$$

Table 6.1: Examples of temporal connectives.

FOMLO expressions are not directly available, but replaced with a number of built-in temporal connectives.

Definition 28 (Temporal query language: syntax)**.** *The* temporal query language $\mathcal{TQL}$ *is induced by the following grammar:*

$$\psi \ ::= \ \omega(\vec{x})(q_1(\vec{y}_1), \ldots, q_n(\vec{y}_n)) \ \mid \ \neg\psi \ \mid \ \psi \wedge \psi$$

where $\omega \in \Omega$ is an n-ary temporal connective with temporal answer variables $\vec{x}$, and every $q_i(\vec{y}_i)$ is a CQ with answer variables $\vec{y}_i$, for $1 \leq i \leq n$. We write $\psi(\vec{x}, \vec{y})$, to denote a $\mathcal{TQL}$ query ψ with temporal answer variables $\vec{x}$ and CQ answer variables $\vec{y}$.

An answer to a $\mathcal{TQL}$ query is a pair of sequences of time instants from T and individual names from N_I, which substituted for the respective temporal and CQ answer variables must satisfy the query. The answer variables of both types can be shared among different CQs and temporal connectives occurring in the query, thus facilitating descriptions of complex dependencies between temporal data (*cf*. Example 7). Like in Chapter 5, we write $\vec{a}|_{\vec{y}_i}$ to denote the subsequence of $\vec{a}$ corresponding to $\vec{y}_i$.

Definition 29 (Temporal query language: semantics)**.** *Let $\psi(\vec{x}, \vec{y})$ be a $\mathcal{TQL}$ query, with $\vec{x} = x_1, \ldots, x_k$ and $\vec{y} = y_1, \ldots, y_l$. For a pair of sequences $(\vec{t}, \vec{a})$, where $\vec{t} = t_1, \ldots, t_k \in T$ and $\vec{a} = a_1, \ldots, a_l \in N_I$, a $(\vec{t}, \vec{a})$-match to ψ in a model $\mathfrak{M} = (T, <, \mathcal{I})$ is a T-substitution π, such that $\pi(x_i) = t_i$, for every $1 \leq i \leq k$, and $\mathfrak{M}, \pi \models_{\vec{a}} \psi$, where the satisfaction relation $\models_{\vec{a}}$ is defined inductively as follows:*

- $\mathfrak{M}, \pi \models_{\vec{a}} \omega(\vec{x_i})(q_1(\vec{y}_1), \ldots, q_n(\vec{y}_n))$ iff $\mathfrak{M}, \pi \models \omega^*$ *(see Definition 26), where* *for every $1 \leq i \leq n$ and any T-substitution π' we set:*

$$\mathfrak{M}, \pi' \models X_i(\pi'(u)) \text{ iff } \mathcal{I}(\pi'(u)) \models q_i[\vec{a}|_{\vec{y}_i}], \tag{†}$$

- $\mathfrak{M}, \pi \models_{\vec{a}} \neg\varphi$ iff $\mathfrak{M}, \pi \not\models_{\vec{a}} \varphi$,
- $\mathfrak{M}, \pi \models_{\vec{a}} \varphi \wedge \psi$ iff $\mathfrak{M}, \pi \models_{\vec{a}} \varphi$ *and* $\mathfrak{M}, \pi \models_{\vec{a}} \psi$.

We write $\mathfrak{M} \models \psi[\vec{t}, \vec{a}]$ whenever there exists a $(\vec{t}, \vec{a})$-match to ψ in $\mathfrak{M}$. We write $\mathcal{T}, \mathcal{A}_T \models \psi[\vec{t}, \vec{a}]$, whenever there exists a $(\vec{t}, \vec{a})$-match to ψ in every model of $\mathcal{T}$ and $\mathcal{A}_T$. In the latter case $\vec{t}, \vec{a}$ is called a certain answer *to ψ w.r.t. $\mathcal{T}, \mathcal{A}_T$.*

Example 7. *We formulate a $\mathcal{TQL}$ query $\psi(x_1, x_2, y)$ requesting all patients y who have been ever diagnosed with some allergy, at some point x_1 were administered a new drug, and at some point x_2, after x_1, had symptoms of an allergic reaction. The precise meaning of the temporal connectives used in the query is as defined in Table 6.1.*

$$\begin{aligned} \psi(x_1, x_2, y) \ ::= \ & \mathbf{sometime}(\exists x.(\mathit{Patient}(y) \wedge \mathit{diagnosedWith}(y, x) \wedge \mathit{Allergy}(x))) \\ & \wedge \mathbf{in}(x_1)(\exists x.(\mathit{administered}(y, x) \wedge \mathit{NewDrug}(x))) \wedge \\ & \wedge \mathbf{after}(x_2, x_1) \\ & \wedge \mathbf{in}(x_2)(\exists x.(\mathit{hasSymptom}(y, x) \wedge \mathit{AllergicReaction}(x))) \end{aligned}$$

Consider the TBox $\mathcal{T}$ containing axioms:

$$\begin{aligned} \mathit{AllergicPatient} &\sqsubseteq \mathit{Patient} \sqcap \exists \mathit{diagnosedWith}.\mathit{Allergy} \\ \mathit{TestPatient} &\sqsubseteq \mathit{Patient} \sqcap \exists \mathit{administered}.\mathit{NewDrug} \end{aligned}$$

and the temporal ABox $\mathcal{A}$ containing time-stamped assertions:

$[1, 5]$: *AllergicPatient*$(john)$	$[2, 3]$: *Patient*(*carl*)
$[1, 2]$: *hasSymptom*$(john, id1)$	$[1, 2]$: *hasSymptom*$(carl, id3)$
$[2, 2]$: *AllergicReaction*(*id1*)	$[2, 2]$: *AllergicReaction*(*id3*)
$[4, 5]$: *TestPatient*(*john*)	$[2, 3]$: *diagnosedWith*(*carl*, *id4*)
$[6, 6]$: *hasSymptom*$(john, id2)$	$[2, 3]$: *Allergy*(*id4*)
$[6, 6]$: *AllergicReaction*(*id2*)	$[5, 5]$: *TestPatient*(*carl*)

Given the time domain of natural numbers $(\mathbb{N}, <)$ there are two certain answers to the query $\psi(x_1, x_2, y)$, namely: $(4, 6, john)$ and $(5, 6, john)$.

6.4.2 Practical query answering

As it turns out, under the introduced semantics the expressive power of $\mathcal{TQL}$ is still too high to provide reasonable guarantees for the worst-case complexity of temporal query answering, and for the possibility of reusing the existing query answering techniques and tools. The level of interaction between the Boolean operators of the temporal language with CQs is sufficient to enable encoding Boolean combinations of conjunctive queries (BCCQs) over DLs, i.e. formulas induced by the grammar:

$$\varphi ::= q \mid \neg\varphi \mid \varphi \wedge \psi.$$

where q is a CQ. The decidability of BCCQs answering over DLs is, to the best of our knowledge, an open problem. Some of the largest classes of queries whose decidability has been studied so far are in fact unions (disjunctions) of CQs [GHLS08] and their syntactic generalization — positive existential queries [OCE08]. In order to render query answering in $\mathcal{TQL}$ practical, we therefore need to employ some means of constraining the language. Quite a trivial fix is to tame the expressiveness of CQs, for instance by considering only CQs without existentially bounded variables — thus a variation of instance queries. Note that by substituting given sequences $\vec{t}, \vec{a}$ for the respective answer variables, one obtains a temporal formula with ABox axioms at the place of atoms. This suggests that , the query answering under such a restriction can be reduced to reasoning with temporalized ABox axioms w.r.t. global TBox. As explained in [LWZ08], for a temporal logic coinciding with PLTL and an arbitrary DL with at least PSPACE-hard satisfiability problem, the complexity of this task remains the same as for the satisfiability in the underlying DL.

A much more interesting way of alleviating the problem of handling BCCQs, however, is to restrict the level of interaction between the operators of the temporal language with those of the embedded CQs, without reducing the expressiveness of the queries. To this end we propose to apply a limited form of the Closed World Assumption (CWA). Although essentially incompatible with the open-world semantics of DLs, a controlled use of CWA is claimed to be justified and beneficial in various application scenarios related to OBDA and Semantic Web reasoning [GM05, CGL+07, LSW12]. In our case, we are interested in restricting $\mathcal{TQL}$ in a way that would enable answering individual CQs under the standard semantics, but at the same time, interpreting negation in front of CQs as Negation-As-Failure, and reducing the problem of answering BCCQs to Boolean operations over the certain answers to CQs. A clean and straightforward method of achieving this effect, advocated and studied in depth in [CGL+07], is to bind every occurrence of a CQ in a $\mathcal{TQL}$ query with the *autoepistemic* **K**-*operator*. Essentially, the operator **K** enforces that a bounded CQ is satisfied in a model, for a given answer, only if this answer is *known* to be certain, or formally:

$$\mathcal{I} \models \mathbf{K}q[\vec{a}] \textit{ iff } \mathcal{T}, \mathcal{A} \models q[\vec{a}]$$

where $\mathcal{I}$ is a model of $\mathcal{T}$ and $\mathcal{A}$. This immediately entails the requested reductions of a limited, closed-world flavor:

$$\mathcal{T}, \mathcal{A} \models \mathbf{K}q[\vec{a}] \textit{ iff } \mathcal{T}, \mathcal{A} \models q[\vec{a}]$$
$$\mathcal{T}, \mathcal{A} \models \neg\mathbf{K}q[\vec{a}] \textit{ iff } \mathcal{T}, \mathcal{A} \not\models q[\vec{a}]$$
$$\mathcal{T}, \mathcal{A} \models \mathbf{K}q_1[\vec{a}] \vee \mathbf{K}q_2[\vec{b}] \textit{ iff } \mathcal{T}, \mathcal{A} \models q_1[\vec{a}] \text{ or } \mathcal{T}, \mathcal{A} \models q_2[\vec{b}]$$

Observe that the set of certain answers to a single CQ is invariant to the possible application of the **K**-operator in front of the query. Thus, the closed-world reasoning, emerging only on the level of Boolean combinations of CQs, does not affect the basic assumption of possible incompleteness of data, inherent to the OBDA paradigm.

Eventually, by replacing every q in $\mathcal{TQL}$ queries with $\mathbf{K}q$, or simply by interpreting it *as if it was bounded* by **K** (as we do below), we obtain the desired, well-behaved temporal query language.

Definition 30 ($\mathcal{TQL}$ semantics with epistemic interpretation of CQs)**.** *The semantics of $\mathcal{TQL}$ with* epistemic interpretation *of embedded CQs is exactly the same as in Definition 29, modulo the replacement of the condition* (†) *with the following one:*

$$\mathfrak{M}, \pi' \models X_i(\pi'(u)) \text{ iff } \mathcal{T}, \mathcal{A}_T(\pi'(u)) \models q_i[\vec{a}|_{\vec{y}_i}]. \tag{‡}$$

To witness the difference between evaluating $\mathcal{TQL}$ queries under the two compared semantics, consider an example involving TBox $\mathcal{T} = \{A \sqsubseteq \neg D,\ B \sqcap C \sqsubseteq D\}$, temporal ABox $\mathcal{A} = \{[1,1] : A(a),\ [1,2] : B(a),\ [2,3] : C(a)\}$ and query $\psi(x,y) ::= \neg\mathbf{in}(x)(D(y))$. Under the original semantics, presented in Definition 29, the query returns a unique certain answer $(1, a)$. On the other hand, by enforcing the epistemic interpretation of the embedded CQ $q(y) ::= D(y)$, as argued above, and setting the time domain of natural numbers, we obtain an infinite set of certain answers $\{(t, a) \mid 2 \neq t \in \mathbb{N}\}$.

Notably, the condition $\mathcal{T}, \mathcal{A}_T(\pi'(u)) \models q_i[\vec{a}|_{\vec{y}_i}]$ in (‡) is an instance of the standard CQ answering problem. Moreover, it is the only point in the revised semantics where DL reasoning is intertwined with reasoning over the temporal language. What follows, is that the most natural algorithm answering $\mathcal{TQL}$

queries can be constructed by augmenting any standard decision procedure for the satisfiability in the temporal language with an oracle answering CQs over the designated snapshots of the ABox *w.r.t.* the TBox. As the decision problem in FOMLO is known to be PSPACE-complete [Rey10], we thus obtain an upper bound on the combined complexity of answering $\mathcal{TQL}$ queries.

Theorem 14 (Combined complexity of $\mathcal{TQL}$ query answering)**.** *Let ψ be a $\mathcal{TQL}$ query over a temporal ABox $\mathcal{A}_T$ w.r.t. TBox $\mathcal{T}$, where ABox and TBox axioms are expressed in a DL language $\mathcal{L}$. The combined complexity of deciding $\mathcal{T}, \mathcal{A}_T \models \psi[\vec{t}, \vec{a}]$, for a pair of sequences $\vec{t}, \vec{a}$, under the epistemic interpretation of the CQs embedded in ψ, is in* $\text{PSPACE}^{QA(\mathcal{L})}$*, where $QA(\mathcal{L})$ is an oracle answering CQs in $\mathcal{L}$.*

This seemingly unsurprising result has some significant theoretical and practical implications. On the theoretical side, it guarantees that answering $\mathcal{TQL}$ queries under the epistemic interpretation of CQs remains decidable, as long as answering CQs over the respective DLs is decidable. Moreover, it establishes a bridge for an immediate transfer of the combined complexity results. For instance, when $\mathcal{L} = \mathcal{ALC}$, answering $\mathcal{TQL}$ queries is in $\text{PSPACE}^{\text{EXPTIME}}$, thus effectively in EXPTIME, as the combined complexity of CQ answering in $\mathcal{ALC}$ is EXPTIME-complete [Lut08]. Analogically, for $\mathcal{L} = \mathcal{SHIQ}$, the problem is in $\text{PSPACE}^{\text{2EXPTIME}}$, and effectively in 2EXPTIME. In general, the combined complexity of answering $\mathcal{TQL}$ queries for an arbitrary DL $\mathcal{L}$ is equal to the complexity of answering CQs in $\mathcal{L}$, provided that the latter problem is at least PSPACE-hard. This observation naturally generalizes over query languages based on combinations of FOMLO with arbitrary classes of first-order queries. Whenever the complexity of answering $\mathcal{Q}$-queries over $\mathcal{L}$, for a given query class $\mathcal{Q}$ and a DL $\mathcal{L}$, is at least PSPACE-hard then answering the resulting temporal queries over $\mathcal{L}$ remains in the same complexity class. Otherwise it is PSPACE-complete. This demonstrates that the temporalization technique employed here yields computationally cheap, yet expressive, temporal query languages over temporal ABoxes. In fact, temporalization of query languages for expressive DLs, subsuming $\mathcal{ALC}$, comes for free.

From the practical perspective, the restricted interaction between the temporal component and CQs, reflected in Theorem 14, promises relatively straightforward implementations of $\mathcal{TQL}$ query engines based on existing technologies, e.g.: temporal theorem provers and CQ answering tools. Roughly, to determine whether a candidate answer to a query ψ is certain for $\mathcal{T}, \mathcal{A}_T$, it suffices to check whether the direct rendering of ψ into FOMLO is satisfiable, where every CQ embedded in ψ is seen as a predicate variable, whose truth-value in a given time instant is determined by a call to an external CQ answering tool over the respective snapshot of $\mathcal{A}_T$ *w.r.t.* $\mathcal{T}$.[2]

Some further interesting prospects concern answering $\mathcal{TQL}$ queries over the DL-Lite family of DLs, enjoying the FO-rewritability property [CDGL+07]. It is known that CQ answering in DL-Lites can be carried out efficiently using highly optimized RDBMSs. In a nutshell, for TBox $\mathcal{T}$, ABox $\mathcal{A}$ and a CQ q, one can always find a first-order query $q^{\mathcal{T}}$, such that for every $\vec{a}$ it is the case that $\mathcal{T}, \mathcal{A} \models q[\vec{a}]$ *iff* $\mathcal{A} \models q^{\mathcal{T}}[\vec{a}]$, where the latter problem can be solved directly in an RDBMS. Clearly, an analogical approach should enable rewriting a $\mathcal{TQL}$ query over $\mathcal{T}, \mathcal{A}_T$ into a temporal first-order formula, which could be then efficiently encoded and evaluated as a TSQL2 query [BCST96] over a temporal database $\mathcal{A}_T$. Although providing precise definition of such a translation and proving its correctness is left as future work, we expect it to be straightforward given that every $\mathcal{TQL}$ query corresponds to a temporal formula with embedded CQs, where each CQ q can be replaced with the corresponding first-order formula $q^{\mathcal{T}}$ obtained by means of established FO-rewritability techniques.

6.5 Temporal metalanguage

The practice of representing and reasoning with temporal information on the Semantic Web, for instance in the field of health care support [SMOD08], suggests that the presented data model and query language might not be sufficiently flexible for real-life applications. The reason for this shortcoming is the general inability of using semantically rich descriptions of the time domain, based on semantically rich temporal vocabularies. In particular, we are incapable of directly expressing typical temporal patterns occurring in temporal queries and constraints used in clinical applications, such as:

1. Visit 17 must occur *at least 1 week but no later than 4 weeks after the end of 2003 ragweed season*.
2. Administer Rapamune *1 week from* Visit 0 *daily for 84 days*.
3. The vital signs of the participant should be obtained at *routine time points starting at 10 minutes post* infusion, then *at 20-minute intervals until* the participant is discharged.

[2] For time domains based on natural numbers and integers, FOMLO formulas can be translated into PLTL [Rey10], and thus decided using off-shelf PLTL provers, such as listed in `http://www.cs.man.ac.uk/~schmidt/tools/`. CQ answering in selected DLs is supported by such systems as QuOnto, REQUIEM, Presto.

4. Administer study medication at *weekly intervals for 3 months*.

5. The first and second blood draws are *10 days apart*, and the third draw is *11-14 days after* the second.

Also, there is no way of supporting temporal annotations whose meaning could be described in terms of abstract temporal concepts. For instance, in certain scenarios one might need to qualify data with annotations denoting:

1. a time interval from May, the 15th, 2005, until some day in June 2005:

2. a time point during 2008 ragweed season :

3. a periodic event recurring at least 3 times, each consecutive day, starting some time in January 2006:

A remarkable feature of such rich semantic descriptions of time is that they facilitate reasoning with incomplete data, where the incompleteness occurs also in the temporal dimension. Such characteristic is conceptually vital considering the open-world philosophy of Semantic Web. In most of the existing applications, such descriptions are represented using OWL-based time ontologies [HP04, Grü10][3] and manipulated by ad-hoc, hybrid reasoning architectures. Those architectures retrieve possibly complex temporal information encoded in time ontologies and process it with external, application-specific tools, thus sacrificing some theoretical rigor and formal transparency of the provided inferences [OD11]. Arguably, OWL semantics is not adequate for supporting genuine temporal reasoning, hence the interpretation of temporal descriptions contained in time ontologies is always to some extent arbitrary.

A natural solution to this problem on the grounds of our framework is to extend the underlying temporal language with additional constructs enabling ontological-style axiomatization of the background knowledge about the time domain, interpreted directly over temporal semantics. For instance, we want to be able to represent concepts involved in the Gregorian calendar, and further employ them on the query level and on the level of data annotations, so that the querying process abides logically to formal calendric constraints. Technically, we achieve this by augmenting FOMLO with integer periodicity constraints and additional monadic predicates — constructs known to be sufficiently expressive to represent a number of interesting temporal concepts, and yet not increasing the complexity of reasoning [Dem06]. Naturally, there are alternative temporal formalisms, e.g. calendar logics [OG98] or metric logics [HR06],

[3]See also http://motools.sourceforge.net/timeline/timeline.html.

which could possibly succeed in this task. For that reason, we do not see our proposal as definitive, but nevertheless, given its simplicity and intuitiveness, we consider it a strong proof of concept.

In this section we restrict our attention to more well-behaved time domains based on countably infinite sets of linearly ordered time points. Effectively, without loss of generality we can assume that for $(T, <)$, $T = \mathbb{N}$ and $<$ is the usual *smaller-than* relation. Let *Var* be a countably infinite set of variables. The following definition of periodicity constraints is based on [TCR94] and [Dem06].

Definition 31 (Periodicity constraints)**.** *A (simple)* periodicity constraint *over* $\mathbb{N}$ *is an expression of one of the following forms:*

$$x \equiv_k c \mid x \equiv_k y + c \mid \bigcirc x \equiv_k y + c$$

where $x, y \in$ *Var and* $k, c \in \mathbb{Z}$*. By PC we denote the set of all periodicity constraints over* Var*. A* valuation v *is a mapping* $Var \times \mathbb{N} \mapsto \mathbb{Z}$*. The satisfaction relation* $\models_{PC}$ *for periodicity constraints w.r.t. a valuation* σ *and* $n \in \mathbb{N}$ *is defined as follows:*

- $\sigma, n \models_{PC} x \equiv_k c$ iff *there exists* $m \in \mathbb{Z}$ *such that* $\sigma(n, x) = m \times k + c$,
- $\sigma, n \models_{PC} x \equiv_k y + c$ iff *there exists* $m \in \mathbb{Z}$ *such that* $\sigma(n, x) - \sigma(n, y) = m \times k + c$,
- $\sigma, n \models_{PC} \bigcirc x \equiv_k y+c$ iff *there exists* $m \in \mathbb{Z}$ *such that* $\sigma(n+1, x) - \sigma(n, y) = m \times k + c$.

Intuitively, the variables correspond to different measurable time units obtaining different values in particular time points. For instance, for a variable *month* $\in$ *Var*, the constraint *month* $\equiv_{12} 1$ states that the value of variable *month* modulo 12 is 1, while $\bigcirc$*month* $\equiv_{12}$ *month* $+ 1$ means that the value of *month* in the next time point is equal to *month* + 1, modulo 12. Next we define an extension of FOMLO with periodicity constraints and monadic predicates, where N_P is a countably infinite set of the predicate names.

Definition 32 (FOMLO^{PC})**.** *Let* $(T, <)$ *be a time domain. A FOMLO*^{PC}*-formula is an expression constructed according to the grammar of FOMLO (see Definition 24) extended with two additional types of atoms:*

$$P(u) \mid \delta(u)$$

where $u \in T \cup T_V$, $P \in N_P$ *and* $\delta \in PC$. *An interpretation is a pair* $\mathfrak{M} = (T, <, \cdot^{\mathcal{J}}, \sigma)$, *where* $\cdot^{\mathcal{J}}$ *is an interpretation function, such that* $P^{\mathcal{J}} \subseteq T$, *for every* $P \in N_P$, *and* σ *is a valuation for periodicity constraints. For an interpretation* $\mathfrak{M}$ *and a* T*-substitution* π, *the satisfaction relation for FOMLO*^{PC}*-formulas is defined exactly as for FOMLO-formulas (see Definition 26), including two additional conditions:*

- $\mathfrak{M}, \pi \models P(u)$ iff $\pi(u) \in P^{\mathcal{J}}$,
- $\mathfrak{M}, \pi \models \delta(u)$ iff $\sigma, \pi(u) \models_{PC} \delta$.

We intend to model (axiomatize) the time domain by means of $\mathcal{TL}^{PC}$ formulas, in the same fashion as DL TBoxes are used for modeling the object domain.

Definition 33 (CBox)**.** *A CBox* $\mathcal{C}$ *is a set of closed FOMLO*^{PC}*-formulas without any predicate variables. We say that an interpretation* $\mathfrak{M} = (T, <, \cdot^{\mathcal{J}}, \sigma)$ *is a model of* $\mathcal{C}$, *written* $\mathfrak{M} \models \mathcal{C}$, *if for every* $\varphi \in \mathcal{C}$ *there exists a* T*-substitution* π *such that* $\mathfrak{M}, \pi \models \varphi$.

Since we work with discrete domains, we make use of a handy *next-time* operator $+1$ applicable to terms inside predicate formulas of type $X(v)$. We write $X(v + 1)$ as an abbreviation for $X(y) \wedge \exists y.(v < y \wedge \neg\exists z.(v < z < y))$. The expressiveness of FOMLO^{PC}-formulas is high enough to enable modeling formal calendars and a number of other interesting temporal concepts. For a detailed discussion we refer to [Dem06]. Here, as a small example, we model a fragment of the Gregorian calendar, covering days and months of common years.[4] To ease the presentation, we assume the following abbreviations:

$$
\begin{aligned}
X &:= \bigcirc month \equiv_{12} month \\
Y &:= \bigcirc month \equiv_{12} month + 1 \\
Z &:= \bigcirc day \equiv_{31} day \\
V &:= \bigcirc day \equiv_{31} day + 1
\end{aligned}
$$

First, we set the beginning and the end of each month:

$$
\begin{aligned}
&\forall x.(((month \equiv_{12} 1)(x) \wedge Y(x)) \rightarrow ((day \equiv_{31} 31)(x) \wedge (day \equiv_{31} 1)(x + 1))) \\
&\forall x.(((month \equiv_{12} 2)(x) \wedge Y(x)) \rightarrow ((day \equiv_{31} 28)(x) \wedge (day \equiv_{31} 1)(x + 1))) \\
&\forall x.(((month \equiv_{12} 3)(x) \wedge Y(x)) \rightarrow ((day \equiv_{31} 31)(x) \wedge (day \equiv_{31} 1)(x + 1))) \\
&\forall x.(((month \equiv_{12} 4)(x) \wedge Y(x)) \rightarrow ((day \equiv_{31} 30)(x) \wedge (day \equiv_{31} 1)(x + 1))) \\
&\qquad \cdots
\end{aligned}
\tag{6.1}
$$

[4] In general, representation of a calendar based on integer periodicity constraints requires also additionally marking its beginning and end, which we omit here.

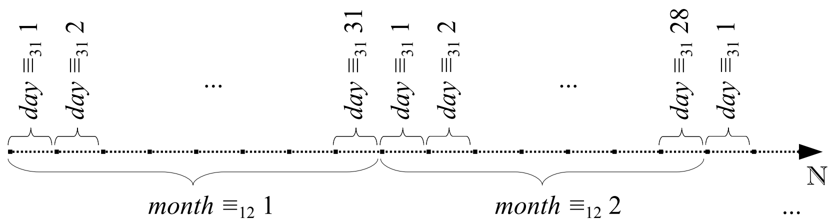

Figure 6.1: A temporal model satisfying day-month calendar constraints.

Next, we enforce the correct consecution of months:

$$\forall x.(Y(x) \lor \exists y.(x < y \land Y(y) \land \forall z.(x \leq z < y \rightarrow X(z)))) \quad (6.2)$$

and the correct consecution of days:

$$\forall x.(V(x) \lor Y(x) \lor \exists y.(x < y \land (V(y) \lor Y(y)) \land \forall z.(x \leq z < y \rightarrow Z(z))) \quad (6.3)$$

With little effort, one can verify that the models satisfying the CBox containing formulas 6.1–6.3 are exactly those that encode the specified fragment of the Gregorian calendar, such as depicted in Figure 6.1. Note, that the minimal granularity of the calendar is not fixed, leaving significant space for incomplete temporal descriptions. Further, it is straightforward to define a number of casual temporal predicates, allowing for more abstract qualifications of time, e.g.:

$$\forall x.(January(x) \leftrightarrow (month \equiv_{12} 1)(x))$$
$$\forall x.(BeginningOfAcademicYear(x) \leftrightarrow ((month \equiv_{12} 9)(x) \land (day \equiv_{12} 1)(x)))$$

Given the background temporal terminology in the form of a CBox, the next step is to enable semantic annotations of data. To this end, we revise the definition of temporal ABoxes (see Definition 22) by replacing concrete timestamps of the form $[t_1, t_2]$ with special predicate names describing all time points in which a piece of data holds.

Definition 34 (Annotated ABox). *Let $N_A \subseteq N_P$ be a designated set of predicate names called* annotations*. A* temporally annotated assertion *is an expression of the form:*

$$\tau : \alpha$$

where α is an ABox assertion and $\tau \in N_A$. A temporally annotated ABox $\mathcal{A}_T$ *is a finite set of temporally annotated assertions. For an interpretation $\mathfrak{M} = (T, <, \mathcal{I}, \cdot^{\mathcal{J}}, \sigma)$ and $t \in T$, a t-snapshot of $\mathcal{A}_T$ is the ABox $\mathcal{A}_T(t) = \{\alpha \mid \tau : \alpha \in \mathcal{A}_T \text{ and } t \in \tau^{\mathcal{J}}\}$.*

The notion of snapshot semantics (see Definition 23) is revised accordingly.

Definition 35 (Snapshot semantics)**.** *Let $\mathcal{T}$ a TBox, $\mathcal{C}$ a CBox and $\mathcal{A}_T$ a temporally annotated ABox. An interpretation $\mathfrak{M} = (T, <, \mathcal{I}, \cdot^{\mathcal{J}}, \sigma)$ is called a temporal model of $\mathcal{T}, \mathcal{C}, \mathcal{A}_T$* iff *$(T, <, \mathcal{I})$ is a model of $\mathcal{T}, \mathcal{A}_T$ (see Definition 23) and $(T, <, \cdot^{\mathcal{J}}, \sigma)$ is a model of $\mathcal{C}$.*

Clearly, the meaning of concrete timestamps can be easily captured by using annotations, defined in the CBox as follows:

Temporal assertion:	$\Leftrightarrow$	Annotated assertion $\tau : \alpha$ + CBox axiom:
$[t_1, t_2] : \alpha$	$\Leftrightarrow$	$\forall x.(\tau(x) \leftrightarrow t_1 \leq x \leq t_2)$,
$[-\infty, t_1] : \alpha$	$\Leftrightarrow$	$\forall x.(\tau(x) \leftrightarrow x \leq t_1)$,
$[t_1, +\infty] : \alpha$	$\Leftrightarrow$	$\forall x.(\tau(x) \leftrightarrow t_1 \leq x)$,
$[-\infty, +\infty] : \alpha$	$\Leftrightarrow$	$\forall x.(\tau(x))$.

Furthermore, one can model a number of complex, possibly incomplete descriptions, useful for annotating data in many practical scenarios, such as mentioned in the beginning of this section. For instance:

1. τ is a time interval from May, the 15th, until some day in June:

$$\exists x, y.((month \equiv_{12} 5)(x) \wedge (day \equiv_{31} 15)(x) \wedge (month \equiv_{12} 6)(y) \wedge \\ \forall z.(\tau(z) \leftrightarrow x \leq z \leq y)),$$

2. τ is a time point during some ragweed season:

$$\exists x.(\tau(x) \wedge RagweedSeason(x)),$$

3. τ is a periodic event recurring at least 3 times, each consecutive day, starting some time in January:

$$\exists x_1, x_2, x_3.(dayBefore(x_1, x_2) \wedge dayBefore(x_2, x_3) \wedge \\ \tau(x_1) \wedge \tau(x_2) \wedge \tau(x_3)),$$

   where the following abbreviations are involved:

$$dayBefore(x,y) := \exists z.(x \leq z \wedge X(z) \wedge \forall u.(x < u < z \to Y(u)) \wedge$$
$$\exists v.(z < v \wedge X(v) \wedge \forall u.(z < u < v \to Y(u)) \wedge z < y \leq v)),$$
$$X := \bigcirc day \equiv_{31} day + 1,$$
$$Y := \bigcirc day \equiv_{31} day.$$

Finally, we extend our temporal query language by grounding it in FOMLO^{PC} and its semantics, in the place of the original FOMLO, underpinning $\mathcal{TQL}$.

Definition 36 ($\mathcal{TQL}^M$: syntax)**.** *The* temporal query language with meta-level descriptions $\mathcal{TQL}^M$ *is defined exactly as $\mathcal{TQL}$ (see Definitions 27 and 28), except for that temporal connectives are based on FOMLO^{PC} (instead of FOMLO).*

Definition 37 ($\mathcal{TQL}^M$: semantics)**.** *Let $\psi(\vec{x},\vec{y})$ be a $\mathcal{TQL}^M$ query, with $\vec{x} = x_1,\ldots,x_k$ and $\vec{y} = y_1,\ldots,y_l$. For a pair of sequences $(\vec{t},\vec{a})$, where $\vec{t} = t_1,\ldots,t_k \in T$ and $\vec{a} = a_1,\ldots,a_l \in N_I$, a $(\vec{t},\vec{a})$-match to ψ in a model $\mathfrak{M} = (T,<,\mathcal{I},\cdot^{\mathcal{J}},\sigma)$ is a T-substitution π, such that $\pi(x_i) = t_i$, for every $1 \leq i \leq k$, and $\mathfrak{M},\pi \models_{\vec{a}} \psi$, where the satisfaction relation $\models_{\vec{a}}$ is defined exactly as in the case of $\mathcal{TQL}$ (see Definition 29 resp. 30).*

We write $\mathfrak{M} \models \psi[\vec{t},\vec{a}]$ whenever there exists a $(\vec{t},\vec{a})$-match to ψ in $\mathfrak{M}$. We write $\mathcal{T},\mathcal{C},\mathcal{A}_T \models \psi[\vec{t},\vec{a}]$, whenever there exists a $(\vec{t},\vec{a})$-match to ψ in every model of $\mathcal{T}$, $\mathcal{C}$ and $\mathcal{A}_T$. In the latter case $\vec{t},\vec{a}$ is called a certain answer *to ψ w.r.t. $\mathcal{T},\mathcal{C},\mathcal{A}_T$.*

As an example, we formalize a simple temporal constraint, of a pattern commonly occurring in clinical applications.

Example 8. *Suppose clinical data in some hospital record must satisfy the constraint "Administer Rapamune 1 day from Visit 0". We model the constraint as a $\mathcal{TQL}^M$ query $\psi(x,y)$:*

$$\psi(x,y) ::= \mathbf{in}(x)(Patient(y) \wedge hasAppointment(y, visit\text{-}0)) \wedge$$
$$\neg\mathbf{onNextDay}(x)(\exists z.(administered(y,z) \wedge Rapamune(z)))$$

where the connective **in** *is defined in Table 6.1, and* **onNextDay** *below, using the abbreviation $dayBefore$, defined earlier in this section:*

$$\mathbf{onNextDay}(x)(X_1) \quad \triangleq \quad \exists y.(dayBefore(x,y) \wedge X_1(y))$$

Under the epistemic semantics, discussed in the previous section, every certain answer to query $\psi(x,y)$ marks a violation to the declared constraint, i.e. a situation such that there exists a time point x in which patient y had a visit 0, but Rapamune was not administered to the patient on the following day.

Importantly, the complexity of $\mathcal{TQL}^M$ query answering over extended representations, given the epistemic interpretation of the embedded CQs, does not increase as compared to $\mathcal{TQL}$. The argument for this claim is again straightforward, and analogical to the one proving Theorem 14. It is known that the decision problem for PLTL with integer periodicity constraints is PSPACE-complete [Dem06]. Moreover, whenever countably infinite time domains are considered, any FOMLO^{PC} formula can be reduced to PLTL with integer periodicity constraints [Rey10]. Consequently, we can again use an oracle for answering CQs over the underlying DL $\mathcal{L}$ to augment an arbitrary PSPACE-complete decision procedure for PLTL with integer periodicity constraints.

Theorem 15 (Combined complexity of $\mathcal{TQL}^M$ query answering)**.** *Let ψ be a $\mathcal{TQL}^M$ query over a temporally annotated ABox $\mathcal{A}_T$ w.r.t. TBox $\mathcal{T}$ and CBox $\mathcal{C}$, where ABox and TBox axioms are expressed in a DL language $\mathcal{L}$. The combined complexity of deciding $\mathcal{T}, \mathcal{C}, \mathcal{A}_T \models \psi[\vec{t}, \vec{a}]$, for a pair of sequences $\vec{t}, \vec{a}$, under the epistemic interpretation of the CQs embedded in ψ, and w.r.t. to countably infinite time domains, is in PSPACE$^{QA(\mathcal{L})}$, where $QA(\mathcal{L})$ is an oracle answering CQs in $\mathcal{L}$.*

6.6 Related work and discussion

The design of the language $\mathcal{TQL}$ follows closely the principles of query languages for temporal databases, as outlined in e.g. [CT98]. In the general TDB setup, the query component $\mathcal{Q}$ is based on the full first-order logic, while temporal operators defined in FOMLO can be nested within each other. Hence, the resulting language is expressively equivalent to the temporal first-order logic. In $\mathcal{TQL}$, we are deliberately constraining the query component and disallow nesting of operators in order to enable practical decoupling of the DL-level from the temporal-level reasoning. In the context of the SW, similar approaches have been proposed to deal with time-stamped RDF data [GHV07, Mot12] and OWL axioms [Mot12]. Both contributions, however, lack the generality of our proposal. The temporal component of the query languages is in both cases highly restricted in order to ensure finite answer sets. In particular, Motik [Mot12] introduces a specially fixed number of most practical temporal operators that can be combined with the data-level queries. All these can be easily restated in FOMLO, and so the language of [Mot12] can be easily defined using the $\mathcal{TQL}$ mechanism.

As explained in Section 6.5, most of the existing solutions for supporting rich temporal vocabularies are purely technology-driven, thus lacking proper logical foundations. An alternative framework, motivated by similar observations to ours and based on the use of formal annotation languages, have been proposed by Zimmerman et al. [ZLPS12]. There, however, the annotation language is a non-standard, task-specific formalism, which cannot be directly translated into temporal logics or OWL. Yet another approach to representing qualitative temporal information in OWL has been addressed in [BSP11]. However, this proposal, based on so-called 4D-fluents, is incommensurable with the standard temporal database philosophy, and thus does not naturally facilitate integration with existing OBDA technologies, which is one of our key motivations here.

An interesting open challenge is a potential integration of the framework with temporal DLs [LWZ08], mentioned in Section 6.2. We believe that our choice of standard temporal semantics and logic-based query formalism, renders such a prospect quite realistic. The framework studied in this chapter is focused on querying temporal data with respect to a fixed, time-invariant terminology (TBox). A natural extension to this approach is to introduce means of querying temporal ABoxes in presence of temporal constraints occurring on the intensional, terminological level. Temporal DLs are a family of two-dimensional DLs, developed intensively in the recent years, intended specifically for the representation of this kind of terminologies. By allowing operators of temporal logics to occur in DL concepts, TDLs enable, for instance, to express the following axiom:

$$\mathit{Patient} \sqcap \exists \mathit{diagnosedWith}.\mathit{Allergy} \sqsubseteq \mathit{AllergicPatient}\ \mathcal{U}\ \forall \mathit{diagnosedWith}.\neg \mathit{Allergy}$$

The axiom states that whenever a patient is diagnosed with an allergy, she should be considered an allergic patient until ($\mathcal{U}$) she is diagnosed with no allergies. Interestingly, TDL TBoxes are interpreted over the same type of semantic structures as used in our framework, i.e. tuples $\mathfrak{M} = (T, <, \mathcal{I})$. This means, that from the formal perspective integration of $\mathcal{TQL}$ with TDLs can be achieved seamlessly. Obviously, query answering in such setting should likely be computationally more expensive, considering that already the satisfiability problem in TDLs is usually harder than in the underlying DLs. So far the only query language for TDLs has been proposed by Artale et al. [AFWZ02]. Differently than here, the queries are defined as unions of CQs, where the atomic predicates can be possibly preceded by temporal operators. As a consequence, the reuse of existing CQ answering techniques is not directly possible within this approach.

6.7 Conclusion

We believe that the framework proposed in this chapter marks a first promising step towards establishing a generic approach to representing and querying temporal data under DL ontologies. Naturally, a number of important problems, which we merely touched upon, are left open to future research. Among others, it is critical to conduct a systematic study of possible ways of restricting the $\mathcal{TQL}$-like languages, in order to turn the query answering problem feasible in practice. The use of epistemic semantics, suggested here, is only one of possible options. Other might involve more fine-grained constraints on the expressiveness of the temporal component, the first-order component or both. We also advocate a study of possible extensions of the framework towards integration with temporal DLs.

Admittedly, more complex temporal patterns result in quite verbose and involved FOMLO/FOMLO^{PC}-formulas. Hence, from the pragmatic perspective, it would be highly desirable to identify the most useful, and ideally expressively complete, set of temporal connectives, and compile it out directly in the syntax of $\mathcal{TQL}/\mathcal{TQL}^M$. Further, for a better compatibility with the current Semantic Web practices, it is necessary to reconsider the relationship between OWL time ontologies and our direct, temporal logic-based representation of temporal information. Arguably, a translation mechanism from the former to the latter could largely facilitate the reusability of existing OWL-based representations of temporal information.

CHAPTER
SEVEN

SUMMARY

7.1 Conclusions

In this thesis, we have studied various Description Logic-based knowledge systems, logic languages, and application scenarios dealing with formal representation of context and different types of contextual reasoning. On the way, we have delivered a number of specific results and insights regarding the addressed cases, which we believe cast new and valuable light on the general problem of contextuality of knowledge within the paradigm of DLs. It is our deep hope, however, that the presented material offers even more than the sum of its parts — namely, that what emerges from all those individual contributions is a lucid image of the context framework, which conceptually unifies all our investigations, and which brings about a fresh, generic and highly explicatory perspective on the problem of reasoning with contexts in DLs. Let us summarize this perspective by restating the answers to our main research questions formulated in the opening of this thesis.

What theory of contexts is adequate for integration with the DL-based knowledge representation paradigm?

We have committed ourselves to McCarthy's theory of contexts, the main reason being, that this theory offers a very instrumental view on contexts. Basically, following McCarthy, we consider contexts simply as formal objects, describable in first-order languages. Such level of abstraction allows us to keep our approach philosophically neutral and application-agnostic, which in turn makes it easy to adapt it to and operationalize in very diverse use-cases and formal systems.

How should such theory be technically reinterpreted and implemented on the grounds of DL semantics, syntax, and the general philosophy and methodology of DL-based knowledge representation?

Our key proposal has been to interpret contexts as possible worlds of a second semantic dimension, added to the standard DL interpretations. Further, we have advocated the use of a second language for expressing knowledge about contexts and for facilitating context-aware management of the object-level representation. Effectively, this has lead us to study a number of two-dimensional, two-sorted knowledge representation systems and languages, from technically involved DLs of Context, designed in the style of product-like combinations of DLs with modal logics (Chapter 3), to different sorts of lightweight query languages for querying contextualized knowledge (Chapters 4-6). It is worth emphasizing that the general knowledge modeling methodology, characteristic to DLs, has been practically unaltered. In all scenarios considered, the object knowledge (domain data) has been always represented in the standard DL fashion. Moreover, in three out of four cases (Chapters 3-5), the context-level representation is also based on the standard DL languages. Only when contexts are identified with time points (Chapter 6), we have used linear temporal logics for expressing context-level knowledge. The only novel aspect of the representation methodology is the use of special operators and annotations for intertwining the object- and context-level constraints, which is a necessary addition if one wants to explicitly model any form of context-dependency at all.

What are the formal properties of the resulting framework?

In general, we have learnt that there are different ways of constructing formalisms capable of supporting representation of contextualized knowledge and contextual reasoning, in a way adhering to the principles of our adopted theory of contexts. Hence, it is not possible to speak about the concrete properties of the proposed context framework, but merely about the properties of the particular formalisms, which turn out to vary from one scenario to the other. The most important and general conclusion drawn from our investigations is that the expressiveness of a two-level, object-context representation formalism, and computational complexity of reasoning in it depends on two major factors: the individual characteristics of the involved object and context languages, which are directly inherited by the formalism, and the level of the syntactic and semantic interaction between the two languages. The latter observation is particularly worthwhile, as it clearly indicates a convenient strategy of controlling the properties of the employed formalisms by suitably restricting or relaxing this level of interaction. We have seen, that when this interaction is very liberal, like in the case of DLs of Context (Chapter 3), we obtain languages even more expressive than traditional two-dimensional DLs, such us $(\mathbf{K}_n)_{\mathcal{L}}$ or $\mathbf{S5}_{\mathcal{L}}$, of the complexity up to one exponential higher than of the underlying DLs. On the other hand, more restrictive settings allow for straightforward reductions of certain reasoning problems into two separate problem modules, corresponding to the two levels of representation (Chapters 5-6), or even more, for direct reductions to the underlying DLs (Chapter 4). Another valuable observation which we have made is that adding an explicit context language to a two-dimensional formalism does not have to introduce an additional cost in the complexity of reasoning, and yet, it might substantially benefit the modeling capabilities of the formalism. In many cases this cost is already hidden in the shift from one-dimensional to two-dimensional semantics. Concluding, reasoning with contexts in DLs does not have to be in general very expensive, as many interesting aspects of contextuality can be captured with the use of very moderate means of expressiveness.

How and to what extent can such a framework be applied in and adapted to different application scenarios, motivated by use-cases and problems observed in the practice of the Semantic Web?

We have successfully applied our framework in several practical scenarios, effectively proposing novel solutions to such problems as: representation of inherently contextualized knowledge, ontology integration, knowledge selection, provenance querying/verification, temporal querying. Since the framework provides a constant conceptual basis for all these cases, we have found it relatively straightforward to transfer observations, results and the style of defining syntaxes and semantics across different settings. We are therefore convinced that the proposed context framework proves its merits as a powerful methodological tool. Essentially, what needs to be decided in each new scenario is what constitutes a context there, and in what way the object knowledge depends on the context dimension. Given this starting point, the formalization of the problem follows quite naturally. Obviously, our context framework will not suit problems in which contexts are understood very differently than in McCarthy's theory. For instance, if a context for a data item is interpreted in a linguistic manner, as the "data surrounding" of this item (i.e. other data known about the same object), then clearly our approach might be of little help. However, whenever contexts are any sort of implicit states about which some information is known, one is in a good position to benefit from the strong regulative capabilities offered by the framework.

7.2 Outlook

In between the lines of our study, we have tried to promote a certain vision of the Semantic Web or, to put it differently, of knowledge representation methodology in a distributed, heterogenous environment such as the Web. Let us now concisely formulate this vision.

As stated in the introduction, the main motivation behind the broad use and standardization of the Semantic Web representation languages is to facilitate machine-understandability of information published on the Web. To turn it into a more practical proviso, a dataset published on the Web by an arbitrary agent should ideally lend itself correctly interpretable by any other agent accessing it independently, so that the original, intended meaning of the data is not lost in the consumption process. That much for the programmatic slogan of the Semantic Web project. The most fundamental premise of our work, which we do hope is undebatable, is that more often than not, information cannot be in fact properly understood without assuming the context in which it is stated. If the context of a published dataset remains opaque to others, the data simply cannot be correctly interpreted, at least not in an automated manner, without investing an effort in discovering and eliciting the context manually. Consequently, if Web data is not accompanied by the corresponding contextual metadata declared explicitly in some standardized formal language, the goal of machine-understandability of information on the Web can never be achieved. We therefore strongly believe that "entering the second dimension" in the philosophy and practice of knowledge representation on the Semantic Web, i.e. explicitly representing the contexts for the published Semantic Web knowledge, is necessary and highly desirable. This, fortunately, is more and more frequently realized by the community. It is therefore our recommendation that this process should start to be treated more systematically and with a long-term perspective in mind, ideally leading to some forms of standardization or at least good practice guidelines, accepted by the community.

We are realistic in that the more elaborate languages and representation systems proposed in this thesis might have little chances of becoming future standards, cordially employed in practical applications. Admittedly, the cost of introducing a new standard for the Semantic Web is very high in terms of resources, and even more than that, achieving community consensus. However, recording the context of a dataset by means of OWL/RDF(S) assertions about the dataset's identifier does not seem such a radical proposal, and in fact, it is already performed locally by some data providers for their own use. What should be settled then are merely some technical details regarding the precise ways of encoding such information and some generic upper vocabularies to be commonly used for describing such metadata. Once the annotated datasets become common and represented in a consistent way, in the next step the community can start developing practical formalisms and tools for managing data in a context-sensitive manner. In this respect, we consider our approach to designing query languages, presented in Chapters 5-6, very promising. It is modular and easily generalizable towards different forms of annotations, offers a reasonably low implementation overhead, and most importantly, it supports the construction of lightweight, yet flexible formalisms, which can be readily applied in a number of real-life scenarios. Based on these observations, we believe that the development of formal foundations and tools for practical query languages over annotated data is the most interesting and vital direction for future work. Following this track, it would be particularly appealing to investigate the cases with data expressed in DL-*Lite* languages, which allow for efficient query answering using Relational Database Management System technologies.

On the theoretical side, this thesis leaves a number of other questions open. For instance: What is the complexity of the satisfiability problem in DLs of Context for other DL languages used on the object/context level, not addressed here? In particular, we have not generalized the results towards the prominent DL $\mathcal{SROIQ}$, underlying OWL 2 DL language. Further, if the object language is sufficiently restricted, is it possible to identify a DL of Context in which contextualization of roles would not lead to undecidability? Is answering temporal queries over temporal DL TBoxes decidable? If so, what is the complexity of this task and what would be the practical algorithms for answering such queries? And many others. In general, our thesis provides a number of interesting insights into how the theoretical work on two-dimensional logics can be fruitfully transferred into more practical environment and help clarify and solve some real-life problems. As the history of DLs shows, tightening the links between the theory and practice of knowledge representation can always benefit both sides. We are convinced that this kind of exchange of expertise between the theoretical and practical communities working on diverse aspects of representing and reasoning with two-dimensional knowledge can be still pushed forward a long way, and thus it should remain in the spotlight of active research efforts on both sides.

APPENDIX

A

PROOFS

Below we present full proofs of the complexity results sketched in Section 3.5.

A.1 2EXPTIME upper bound

First we demonstrate decidability and the implied 2EXPTIME upper complexity bound for the knowledge base satisfiability problem in $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$.

Theorem (Upper bound). *Satisfiability of a knowledge base in $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$, for $\mathcal{L}_O = \mathcal{L}_C = \mathcal{SHIO}$, any combination of context operators $\mathfrak{F}_1/\mathfrak{F}_2$ and for local interpretation of object roles, is decidable in* 2EXPTIME.

Proof. Let $\mathcal{K} = (\mathcal{C}, \mathcal{O})$ be a knowledge base in $\mathfrak{C}_{\mathcal{L}_O}^{\mathcal{L}_C}$, for $\mathcal{L}_O = \mathcal{L}_C = \mathcal{SHIO}$ with context operators $\mathfrak{F}_1$ and $\mathfrak{F}_2$. We devise a quasimodel elimination algorithm which decides satisfiability of $\mathcal{K}$ in at most double exponential time in the size of $\mathcal{K}$.

By $\cdot^-$ we denote the inverse constructor for roles and assume that $(r^-)^- = r$ (resp. $(\boldsymbol{r}^-)^- = \boldsymbol{r}$). Let f be a set of $\mathcal{SHIO}$ axioms. Then by $\sqsubseteq_f^*$ we denote the reflexive-transitive closure of $\sqsubseteq$ on $\{r \sqsubseteq s, s^- \sqsubseteq r^- \mid r \sqsubseteq s \in f\}$ (resp. $\{\boldsymbol{r} \sqsubseteq \boldsymbol{s}, \boldsymbol{s}^- \sqsubseteq \boldsymbol{r}^- \mid \boldsymbol{r} \sqsubseteq \boldsymbol{s} \in f\}$). W.l.o.g. we assume that none of the constructors $[\boldsymbol{r}.\cdot]$, $[\cdot]$, $\forall r.\cdot$, $\forall \boldsymbol{r}.\cdot$, $\sqcup$ occur in $\mathcal{K}$. Further, we apply the following replacements of all respective (sub)formulas with their equivalents:

$$\begin{array}{lll} C(a) & \Rightarrow & \{a\} \sqsubseteq C, \\ r(a,b) & \Rightarrow & \{a\} \sqsubseteq \exists r.\{b\}, \\ \boldsymbol{C}(\boldsymbol{a}) & \Rightarrow & \{\boldsymbol{a}\} \sqsubseteq \boldsymbol{C}, \\ \boldsymbol{r}(\boldsymbol{a},\boldsymbol{b}) & \Rightarrow & \{\boldsymbol{a}\} \sqsubseteq \exists \boldsymbol{r}.\{\boldsymbol{b}\}. \end{array}$$

To shorten the proof, we consider satisfiability with respect to simplified semantics based only on models $\mathfrak{M} = (\Theta, \mathfrak{C}, \cdot^{\mathcal{J}}, \Delta, \{\cdot^{\mathcal{I}(i)}\}_{i\in\mathfrak{C}})$ with $\Theta = \mathfrak{C}$, abbreviated to $\mathfrak{M}^\star = (\Theta, \cdot^{\mathcal{J}}, \Delta, \{\cdot^{\mathcal{I}(i)}\}_{i\in\Theta})$. It can be easily shown that $\mathcal{K}$ is satisfiable *iff* $\mathcal{K}^\star$ is satisfiable w.r.t. the simplified semantics, where $\mathcal{K}^\star$ is obtained from $\mathcal{K}$ by introducing a fresh context concept name ***Context*** in the following positions:

- $\langle \boldsymbol{r}.(\boldsymbol{Context} \sqcap \boldsymbol{C})\rangle$ and $\langle \boldsymbol{Context} \sqcap \boldsymbol{C}\rangle$ in every operator $\langle \boldsymbol{r}.\boldsymbol{C}\rangle$ and $\langle \boldsymbol{C}\rangle$ occurring in $\mathcal{K}$, respectively,
- $(\boldsymbol{Context} \sqcap \boldsymbol{C}) : \varphi$ in every axiom $\boldsymbol{C} : \varphi \in \mathcal{O}$,
- $(\boldsymbol{Context} \sqcap \{\boldsymbol{c}\}) : \varphi$ in every axiom $\boldsymbol{c} : \varphi \in \mathcal{O}$, followed by an extra axiom $\{\boldsymbol{c}\} \sqsubseteq \boldsymbol{Context}$ added to $\mathcal{C}$.

Then $\mathcal{K}^\star$ is satisfied in $\mathfrak{M}^\star = (\Theta, \cdot^{\mathcal{J}}, \Delta, \{\cdot^{\mathcal{I}(i)}\}_{i\in\Theta})$ *iff* $\mathcal{K}$ is satisfied in $\mathfrak{M} = (\Theta, \mathfrak{C}, \cdot^{\mathcal{J}}, \Delta, \{\cdot^{\mathcal{I}(i)}\}_{i\in\mathfrak{C}})$, where $\mathfrak{C} = \boldsymbol{Context}^{\mathcal{J}}$.

Finally, the following notation is used to mark the sets of symbols of particular type occurring in $\mathcal{K}$:

$con_c(\mathcal{K})$:	all context concepts, closed under negation,
$con_o(\mathcal{K})$:	all object concepts, closed under negation,
$rol_c(\mathcal{K})$:	all context roles,
$rol_c^+(\mathcal{K}) \subseteq rol_c(\mathcal{K})$:	all transitive context roles,
$rol_o(\mathcal{K})$:	all object roles,
$rol_o^+(\mathcal{K}) \subseteq rol_o(\mathcal{K})$:	all transitive object roles,
$ind_c(\mathcal{K})$:	all context individual names,
$ind_o(\mathcal{K})$:	all object individual names,
$sub_o(\mathcal{K})$:	all axioms from $\{\varphi \mid \boldsymbol{C} : \varphi \in \mathcal{O}$ for any $\boldsymbol{C}\}$.

A *context type* for $\mathcal{K}$ is a subset $c \subseteq con_c(\mathcal{K})$, where:

- $\boldsymbol{C} \in c$ iff $\neg\boldsymbol{C} \notin c$, for all $\boldsymbol{C} \in con_c(\mathcal{K})$,
- $\boldsymbol{C} \sqcap \boldsymbol{D} \in c$ iff $\{\boldsymbol{C}, \boldsymbol{D}\} \subseteq c$, for all $\boldsymbol{C} \sqcap \boldsymbol{D} \in con_c(\mathcal{K})$.

An *object type* for $\mathcal{K}$ is a subset $t \subseteq con_o(\mathcal{K})$, where:

- $C \in t$ iff $\neg C \notin t$, for all $C \in con_o(\mathcal{K})$,
- $C \sqcap D \in t$ iff $\{C, D\} \subseteq t$, for all $C \sqcap D \in con_o(\mathcal{K})$.

Definition 38 (matching object role-successor)**.** *Let t, t' be two object types for $\mathcal{K}$ and $f \subseteq \mathrm{sub}_o(\mathcal{K})$. For any $r \in rol_o(\mathcal{K})$, t' is a* matching r-successor *for t under f* iff *the following conditions are satisfied:*

- $\{\neg C \mid \neg\exists r.C \in t\} \subseteq t'$ *and* $\{\neg C \mid \neg\exists r^-.C \in t'\} \subseteq t$,
- *if* $r \in rol_o^+(\mathcal{K})$ *then* $\{\neg\exists r.C \in t\} \subseteq t'$ *and* $\{\neg\exists r^-.C \in t'\} \subseteq t$,
- *t' is a matching s-successor for t under f, for every $s \in rol_o(\mathcal{K})$ such that $r \sqsubseteq_f^* s$,*
- *t is a matching s-successor for t' under f, for every $s \in rol_o(\mathcal{K})$ such that $r^- \sqsubseteq_f^* s$.*

A *quasistate* for $\mathcal{K}$ is a tuple $q = \langle c_q, f_q, O_q \rangle$, where c_q is a context type for $\mathcal{K}$, $f_q \subseteq sub_o(\mathcal{K})$ and O_q is a non-empty set of object types for $\mathcal{K}$. We say that q is *saturated iff* for every $t \in O_q$:

(qS) if $\exists r.D \in t$ then t has a matching r-successor $t' \in O_q$ under f_q.

We call q *coherent iff* the following conditions hold:

(qC1) for every $a \in ind_o(\mathcal{K})$ there exists a unique $t \in O_q$ such that $\{a\} \in t$,

(qC2) for every $\boldsymbol{C} : \varphi \in \mathcal{O}$, if $\boldsymbol{C} \in c_q$ then $\varphi \in f_q$,

(qC3) for every $C \sqsubseteq D \in f_q$ and $t \in O_q$, if $C \in t$ then $D \in t$,

(qC4) for every $t \in O_q$ and $\neg\langle \boldsymbol{C} \rangle D \in t$, if $\boldsymbol{C} \in c_q$ then $\neg D \in t$.

A *linkage* between two quasistates $q = \langle c_q, f_q, O_q \rangle$ and $q' = \langle c'_q, f'_q, O'_q \rangle$ for $\mathcal{K}$ is a mapping $\lambda = g \cup h$, where $g : O_q \mapsto O'_q$ and $h : O'_q \mapsto O_q$, such that for every $a \in ind_o(\mathcal{K})$ and $t \in O_q \cup O'_q$, $\{a\} \in t$ *iff* $\{a\} \in \lambda(t)$.

Definition 39 (matching $\mathfrak{F}_1$-successor)**.** *Let $q = \langle c_q, f_q, O_q \rangle$ and $q' = \langle c'_q, f'_q, O'_q \rangle$ be two quasistates for $\mathcal{K}$. Then q' is a* matching $\mathfrak{F}_1$-successor *for q via a linkage λ* iff *for every $t \in O_q \cup O'_q$, $\{\langle \boldsymbol{C} \rangle D, \neg\langle \boldsymbol{C} \rangle D \in t\} = \{\langle \boldsymbol{C} \rangle D, \neg\langle \boldsymbol{C} \rangle D \in \lambda(t)\}$.*

Definition 40 (matching $\mathfrak{F}_2$-successor)**.** *Let $q = \langle c_q, f_q, O_q \rangle$ and $q' = \langle c'_q, f'_q, O'_q \rangle$ be two quasistates for $\mathcal{K}$. For any $\boldsymbol{r} \in rol_c(\mathcal{K})$, q' is a* matching $\boldsymbol{r}$-successor *for q via a linkage λ* iff *q' is a matching $\mathfrak{F}_1$-successor for q via λ and the following conditions are satisfied:*

- $\{\neg \boldsymbol{C} \mid \neg\exists \boldsymbol{r}.\boldsymbol{C} \in c_q\} \subseteq c'_q$ *and* $\{\neg \boldsymbol{C} \mid \neg\exists \boldsymbol{r}^-.\boldsymbol{C} \in c'_q\} \subseteq c_q$,
- *if* $\boldsymbol{r} \in rol_c^+(\mathcal{K})$ *then* $\{\neg\exists \boldsymbol{r}.\boldsymbol{C} \in c_q\} \subseteq c'_q$ *and* $\{\neg\exists \boldsymbol{r}^-.\boldsymbol{C} \in c'_q\} \subseteq c_q$,
- *for every* $t \in O_q$ *and* $t' \in O'_q$, $\{\neg D \mid \neg\langle \boldsymbol{r}.\boldsymbol{C}\rangle D \in t, \boldsymbol{C} \in c'_q\} \subseteq \lambda(t)$, $\{\neg D \mid \neg\langle \boldsymbol{r}.\boldsymbol{C}\rangle D \in \lambda(t'), \boldsymbol{C} \in c'_q\} \subseteq t'$, $\{\neg D \mid \neg\langle \boldsymbol{r}^-.\boldsymbol{C}\rangle D \in \lambda(t), \boldsymbol{C} \in c_q\} \subseteq t$ *and* $\{\neg D \mid \neg\langle \boldsymbol{r}^-.\boldsymbol{C}\rangle D \in t', \boldsymbol{C} \in c_q\} \subseteq \lambda(t')$,
- *for every* $t \in O_q$ *and* $t' \in O'_q$, *if* $\boldsymbol{r} \in rol_c^+(\mathcal{K})$ *then* $\{\neg\langle \boldsymbol{r}.\boldsymbol{C}\rangle D \in t\} \subseteq \lambda(t)$, $\{\neg\langle \boldsymbol{r}.\boldsymbol{C}\rangle D \in \lambda(t')\} \subseteq t'$, $\{\neg\langle \boldsymbol{r}^-.\boldsymbol{C}\rangle D \in \lambda(t)\} \subseteq t$ *and* $\{\neg\langle \boldsymbol{r}^-.\boldsymbol{C}\rangle D \in t'\} \subseteq \lambda(t)$,
- q' *is a matching* $\boldsymbol{s}$*-successor for* q *via* λ *for every* $\boldsymbol{s} \in rol_c(\mathcal{K})$ *such that* $\boldsymbol{r} \sqsubseteq^*_{\mathcal{C}} \boldsymbol{s}$,
- q *is a matching* $\boldsymbol{s}$*-successor for* q' *via* λ *for every* $\boldsymbol{s} \in rol_c(\mathcal{K})$ *such that* $\boldsymbol{r}^- \sqsubseteq^*_{\mathcal{C}} \boldsymbol{s}$.

A set of quasistates Q is *saturated iff* for every quasistate $q \in Q$, with $q = \langle c_q, f_q, O_q\rangle$:

(QS1) for every $\exists \boldsymbol{r}.\boldsymbol{C} \in c_q$ there is a matching $\boldsymbol{r}$-successor for q in Q via some linkage λ,

(QS2) for every $t \in O_q$ and $\langle \boldsymbol{C}\rangle D \in t$ there is a matching $\mathfrak{F}_1$-successor $q' = \langle c'_q, f'_q, O'_q\rangle$ for q in Q via some linkage λ, such that $\boldsymbol{C} \in c'_q$ and $D \in \lambda(t)$,

(QS3) for every $t \in O_q$ and $\langle \boldsymbol{r}.\boldsymbol{C}\rangle D \in t$ there is a matching $\boldsymbol{r}$-successor $q' = \langle c'_q, f'_q, O'_q\rangle$ for q in Q via some linkage λ, such that $\boldsymbol{C} \in c'_q$ and $D \in \lambda(t)$.

A *quasimodel* $\mathfrak{N}$ for $\mathcal{K}$ is a non-empty, saturated set of saturated and coherent quasistates for $\mathcal{K}$ satisfying the following conditions:

(M1) for every $\boldsymbol{c} \in ind_c(\mathcal{K})$ there is a unique $q \in \mathfrak{N}$, with $q = \langle c_q, f_q, O_q\rangle$, such that $\{\boldsymbol{c}\} \in c_q$,

(M2) for every $\boldsymbol{C} \sqsubseteq \boldsymbol{D} \in \mathcal{C}$ and $q \in \mathfrak{N}$, with $q = \langle c_q, f_q, O_q\rangle$, if $\boldsymbol{C} \in c_q$ then $\boldsymbol{D} \in c_q$.

We can now prove the quasimodel lemma.

Lemma 1. *There is a quasimodel for* $\mathcal{K}$ iff *there is an* $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$*-model of* $\mathcal{K}$.

Proof. The key observation which we exploit in this proof is that the constraints (QS1)-(QS3) imposed on quasimodels ensure existence of certain specific quasistates, which represent successors in the context dimension, and existence of special linkage relations allowing for a proper choice of types for the same object in different contexts. To ease reference to these elements we amend the corresponding conditions with the following naming conventions:

(QS1*) in such case call q' a *witness* for $(\exists \boldsymbol{r}.\boldsymbol{C}, q)$ and a linkage λ, enforced by the condition, a *witnessing linkage*,

(QS2*) in such case call q' a *witness* for $(\langle \boldsymbol{C} \rangle D, t, q)$ and a linkage λ, enforced by the condition, a *witnessing linkage*,

(QS3*) in such case call q' a *witness* for $(\langle \boldsymbol{r}.\boldsymbol{C} \rangle D, t, q)$ and a linkage λ, enforced by the condition, a *witnessing linkage*.

$(\Rightarrow)$ Suppose $\mathfrak{N}$ is a quasimodel for $\mathcal{K} = (\mathcal{C}, \mathcal{O})$. We sketch the construction of an $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-model $\mathfrak{M} = (\Theta, \cdot^{\mathcal{J}}, \Delta, \{\cdot^{\mathcal{I}(i)}\}_{i \in \Theta})$ of $\mathcal{K}$. We start by constructing an interpretation of the context dimension $(\Theta, \cdot^{\mathcal{J}})$. First, for every $\boldsymbol{c} \in ind_c(\mathcal{K})$ and $q \in \mathfrak{N}$ such that $\{\boldsymbol{c}\} \in q$, add q to Θ and set $\boldsymbol{c}^{\mathcal{J}} = q$. In case $ind_c(\mathcal{K}) = \emptyset$ set $\Theta = \{q\}$ for any $q \in \mathfrak{N}$. Then iteratively extend $(\Theta, \cdot^{\mathcal{J}})$ as follows. For every $q \in \Theta$, with $q = \langle c_q, f_q, O_q \rangle$:

- for every $\exists \boldsymbol{r}.\boldsymbol{C} \in c_q$ pick a witness q' for $(\exists \boldsymbol{r}.\boldsymbol{C}, q)$ from $\mathfrak{N}$, add it to Θ and set $(q, q') \in \boldsymbol{r}^{\mathcal{J}}$,
- for every $t \in O_q$ and $\langle \boldsymbol{C} \rangle D \in t$ pick a witness q' for $(\langle \boldsymbol{C} \rangle D, t, q)$ from $\mathfrak{N}$ and add it to Θ,
- for every $t \in O_q$ and $\langle \boldsymbol{r}.\boldsymbol{C} \rangle D \in t$ pick a witness q' for $(\langle \boldsymbol{r}.\boldsymbol{C} \rangle D, t, q)$ from $\mathfrak{N}$, add it to Θ and set $(q, q') \in \boldsymbol{r}^{\mathcal{J}}$.

Further, we extend the interpretation of roles by iteratively saturating the following steps. For every $q, q', q'' \in \Theta$ and $\boldsymbol{r}, \boldsymbol{s} \in rol_c(\mathcal{K})$:

- if $(q, q') \in \boldsymbol{r}^{\mathcal{J}}$ then set $(q', q) \in (\boldsymbol{r}^{-})^{\mathcal{J}}$,
- if $(q, q') \in \boldsymbol{r}^{\mathcal{J}}$ and $\boldsymbol{r} \sqsubseteq^*_{\mathcal{C}} \boldsymbol{s}$ then set $(q, q') \in \boldsymbol{s}^{\mathcal{J}}$,
- if $\boldsymbol{r} \in rol^+_c(\mathcal{K})$ and $(q, q'), (q', q'') \in \boldsymbol{r}^{\mathcal{J}}$ then set $(q, q'') \in \boldsymbol{r}^{\mathcal{J}}$.

Finally, for every $\boldsymbol{A} \in con_c(\mathcal{K})$ set $\boldsymbol{A}^{\mathcal{J}} = \{q \in \Theta \mid \boldsymbol{A} \in c_q\}$.

By structural induction it follows that all complex context concepts are satisfied by $\mathfrak{M}$ in the expected contexts. In particular, all role restrictions must be satisfied due to an adequate interpretation of context roles, ensuring that:

- role names and their inverses are interpreted as relations which are inverses of each other,
- transitive roles are interpreted as transitive relations,
- the role hierarchies entailed by $\mathcal{C}$ are respected.

Above properties are guaranteed by Definition 40 and the construction of the model. Consequently, since $\mathfrak{N}$ satisfies conditions (M1), (M2), all axioms from the context knowledge base $\mathcal{C}$ must be satisfied. Next we turn to the object dimension.

A *run* ρ through Θ is a choice function which for every $q \in \Theta$ selects an object type $\rho(q) \in O_q$. Runs are used for representing the behavior of object individuals across contexts. The easiest way to properly constrain this behavior is by employing the witnessing linkages introduced in conditions (QS1)-(QS3). Note that the way the interpretation $(\Theta, \cdot^{\mathcal{J}})$ is constructed ensures that for every two contexts there exists a witnessing linkage we can refer to in order to align the interpretations of object individuals inhabiting these contexts. A set of runs $\mathfrak{R}$ is *coherent iff* the following conditions are satisfied. For every $q, q' \in \Theta$, with $q = \langle c_q, f_q, O_q \rangle$ and $q' = \langle c'_q, f'_q, O'_q \rangle$ and λ being the witnessing linkage between q and q':

- for every $a \in ind_o(\mathcal{K})$, there is exactly one run $\rho_{a,q} \in \mathfrak{R}$ such that $\{a\} \in \rho_{a,q}(q)$,
- for every $\rho \in \mathfrak{R}$, $\lambda(\rho(q)) = \rho(q')$,
- for every $t \in O_q$ and $t' \in O'_q$, if $\lambda(t) = t'$ then there exists $\rho \in \mathfrak{R}$, such that $\rho(q) = t$ and $\rho(q') = t'$.

We let $\Delta = \mathfrak{R}$, for a coherent set of runs $\mathfrak{R}$ through Θ, and for every $q \in \Theta$, with $q = \langle c_q, f_q, O_q \rangle$, we fix the corresponding interpretation function $\cdot^{\mathcal{I}(q)}$ as follows:

- for every individual name $a \in ind_o(\mathcal{K})$ set $a^{\mathcal{I}(q)} = \rho_{a,q}(q)$,
- for every concept name $A \in con_o(\mathcal{K})$ set $A^{\mathcal{I}(q)} = \{\rho \in \mathfrak{R} \mid A \in \rho(q)\}$,
- for every role $r \in rol_o(\mathcal{K})$, $\rho \in \mathfrak{R}$ and $\exists r.D \in \rho(q)$ pick $\rho' \in \mathfrak{R}$ such that $\rho'(q)$ is a matching r-successor for $\rho(q)$ under f_q and set $(\rho, \rho') \in r^{\mathcal{I}(q)}$.

Note that by aligning runs with the witnessing linkages we automatically ensure that each object obtains compatible interpretations in every two related contexts. In particular, whenever $d \in (\langle \boldsymbol{r}.\boldsymbol{C} \rangle D)^{\mathcal{I}(q)}$ for some $d \in \Delta$ and $q \in \Theta$, there has to exist a context $q' \in \boldsymbol{C}^{\mathcal{J}}$ accessible from q through $\boldsymbol{r}$ in which $d \in D^{\mathcal{I}(q')}$. By the same token, whenever $d \in (\langle \boldsymbol{C} \rangle D)^{\mathcal{I}(q)}$, there must be a context $q' \in \boldsymbol{C}^{\mathcal{J}}$ such that $d \in D^{\mathcal{I}(q')}$.

Further, as before, we extend the interpretation of roles by iteratively saturating the following steps. For every $q \in \Theta$, with $q = \langle c_q, f_q, O_q \rangle$, every $\rho, \rho', \rho'' \in \mathfrak{R}$ and $r, s \in rol_o(\mathcal{K})$:

- if $(\rho, \rho') \in r^{\mathcal{I}(q)}$ then set $(\rho', \rho) \in (r^-)^{\mathcal{I}(q)}$,
- if $(\rho, \rho') \in r^{\mathcal{I}(q)}$ and $r \sqsubseteq^*_{f_q} s$ then set $(\rho, \rho') \in s^{\mathcal{I}(q)}$,
- if $r \in rol^+_o(\mathcal{K})$ and $(\rho, \rho'), (\rho, \rho'') \in r^{\mathcal{I}(q)}$ then set $(\rho, \rho'') \in r^{\mathcal{I}(q)}$.

Similarly as in the context dimension, Definition 38 along with way the model is constructed ensure an adequate interpretation of all roles. Consequently, by structural induction it is not difficult to see that all object concepts are satisfied by $\mathfrak{M}$ as expected and thus, since $\mathfrak{N}$ satisfies conditions (qC1)-(qC4), all axioms from the object knowledge base $\mathcal{O}$ must be also satisfied.

($\Leftarrow$) This direction is straightforward. Let $\mathfrak{M} = (\Theta, \cdot^{\mathcal{J}}, \Delta, \{\cdot^{\mathcal{I}(i)}\}_{i \in \Theta})$ be a $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-model of $\mathcal{K}$. We construct a quasimodel $\mathfrak{N}$ for $\mathcal{K}$ as follows. Let $\mathbf{t}$ be a function mapping every context from $\mathfrak{f}$ to its type determined by the interpretation $\mathfrak{M}$, i.e., for every $c \in \Theta$, set $\mathbf{t}(c) = \langle t_c, f_c \rangle$ where t_c and f_c have to satisfy the constraints:

- $\boldsymbol{C} \in t_c$ iff $c \in \boldsymbol{C}^{\mathcal{J}}$, for every $\boldsymbol{C} \in con_c(\mathcal{K})$,
- $\varphi \in f_c$ iff $\mathcal{I}, c \models \varphi$, for every $\varphi \in sub_o(\mathcal{K})$.

In the same way we use $\mathbf{t}$ to denote object types for objects. For every object-context pair $\langle d, c \rangle \in \Delta \times \Theta$ we define $\mathbf{t}(d, c)$ as:

- $C \in \mathbf{t}(d, c)$ iff $d \in C^{\mathcal{I}(c)}$, for every $C \in con_o(\mathcal{K})$,

Further, for every $c \in \Theta$ let $O_c = \{\mathbf{t}(d, c) \mid d \in \Delta\}$ be the set of object types represented in the context c. We can then define a quasistate for every $c \in \mathfrak{C}$ as $q_c = \langle t_c, f_c, O_c \rangle$, where $\mathbf{t}(c) = \langle t_c, f_c \rangle$. Finally, let $\mathfrak{N} = \{q_c \mid c \in \mathfrak{C}\}$. Clearly $\mathfrak{N}$ is a quasimodel for $\mathcal{K}$. In particular, it is guaranteed that for all existential restrictions and context operators occurring in the context and object types from the quasistates, there must exist suitable witnesses and witnessing linkages, and thus that all conditions constituting quasimodels have to be satisfied. ❑

The basic, brute-force algorithm deciding whether a quasimodel for $\mathcal{K}$ exists is a straightforward extension of the Pratt-style type elimination method, similar to [KWZG03, Theorem 6.61]. We start by enumerating the set $\mathfrak{N}$ of all possible quasistates. Further, we enumerate all possible mappings $\gamma : ind_c(\mathcal{K}) \mapsto \mathfrak{N}$. The algorithm proceeds in two steps:

1. select a mapping γ, and for every $\boldsymbol{c} \in ind_c(\mathcal{K})$ eliminate all quasistates $q \in \mathfrak{N}$ such that $q \neq \gamma(\boldsymbol{c})$, with $q = \langle c_q, f_q, O_q \rangle$ and $\{\boldsymbol{c}\} \in c_q$,

2. iteratively eliminate all quasistates and object types from the quasistates which violate any of the conditions (qS), (qC1)-(qC4), (QS1)-(QS3), (M1)-(M2).

It succeeds *iff* the following conditions are met:

- no more object types nor quasistates can be eliminated,

- there is at least one quasistate left and every quasistate contains at least one object type.

In such case the result of elimination is clearly a quasimodel and the search is finished with the answer "$\mathcal{K}$ is satisfiable". Else, if all quasistates get eliminated, the algorithm selects another mapping γ and repeats the elimination procedure. If none of the mappings allow for a successful termination then clearly no quasimodel exists and the algorithm returns "$\mathcal{K}$ is unsatisfiable".

The whole algorithm runs in double exponential time in the size of $\mathcal{K}$. To show this, we observe that the following (very liberally estimated) inequalities hold. By $\ell(\mathcal{K})$ we denote the size of $\mathcal{K}$, measured in the number of symbols used, and by $|X|$ — the number of elements of set X:

$$|con_c(\mathcal{K}) \cup con_o(\mathcal{K})| \leq 2\ell(\mathcal{K}),$$
$$|ind_c(\mathcal{K})| \leq \ell(\mathcal{K}),\ \ |sub_o(\mathcal{K})| \leq \ell(\mathcal{K}),$$

size of a quasistate:
$$\ell(q) \leq \ell(\mathcal{K}) \cdot (|con_c(\mathcal{K})| + |sub_o(\mathcal{K})| + 2^{|con_o(\mathcal{K})|}) \leq \ell(\mathcal{K}) \cdot (2\ell(\mathcal{K}) + \ell(\mathcal{K}) + 2^{2\ell(\mathcal{K})}),$$

number of quasistates in a quasimodel:
$$|\mathfrak{N}| = 2^{|con_c(\mathcal{K})|} \cdot 2^{|sub_o(\mathcal{K})|} \cdot 2^{2^{|con_o(\mathcal{K})|}} = 2^{2\ell(\mathcal{K})} \cdot 2^{\ell(\mathcal{K})} \cdot 2^{2^{2\ell(\mathcal{K})}}.$$

Since deciding whether a quasistate can be eliminated at a given stage, in particular checking if there exist appropriate witnesses for it (QS1)-(QS3), cannot take more than $\ell(q)^2 \cdot |\mathfrak{N}|$ steps, therefore a single run of the elimination procedure takes no more than $(\ell(q) \cdot |\mathfrak{N}|)^2$ steps. Finally, there can be at most $|\mathfrak{N}|^{|ind_c(\mathcal{K})|}$ different mappings γ, hence the whole procedure must terminate in time belonging to $O(2^{2^{\ell(\mathcal{K})}})$. ❑

A.2 2EXPTIME lower bound

Next, we derive the lower bound for the concept satisfiability problem in the logic $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$, which carries over to several other logics discussed in Chapter 3, including $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$. We start by making an observation, which is especially useful in the proof, that $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$ is Kripke-complete w.r.t. the class of infinite intransitive trees with a constant branching factor, determined by the number of context modalities.

Proposition 5. *A concept C is satisfiable w.r.t. a global TBox $\mathcal{T}$ in $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$* iff *it is satisfied w.r.t. $\mathcal{T}$ in some model $\mathfrak{M} = (\mathfrak{W}, \{<_i\}_{1\leq i\leq n}, \Delta, \{\cdot^{\mathcal{I}(w)}\}_{w\in\mathfrak{W}})$, such that $\langle \mathfrak{W}, \bigcup\{<_i\}_{1\leq i\leq n}\rangle$ is a tree, every world in $\mathfrak{W}$ has exactly one $<_i$-successor, for each $i \in (1, n)$, and for $i \neq j$, $<_i$- and $<_j$-successors are different.*

Models based on such trees can be easily obtained from arbitrary $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$-models by using the standard unraveling technique. Thus, in what follows, we focus exclusively on $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$*-tree-models*.

Theorem. *Deciding concept satisfiability in $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$ w.r.t. global TBoxes and only with local roles is* 2EXPTIME*-hard.*

The proof is based on reduction of the word problem of an exponentially space-bounded *Alternating Turing Machine* (ATM), which is known to be 2EXPTIME-hard [CKS81].

Alternating Turing Machines.

An ATM is a tuple $\mathcal{M} = (Q, \Sigma, \Gamma, q_0, \delta)$, where:

- Q is a set of states containing pairwise disjoint sets of *existential states* $Q_\exists$, *universal states* $Q_\forall$, and *halting states* $\{q_a, q_r\}$, where q_a is an *accepting* and q_r a *rejecting* state;

- Σ is an *input alphabet* and Γ a *working alphabet*, containing the *blank symbol* $\varnothing$, such that $\Sigma \subseteq \Gamma$;
- $q_0 \in Q_\exists \cup Q_\forall$ is the *initial state;*
- δ is a *transition relation*, which to every pair $(q, a) \in (Q_\exists \cup Q_\forall) \times \Gamma$ assigns at least one triple $(q', b, m) \in Q \times \Gamma \times \{l, n, r\}$. The triple describes the transition to state q', involving overwriting of symbol a with b and a shift of the head to the left ($m = l$), to the right ($m = r$) or no shift ($m = n$). If q is a halting state then the set of possible transitions $\delta(q, a)$ for every $a \in \Gamma$ is empty.

A *configuration* of an ATM is given as a sequence $\omega q \omega'$, where $\omega, \omega' \in (\Gamma \setminus \{\varnothing\})^*$ and $q \in Q$, which says that the tape contains the word $\omega\omega'$ (possibly followed by blank symbols), the machine is in state q and the head of the machine is on the leftmost symbol of ω'. A succeeding configuration is defined by transitions δ, where the head of the machine reads and writes the symbols on the tape. A configuration $\omega q \omega'$ is a *halting* one if $q = q_a$ (*accepting configuration*) or if $q = q_r$ (*rejecting configuration*).

Without loss of generality we adopt a somewhat simplified and more convenient setup for ATMs presented in [ALT07]. An *ATM computation tree* of $\mathcal{M}$ is a finite tree whose nodes are labeled with configurations and such that the following conditions are satisfied:

- the root contains the *initial configuration* $q_0\omega$, where ω is of length n,
- every configuration $\omega q \omega'$ on the tree, where $\omega\omega'$ is of length at most 2^n, is succeed by:
  - at least one successor configuration, whenever $q \in Q_\exists$,
  - all successor configurations, whenever $q \in Q_\forall$,
- all leaves are labeled with halting configurations.

A tree is *accepting iff* all the leaves are labeled with accepting configurations and *rejecting* otherwise. An ATM *accepts* an input ω iff there exists an accepting ATM tree with $q_0\omega$ as its initial configuration. The set of all words accepted by an ATM $\mathcal{M}$ is denoted as the language $L(\mathcal{M})$. According to [CKS81, Theorem 3.4], the problem of deciding whether $\omega \in L(\mathcal{M})$, for ω and $\mathcal{M}$ complying to the requirements described above, is 2EXPTIME-hard.

Reduction.

Technically the reduction is quite involved but its conceptual core is straightforward. We use separate **DAlt** modalities for representing symbols of the alphabet and possible transitions. By isolating specific fragments of $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$-tree-models we can thus embed the syntactic structure of an ATM computation tree (see Figure 3.5). At the same time, using special counting concepts, which enable traversing this structure downwards and upwards, we align the succeeding configurations semantically, ensuring they satisfy the constraints of the respective ATM transitions (see Figure A.1).

Let $\mathcal{M} = (Q, \Sigma, \Gamma, q_0, \delta)$ be an ATM and ω the word for which we want to decide whether $\omega \in L(\mathcal{M})$. In the following we will construct a TBox $\mathcal{T}_\mathcal{M}$ and a concept $C_{\mathcal{M},\omega}$, of a total polynomial size in the size of the input, such that $\omega \in L(\mathcal{M})$ *iff* $C_{\mathcal{M},\omega}$ is satisfiable w.r.t. global $\mathcal{T}_\mathcal{M}$ in $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$. The encoding is constructed incrementally and provided with extensive explanations on the way.

First we define the set of **DAlt** modal operators:

alphabet modalities: $\bigcirc_a$, for every $a \in \Gamma$,

transition modalities: $\bigcirc_{q,a,m}$, for every $(q, a, m) \in \Theta$, where $\Theta = \{(q, a, m) \mid (q', b, q, a, m) \in \delta$ for any $b \in \Gamma$ and $q' \in Q\}$,

and introduce the following abbreviations (for any concept B):

$$\begin{aligned}
\Box B &= \bigsqcap_{a\in\Gamma} \bigcirc_a B, \\
\Diamond B &= \bigsqcup_{a\in\Gamma} \bigcirc_a B, \\
\blacksquare B &= \bigsqcap_{(q,a,m)\in\Theta} \bigcirc_{q,a,m} B, \\
\blacklozenge B &= \bigsqcup_{(q,a,m)\in\Theta} \bigcirc_{q,a,m} B.
\end{aligned}$$

In the encoding we use several counters, consisting of a number of inclusions of a total polynomial size, which allow to identify distances on the branches of the same fixed length 2^n. Constraints (A.1)-(A.5) implement an exemplary *downward counter*, based on atomic concepts X_i, for $1 \leq i \leq n$, which simulate bits in a binary number. The counting is initiated on $d \in \Delta$ whenever d instantiates concept $Count_d$. In every successor **DAlt**-world along the alphabet modalities, d becomes then an instance of a concept description, representing the consecutive number, which uniquely determines the distance from the world in which the counting was initiated. The counter turns the full loop, back to $Count_d$, in periods of 2^n.

$$Count_d \equiv \bigsqcap_{j=1}^{n} \neg X_j, \tag{A.1}$$

$$\neg X_i \sqcap \neg X_j \sqsubseteq \Box \neg X_i, \text{ for every } 1 \leq j < i \leq n, \tag{A.2}$$

$$X_i \sqcap \neg X_j \sqsubseteq \Box X_i, \text{ for every } 1 \leq j < i \leq n, \tag{A.3}$$

$$\neg X_j \sqcap X_{j-1} \sqcap \ldots \sqcap X_1 \sqsubseteq \Box X_j, \text{ for every } 1 \leq j \leq n, \tag{A.4}$$

$$X_j \sqcap X_{j-1} \sqcap \ldots \sqcap X_1 \sqsubseteq \Box \neg X_j, \text{ for every } 1 \leq j \leq n. \tag{A.5}$$

An alternative *upward counter*, initiated with $Count_u$ and implemented via template (A.6)-(A.10), behaves exactly the same way, with the only difference that the counting proceeds along the alphabet modalities *up* the branch of the model.

$$Count_u \equiv \bigsqcap_{j=1}^{n} X_j, \tag{A.6}$$

$$\Diamond(X_i \sqcap X_j) \sqsubseteq X_i, \text{ for every } 1 \leq j < i \leq n, \tag{A.7}$$

$$\Diamond(\neg X_i \sqcap X_j) \sqsubseteq \neg X_i, \text{ for every } 1 \leq j < i \leq n, \tag{A.8}$$

$$\Diamond(X_j \sqcap \neg X_{j-1} \sqcap \ldots \sqcap \neg X_1) \sqsubseteq \neg X_j, \text{ for every } 1 \leq j \leq n, \tag{A.9}$$

$$\Diamond(\neg X_j \sqcap \neg X_{j-1} \sqcap \ldots \sqcap \neg X_1) \sqsubseteq X_j, \text{ for every } 1 \leq j \leq n. \tag{A.10}$$

We can now introduce a fresh downward counter $Count_d^{tape}$:

$$Count_d^{tape} \equiv \bigsqcap_{j=1}^{n} \neg R_j, \tag{A.11}$$

and define constraints which encode a single tape on a branch of a model. In (A.12) we define the beginning of such a tape, in (A.13) its end, while with

(A.14)-(A.16) we ensure that there is a unique path connecting the two. Note that whenever an individual d instantiates concept $\mathit{StartTape}$, it becomes an instance of Tape for exactly 2^n succeeding worlds along a unique path of alphabet modalities. We will consider such a path as determining the content of the tape, as presented in Figure 3.5. In fact, in our models we will need only one such individual which will single out the whole structure of the ATM tree. Constraint (A.16) ensures that the blank symbol is followed only by blank symbols on the tape.

$$\mathit{StartTape} \equiv \mathit{Tape} \sqcap \mathit{Count}_d^{tape}, \tag{A.12}$$

$$\mathit{EndTape} \equiv \mathit{Tape} \sqcap \Diamond \mathit{Count}_d^{tape}, \tag{A.13}$$

$$\mathit{Tape} \sqcap \neg \mathit{EndTape} \sqsubseteq \Diamond \mathit{Tape}, \tag{A.14}$$

$$\Diamond(\mathit{Tape} \sqcap \neg \mathit{StartTape}) \sqsubseteq \mathit{Tape}, \tag{A.15}$$

$$\bigcirc_a \mathit{Tape} \sqcap \bigcirc_b \mathit{Tape} \sqsubseteq \bot, \text{ for every } a \neq b \in \Gamma, \tag{A.16}$$

$$\bigcirc_\varnothing(\mathit{Tape} \sqcap \bigcirc_a \mathit{Tape}) \sqsubseteq \bot, \text{ for every } a \neq \varnothing \in \Gamma. \tag{A.17}$$

Further, we implement the transitions by transferring the necessary information downwards or upwards the branches of a $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$-tree-model, as depicted in Figure A.1.

For the downward part, we introduce new concept names Q_q for every $q \in Q$ and $M_{q,a,m}$ for every $(q, a, m) \in \Theta$, as well as a fresh downward counter Count_d^{head} (A.18) for measuring the distance from the original position of the head. The Q_q concepts denote the current state and the position of the head, while the others serve for carrying the information about the following transitions. Information about the transitions is generated depending on whether the state is universal (A.19) or existential (A.20) and then carried to the end of the tape. There the transitions take place (A.21)-(A.22) and new tapes are initiated.

$$\mathit{Count}_d^{head} \equiv \prod_{j=1}^{n} \neg S_j, \tag{A.18}$$

$$\bigcirc_a(Q_q \sqcap Tape) \sqsubseteq \bigcirc_a(\bigsqcap_{(q'b'm)\in\delta(q,a)} M_{q',b,m} \sqcap Count_d^{head}), \tag{A.19}$$
$$\text{for every } a \in \Gamma, q \in Q_\forall,$$

$$\bigcirc_a(Q_q \sqcap Tape) \sqsubseteq \bigcirc_a(\bigsqcup_{(q'b'm)\in\delta(q,a)} M_{q',b,m} \sqcap Count_d^{head}), \tag{A.20}$$
$$\text{for every } a \in \Gamma, q \in Q_\exists,$$

$$M_{q,a,m} \sqsubseteq \Box M_{q,a,m}, \tag{A.21}$$

$$M_{q,a,m} \sqcap EndTape \sqsubseteq \bigcirc_{q,a,m}\Diamond StartTape, \text{ for every } (q,a,m) \in \Theta. \tag{A.22}$$

Note that, once we move along a transition modality, starting a new offspring of the computation, the concepts $M_{q,a,m}$ as well as the counters are not carried along. This is intended, as we want to avoid potential clashes with the information generated on the succeeding tapes. However, we still need to inform the new offsprings about their configurations. To this end we create copies $N_{q,a,m}$ for all concepts $M_{q,a,m}$, which continue to carry their information over the new tape (A.21)-(A.22). Further we introduce a fresh downward counter $Count_d^{*head}$, which proceeds with the counting exactly from the point where the previous head counter terminated (A.25)-(A.27). Finally, the constraints (A.28)-(A.29) introduce some handy abbreviations which will be used for imposing the new configuration.

$$M_{q,a,m} \sqsubseteq \bigcirc_{q,a,m} N_{q,a,m}, \tag{A.23}$$

$$N_{q,a,m} \sqsubseteq \Box N_{q,a,m}, \tag{A.24}$$

$$Count_d^{*head} \equiv \bigsqcap_{j=1}^{n} \neg T_j, \tag{A.25}$$

$$S_i \sqsubseteq \blacksquare T_i, \text{ for every } 1 \leq i \leq n, \tag{A.26}$$

$$\neg S_i \sqsubseteq \blacksquare \neg T_i, \text{ for every } 1 \leq i \leq n, \tag{A.27}$$

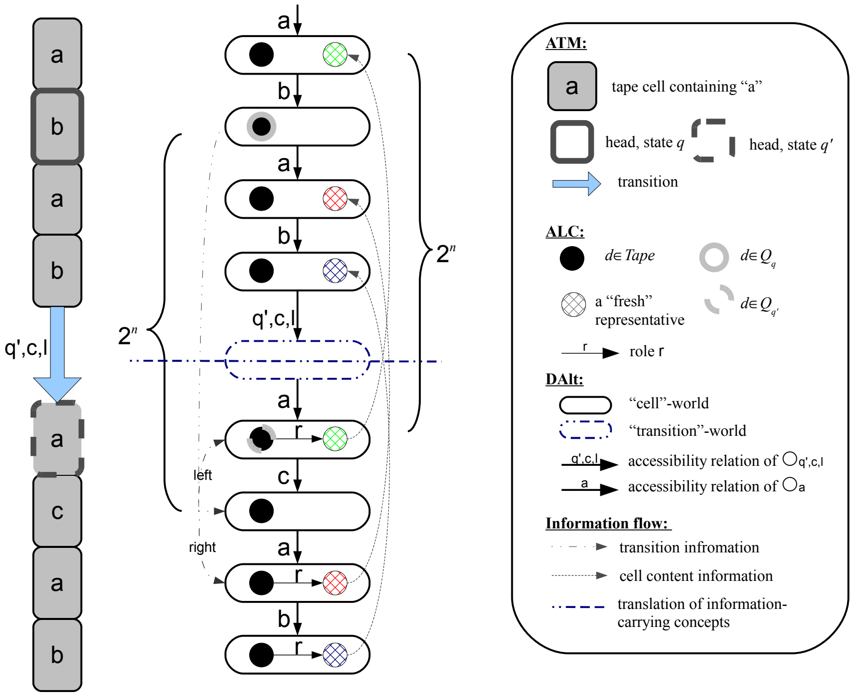

Figure A.1: A transition between succeeding configurations in $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$-tree-models for $n = 2$ and $(q', c, l) \in \delta(q, b)$.

$$Count_d^{*head} - 1 \equiv Head^l, \tag{A.28}$$

$$Count_d^{*head} \equiv Head^n, \tag{A.29}$$

$$Count_d^{*head} + 1 \equiv Head^r. \tag{A.30}$$

The necessary changes in the configuration are imposed through constraints (A.31)-(A.32), which place the head in the appropriate position, marking it with the new state concept, and force the old position to be overwritten with the new symbol. The inclusions (A.33)-(A.34) ensure that the transition does not push the head beyond the tape.

$$N_{q,a,m} \sqcap \mathit{Tape} \sqcap \mathit{Head}^m \sqsubseteq Q_q, \text{ for every } (q,a,m) \in \Theta, \tag{A.31}$$

$$\bigcirc_b (N_{q,a,m} \sqcap \mathit{Tape} \sqcap \mathit{Head}^n) \sqsubseteq \bot, \text{ for every } (q,a,m) \in \Theta \text{ and } b \neq a \in \Gamma, \tag{A.32}$$

$$\mathit{Head}^n \sqcap \mathit{StartTape} \sqsubseteq \neg N_{q,a,l}, \text{ for every } q \in Q, a \in \Gamma, \tag{A.33}$$

$$\mathit{Head}^n \sqcap \mathit{EndTape} \sqsubseteq \neg N_{q,a,r}, \text{ for every } q \in Q, a \in \Gamma. \tag{A.34}$$

In the opposite direction we will transfer the information about the content of the cells which are not meant to change during the transition. This information is carried by newly generated 'representatives', i.e., new r-successors of the individual instantiating Tape. Observe that since our models are tree-shaped, it follows that whenever the representative reaches the 2^n-th ancestor world (upwards the alphabet modalities and one transition modality), it is exactly the world which holds the previous version of the represented cell. This enables us to align the content of the two versions. In a similar way as before, we introduce two fresh upward counters which are synchronized at the point of transition (A.35)-(A.38).

$$\mathit{Count}_u^{cell} \equiv \bigsqcap_{j=1}^{n} U_j, \tag{A.35}$$

$$\mathit{Count}_u^{*cell} \equiv \bigsqcap_{j=1}^{n} V_j, \tag{A.36}$$

$$\blacklozenge U_i \sqsubseteq V_i, \text{ for every } 1 \leq i \leq n, \tag{A.37}$$

$$\blacklozenge \neg U_i \sqsubseteq \neg V_i, \text{ for every } 1 \leq i \leq n. \tag{A.38}$$

At the same time, for each $a \in \Gamma$ we introduce two concept names W_a, S_a, whose interpretation is propagated upwards the alphabet modalities (A.39)-(A.40) and aligned at the transition point (A.41). Constraint (A.42) generates a representative of each cell (except for the one that has been changed, marked with the concept $Head^n$), and equips it with the concept W describing the cell's content. Once this information arrives to the previous version of that cell we prevent the cells from having different content (A.43).

$$\Diamond W_a \sqsubseteq W_a, \text{ for every } a \in \Gamma, \tag{A.39}$$

$$\Diamond S_a \sqsubseteq S_a, \text{ for every } a \in \Gamma, \tag{A.40}$$

$$\blacklozenge W_a \sqsubseteq S_a, \text{ for every } a \in \Gamma, \tag{A.41}$$

$$\bigcirc_a(\mathit{Tape} \sqcap \neg \mathit{Head}^n) \sqsubseteq \bigcirc_a \exists r.(\mathit{Count}_u^{cell} \sqcap W_a), \text{ for every } a \in \Gamma, \tag{A.42}$$

$$\bigcirc_a(S_b \sqcap \mathit{Count}_u^{*cell}) \sqsubseteq \bot, \text{ for every } b \neq a \in \Gamma. \tag{A.43}$$

Finally, it suffices to ensure that nowhere in the model is the rejecting state satisfied.

$$\top \sqsubseteq \neg Q_{q_r} \tag{A.44}$$

This completes the construction of the TBox $\mathcal{T}_\mathcal{M}$. The initial configuration $q_0\omega$ is encoded as concept $C_{\mathcal{M},\omega}$. Let $\omega = a_1 \dots a_n$. For $2 \leq i \leq n$ define recursively:

$$A_i = \bigcirc_{a_i}(\mathit{Tape} \sqcap A_{i+1})$$
$$A_{n+1} = \bigcirc_\varnothing \mathit{Tape}$$

Then $C_{\mathcal{M},\omega} = \bigcirc_{a_1}(\mathit{StartTape} \sqcap Q_{q_0} \sqcap A_2)$. We conclude by demonstrating validity of the target claim:

Lemma 2. $\omega \in L(\mathcal{M})$ iff $C_{\mathcal{M},\omega}$ *is satisfiable w.r.t. global* $\mathcal{T}_\mathcal{M}$ *in* $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$.

Proof. ($\Rightarrow$) Suppose $\omega \in L(\mathcal{M})$ and T is an ATM computation tree accepting ω. We roughly sketch the construction of a model $\mathfrak{M} = (\mathfrak{W}, \{<_x\}_{x \in \Gamma \cup \Theta}, \Delta, \{\cdot^{\mathcal{I}(w)}\}_{w \in \mathfrak{W}})$ of $\mathcal{T}_\mathcal{M}$ satisfying $C_{\mathcal{M},\omega}$.

We assume that each tape associated with a configuration in T is of length exactly 2^n. Let $t(i, \omega q \omega')$ be a function returning the i-th symbol from the tape containing $\omega\omega'$, and $h(\omega q \omega')$ a function returning the position of the head over that tape. Let $q_0\omega$ be the initial configuration and $w \in \mathfrak{W}$ the root of $\mathfrak{M}$. Then for some $d \in \Delta$ set $d \in C_{\mathcal{M},\omega}^{\mathcal{I}(w)}$. Then encode the tape of $q_0\omega$ starting from w, according to the following inductive procedure. Given a tape of $\omega q \omega'$ and the world $w \in \mathfrak{W}$ in which the encoding starts, set $i := 1$ and $x := w$ and proceed recursively until $i = 2^n + 1$:

1. pick $w \in \mathfrak{W}$ such that $x <_{t(i,\omega q \omega')} w$;
2. set $d \in \mathit{Tape}^{\mathcal{I}(w)}$;
3. if $i = 1$ then set $d \in \mathit{StartTape}^{\mathcal{I}(w)}$ and $d \in (\mathit{Count}_d^{tape})^{\mathcal{I}(w)}$;
4. if $i = h(\omega q \omega')$ then set $d \in Q_q^{\mathcal{I}(w)}$, $d \in (\mathit{Count}_d^{head})^{\mathcal{I}(w)}$ and for all transitions (q, a, m) from $\omega q \omega'$ performed on T, $d \in M_{q,a,m}^{\mathcal{I}(w)}$;
5. if $i = 2^n$ then set $d \in \mathit{EndTape}^{\mathcal{I}(w)}$;
6. set $(d, e) \in r^{\mathcal{I}(w)}$ for some fresh $e \in \Delta$, $e \in W_{t(i,\omega q \omega')}^{\mathcal{I}(w)}$ and $e \in (\mathit{Count}_u^{cell})^{\mathcal{I}(w)}$
7. set $i := i + 1$ and $x := w$;

Then for every transition (q, a, m) from $\omega q \omega'$ in T, resulting in the succeeding configuration $\varpi q' \varpi'$, pick the world $w \in \mathfrak{W}$ such that $x <_{q,a,m} w$ and repeat the procedure above for the tape of $\varpi q' \varpi'$ starting from the world w. Once the halting configurations are encoded, fix the interpretations of the bit concepts associated with the respective counters and propagate the interpretations of selected concepts as follows:

- $M_{q,a,m}$ and $N_{q,a,m}$ for every $(q, a, m) \in \Theta$: downwards along relations $<_x$ for all $x \in \Gamma$;
- W_a and S_a for every $a \in \Gamma$: upwards along relations the $<_x$ for all $x \in \Gamma$;

In the worlds representing the transition points, ensure the proper alignment of the interpretations of the concept pairs $M_{q,a,m}$ – $N_{q,a,m}$ and W_a – S_a, as well as the bit concepts of the counters Count_d^{head} – Count_d^{*head} and Count_d^{cell} – Count_d^{*cell}.

($\Leftarrow$) This direction of the claim follows straightforwardly from the reduction. In order to retrieve an ATM tree accepting ω from a $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$-tree-model we only need to pick an individual d, such that $d \in C_{\mathcal{M},\omega}^{\mathcal{I}(w_0)}$ and follow the paths of worlds $w \in \mathfrak{W}$ for which $d \in \mathit{Tape}^{\mathcal{I}(w)}$, just as presented in Figure 3.5. On the way we collect information about the entire configuration. Two important comments are in order. First, note that the reduction is somewhat underconstrained in the sense that the models might represent also some surplus states or transitions. However, the proper computation tree, i.e., the one directly enforced by the encoding, has to appear within this structure. Secondly, we recall that the ATM trees we consider are all finite. Since the transitions in the reduction properly simulate those of an ATM, therefore the trees embedded in $(\mathbf{DAlt}_n)_{\mathcal{ALC}}$-tree-models have to be also finite, even though the models themselves are always infinite. ❑

A.3 NEXPTIME lower bounds

In this section we prove the NEXPTIME lower bound for the satisfiability problems in $\mathbf{S5}_{\mathcal{ALCO}}$ and $\mathfrak{C}^{\mathcal{ALC}}_{\mathcal{ALC}}$, respectively. All the remaining NEXPTIME lower bounds covered in Theorem 10 carry over directly from these two results, possibly involving the correspondence between $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$ with operators $\mathfrak{F}_2$ and $\mathbf{S5}_{\mathcal{L}}$, established in Theorem 2.

Theorem 16. *Deciding concept satisfiability w.r.t. global TBoxes in $\mathbf{S5}_{\mathcal{ALCO}}$, for local interpretation of object roles, is* NEXPTIME*-hard.*

Proof. The result is established by devising a polynomial reduction of the $2^n \times 2^n$ tiling problem, known to be NEXPTIME-complete [Mar99], to concept satisfiability in $\mathbf{S5}_{\mathcal{ALCO}}$ w.r.t. global TBoxes. An instance $\mathfrak{T} = (n, T)$ of the problem is defined as follows: given some $n \in \mathbb{N}$ in unary and a set of tiles $T = \{\tau_0, \dots, \tau_m\}$, decide whether a $2^n \times 2^n$ grid can be tiled with T where the first cell in the grid is tiled with some $\tau_0 \in T$.

Let $\mathfrak{T} = (n, T)$ be an instance of the problem. In the consecutive steps, we define a TBox $\mathcal{T}_{\mathfrak{T}}$ and a concept $C_{\mathfrak{T}}$, such that there exists a tiling for $\mathfrak{T}$ *iff* $C_{\mathfrak{T}}$ is satisfiable w.r.t. $\mathcal{T}_{\mathfrak{T}}$.

First, the inclusions (A.45)-(A.54) enforce a 2^{2n}-long chain of individuals (*Grid*), uniquely identifiable by counting concepts X_i and Y_i, for $i \in (1, 2n)$. Notably, the Y-counter is shifted in the phase w.r.t. the X-counter by exactly 2^n, (i.e.: $X + 2^n = Y$), which further on is utilized for identifying the top-down neighbors in the tiling. Also, every 2^n-th individual, starting from the beginning of the chain, is made an instance of concept *RightEdge*, marking the right edge of the tiling (A.55):

$$\mathit{StartGrid} \equiv \mathit{Grid} \sqcap \prod_{j=1}^{2n} \neg X_j \sqcap \prod_{j=1}^{n} \neg Y_j \sqcap Y_{n+1} \sqcap \prod_{j=n+2}^{2n} \neg Y_j, \tag{A.45}$$

$$\mathit{EndGrid} \equiv \bigsqcap_{j=1}^{2n} X_j, \quad \mathit{Grid} \sqcap \neg \mathit{EndGrid} \sqsubseteq \exists s.\mathit{Grid}, \tag{A.46}$$

$$\neg X_i \sqcap \neg X_j \sqsubseteq \forall s.\neg X_i, \text{ for every } 1 \leq j < i \leq 2n, \tag{A.47}$$

$$X_i \sqcap \neg X_j \sqsubseteq \forall s.X_i, \text{ for every } 1 \leq j < i \leq 2n, \tag{A.48}$$

$$\neg X_j \sqcap X_{j-1} \sqcap \ldots \sqcap X_1 \sqsubseteq \forall s.X_j, \text{ for every } 1 \leq j \leq 2n, \tag{A.49}$$

$$X_j \sqcap X_{j-1} \sqcap \ldots \sqcap X_1 \sqsubseteq \forall s.\neg X_j, \text{ for every } 1 \leq j \leq 2n, \tag{A.50}$$

$$\neg Y_i \sqcap \neg Y_j \sqsubseteq \forall s.\neg Y_i, \text{ for every } 1 \leq j < i \leq 2n, \tag{A.51}$$

$$Y_i \sqcap \neg Y_j \sqsubseteq \forall s.Y_i, \text{ for every } 1 \leq j < i \leq 2n, \tag{A.52}$$

$$\neg Y_j \sqcap Y_{j-1} \sqcap \ldots \sqcap Y_1 \sqsubseteq \forall s.Y_j, \text{ for every } 1 \leq j \leq 2n, \tag{A.53}$$

$$Y_j \sqcap Y_{j-1} \sqcap \ldots \sqcap Y_1 \sqsubseteq \forall s.\neg Y_j, \text{ for every } 1 \leq j \leq 2n, \tag{A.54}$$

$$\mathit{RightEdge} \equiv \bigsqcap_{j=1}^{n} X_j. \tag{A.55}$$

Next, by (A.56)-(A.57), the values of the counting concepts are propagated globally across all **S5**-worlds:

$$X_i \sqsubseteq \Box X_i, \quad \neg X_i \sqsubseteq \Box \neg X_i, \text{ for every } 1 \leq i \leq 2n, \tag{A.56}$$

$$Y_i \sqsubseteq \Box Y_i, \quad \neg Y_i \sqsubseteq \Box \neg Y_i, \text{ for every } 1 \leq i \leq 2n. \tag{A.57}$$

Further, we impose the basic coloring constraints over all individuals (A.58), adjust the coloring of all the left-right neighbors: (A.59), and propagate the tile types over all **S5**-worlds (A.60):

$$\top \sqsubseteq (\bigsqcup_{\tau_i} T_i) \sqcap \bigsqcap_{\tau_i \neq \tau_j} \neg(T_i \sqcap T_j), \text{ for every } \tau_i, \tau_j \in T, \tag{A.58}$$

$$T_i \sqcap \neg \mathit{RightEdge} \sqsubseteq \forall s.(\bigsqcup_{right(\tau_i)=left(\tau_j)} T_j), \text{ for every } \tau_i, \tau_j \in T, \tag{A.59}$$

$$T_i \sqsubseteq \Box T_i, \text{ for every } \tau_i \in T. \tag{A.60}$$

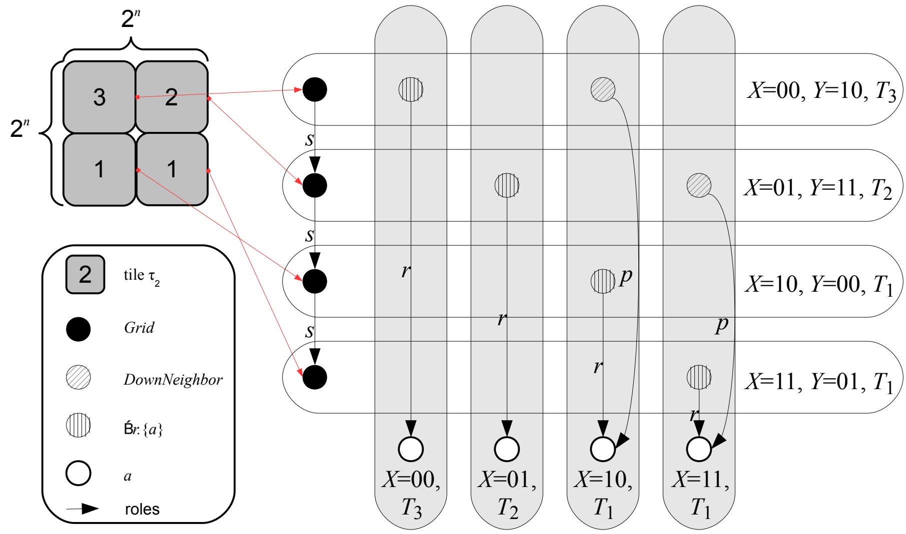

Figure A.2: Encoding of a $2^n \times 2^n$ tiling in an $\mathbf{S5}_{\mathcal{ALCO}}$-model.

The key to the reduction is a suitable use of a single nominal $\{a\}$ (see Figure A.2). By (A.61) every individual in the grid is linked to a via role r in some **S5**-world. There, due to (A.62)-(A.63), the value of the X-counter and the tile type assigned to the individual is forced upon a. Consequently, by assuming rigid individual names,[1] we generate 2^{2n} distinct **S5**-worlds:

$$\mathit{Grid} \sqsubseteq \Diamond \exists r.\{a\}, \tag{A.61}$$

$$X_i \sqsubseteq \forall r.X_i, \quad \neg X_i \sqsubseteq \forall r.\neg X_i, \text{ for every } 1 \leq i \leq 2n, \tag{A.62}$$

[1]Such assumption can be also made explicit by including axiom $\{a\} \sqsubseteq \Box\{a\}$.

$$T_i \sqsubseteq \forall r.T_i, \text{ for every } \tau_i \in T. \tag{A.63}$$

Finally, in every **S5**-world, all individuals are linked to a via p (A.64). Whenever the value of the Y-counter on a grid-individual matches the value of the X-counter on a (A.65), the proper top-down coloring constraints are imposed (A.66):

$$\top \sqsubseteq \exists p.\{a\}, \tag{A.64}$$

$$\mathit{DownNeighbor} \equiv \prod_{j=1}^{2n}((Y_i \sqcap \exists p.X_i) \sqcup (\neg Y_i \sqcap \exists p.\neg X_i)), \tag{A.65}$$

$$T_i \sqcap \mathit{DownNeighbor} \sqsubseteq \forall p. \bigsqcup_{down(\tau_i)=top(\tau_j)} T_j, \text{ for every } \tau_i, \tau_j \in T. \tag{A.66}$$

The TBox $\mathcal{T}_{\mathfrak{T}}$ is defined as the union of the axioms (A.45)-(A.66). It is easy to see that the size of $\mathcal{T}_{\mathfrak{T}}$ is polynomial in the size of the instance $\mathfrak{T}$. Finally, we define the concept $C_{\mathfrak{T}} = \mathit{StartGrid} \sqcap T_0$ and claim that there is a tiling for $\mathfrak{T}$ *iff* $C_{\mathfrak{T}}$ is satisfiable w.r.t. globally interpreted $\mathcal{T}_{\mathfrak{T}}$.
($\Rightarrow$) Let τ be a tiling for $\mathfrak{T}$, i.e. a mapping from $2^n \times 2^n$ to T. Define an $\mathbf{S5}_{\mathcal{ALCO}}$-model $\mathfrak{M} = (\Theta, \mathfrak{C}, \cdot^{\mathcal{J}}, \Delta, \{\cdot^{\mathcal{I}(c)}\}_{c\in\mathfrak{C}})$ for $\mathcal{T}_{\mathfrak{T}}$ satisfying $C_{\mathfrak{T}}$ as follows. First, transform τ into $\pi : 2^{2n} \mapsto T$, such that for every $(x, y) \in 2^n \times 2^n$, $\tau(x, y) = \pi(y * 2^n + x)$. Then, fix $\Theta = \mathfrak{C} = \{c_i \mid i \in (0, 2^{2n})\}$ and $\Delta = \{d_i \mid i \in (0, 2^{2n})\}$ and ensure that the following interpretation constraints are satisfied:

- $a^{\mathcal{I}(c)} = d_0$ for $d_0 \in \Delta$ and every $c \in \mathfrak{C}$,
- for $c_0 \in \mathfrak{C}$:
  - $\mathit{Grid}^{\mathcal{I}(c_0)} = \Delta \setminus \{d_0\}$,
  - $\mathit{StartGrid}^{\mathcal{I}(c_0)} = \{d_1 \in \Delta\}$, $\mathit{EndGrid}^{\mathcal{I}(c_0)} = \{d_{2^{2n}} \in \Delta\}$,
  - $\mathit{RightEdge}^{\mathcal{I}(c_0)} = \{d_{2^n * i} \in \Delta\}$,
  - $s^{\mathcal{I}(c_0)} = \{\langle d_i, d_{i+1}\rangle \mid d_i, d_{i+1} \in \Delta, i \geq 1\}$,
- $\{d_i \mid \pi(i) = \tau_j\} \subseteq T_j^{\mathcal{I}(c)}$, for every $c \in \mathfrak{C}$ and $\tau_j \in T$,
- $d_0 \in T_j^{\mathcal{I}(c_i)}$ *iff* $\pi(i) = \tau_j$, for every $i \geq 1$ and $\tau_j \in T$,

- $r^{\mathcal{I}(c_i)} = \{\langle d_i, d_0 \rangle \mid d_i \in \Delta\}$ for $i \geq 1$,
- $p^{\mathcal{I}(c)} = \{\langle d, d_0 \rangle \mid d \in \Delta\}$ for every $c \in \mathfrak{C}$,
- $\mathit{DownNeighbor}^{\mathcal{I}(c_i)} = \{d_{i-2^n} \in \Delta\}$, for every $c_i \in \mathfrak{C}$ and $i \geq 2^n + 1$.

The interpretations can be straightforwardly extended over the counting concepts X_i and Y_i so that $\mathfrak{M}$ is indeed a model for $\mathcal{T}_{\mathfrak{T}}$, where $d_1 \in (C_{\mathfrak{T}})^{\mathcal{I}(c_0)}$.

($\Leftarrow$) Let $\mathfrak{M}$ be an $\mathbf{S5}_{\mathcal{ALCO}}$-model of $\mathcal{T}_{\mathfrak{T}}$ satisfying $C_{\mathfrak{T}}$. Then, a tiling for $\mathfrak{T}$ can be retrieved from $\mathfrak{M}$ by mapping a chain of s-successors, which instantiate concept *Grid* in the **S5**-world in which $C_{\mathfrak{T}}$ is satisfied, on the $2^n \times 2^n$ grid, where the type of a tile in the grid is determined by the unique concept T_i satisfied by the individual in the chain. The coloring constraints have to be satisfied by the construction of the encoding. ❑

Theorem 17. *Deciding concept satisfiability w.r.t. global TBoxes in $\mathfrak{C}^{\mathcal{ALC}}_{\mathcal{ALC}}$, with context operators $\mathfrak{F}_2$ only and for local interpretation of object roles, is* NEXPTIME*-hard.*

Proof. The result is established by reducing the $2^n \times 2^n$ tiling problem. Let $\mathfrak{T} = (n, T)$ be an instance of the problem. In the consecutive steps, we define a TBox $\mathcal{T}_{\mathfrak{T}}$ and a concept $C_{\mathfrak{T}}$, such that there exists a tiling for $\mathfrak{T}$ *iff* $C_{\mathfrak{T}}$ is satisfiable w.r.t. $\mathcal{T}_{\mathfrak{T}}$. Again, the encoding utilizes the possibility of constructing and constraining a "diagonal" in models, as depicted in Figure A.3, representing the whole tiling in a linear projection.

The inclusions (A.67)-(A.72) enforce a 2^{2n}-long chain of individuals, uniquely identifiable by counting concepts X_i, for $i \in (1, 2n)$. Moreover, every 2^n-th individual, starting from the beginning of the chain, is an instance of concept $\mathit{RightEdge}$, marking the right edge of the tiling, while the last 2^n individuals are instances of $\mathit{BottomEdge}$, marking the bottom of the tiling.

$$\mathit{StartGrid} \equiv \prod_{j=1}^{2n} \neg X_j, \quad \mathit{EndGrid} \equiv \prod_{j=1}^{2n} X_j, \quad \neg\mathit{EndGrid} \sqsubseteq \langle \top \rangle \exists r.\top, \tag{A.67}$$

$$\neg X_i \sqcap \neg X_j \sqsubseteq [\top] \forall r.\neg X_i, \text{ for every } 1 \leq j < i \leq 2n, \tag{A.68}$$

$$X_i \sqcap \neg X_j \sqsubseteq [\top] \forall r.X_i, \text{ for every } 1 \leq j < i \leq 2n, \tag{A.69}$$

$$\neg X_j \sqcap X_{j-1} \sqcap \ldots \sqcap X_1 \sqsubseteq [\top] \forall r.X_j, \text{ for every } 1 \leq j \leq 2n, \tag{A.70}$$

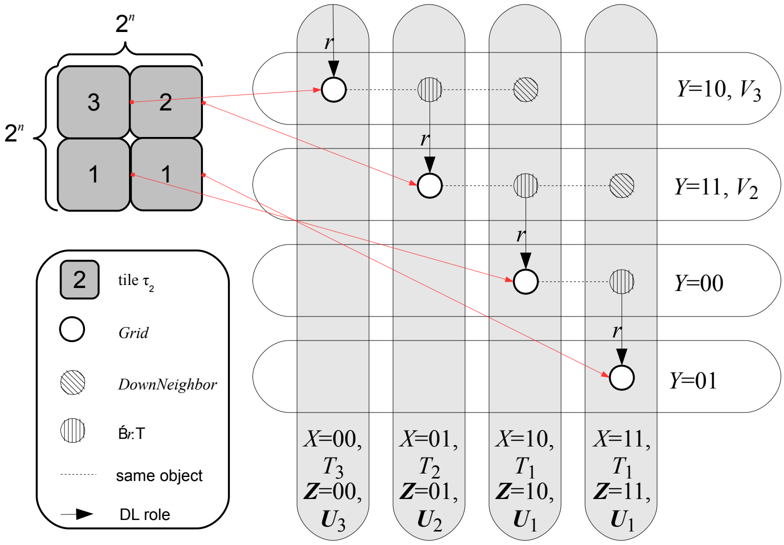

Figure A.3: Encoding of a $2^n \times 2^n$ tiling in an $\mathbf{S5}_{\mathcal{ALC}}^{\mathcal{ALC}}$-model.

$$X_j \sqcap X_{j-1} \sqcap \ldots \sqcap X_1 \sqsubseteq [\top]\forall r.\neg X_j, \text{ for every } 1 \leq j \leq 2n, \tag{A.71}$$

$$\mathit{RightEdge} \equiv \bigsqcap_{j=1}^{n} X_j, \quad \mathit{BottomEdge} \equiv \bigsqcap_{j=n+1}^{2n} X_j. \tag{A.72}$$

The values of these counting concepts are then propagated over all the objects in the given context, by involving an interaction with concepts of the metalanguage $\mathbf{Z}_i$, for $i \in (1, 2n)$ (A.73).

$$\top \sqsubseteq [\mathbf{Z}_i]X_i, \quad \top \sqsubseteq [\neg\mathbf{Z}_i]\neg X_i, \text{ for every } 1 \leq i \leq 2n. \tag{A.73}$$

Each individual is required to satisfy exactly one concept T_i, representing a tile type $\tau_i \in T$ (A.74). This type is then propagated to all individuals in the given world (A.75-A.76) and used to adjust the coloring of the left-right neighbors (A.77).

$$\top \sqsubseteq (\bigsqcup_{\tau_i} T_i) \sqcap \bigsqcap_{\tau_i \neq \tau_j} \neg(T_i \sqcap T_j), \text{ for every } \tau_i, \tau_j \in T, \tag{A.74}$$

$$\top \sqsubseteq [\boldsymbol{U}_i]T_i, \text{ for every } \tau_i \in T. \tag{A.75}$$

$$\top \sqsubseteq [\neg\boldsymbol{U}_i]\neg T_i, \text{ for every } \tau_i \in T, \tag{A.76}$$

$$T_i \sqcap \neg \mathit{RightEdge} \sqsubseteq [\top]\forall r.(\bigsqcup_{right(\tau_i)=left(\tau_j)} T_j), \text{ for every } \tau_i, \tau_j \in T. \tag{A.77}$$

For each individual we identify the counter of its down neighbor and encode this value rigidly across all **S5**-worlds by means of concepts Y_i (A.78-A.83). In the same manner, the tile type is propagated (A.84).

$$\neg X_i \sqcap \neg X_j \sqsubseteq \forall r.[\top]\neg Y_i, \text{ for every } n+1 \leq j < i \leq 2n, \tag{A.78}$$

$$X_i \sqcap \neg X_j \sqsubseteq \forall r.[\top]Y_i, \text{ for every } n+1 \leq j < i \leq 2n, \tag{A.79}$$

$$\neg X_j \sqcap X_{j-1} \sqcap \ldots \sqcap X_{(n+1)} \sqsubseteq \forall r.[\top]Y_j, \text{ for every } n+1 \leq j \leq 2n, \tag{A.80}$$

$$X_j \sqcap X_{j-1} \sqcap \ldots \sqcap X_{(n+1)} \sqsubseteq \forall r.[\top]\neg Y_j, \text{ for every } n+1 \leq j \leq 2n, \tag{A.81}$$

$$X_i \sqsubseteq \forall r.[\top]Y_i, \text{ for every } 1 \leq i \leq n, \tag{A.82}$$

$$\neg X_i \sqsubseteq \forall r.[\top]\neg Y_i, \text{ for every } 1 \leq i \leq n, \tag{A.83}$$

$$\neg \mathit{BottomEdge} \sqcap T_i \sqsubseteq \forall r.[\top]V_i, \text{ for every } \tau_i \in T. \tag{A.84}$$

Finally, the up-down coloring constraints are enforced whenever the value of Y_i's agrees with the X_i-counter. (A.85-A.86).

$$\mathit{DownNeighbor} \equiv \bigsqcap_{1 \leq i \leq 2n} ((X_i \sqcap Y_i) \sqcup (\neg X_i \sqcap \neg Y_i)), \tag{A.85}$$

$$\mathit{DownNeighbor} \sqcap V_i \sqsubseteq \bigsqcap_{down(\tau_i) \neq up(\tau_j)} \neg T_j, \text{ for every } \tau_i \in T. \tag{A.86}$$

The TBox $\mathcal{T}_{\mathfrak{T}}$ is defined as the union of the axioms (A.67)-(A.86). It is easy to see that the size of $\mathcal{T}_{\mathfrak{T}}$ is polynomial in the size of the instance $\mathfrak{T}$. Finally, we define the concept $C_{\mathfrak{T}} = \exists r.(\mathit{StartGrid} \sqcap T_0)$ and claim that there is a tiling for $\mathfrak{T}$ *iff* $C_{\mathfrak{T}}$ is satisfiable w.r.t. $\mathcal{T}_{\mathfrak{T}}$.

($\Rightarrow$) Let τ be a tiling for $\mathfrak{T}$, i.e. a mapping from $2^n \times 2^n$ to T. Define an $\mathfrak{C}^{\mathcal{ALC}}_{\mathcal{ALC}}$-model $\mathfrak{M} = (\Theta, \mathfrak{C}, \cdot^{\mathcal{J}}, \Delta, \{\cdot^{\mathcal{I}(c)}\}_{c\in\mathfrak{C}})$ for $\mathcal{T}_{\mathfrak{T}}$ satisfying $C_{\mathfrak{T}}$ as follows. First, transform τ into $\pi : 2^{2n} \mapsto T$, such that for every $(x, y) \in 2^n \times 2^n$, $\tau(x, y) = \pi(y * 2^n + x)$. Then, fix $\Theta = \mathfrak{C} = \{c_i \mid i \in (1, 2^{2n})\}$ and $\Delta = \{d_i \mid i \in (0, 2^{2n})\}$ and ensure that the following interpretation constraints are satisfied:

- $r^{\mathcal{I}(c_i)} = \{(d_{i-1}, d_i) \mid d_{i-1}, d_i \in \Delta\}$,
- $\mathit{StartGrid}^{\mathcal{I}(c_1)} = \{d_1 \in \Delta\}$, $\mathit{EndGrid}^{\mathcal{I}(c_{2^{2n}})} = \{d_{2^{2n}} \in \Delta\}$,
- for every $c_i \in \mathfrak{C}$, $\mathit{DownNeighbor}^{\mathcal{I}(c_i)} = \{d_{i-2^n} \in \Delta\}$,
- for every $\tau_j \in T$ and $i \in (1, 2^{2n})$, $T_j^{\mathcal{I}(c_i)} = \Delta$, if $\pi(i) = \tau_j$, and else $T_j^{\mathcal{I}(c_i)} = \emptyset$.

The interpretations can be straightforwardly extended over the remaining concepts so that $\mathfrak{M}$ is indeed a model for $\mathcal{T}_{\mathfrak{T}}$, where $d_0 \in (C_{\mathfrak{T}})^{\mathcal{I}(c_1)}$.

($\Leftarrow$) Let $\mathfrak{M}$ be an $\mathfrak{C}^{\mathcal{ALC}}_{\mathcal{ALC}}$-model of $\mathcal{T}_{\mathfrak{T}}$ satisfying $C_{\mathfrak{T}}$. Then, a tiling for $\mathfrak{T}$ can be retrieved from $\mathfrak{M}$ by mapping the diagonal of the model on the $2^n \times 2^n$ grid, where the type of a tile in the grid is determined by the unique concept T_i satisfied by the individual in the chain. The coloring constraints have to be satisfied by the construction of the encoding. ❑

A.4 NEXPTIME/EXPTIME upper bounds

In this section we derive some NEXPTIME and EXPTIME upper bounds which transfer directly to the respective decision problems described in Theorems 10 and 11. Specifically, we prove: 1) the NEXPTIME upper bound for $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{SHIO}}$ with object axioms of the form $\boldsymbol{C} : \varphi$, where φ is a formula constructed according to the grammar:

$$\phi \mid \neg\phi \mid \phi \wedge \phi \mid \langle C \rangle \phi$$

such that ϕ is an axiom over the object language of $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{SHIO}}$ and where $\mathcal{L}_C \in \{\mathcal{SHIO}, \mathcal{EL}^{++}\}$; 2) the EXPTIME upper bound for $\mathfrak{C}^{\mathcal{EL}^{++}}_{\mathcal{SHI}}$.

The decision procedures devised here are essentially variants of the type-based techniques, commonly used in proving the complexity results for the satisfiability problem in various modal logics and their combinations [KWZG03]. First, we introduce a number of notational conventions and auxiliary results that should ease the layout of the target proofs. Whenever necessary we distinguish between the languages under consideration.

Consider a knowledge base $\mathcal{K} = (\mathcal{C}, \mathcal{O})$ in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$. We use the following notation to mark the sets of symbols of particular type occurring in $\mathcal{K}$:

- $con_c(\mathcal{K})$: the set of all context language concepts, closed under negation ($\mathcal{L}_C = \mathcal{SHIO}$),
- $con_c(\mathcal{K})$: the set of all context language concepts ($\mathcal{L}_C = \mathcal{EL}^{++}$),
- $con_c^{op}(\mathcal{K}) \subseteq con_c(\mathcal{K})$: the set of all context language concepts occurring inside the context operators,
- $con_o(\mathcal{K})$: the set of all object language concepts, closed under negation,
- $rol_c(\mathcal{K})$: the set of all context language roles,
- $rol_c^+(\mathcal{K}) \subseteq rol_c(\mathcal{K})$: the set of all context language transitive roles ($\mathcal{M} = \mathcal{SHIO}$),
- $rol_o(\mathcal{K})$: the set of all object roles,
- $rol_o^+(\mathcal{K}) \subseteq rol_o(\mathcal{K})$: the set of all transitive object roles,
- $obj_o(\mathcal{K})$: the set of object individual names,
- $sub_o(\mathcal{K})$: the set of all object (sub)formulas, closed under negation.

By $\cdot^-$ we denote the inverse constructor for roles and assume that $(r^-)^- = r$ (resp. $(\boldsymbol{r}^-)^- = \boldsymbol{r}$). Let f be a set of $\mathcal{SHIO}$ formulas. Then by $\sqsubseteq^*_f$ we denote the reflexive-transitive closure of $\sqsubseteq$ on $\{r \sqsubseteq s, s^- \sqsubseteq r^- \mid r \sqsubseteq s \in f\}$ (resp. $\{\boldsymbol{r} \sqsubseteq \boldsymbol{s}, \boldsymbol{s}^- \sqsubseteq \boldsymbol{r}^- \mid \boldsymbol{r} \sqsubseteq \boldsymbol{s} \in f$). Without loss of generality we assume that neither $[\cdot]$, $\forall$ nor $\sqcup$ occur in $\mathcal{K}$. Further, in order to reduce the syntactic load in the considered cases, whenever possible we apply the following replacements of all the respective formulas with their equivalents:

$$
\begin{array}{llll}
a : C & \Rightarrow & \{a\} \sqsubseteq C, & (\mathcal{L}_O = \mathcal{SHIO}) \\
r(a,b) & \Rightarrow & \{a\} \sqsubseteq \exists r.\{b\}, & (\mathcal{L}_O = \mathcal{SHIO}) \\
\boldsymbol{a} : \boldsymbol{C} & \Rightarrow & \{\boldsymbol{a}\} \sqsubseteq \boldsymbol{C}, & \\
\boldsymbol{r}(\boldsymbol{a},\boldsymbol{b}) & \Rightarrow & \{\boldsymbol{a}\} \sqsubseteq \exists \boldsymbol{r}.\{\boldsymbol{b}\}, & \\
\mathrm{dom}(\boldsymbol{r}) \sqsubseteq \boldsymbol{C} & \Rightarrow & \exists \boldsymbol{r}.\top \sqsubseteq \boldsymbol{C}, & (\mathcal{L}_C = \mathcal{EL}^{++})
\end{array}
$$

An *object type* for $\mathcal{K}$ is a subset $t_o \subseteq con_o(\mathcal{K})$, where:

- $\neg\top \notin t_o$ and $\bot \notin t_o$,
- $C \in t_o$ iff $\neg C \notin t_o$, for all $C \in con_o(\mathcal{K})$,
- $C \sqcap D \in t_o$ iff $\{C, D\} \subseteq t_o$, for all $C \sqcap D \in con_o(\mathcal{K})$,
- $\{\exists s.C \mid \exists r.C \in t_c\} \subseteq t_c$, for every $s \in rol_c(\mathcal{K})$ such that $r \sqsubseteq^*_{\mathcal{C}} s$,
- $\{\neg\exists s.C \mid \neg\exists r.C \in t_c\} \subseteq t_c$, for every $\boldsymbol{s} \in rol_c(\mathcal{K})$ such that $s \sqsubseteq^*_{\mathcal{C}} r$.

The set of all object types for $\mathcal{K}$ is denoted by Π. An *object formula type* for $\mathcal{K}$ is a subset $f \subseteq sub_o(\mathcal{K})$, where:

- $\varphi \in f$ iff $\neg\varphi \notin f$, for all $\varphi \in sub_o(\mathcal{K})$,
- $\varphi \wedge \psi \in f$ iff $\{\varphi, \psi\} \subseteq f$, for all $\varphi \wedge \psi \in sub_o(\mathcal{K})$,

The set of all object formula types for $\mathcal{K}$ is denoted by Φ. A *context type* for $\mathcal{K}$ is a subset $t_c \subseteq con_c(\mathcal{K})$, where:

- $\neg\top \notin t_c$ and $\bot \notin t_c$,
- $\boldsymbol{C} \in t_c$ iff $\neg\boldsymbol{C} \notin t_c$, for all $\boldsymbol{C} \in con_c(\mathcal{K})$, $(\mathcal{L}_C = \mathcal{SHIO})$
- $\boldsymbol{C} \sqcap \boldsymbol{D} \in t_c$ iff $\{\boldsymbol{C}, \boldsymbol{D}\} \subseteq t_c$, for all $\boldsymbol{C} \sqcap \boldsymbol{D} \in con_c(\mathcal{K})$,
- $\{\exists \boldsymbol{s}.\boldsymbol{C} \mid \exists \boldsymbol{r}.\boldsymbol{C} \in t_c\} \subseteq t_c$, for every $\boldsymbol{s} \in rol_c(\mathcal{K})$ such that $\boldsymbol{r} \sqsubseteq^*_{\mathcal{C}} \boldsymbol{s}$,
- $\{\neg\exists \boldsymbol{s}.\boldsymbol{C} \mid \neg\exists \boldsymbol{r}.\boldsymbol{C} \in t_c\} \subseteq t_c$, for every $\boldsymbol{s} \in rol_c(\mathcal{K})$ such that $\boldsymbol{s} \sqsubseteq^*_{\mathcal{C}} \boldsymbol{r}$.

The set of all context types for $\mathcal{K}$ is denoted by Ξ.

The following two definitions introduce the notions of *matching object role-successor* and *matching* **S5**-*successor*, used in the proofs for reconstructing the role relationships and accessibility relations between individuals in the object dimension.

Definition 41 (matching object role-successor). *Let t_o, t'_o be two object types for $\mathcal{K}$. For any $r \in rol_o(\mathcal{K})$, t'_o is a* matching r-successor *for t_o under $f \subseteq \mathrm{sub}_o(\mathcal{K})$* iff *the following conditions are satisfied:*

- $\{\neg C \mid \neg\exists r.C \in t_o\} \subseteq t'_o$ *and* $\{\neg C \mid \neg\exists r^-.C \in t'_o\} \subseteq t_o$,
- *if* $\{r, r^-\} \cap rol^+_o(\mathcal{K}) \neq \emptyset$ *then* $\{\neg\exists r.C \in t_o\} \subseteq t'_o$ *and* $\{\neg\exists r^-.C \in t'_o\} \subseteq t_o$,
- t'_o *is a matching s-successor for every $s \in rol_o(\mathcal{K})$ such that $r \sqsubseteq^*_f s$.*

Definition 42 (matching **S5**-successor). *For any object type t_o for $\mathcal{K}$, let $\mathbf{m}(t_o)$ denote the set of all object concepts containing context operators in t_o, i.e.: $\mathbf{m}(t_o) = \{\langle \boldsymbol{C} \rangle D, \neg\langle \boldsymbol{C} \rangle D \in t_o \mid \boldsymbol{C} \in con_c(\mathcal{K}), D \in con_o(\mathcal{K})\}$. Then, two object types t_o, t'_o for $\mathcal{K}$ are* matching **S5**-successors iff $\mathbf{m}(t_o) = \mathbf{m}(t'_o)$.

The analogous definition of matching role-successor applies to metalanguage roles in $\mathcal{L}_C = \mathcal{SHIO}$:

Definition 43 (matching metalanguage role-successor). *Let t_c, t'_c be two context types for $\mathcal{K}$. For any $\boldsymbol{r} \in rol_c(\mathcal{K})$, t'_c is a* matching $\boldsymbol{r}$-successor *for t_c under $\mathcal{C}$* iff *the following conditions are satisfied:*

- $\{\neg \boldsymbol{C} \mid \neg\exists \boldsymbol{r}.\boldsymbol{C} \in t_c\} \subseteq t'_c$ *and* $\{\neg \boldsymbol{C} \mid \neg\exists \boldsymbol{r}^-.\boldsymbol{C} \in t'_c\} \subseteq t_c$,
- *if* $\{\boldsymbol{r}, \boldsymbol{r}^-\} \cap rol^+_c(\mathcal{K}) \neq \emptyset$ *then* $\{\neg\exists \boldsymbol{r}.\boldsymbol{C} \in t_c\} \subseteq t'_c$ *and* $\{\neg\exists \boldsymbol{r}^-.\boldsymbol{C} \in t'_c\} \subseteq t_c$,
- t'_c *is a matching $\boldsymbol{s}$-successor for every $\boldsymbol{s} \in rol_c(\mathcal{K})$ such that $\boldsymbol{r} \sqsubseteq^*_{\mathcal{C}} \boldsymbol{s}$.*

Definition 44 ($\mathcal{C}$-admissibility). *Let S be a set of context types for $\mathcal{K} = (\mathcal{C}, \mathcal{O})$. We say that S is $\mathcal{C}$-*admissible iff *there exists a model $(S, \cdot^{\mathcal{J}})$ for $\mathcal{C}$, such that for every $t_c \in S$ and $\boldsymbol{C} \in con_c(\mathcal{K})$, $t_c \in \boldsymbol{C}^{\mathcal{J}}$* iff $\boldsymbol{C} \in t_c$.

Theorem 18 ($\mathcal{C}$-admissibility). *Let $S_\times$ be a multiset of context types for $\mathcal{K} = (\mathcal{C}, \mathcal{O})$, where $\mathcal{C}$ is a knowledge base in $\mathcal{L}_C \in \{\mathcal{SHIO}, \mathcal{EL}^{++}\}$, such that:*

- *S is the underlying set of elements of $S_\times$,*
- *for every $\boldsymbol{a} \in obj_c(\mathcal{K})$ and $t_c, t'_c \in S_\times$, if $\{\boldsymbol{a}\} \in t_c \cap t'_c$ then $t_c = t'_c$.*

Then the following two conditions are equivalent:

1. *There exists a model $(S_\times, \cdot^{\mathcal{J}})$ for $\mathcal{C}$, such that for every $t_c \in S_\times$ and $\boldsymbol{C} \in con_c(\mathcal{K})$, $t_c \in \boldsymbol{C}^{\mathcal{J}}$* iff $\boldsymbol{C} \in t_c$.
2. *S is $\mathcal{C}$-admissible.*

Proof. Intuitively, since neither of $\mathcal{L} \in \{\mathcal{SHIO}, \mathcal{EL}^{++}\}$ involves cardinality restrictions, it is straightforward to turn a model for $\mathcal{C}$ implied by condition (1) into a model implied by (2) (from Definition 44), and vice versa. This can be done simply by collapsing (resp. duplicating) individuals which realize the same type in the model. Formally, we demonstrate this by establishing a direct correspondence between both type of models. Let $\pi : S_\times \mapsto S$ be a surjective mapping, such that for every $t_c \in S_\times$, $\pi(t_c) = t_c$. Then $(S_\times, \cdot^{\mathcal{J}_\times})$ is a model implied by (1) *iff* $(S, \cdot^{\mathcal{J}})$ is a model implied by (2), provided that for every for every $t_c, t'_c \in S_\times$ the following conditions are satisfied:

- $t_c \in \boldsymbol{C}^{\mathcal{J}_\times}$ *iff* $\pi(t_c) \in \boldsymbol{C}^{\mathcal{J}}$,
- $\langle t_c, t'_c \rangle \in \boldsymbol{r}^{\mathcal{J}_\times}$ *iff* $\langle \pi(t_c), \tau(t'_c) \rangle \in \boldsymbol{r}^{\mathcal{J}}$.

By structural induction over constructs of $\mathcal{L}_C$ it is easy to find out that the models are bisimilar, and thus satisfy exactly the same formulas from $\mathcal{C}$. ❑

As a consequence of Definition 44 and Theorem 18, satisfiability of $\mathcal{C}$ can be reduced to the problem of finding a $\mathcal{C}$-admissible set of context types. The following theorems provide effectively verifiable, language-specific conditions for deciding whether a given set of context types is $\mathcal{C}$-admissible.

Theorem 19 (Deciding $\mathcal{C}$-admissibility in $\mathcal{SHIO}$)**.** *Let S be a set of context types for $\mathcal{K} = (\mathcal{C}, \mathcal{O})$, where $\mathcal{C}$ is expressed in $\mathcal{SHIO}$. Then, S is $\mathcal{C}$-admissible* iff *the following conditions are satisfied:*

1. *for every $\boldsymbol{C} \sqsubseteq \boldsymbol{D} \in \mathcal{C}$ and $t_c \in S$, if $\boldsymbol{C} \in t_c$ then $\boldsymbol{D} \in t_c$,*
2. *for every $\boldsymbol{a} \in obj_c(\mathcal{K})$ there is a unique $t_c \in S$ such that $\{\boldsymbol{a}\} \in t_c$,*
3. *for every $\exists \boldsymbol{s}.\boldsymbol{C} \in con_c(\mathcal{K})$ and $t_c \in S$ with $\exists \boldsymbol{s}.\boldsymbol{C} \in t_c$, there is $t'_c \in S$, such that $\boldsymbol{C} \in t'_c$ and t'_c is a matching $\boldsymbol{s}$-successor for t_c under $\mathcal{C}$.*

The conditions can be effectively verified in a time at most exponential in the size of $\mathcal{K}$.

Proof. First, we construct a $\mathcal{SHIO}$-model $(S, \cdot^{\mathcal{J}})$ for $\mathcal{C}$ implied by $\mathcal{C}$-admissibility of S, as follows. For every $t_c, t'_c \in S$:

- $\boldsymbol{a}^{\mathcal{J}} = t_c$ *iff* $\{\boldsymbol{a}\} \in t_c$, for every $\boldsymbol{a} \in obj_c(\mathcal{K})$,
- $t_c \in \boldsymbol{C}^{\mathcal{J}}$ *iff* $\boldsymbol{C} \in t_c$, for every $\boldsymbol{C} \in con_c(\mathcal{K})$,
- $\langle t_c, t'_c \rangle \in \boldsymbol{s}^{\mathcal{J}}$ *iff* t'_c is a matching $\boldsymbol{s}$-successor for t_c under $\mathcal{C}$, for every $\boldsymbol{s} \in rol_c(\mathcal{K})$.

($\Leftarrow$) We show that $(S, \cdot^{\mathcal{J}})$ is indeed a model for $\mathcal{C}$. Observe that the respective conditions in the theorem guarantee that:

1. all GCIs are satisfied,
2. all individual names (and so the nominals) are given unique interpretations,
3. all individuals satisfying existential restrictions obtain proper role successors, and moreover, by Def. 43, it is ensured that:
   - role names and their inverses are interpreted as relations which are inverses of each other,
   - transitive roles are interpreted as transitive relations,
   - the role hierarchies entailed by $\mathcal{C}$ are respected,
   - for every $t_c \in S$, and $\boldsymbol{r} \in rol_c(\mathcal{K})$ all concepts of the form $\neg\exists \boldsymbol{r}.\boldsymbol{C} \in t_c$ are satisfied in the model.

The first two points are clear by the construction of the model and the conditions 1 and 2 in the theorem. The third one follows from the construction of the model, definition of context type (DCT) and of matching metalanguage role successor (Def. 43). We proceed by induction. Consider any $t_c, t'_c \in S$, such that t'_c is a matching $\boldsymbol{s}$-successor for t_c under $\mathcal{C}$ for some $\boldsymbol{s} \in rol_c(\mathcal{K})$ at the top level of the role hierarchy. Then $\langle t_c, t'_c \rangle \in \boldsymbol{s}^{\mathcal{J}}$ and, by Def. 43, t_c has to be a matching $\boldsymbol{s}^-$-successor for t'_c under $\mathcal{C}$, and thus $\langle t'_c, t_c \rangle \in (\boldsymbol{s}^-)^{\mathcal{J}}$. In both cases all concepts of the form $\neg\exists \boldsymbol{s}.\boldsymbol{C} \in t_c$ and $\neg\exists \boldsymbol{s}^-.\boldsymbol{C} \in t'_c$ need to be satisfied. Also, by the construction of the model, it is ensured that for all $t_c \in S$ all concepts $\exists \boldsymbol{s}.\boldsymbol{C} \in t_c$ are satisfied as well. Further, suppose $\boldsymbol{s}$ is a transitive role. Then for every t''_c which is a matching $\boldsymbol{s}$-successor for t'_c under $\mathcal{C}$, t''_c has to be also a matching $\boldsymbol{s}$-successor for t_c under $\mathcal{C}$ and so $\langle t_c, t''_c \rangle \in \boldsymbol{s}^{\mathcal{J}}$, which inductively extends over the whole interpretation of $\boldsymbol{s}$, rendering it a transitive relation. In such case, Def. 43 guarantees that the model satisfies all $\neg\exists \boldsymbol{s}.\boldsymbol{C} \in t_c$ and $\neg\exists \boldsymbol{s}^-.\boldsymbol{C} \in t''_c$.

Now, suppose that for some role $\boldsymbol{r}$ there is $\boldsymbol{s} \sqsubseteq^*_{\mathcal{C}} \boldsymbol{r}$ and let t'_c be a matching $\boldsymbol{s}$-successor for t_c under $\mathcal{C}$, for some $t_c, t'_c \in S$. Then by Def. 43, t'_c must be also a matching $\boldsymbol{r}$-successor for t_c under $\mathcal{C}$, and so by the construction of the model $\langle t_c, t'_c \rangle \in \boldsymbol{r}^{\mathcal{J}}$ and $\langle t'_c, t_c \rangle \in (\boldsymbol{r}^-)^{\mathcal{J}}$, which fulfills the semantics of the role inclusion. Finally, suppose $\boldsymbol{s}$ is a transitive role and $\langle t_c, t'_c \rangle, \langle t'_c, t''_c \rangle \in \boldsymbol{s}^{\mathcal{J}}$, for some $t_c, t'_c, t''_c \in S$. Since, as argued above, t''_c must be also a matching

$\boldsymbol{s}$-successor for t_c under $\mathcal{C}$, it follows that $\langle t_c, t''_c\rangle \in \boldsymbol{s}^{\mathcal{J}}$. But then, by (DCT), for every concept of the form $\neg\exists\boldsymbol{r}.\boldsymbol{C} \in t_o$, there already is $\neg\exists\boldsymbol{s}.\boldsymbol{C} \in t_o$, and consequently, by transitivity of $\boldsymbol{s}$, also $\neg\exists\boldsymbol{s}.\boldsymbol{C} \in t'_o$. Therefore, it is also the case that t''_c is a matching $\boldsymbol{r}$-successor for t_c under $\mathcal{C}$ and $\langle t_c, t''_c\rangle \in \boldsymbol{r}^{\mathcal{J}}$. Clearly, all concepts of the form $\neg\exists\boldsymbol{r}.\boldsymbol{C} \in t_c$ and $\neg\exists\boldsymbol{r}^-.\boldsymbol{C} \in t''_c$ are satisfied in the model. By induction, the argument carries over to all roles in the hierarchy.

($\Rightarrow$) We demonstrate that $(S, \cdot^{\mathcal{J}})$, constructed as above, satisfies the conditions stated in the theorem. The first two are immediate. For the third one, suppose that for some $\exists\boldsymbol{s}.\boldsymbol{C} \in con_c(\mathcal{K})$ and $t_c \in S$ there is $\exists\boldsymbol{s}.\boldsymbol{C} \in t_c$. Clearly, by the semantics, there must be a $t'_c \in S$, such that $\langle t_c, t'_c\rangle \in \boldsymbol{s}^{\mathcal{J}}$ and $t'_c \in \boldsymbol{C}^{\mathcal{J}}$, and thus with $\boldsymbol{C} \in t'_c$. We show that in such case t'_c is a matching $\boldsymbol{s}$-successor for t_c under $\mathcal{C}$, i.e. that the three conditions in Def. 43 are satisfied. The first one is obvious. For the second one, suppose that $\boldsymbol{s}$ is transitive and some $\neg\exists\boldsymbol{s}.\boldsymbol{D}$ is satisfied in t_c. Then it must be the case that either t'_c has no $\boldsymbol{s}$-successors in the model (in such case $\neg\exists\boldsymbol{s}.\boldsymbol{D}$ is vacuously satisfied in t'_c) or it has some $\boldsymbol{s}$-successors. In the latter case, by transitivity of $\boldsymbol{s}$, such successors have to satisfy all $\boldsymbol{D}$ such that $t_c \in (\neg\exists\boldsymbol{s}.\boldsymbol{D})^{\mathcal{J}}$. It follows that all such $\neg\exists\boldsymbol{s}.\boldsymbol{D}$ have to be satisfied also in t'_c, and so the condition holds. Finally, by induction over the role hierarchy, t'_c is clearly a matching $\boldsymbol{r}$-successor for t_c under $\mathcal{C}$, for all $\boldsymbol{r}$ such that $\boldsymbol{s} \sqsubseteq^*_{\mathcal{C}} \boldsymbol{r}$.

Observe that the size of $\sqsubseteq^*_{\mathcal{C}}$ is at most polynomial in $\ell(\mathcal{K})$, while $|S| \leq 2^{\ell(\mathcal{K})}$ and $|t_c| \leq \ell(\mathcal{K})$ for every $t_c \in S$ (see also the proof of Lemma 3). Thus, deciding the conditions specified in the theorem cannot take more than a polynomial time in the size of S and, exponential in $\ell(\mathcal{K})$. ❑

In order to formulate a similar claim for $\mathcal{L}_C = \mathcal{EL}^{++}$ we require some additional notation. We write $\mathcal{C} \vdash \boldsymbol{r} \sqsubseteq \boldsymbol{s}$ *iff* $\boldsymbol{r} = \boldsymbol{s}$ or $\mathcal{C}$ contains role inclusions $\boldsymbol{r}_1 \sqsubseteq \boldsymbol{r}_2, \ldots, \boldsymbol{r}_{n-1} \sqsubseteq \boldsymbol{r}_n$ with $\boldsymbol{r}_1 = \boldsymbol{r}$ and $\boldsymbol{r}_n = \boldsymbol{s}$. Further, we write $\mathcal{C} \vdash \mathrm{ran}(\boldsymbol{r}) \sqsubseteq \boldsymbol{C}$ if there is a role name $\boldsymbol{s}$ with $\mathcal{C} \vdash \boldsymbol{r} \sqsubseteq \boldsymbol{s}$ and $\mathrm{ran}(\boldsymbol{s}) \sqsubseteq \boldsymbol{C} \in \mathcal{C}$.

Let $X \subseteq con_c(\mathcal{K})$. Then by $X^{\sqsubseteq}_{\mathcal{C}}$ we denote the closure of X under subsumption in $\mathcal{C}$ w.r.t. $con_c(\mathcal{K})$, i.e.:

- $X \subseteq X^{\sqsubseteq}_{\mathcal{C}}$,
- if $\boldsymbol{C}, \boldsymbol{D} \in X^{\sqsubseteq}_{\mathcal{C}}$ then $\boldsymbol{C} \sqcap \boldsymbol{D} \in X^{\sqsubseteq}_{\mathcal{C}}$, for every $\boldsymbol{C} \sqcap \boldsymbol{D} \in con_c(\mathcal{K})$,
- for every $\boldsymbol{C} \in X^{\sqsubseteq}_{\mathcal{C}}$ and $\boldsymbol{D} \in con_c(\mathcal{K})$, if $\mathcal{C} \models \boldsymbol{C} \sqsubseteq \boldsymbol{D}$ then $\boldsymbol{D} \in X^{\sqsubseteq}_{\mathcal{C}}$.

Since the subsumption problem in $\mathcal{EL}^{++}$ is tractable [BBL08], it is clear that $X^{\sqsubseteq}_{\mathcal{C}}$ can be computed in a polynomial time. By an abuse of notation we write $\boldsymbol{C}^{\sqsubseteq}_{\mathcal{C}}$, whenever $X = \{\boldsymbol{C}\}$ for any $\boldsymbol{C}$.

Algorithm 1 computes a $\mathcal{C}$-admissible set $S_{\mathcal{C},\Omega}$ of context types for $\mathcal{K}$, provided such set exists at all, and its subset $U_{\mathcal{C},\Omega}$. The subscript $\Omega \subseteq con_c(\mathcal{K})$ denotes an extra set of concepts which must be also satisfied in the model corresponding to $S_{\mathcal{C},\Omega}$. This parameter and the set $U_{\mathcal{C},\Omega}$ are necessary later on, when satisfiability of the whole knowledge base $\mathcal{K}$ is considered. In the special case, for $\Omega = \emptyset$, $S_{\mathcal{C},\Omega}$ corresponds exactly to the canonical model of $\mathcal{C}$. That is, for every $\mathcal{EL}^{++}$ concept $\boldsymbol{C}$, there exists $Y \in S_{\mathcal{C},\Omega}$ with $\boldsymbol{C} \in Y$ *iff* $\boldsymbol{C}$ is satisfied in every model of $\mathcal{C}$, provided such models exist. This dramatically reduces the search space for $\mathcal{C}$-admissible sets of context types for $\mathcal{M} = \mathcal{EL}^{++}$, which paves the way to the EXPTIME upper bound for $\mathfrak{C}^{\mathcal{EL}^{++}}_{\mathcal{SHI}}$.

Theorem 20 (Semi-deciding $\mathcal{C}$-admissibility in $\mathcal{EL}^{++}$)**.** *Let $\mathcal{K} = (\mathcal{C}, \mathcal{O})$ be a knowledge base, where $\mathcal{C}$ is expressed in $\mathcal{EL}^{++}$, and S be a set of context types for $\mathcal{K}$. Then S is $\mathcal{C}$-admissible if the following conditions are satisfied:*

1. *$S = S_{\mathcal{C},\Omega}$, where $S_{\mathcal{C},\Omega}$ is computed by Algorithm 1 for some $\Omega \subseteq \mathrm{con}_c(\mathcal{K})$,*
2. *$S_{\mathcal{C},\Omega}$ is non-empty.*

For a fixed Ω, the algorithm runs in a time polynomial in the size of $\mathcal{K}$.

Proof. Suppose $S_{\mathcal{C},\Omega}$ is non-empty. Then clearly, every element of $S_{\mathcal{C},\Omega}$ is a context type for $\mathcal{K}$. We construct an $\mathcal{EL}^{++}$-model $(S_{\mathcal{C},\Omega}, \cdot^{\mathcal{J}})$ for $\mathcal{C}$, implied by $\mathcal{C}$-admissibility of $S_{\mathcal{C},\Omega}$, as follows. For every $t_c, t'_c \in S$ fix:

- $\boldsymbol{a}^{\mathcal{J}} = t_c$ *iff* $\{\boldsymbol{a}\} \in t_c$, for every $\boldsymbol{a} \in obj_c(\mathcal{K})$,
- $t_c \in \boldsymbol{C}^{\mathcal{J}}$ *iff* $\boldsymbol{C} \in t_c$, for every $\boldsymbol{C} \in con_c(\mathcal{K})$,
- $\langle t_c, t'_c\rangle \in \boldsymbol{s}^{\mathcal{J}}$ *iff* $\{\boldsymbol{D} \mid \mathcal{C} \vdash \mathrm{ran}(\boldsymbol{s}) \sqsubseteq \boldsymbol{D}\} \subseteq t'_c$.

Extend $\cdot^{\mathcal{J}}$ inductively over all roles by ensuring that for every $t_c, t'_c, t''_c \in S$:

- if $\langle t_c, t'_c\rangle \in \boldsymbol{r}$ and $\langle t'_c, t''_c\rangle \in \boldsymbol{s}$ then $\langle t_c, t''_c\rangle \in \boldsymbol{r} \circ \boldsymbol{s}$, for every $\boldsymbol{r} \circ \boldsymbol{s} \in rol_c(\mathcal{K})$,
- if $\langle t_c, t'_c\rangle \in (\boldsymbol{r}_1 \circ \ldots \circ \boldsymbol{r}_n)^{\mathcal{J}}$ then $\langle t_c, t'_c\rangle \in \boldsymbol{s}^{\mathcal{J}}$, for every $\boldsymbol{r}_1 \circ \ldots \circ \boldsymbol{r}_n \sqsubseteq \boldsymbol{s} \in \mathcal{C}$.

It is not hard to verify, that $(S_{\mathcal{C},\Omega}, \cdot^{\mathcal{J}})$ defined in this way is indeed a model for $\mathcal{C}$. In particular, by the construction of the model and definition of the algorithm, it is guaranteed that all GCIs are satisfied (by closure of the generated types under $\sqsubseteq$ in $\mathcal{C}$), individual names obtain unique interpretations (by merging types containing the same nominals) and that all individuals satisfying existential restrictions obtain proper successors. The only issue requiring more

Algorithm 1 Computation of a set of context types for $\mathcal{K}$ in $\mathcal{EL}^{++}$.

Require: (context) ontology $\mathcal{C}$, a set of concepts $\Omega \subseteq con_c(\mathcal{K})$
Ensure: two sets of context types $S_{\mathcal{C},\Omega}$ and $U_{\mathcal{C},\Omega}$

$S := \emptyset$, $U := \emptyset$, $Marked := \emptyset$
if $obj_c(\mathcal{K}) = \emptyset$ and $\Omega = \emptyset$ **then**
  add $\top^{\sqsubseteq}_{\mathcal{C}}$ to S
else
  for all $\boldsymbol{a} \in obj_c(\mathcal{K})$ **do**
    add $\{\boldsymbol{a}\}^{\sqsubseteq}_{\mathcal{C}}$ to S
  end for
  for all $\boldsymbol{C} \in \Omega$ **do**
    add $\boldsymbol{C}^{\sqsubseteq}_{\mathcal{C}}$ to S and to U
  end for
end if
while applicable **do**
  for all $Y \in S$ and $\exists \boldsymbol{s}.\boldsymbol{C} \in Y$ **do**
    if $\exists \boldsymbol{s}.\boldsymbol{C} \notin Marked$ **then**
      add $(\{\boldsymbol{C}\} \cup \{\boldsymbol{D} \mid \mathcal{C} \vdash \mathrm{ran}(\boldsymbol{s}) \sqsubseteq \boldsymbol{D}\})^{\sqsubseteq}_{\mathcal{C}}$ to S and add $\exists \boldsymbol{s}.\boldsymbol{C}$ to $Marked$
    end if
  end for
  for all $\boldsymbol{a} \in obj_c(\mathcal{K})$ and $Y, Z \in S$ **do**
    if $\{\boldsymbol{a}\} \in Y \cap Z$ **then**
      replace Y and Z in S with $(Y \cup Z)^{\sqsubseteq}_{\mathcal{C}}$
    end if
    if $Y \in U$ **or** $Z \in U$ **then**
      remove Y, Z from U and add $(Y \cup Z)^{\sqsubseteq}_{\mathcal{C}}$ to U
    end if
  end for
end while
if $\bot \notin Y$ for every $Y \in S$ **then**
  $S_{\mathcal{C},\Omega} := S$ and $U_{\mathcal{C},\Omega}$
else
  $S_{\mathcal{C},\Omega} := \emptyset$ and $U_{\mathcal{C},\Omega} = \emptyset$
end if

attention is the satisfaction of role ranges. Clearly, the ranges of roles included in existential restrictions are respected by the definition of the algorithm. For the inductive extension of $\cdot^{\mathcal{J}}$ over the remaining roles, we resort to the syntactic restriction permitting tractable reasoning in $\mathcal{EL}^{++}$, which has been identified in [BBL08]. The restriction states:

If $\boldsymbol{r}_1 \circ \ldots \circ \boldsymbol{r}_n \sqsubseteq \boldsymbol{s} \in \mathcal{C}$ with $n \geq 1$ and $\mathcal{C} \vdash \text{ran}(\boldsymbol{s}) \sqsubseteq \boldsymbol{D}$, then $\mathcal{C} \vdash \text{ran}(\boldsymbol{r}_n) \sqsubseteq \boldsymbol{D}$.

It immediately follows, that whenever $\langle t_c, t'_c \rangle \in \boldsymbol{s}^{\mathcal{J}}$ is included in $\cdot^{\mathcal{J}}$, for any $t_c, t'_c \in S_{\mathcal{C}}$, because of some role inclusion $\boldsymbol{r}_1 \circ \ldots \circ \boldsymbol{r}_n \sqsubseteq \boldsymbol{s} \in \mathcal{C}$, it is the case, that $\{\boldsymbol{D} \mid \mathcal{C} \vdash \text{ran}(\boldsymbol{s}) \sqsubseteq \boldsymbol{D}\} \subseteq t'_c$ if $\{\boldsymbol{D} \mid \mathcal{C} \vdash \text{ran}(\boldsymbol{r}_n) \sqsubseteq \boldsymbol{D}\} \subseteq t'_c$. But then, by induction, it is easy to see that the appropriate range restrictions are carried over from the roles occurring in some existential restrictions, which are sufficiently handled by the algorithm.

Observe, that the number of distinct concepts of the form $\exists \boldsymbol{r}.\boldsymbol{C}$ occurring in $S_{\mathcal{C},\Omega}$ is linearly bounded by the size of $\mathcal{C}$, and thus, computing $S_{\mathcal{C},\Omega}$ must terminate in a time polynomial in the size of $\mathcal{K}$. ❑

Theorem 21 (Satisfiability as $\mathcal{C}$-admissibility in $\mathcal{EL}^{++}$)**.** *Let $\mathcal{K} = (\mathcal{C}, \mathcal{O})$ be a knowledge base, where $\mathcal{C}$ is expressed in $\mathcal{EL}^{++}$, and let $\Omega \subseteq con_c(\mathcal{K})$. Then $\mathcal{C}$ is satisfied in some model which also satisfies every $\boldsymbol{C} \in \Omega$* iff *$S_{\mathcal{C},\Omega}$, computed by Algorithm 1, is non-empty.*

Proof. ($\Rightarrow$) Let $(\mathfrak{C}, \cdot^{\mathcal{J}})$ be a model of $\mathcal{C}$ satisfying every $\boldsymbol{C} \in \Omega$. Define a mapping $\tau : \mathfrak{C} \mapsto \Xi$, such that for every $c \in \mathfrak{C}$ and $\boldsymbol{C} \in con_c(\mathcal{K})$, $\boldsymbol{C} \in \tau(c)$ *iff* $c \in \boldsymbol{C}^{\mathcal{J}}$. Now, let $S = \{\tau(c) \mid c \in \mathfrak{C}\}$. Observe, that the algorithm generating the context types from $S_{\mathcal{C},\Omega}$ is deterministic and enforces only the necessary consequences of $\mathcal{C}$ and the semantics of $\mathcal{EL}^{++}$. Hence, for every $t_c \in S_{\mathcal{C},\Omega}$ there must be some $t'_c \in S$, such that $t_c \subseteq t'_c$. Obviously, no $t_c \in S$ contains $\perp$. Thus, whenever $\mathcal{C}$ has a model satisfying all concepts from Ω, there has to exist a non-empty output from the algorithm.

($\Leftarrow$) Suppose $S_{\mathcal{C},\Omega}$ is non-empty. By the construction of $S_{\mathcal{C},\Omega}$, for every $\boldsymbol{C} \in \Omega$ there exists a type $t_c \in S_{\mathcal{C},\Omega}$, such that $\boldsymbol{C} \in t_c$. Thus, by Algorithm 1, Theorem 20 and Definition 44 there has to exist a model of $\mathcal{C}$ satisfying every $\boldsymbol{C} \in \Omega$. ❑

A *context structure* $\langle S, \mathfrak{S} \rangle$ for $\mathcal{K}$ is a pair consisting of a set $S \subseteq \Xi$ of context types for $\mathcal{K}$ and a non-empty set $\mathfrak{S}$ of tuples of the form $\langle t_c, f, \nu \rangle$, where $t_c \in S$, $f \subseteq sub_o(\mathcal{K})$ is an object formula type for $\mathcal{K}$, $\nu : obj_o(\mathcal{K}) \mapsto \Pi$ assigns unique object types to individual object names, and such that the following conditions are satisfied:

(CS1) for every $\boldsymbol{a} \in obj_c(\mathcal{K})$ there is a unique $t_c \in S$ such that $\{\boldsymbol{a}\} \in t_c$, and at most one $\langle t_c, f, \nu\rangle \in \mathfrak{S}$. If $\boldsymbol{a} : \varphi \in \mathcal{O}$, for any φ, then such $\langle t_c, f, \nu\rangle \in \mathfrak{S}$ must exist,

(CS2) S is $\mathcal{C}$-admissible,

(CS3) for every $\langle t_c, f, \nu\rangle \in \mathfrak{S}$ and $\boldsymbol{C} : \varphi \in \mathcal{O}$, if $\boldsymbol{C} \in t_c$ then $\varphi \in f$,

In the case of languages with full object formulas, the following requirement has to be also satisfied:

(CS4) for every $\langle t_c, f, \nu\rangle \in \mathfrak{S}$ it holds that:

- if $\boldsymbol{C} \in t_c$ and $\varphi \in f$ then $\langle \boldsymbol{C}\rangle\varphi \in f$, for every $\langle \boldsymbol{C}\rangle\varphi \in sub_o(\mathcal{K})$,
- for every $\langle \boldsymbol{C}\rangle\varphi \in f$ there is $\langle t'_c, f', \nu'\rangle \in \mathfrak{S}$, such that $\boldsymbol{C} \in t'_c$ and $\varphi \in f'$,
- for every $\neg\langle \boldsymbol{C}\rangle\neg\varphi \in f$ and $\langle t'_c, f', \nu'\rangle \in \mathfrak{S}$, if $\boldsymbol{C} \in t'_c$ then $\varphi \in f'$.

Intuitively, a context structure contains all the pieces necessary for reconstructing a single $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-interpretation. However, not all such interpretations might correspond to a genuine $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{L}_O}$-model. To find exactly the proper ones, some additional conditions need to be imposed. These are introduced in the notion of quasimodel candidate, and further, in the notions of quasimodel associated with specific logics under consideration.

Definition 45 (Quasimodel candidate)**.** *A* quasimodel candidate $\mathfrak{Q}^S_{\mathfrak{S}}$ *for $\mathcal{K}$, where $\langle S, \mathfrak{S}\rangle$ is a context structure for $\mathcal{K}$, is a set of pairs $\langle k, t_o\rangle$, such that $k \in \mathfrak{S}$, $t_o \in \Pi$, satisfying the following conditions:*

(QC1) for every $k \in \mathfrak{S}$, with $k = \langle t_c, f, \nu\rangle$, and $a \in obj_o(\mathcal{K})$, $\langle k, \nu(a)\rangle \in \mathfrak{Q}^S_{\mathfrak{S}}$.

For every $\langle k, t_o\rangle \in \mathfrak{Q}^S_{\mathfrak{S}}$, with $k = \langle t_c, f, \nu\rangle$:

(QC2) if $\neg\langle \boldsymbol{C}\rangle\neg D \in t_o$ and $\boldsymbol{C} \in t_c$ then $D \in t_o$, for all $\neg\langle \boldsymbol{C}\rangle\neg D \in con_o(\mathcal{K})$,

(QC3) if $\boldsymbol{C} \in t_c$ and $D \in t_o$ then $\langle \boldsymbol{C}\rangle D \in t_o$, for all $\langle \boldsymbol{C}\rangle D \in con_o(\mathcal{K})$,

(QC4) for every $k' \in \mathfrak{S}$, there is some $\langle k', t'_o\rangle \in \mathfrak{Q}^S_{\mathfrak{S}}$ such that t_o, t'_o are matching **S5**-successors,

(QC5) if $\langle \boldsymbol{C}\rangle D \in t_o$ then there is $\langle k', t'_o\rangle \in \mathfrak{Q}^S_{\mathfrak{S}}$, such that $k' = \langle t'_c, f', \nu'\rangle$,$\boldsymbol{C} \in t'_c$, $D \in t'_o$ and t_o, t'_o are matching **S5**-successors. Moreover, if $t_o \neq t'_o$ then $k' \neq k$,

(QC6) for every $\exists r.C \in t_o$ there is $\langle k, t'_o \rangle \in \mathfrak{Q}^S_{\mathfrak{S}}$, such that $C \in t'_o$ and t'_o is a matching r-successor for t_o under f.

Lemma 3 (Quasimodel candidate space bound)**.** *The size of a quasimodel candidate is exponentially bounded in the size of $\mathcal{K}$.*

Proof. By $\ell(\mathcal{K})$ we denote the size of $\mathcal{K}$, measured in the number of symbols used, and by $|X|$ — the number of elements of set X. We observe that the following (very liberally estimated) inequalities hold:

$$|con_c(\mathcal{K})| \leq 2\ell(\mathcal{K}), \ \ |con_o(\mathcal{K})| \leq 2\ell(\mathcal{K}), \ \ |sub_o(\mathcal{K})| \leq 2\ell(\mathcal{K}), \ \ |obj_o(\mathcal{K})| \leq \ell(\mathcal{K})$$
$$|\Pi| \leq 2^{con_o(\mathcal{K})} \leq 2^{2\ell(\mathcal{K})}, \quad |\Xi| \leq 2^{con_c(\mathcal{K})} \leq 2^{2\ell(\mathcal{K})}, \quad |\Phi| \leq 2^{sub_o(\mathcal{K})} \leq 2^{2\ell(\mathcal{K})}$$
$$|\Pi^{|obj_o(\mathcal{K})|}| = |\Pi|^{|obj_o(\mathcal{K})|} \leq 2^{2\ell(\mathcal{K})^2}$$
$$|\mathfrak{S}| \leq |\Xi| \cdot |\Phi| \cdot |\Pi^{|obj_o(\mathcal{K})|}| \leq 2^{2\ell(\mathcal{K})^2+4\ell(\mathcal{K})}$$
$$|\mathfrak{Q}^S_{\mathfrak{S}}| \leq |\mathfrak{S}| \cdot |\Pi| \leq 2^{2\ell(\mathcal{K})^2+6\ell(\mathcal{K})}$$

Since the maximum size of a single tuple in a quasimodel candidate is polynomial in $\ell(\mathcal{K})$ therefore the maximum size of a quasimodel is never greater than $2^{p(\ell(\mathcal{K}))}$, where p is a fixed polynomial. ❑

The structure of the proofs:

1. definition of a quasimodel
2. the quasimodel lemma
3. an algorithm with a specified time resource bound

Theorem 22. *Knowledge base satisfiability in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{SHIO}}$, for $\mathcal{L}_C \in \{\mathcal{SHIO}, \mathcal{EL}^{++}\}$, with full object formulas and only local roles is in* NEXPTIME*.*

Proof. We begin by defining the relevant notion of quasimodel.

Definition 46 (Quasimodel)**.** *A quasimodel candidate $\mathfrak{Q}^S_{\mathfrak{S}}$ for $\mathcal{K}$, where $\mathcal{K}$ is a knowledge base in $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{SHIO}}$, for $\mathcal{L}_C \in \{\mathcal{SHIO}, \mathcal{EL}^{++}\}$, with full object formulas and only local roles, is called a* quasimodel *for $\mathcal{K}$* iff *the following conditions are satisfied:*

(QM1) for every $\langle k, t_o \rangle \in \mathfrak{Q}^S_{\mathfrak{S}}$, with $k = \langle t_c, f, \nu \rangle$, and $a \in obj_o(\mathcal{K})$, $\{a\} \in t_o$ *iff* $t_o = \nu(a)$.

For every $k \in \mathfrak{S}$ with $k = \langle t_c, f, \nu \rangle$:

(QM2) $C \sqsubseteq D \in f$ *iff* for every $\langle k, t_o \rangle \in \mathfrak{Q}^S_{\mathfrak{S}}$ if $C \in t_o$ then $D \in t_o$, for every $C \sqsubseteq D \in sub_o(\mathcal{K})$.

(QM3) $\neg(C \sqsubseteq D) \in f$ *iff* there is $\langle k, t_o \rangle \in \mathfrak{Q}^S_{\mathfrak{S}}$ such that $\{C, \neg D\} \subseteq t_o$, for every $\neg(C \sqsubseteq D) \in sub_o(\mathcal{K})$,

(QM4) $\neg(r \sqsubseteq s) \in f$ *iff* $r \not\sqsubseteq^*_f s$ and there is $\langle k, t_o \rangle, \langle k, t'_o \rangle \in \mathfrak{Q}^S_{\mathfrak{S}}$ such that t'_o is a matching r-successor for t_o under f, for every $\neg(r \sqsubseteq s) \in sub_o(\mathcal{K})$.

Next, we show the correspondence between quasimodels and models.

Lemma 4 (Quasimodel lemma)**.** *A knowledge base $\mathcal{K}$ has an $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{SHIO}}$-model, for $\mathcal{L}_C \in \{\mathcal{SHIO}, \mathcal{EL}^{++}\}$,* iff *there is a quasimodel for $\mathcal{K}$.*

Proof. ($\Rightarrow$) Let $\mathfrak{M} = (\Theta, \mathfrak{C}, \cdot^{\mathcal{J}}, \Delta, \{\cdot^{\mathcal{I}(c)}\}_{c \in \mathfrak{C}})$ be a model for $\mathcal{K}$. First, we fix a mapping τ from Θ, $\mathfrak{C}$ and Δ to the corresponding context/object types. For every $c \in \Theta$ set $\tau(c) = t_c$, such that:

- $\boldsymbol{C} \in t_c$ *iff* $c \in \boldsymbol{C}^{\mathcal{J}}$, for every $\boldsymbol{C} \in con_c(\mathcal{K})$,

For every $\langle c, d \rangle \in \mathfrak{C} \times \Delta$ set $\tau(\langle c, d \rangle) = t_o$, such that:

- $C \in t_o$ *iff* $d \in C^{\mathcal{I}(c)}$, for every $C \in con_o(\mathcal{K})$.

Further, for every $c \in \mathfrak{C}$ set $\tau(c) = \langle t_c, f, \nu \rangle$, such that the following correspondences hold:

- $\boldsymbol{C} \in t_c$ *iff* $c \in \boldsymbol{C}^{\mathcal{J}}$, for every $\boldsymbol{C} \in con_c(\mathcal{K})$,
- $\varphi \in f$ *iff* $\mathfrak{M}, c \models \varphi$, for every $\varphi \in sub_o(\mathcal{K})$,
- $\nu(a) = \tau(\langle c, a^{\mathcal{I}(c)} \rangle)$ for every $a \in obj_o(\mathcal{K})$.

Fix $S = \{\tau(c) \mid c \in \Theta\}$ and $\mathfrak{S} = \{\tau(c) \mid c \in \mathfrak{C}\}$. By Theorem 18, $\langle S, \mathfrak{S} \rangle$ is a proper context structure satisfying all conditions (**CS**x). Next, define the quasimodel $\mathfrak{Q}^S_{\mathfrak{S}} = \{\langle \tau(c), \tau(\langle c, d \rangle) \rangle \mid \langle c, d \rangle \in \mathfrak{C} \times \Delta, \}$. It is easy to see, that all conditions (**QC1**)-(**QC5**) and (**QM1**)-(**QM3**) have to be satisfied. Since the notion of matching role successor is exactly the same for the object and meta-language roles in $\mathcal{SHIO}$, the satisfaction of (**QC6**) can be demonstrated by the same argument as used in the proof of Theorem 19. Finally, for (**QM4**), observe that whenever $c \models \neg(r \sqsubseteq s)$ holds in $c \in \mathfrak{C}$, then there have to be $d, d' \in \Delta$, such that $\langle d, d' \rangle \in r^{\mathcal{J}}$ and $\langle d, d' \rangle \notin s^{\mathcal{J}}$ and so that the condition (**QM4**) has to be satisfied in $\mathfrak{Q}^S_{\mathfrak{S}}$ defined as above.

($\Leftarrow$) Let $\mathfrak{Q}^S_{\mathfrak{S}}$ be a quasimodel for $\mathcal{K}$. In the following steps we define a model $\mathfrak{M} = (\Theta, \mathfrak{C}, \cdot^{\mathcal{J}}, \Delta, \{\cdot^{\mathcal{I}(c)}\}_{c \in \mathfrak{C}})$ for $\mathcal{K}$. The interpretation of the context dimension follows immediately from the definition of the context structure. Since $\langle S, \mathfrak{S}\rangle$ is a context structure, then S must be a $\mathcal{C}$-admissible set of context types. But then, by Theorem 18, for any multiset $S_\times$ such that $S \subseteq S_\times$, there must be some interpretation function $\cdot^{\mathcal{J}}$ such that $(S_\times, \cdot^{\mathcal{J}})$ is a model for $\mathcal{C}$. Fix $\mathfrak{C} = \{k^{t_o} \mid \langle k, t_o\rangle \in \mathfrak{Q}^S_{\mathfrak{S}}\}$ and $\Theta = \{t^k_c \mid k = \langle t_c, f, \nu\rangle \in \mathfrak{C}\} \cup \{t_c \in S \mid \langle t_c, f, \nu\rangle \notin \mathfrak{C}$ for any $f, \nu\}$. Then clearly, $(\Theta, \cdot^{\mathcal{J}})$ is also a model for $\mathcal{C}$. The tuple $(\Theta, \mathfrak{C}, \cdot^{\mathcal{J}})$ is incorporated into $\mathfrak{M}$.

Now, consider the object dimension. For every $k \in \mathfrak{C}$, we fix the set of object types $T_k = \{t_o \mid \langle k, t_o\rangle \in \mathfrak{Q}^S_{\mathfrak{S}}\}$ realized in this context. A *run* ρ through $\mathfrak{Q}^S_{\mathfrak{S}}$ is a function which to every $k \in \mathfrak{C}$ assigns a single type from T_k, such that:

- for every $k, k' \in \mathfrak{C}$ it is the case that $\rho(k), \rho(k')$ are matching **S5**-successors,
- for every $k \in \mathfrak{C}$, if $\langle \boldsymbol{C}\rangle D \in \rho(k)$ then there is $k' \in \mathfrak{C}$, such that $k' = \langle t'_c, f', \nu'\rangle$, $\boldsymbol{C} \in t'_c$ and $D \in \rho(k')'$.

A set $\mathfrak{R}$ of runs through $\mathfrak{Q}^S_{\mathfrak{S}}$ is called *coherent iff* the following conditions are satisfied:

- for every $k \in \mathfrak{C}$ and $t_o \in T_k$, there is a $\rho \in \mathfrak{R}$ such that $\rho(k) = t_o$,
- for every $a \in obj_o(\mathcal{K})$ and $k \in \mathfrak{C}$, with $k = \langle t_o, f, \nu\rangle$, there is a unique $\rho \in \mathfrak{R}$, such that $\rho(k) = \nu(a)$,

Next, we define the interpretation of the object dimension as follows. First, fix the object domain as:

- $\Delta := \mathfrak{R}$

Then, for every $k \in \mathfrak{C}$, with $k = \langle t_c, f, \nu\rangle$, and $\rho, \rho' \in \Delta$ set the interpretation function as:

- $a^{\mathcal{I}(k)} = \rho$ *iff* $\nu(a) = \rho(k)$, for every $a \in obj_o(\mathcal{K})$,
- $\rho \in C^{\mathcal{I}(k)}$ *iff* $C \in \rho(k)$, for every $C \in con_o(\mathcal{K})$,
- $\langle \rho, \rho'\rangle \in r^{\mathcal{I}(k)}$ *iff* $\rho'(k)$ is a matching r-successor for $\rho(k)$ under f.

As all the conditions (**CS**x), (**QC**x) and (**QM**x) are satisfied by the assumption, it is not difficult to verify that $\mathfrak{M} = (\Theta, \mathfrak{C}, \cdot^{\mathcal{J}}, \Delta, \{\cdot^{\mathcal{I}(c)}\}_{c\in\mathfrak{C}})$, defined as above, is indeed an $\mathfrak{C}^{\mathcal{L}_C}_{\mathcal{SHIO}}$-model for $\mathcal{K}$, for $\mathcal{L}_C \in \{\mathcal{SHIO}, \mathcal{EL}^{++}\}$. Again for the interpretation of the roles (including inverses and transitive roles) and satisfaction of the role hierarchy we apply the same argument as used for the case of the context ontology in the proof of Theorem 19. ❑

By Lemmas 3 and 4, the simplest brute-force NEXPTIME algorithm for checking satisfiability of $\mathcal{K}$ first guesses a quasimodel and then checks whether all conditions (**CS**x), (**QC**x) and(**QM**x) are satisfied. Clearly, such a check can be accomplished in a polynomial time in the size of the quasimodel, and thus in at most an exponential time in the size of $\mathcal{K}$. ❑

Theorem 23. *Knowledge base satisfiability in $\mathfrak{C}^{\mathcal{EL}^{++}}_{\mathcal{SHI}}$, with only local roles, is in* EXPTIME.

Proof. Again, we start by defining the relevant notion of quasimodel.

Definition 47 (Quasimodel)**.** *A quasimodel candidate $\mathfrak{Q}^S_{\mathfrak{S}}$ for $\mathcal{K}$, where $\mathcal{K}$ is a knowledge base in $\mathfrak{C}^{\mathcal{EL}^{++}}_{\mathcal{SHI}}$, with only local roles, is called a* quasimodel iff *it satisfies the following conditions:*

(QM1) for every $k, k' \in \mathfrak{S}$, with $k = \langle t_c, f, \nu \rangle$ and $k' = \langle t'_c, f', \nu' \rangle$, and $t_o \in \Pi$, whenever $t_c = t'_c$ then $\langle k, t_o \rangle \in \mathfrak{Q}^S_{\mathfrak{S}}$ *iff* $\langle k', t_o \rangle \in \mathfrak{Q}^S_{\mathfrak{S}}$,

(QM2) for every $\langle k, t_o \rangle \in \mathfrak{Q}^S_{\mathfrak{S}}$, with $k = \langle t_c, f, \nu \rangle$, and $C \sqsubseteq D \in f$, if $C \in t_o$ then $D \in t_o$.

For every $k \in \mathfrak{S}$ with $k = \langle t_c, f, \nu \rangle$:

(QM3) if $a : C \in f$ then $C \in \nu(a)$,

(QM4) if $r(a, b) \in f$ then $\nu(b)$ is a matching r-successor for $\nu(a)$ under f,

(QM5) (**rigid object names**) for every $a \in obj_o(\mathcal{K})$ and $k' \in \mathfrak{S}$ with $k' = \langle t'_c, f', \nu' \rangle$, $\nu(a)$ and $\nu'(a)$ are matching **S5**-successors,

(QM6) (**rigid object names**) for every $a \in obj_o(\mathcal{K})$ and $\langle \boldsymbol{C} \rangle D \in \nu(a)$ there is $\langle t'_c, f', \nu' \rangle \in \mathfrak{S}$, such that $\boldsymbol{C} \in t'_c$, $D \in \nu'(a)$.

Next, we prove the corresponding quasimodel lemma.

Lemma 5 (Quasimodel lemma)**.** *A knowledge base $\mathcal{K}$ has an $\mathfrak{C}^{\mathcal{EL}^{++}}_{\mathcal{SHI}}$-model* iff *there is a quasimodel for $\mathcal{K}$.*

Proof. The proof is slightly more involved than in the case of Lemma 4, as here we need to restrict the space of possible quasimodels only to those built over the minimal context structures, generated by Algorithm 1.

(⇒) Let $\mathfrak{M} = (\Theta, \mathfrak{C}, \cdot^{\mathcal{J}}, \Delta, \{\cdot^{\mathcal{I}(c)}\}_{c\in\mathfrak{C}})$ be a model for $\mathcal{K}$. As in the proof of Lemma 4, we first fix a mapping τ from $\mathfrak{C}$ and Δ to the corresponding context/object types. For every $c \in \Theta$ set $\tau(c) = t_c$, such that:

- $\boldsymbol{C} \in t_c$ *iff* $c \in \boldsymbol{C}^{\mathcal{J}}$, for every $\boldsymbol{C} \in con_c(\mathcal{K})$,

For every $\langle c, d\rangle \in \mathfrak{C} \times \Delta$ set $\tau(\langle c, d\rangle) = t_o$, such that:

- $C \in t_o$ *iff* $d \in C^{\mathcal{I}(c)}$, for every $C \in con_o(\mathcal{K})$.

Further, for every $c \in \mathfrak{C}$ set $\tau(c) = \langle t_c, f, \nu\rangle$, such that the following correspondences hold:

- $\boldsymbol{C} \in t_c$ *iff* $c \in \boldsymbol{C}^{\mathcal{J}}$, for every $\boldsymbol{C} \in con_c(\mathcal{K})$,
- $\varphi \in f$ *iff* $\mathfrak{M}, c \models \varphi$, for every $\varphi \in sub_o(\mathcal{K})$,
- $\nu(a) = \tau(\langle c, a^{\mathcal{I}(c)}\rangle)$ for every $a \in obj_o(\mathcal{K})$.

Define set $\Omega = \{\boldsymbol{C} \mid (\langle \boldsymbol{C}\rangle D)^{\mathcal{I}(c)} \neq \emptyset$ for any $\langle \boldsymbol{C}\rangle D \in con_o(\mathcal{K})$ and $c \in \mathfrak{C}\}$. The set contains all those metalanguage concepts whose satisfaction is enforced by means of object concepts containing context operators which are actually satisfied in the model. Then compute the set $S_{\mathcal{C},\Omega}$. Observe, that by Theorem 21, $S_{\mathcal{C},\Omega}$ has to be non-empty. Next, for every $t_c \in S_{\mathcal{C},\Omega}$, set $f(t_c) = \{\varphi \mid \boldsymbol{C} : \varphi \in \mathcal{O}, \boldsymbol{C} \in t_c\}$ and define the context structure $\langle S, \mathfrak{S}\rangle$ and the quasimodel $\mathfrak{Q}^S_{\mathfrak{S}}$ by applying the following steps.

1. Set $S = S_{\mathcal{C},\Omega}$, $\mathfrak{S} := \emptyset$, $\mathfrak{Q}^S_{\mathfrak{S}} := \emptyset$, and $T_{t_c} := \emptyset$, for every $t_c \in S_{\mathcal{C},\Omega}$.
2. For every $c \in \mathfrak{C}$ and every $t_c \in S_{\mathcal{C},\Omega}$:
   - let $\tau(c) = \langle t'_c, f', \nu\rangle$. If $t_c \subseteq t'_c$, then add $\langle t_c, f(t_c), \nu\rangle$ to $\mathfrak{S}$ and for all $d \in \Delta$ add $\tau(\langle c, d\rangle)$ to T_{t_c}.
3. For every $k \in \mathfrak{S}$, with $k = \langle t_c, f, \nu\rangle$, and $t_o \in T_{t_c}$, add $\langle k, t_o\rangle$ to $\mathfrak{Q}^S_{\mathfrak{S}}$.

It is not difficult to verify that all conditions (**CS**x), (**QC**x) and (**QM1**)-(**QM4**) have to be satisfied by $\langle S, \mathfrak{S}\rangle$ and $\mathfrak{Q}^S_{\mathfrak{S}}$. Conditions (**QM5**) and (**QM6**) are special variants of (**QC4**) and (**QC5**) and impose rigid name assumption on individual object names, i.e. the requirement that for every $a \in obj_o(\mathcal{K})$ and $c, c' \in \mathfrak{C}$, it is always the case that $a^{\mathcal{I}(c)} = a^{\mathcal{I}(c')}$. It is not hard to see that whenever an $\mathfrak{C}_{\mathcal{SHI}}^{\mathcal{EL}^{++}}$-model satisfies this constraint, (**QM5**) and (**QM6**) are also satisfied in the corresponding quasimodel.

($\Leftarrow$) Let $\mathfrak{Q}_{\mathfrak{S}}^{S}$ be a quasimodel for $\mathcal{K}$. In order to construct a model $\mathfrak{M} = (\Theta, \mathfrak{C}, \cdot^{\mathcal{J}}, \Delta, \{\cdot^{\mathcal{I}(c)}\}_{c \in \mathfrak{C}})$ for $\mathcal{K}$ we proceed in a similar manner as in the proof of Lemma 4. In the case without the rigid name assumption, we first fix the interpretation of the context dimension $(\Theta, \mathfrak{C}, \cdot^{\mathcal{J}})$ and define runs through $\mathfrak{Q}_{\mathfrak{S}}^{S}$ as before, and impose only one coherency condition on sets of runs. We say that a set $\mathfrak{R}$ of runs through $\mathfrak{Q}_{\mathfrak{S}}^{S}$ is called *coherent iff* the following condition is satisfied:

- for every $k \in \mathfrak{C}$ and $t_o \in T_k$, there is a $\rho \in \mathfrak{R}$ such that $\rho(k) = t_o$.

Next, we define the interpretation of the object dimension as follows. First, fix the object domain as:

- $\Delta := \mathfrak{R}$

Then, for every $k \in \mathfrak{C}$, with $k = \langle t_c, f, \nu \rangle$, and $\rho, \rho' \in \Delta$ set the interpretation function as:

- $\rho \in C^{\mathcal{I}(k)}$ *iff* $C \in \rho(k)$, for every $C \in con_o(\mathcal{K})$,
- $\langle \rho, \rho' \rangle \in r^{\mathcal{I}(k)}$ *iff* $\rho'(k)$ is a matching r-successor for $\rho(k)$ under f.

Finally, we fix the interpretation of the individual object names. For every $k \in \mathfrak{C}$, with $k = \langle t_c, f, \nu \rangle$, and $a \in obj_o(\mathcal{K})$:

- $a^{\mathcal{I}(c)} = \rho(k)$, for some unique $\rho \in \mathfrak{R}$ such that $\nu(a) = \rho(k)$.

When the rigid name assumption applies, first pick any $k \in \mathfrak{S}$, with $k = \langle t_c, f, \nu \rangle$, and then remove all k' from $\mathfrak{S}$, with $k' = \langle t'_c, f', \nu' \rangle$, and $\langle k', t_o \rangle$ from $\mathfrak{Q}_{\mathfrak{S}}^{S}$, such that for some $a \in obj_o(\mathcal{K})$, $\nu'(a)$ and $\nu(a)$ are not matching **S5**-successors. By conditions (**QM5**) and (**QM6**) the resulting sets $\mathfrak{S}$ and $\mathfrak{Q}_{\mathfrak{S}}^{S\,\prime}$ must be still a context structure and a quasimodel, respectively. Then we formulate the coherency conditions as:

- for every $k \in \mathfrak{C}$ and $t_o \in T_k$, there is a $\rho \in \mathfrak{R}$ such that $\rho(k) = t_o$,
- (**rigid object names**) for every $a \in obj_o(\mathcal{K})$ there is a unique run $\rho_a \in \mathfrak{R}$ such that for every $k \in \mathfrak{C}$, with $k = \langle t_c, f, \nu \rangle$, it is the case that $\nu(a) = \rho_a(k)$.

The construction of the model remains the same as in the case without the assumption, with the only difference in assigning the interpretation to individual names. For every $k \in \mathfrak{C}$ and $a \in obj_o(\mathcal{K})$:

- $a^{\mathcal{I}(c)} = \rho_a(k)$.

In both cases, one can see by the construction that the resulting interpretation $\mathfrak{M} = (\Theta, \mathfrak{C}, \cdot^{\mathcal{J}}, \Delta, \{\cdot^{\mathcal{I}(c)}\}_{c\in\mathfrak{C}})$, is an $\mathfrak{C}^{\mathcal{EL}^{++}}_{\mathcal{SHI}}$-model for $\mathcal{K}$. ❑

The EXPTIME procedure, which we sketch here, finds a quasimodel for $\mathcal{K}$, whenever it exists. This, by Lemma 5, provides a decision for the satisfiability of $\mathcal{K}$. The procedure combines computations of Algorithm 1 with the type elimination technique. It involves two non-deterministic steps marked with (♣) and (♠) below, which, as shown later, can be reduced to a deterministic computation over exponentially many possible choice.

Let $f(t_c) = \{\varphi \mid \boldsymbol{C} : \varphi \in \mathcal{O} \text{ and } \boldsymbol{C} \in t_c\}$, for every $t_c \in \Xi$. Start by (♣) picking a subset $P \subseteq con^{op}_c(\mathcal{K})$, fixing $\Omega = P \cup \{\{\boldsymbol{a}\} \mid \boldsymbol{a} : \varphi \in \mathcal{O}\}$ and computing $S_{\mathcal{C},\Omega}$ and $U_{\mathcal{C},\Omega}$. Then fix a context structure $\langle S, \mathfrak{S}\rangle$ as follows:

- Set $S := S_{\mathcal{C},\Omega}$ and $\mathfrak{S}$,
- (♠) for every $t_c \in U_{\mathcal{C},\Omega}$ and $\boldsymbol{a} \in obj_c(\mathcal{K})$, such that $\{\boldsymbol{a}\} \in t_c$ add a single tuple $\langle t_c, f(t_c), \nu\rangle \in \mathfrak{S}$, for some unique mapping $\nu : obj_c(\mathcal{K}) \mapsto \Pi$;
- for every $t_c \in U_{\mathcal{C},\Omega}$, such that $\{\boldsymbol{a}\} \notin t_c$ for every $\boldsymbol{a} \in obj_c(\mathcal{K})$, add $\langle t_c, f(t_c), \nu\rangle \in \mathfrak{S}$, for every mapping $\nu : obj_c(\mathcal{K}) \mapsto \Pi$.

Then define a set $\mathfrak{Q}^S_{\mathfrak{S}} = \{\langle k, t_o\rangle \mid k \in \mathfrak{S}, t_o \in \Pi\}$ and proceed with elimination of elements of $\mathfrak{Q}^S_{\mathfrak{S}}$ and $\mathfrak{S}$:

- for (**QC1**): eliminate k from $\mathfrak{S}$, whenever it violates the condition. Subsequently, eliminate every $\langle k, t_o\rangle$ from $\mathfrak{Q}^S_{\mathfrak{S}}$,
- for (**QC2**)-(**QC6**): eliminate $\langle k, t_o\rangle$ from $\mathfrak{Q}^S_{\mathfrak{S}}$, whenever it violates any of the conditions,
- for (**QM1**): eliminate $\langle k, t_o\rangle$ from $\mathfrak{Q}^S_{\mathfrak{S}}$, whenever $\langle k, t_o\rangle$ gets eliminated,
- for (**QM2**): eliminate $\langle k, t_o\rangle$ from $\mathfrak{Q}^S_{\mathfrak{S}}$, whenever it violates the condition,
- for (**QM3**)-(**QM4**), (**rigid name assumption** (**QM3**)-(**QM4**)): eliminate k from $\mathfrak{S}$, whenever it violates any of the conditions. Subsequently, eliminate every $\langle k, t_o\rangle$ from $\mathfrak{Q}^S_{\mathfrak{S}}$.

Let $\mathfrak{Q}^S_{\mathfrak{S}}$ be the result of the elimination. If $\mathfrak{Q}^S_{\mathfrak{S}} \neq \emptyset$ and $\langle S, \mathfrak{S}\rangle$ is a context structure, then clearly $\mathfrak{Q}^S_{\mathfrak{S}}$ is a quasimodel. In such case, by Lemma 5, the algorithm returns "$\mathcal{K}$ *is satisfiable*". Else, the algorithm repeats the elimination procedure using a different subset $P \subseteq con^{op}_c(\mathcal{K})$ in step (♣) and/or a different set of mappings $\nu : obj_c(\mathcal{K}) \mapsto \Pi$ in step (♠). Note, that by Theorem 21, and the fact that only concepts in $con^{op}_c(\mathcal{K})$ might occur inside the context operators, it follows that if a quasimodel for $\mathcal{K}$ exists, there has to exist also a quasimodel based on a context structure corresponding to one of the pairs $S_{\mathcal{C},\Omega}, U_{\mathcal{C},\Omega}$ computed by Algorith 1. Thus, if for all such combinations the procedure fails to find a quasimodel, then evidently such quasimodel does not exist and, by Lemma 5, the procedure returns "$\mathcal{K}$ *is unsatisfiable*".

It is easy to see, that at the start of the elimination the exponential space-bound for quasimodel candidates, Lemma 3, applies also to $\mathfrak{Q}^S_{\mathfrak{S}}$. Consequently, a single run of the elimination procedure cannot take more than an exponential time in order to terminate. Further, observe that the non-deterministic steps (♣) and (♠) can be replaced by a deterministic enumeration of all possible choices. In both cases there are at most exponentially many of them. For (♣) it is $2^{|con^{op}_c(\mathcal{K})|}$, while for (♠) — $(|\Pi|^{|obj_o(\mathcal{K})|})^{|obj_c(\mathcal{K})|}$, which all together results in at most $2^{\ell\mathcal{K}} \cdot 2^{2\ell\mathcal{K}^3} = 2^{2\ell\mathcal{K}^3+\ell\mathcal{K}}$ possible sets $\mathfrak{Q}^S_{\mathfrak{S}}$ to perform elimination on. Therefore, the algorithm has to return the correct answer in a time at most exponential in the size of $\mathcal{K}$. ❑

BIBLIOGRAPHY

[ACKZ09] Alessandro Artale, Diego Calvanese, Roman Kontchakov, and Michael Zakharyaschev, *The DL-Lite Family and Relations*, Journal of Artificial Intelligence Research **36** (2009), 1–69.

[AdR01] Carlos Areces and Maarten de Rijke, *From description to hybrid logics, and back*, Advances in Modal Logic (F. Wolter, H. Wansing, M. de Rijke, and M. Zakharyaschev, eds.), vol. 3, CSLI Publications, 2001, pp. 17–36.

[AFWZ02] Alessandro Artale, Enrico Franconi, Frank Wolter, and Michael Zakharyaschev, *A temporal description logic for reasoning over conceptual schemas and queries*, Proceedings of the European Conference on Logics in Artificial Intelligence, JELIA '02, 2002.

[AKK+03] Marcelo Arenas, Vasiliki Kantere, Anastasios Kementsietsidis, Iluju Kiringa, Renée J. Miller, and John Mylopoulos, *The hyperion project: from data integration to data coordination*, SIGMOD Record **32** (2003), no. 3, 53–58.

[AKL+07] Alessandro Artale, Roman Kontchakov, Carsten Lutz, Frank Wolter, and Michael Zakharyaschev, *Temporalising tractable description logics*, Proceedings of the International Symposium on Temporal Representation and Reasoning (TIME-07), 2007.

[ALT07] Alessandro Artale, Carsten Lutz, and David Toman, *A description logic of change*, Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-07), 2007.

[AS96] Varol Akman and Mehmet Surav, *Steps toward formalizing context*, AI Magazine **17** (1996), 55–72.

[BBG00] Massimo Benerecetti, Paolo Bouquet, and Chiara Ghidini, *Contextual reasoning distilled*, Philosophical Foundations of Artificial Intelligence. A special issue of the journal of Experimental and Theoretical AI (JETAI) **12** (2000), no. 3, 279–305.

[BBG08] Massimo Benerecetti, Paolo Bouquet, and Chiara Ghidini, *The dimensions of context dependence*, Perspectives on Context (P. Bouquet, L. Serafini, and R. H. Thomason, eds.), CSLI Publications, 2008, pp. 1–18.

[BBL05] Franz Baader, Sebastian Brandt, and Carsten Lutz, *Pushing the $\mathcal{EL}$ envelope*, Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-05), 2005.

[BBL08] Franz Baader, Sebastian Brandt, and Carsten Lutz, *Pushing the $\mathcal{EL}$ envelope further*, Proceedings of the Workshop on OWL: Experiences and Directions (OWLED-08 DC) (Kendall Clark and Peter F. Patel-Schneider, eds.), 2008.

[BBM95] Saša Buvač, Vanja Buvac, and Ian A. Mason, *Metamathematics of contexts*, Fundamenta Informaticae **23** (1995), 412–419.

[BCC+11] Khalid Belhajjame, James Cheney, David Corsar, Daniel Garijo, Stian Soiland-Reyes, Stephan Zednik, and Jun Zhao, *The PROV Ontology: Model and formal semantics*, Tech. report, W3C Draft: http://dvcs.w3.org/hg/prov/raw-file/default/ontology/ProvenanceFormalModel.html, 2011.

[BCM+03] Franz Baader, Diego Calvanese, Deborah L. Mcguinness, Daniele Nardi, and Peter F. Patel-Schneider, *The description logic handbook: theory, implementation, and applications*, Cambridge University Press, 2003.

[BCST96] Michael H. Böhlen, Jan Chomicki, Richard T. Snodgrass, and David Toman, *Querying TSQL2 Databases with Temporal Logic*, Proceedings of the International Conference on Extending Database Technology (EDBT-96), 1996.

[BGGB03] Paolo Bouquet, Chiara Ghidini, Fausto Giunchiglia, and Enrico Blanzieri, *Theories and uses of context in knowledge representation and reasoning*, Journal of Pragmatics (2003).

[BGK+02] Philip A. Bernstein, Fausto Giunchiglia, Anastasios Kementsietsidis, John Mylopoulos, Luciano Serafini, and Ilya Zaihrayeu, *Data management for peer-to-peer computing : A vision*, International Workshop on the Web and Databases (WebDB-02), 2002.

[BGvH+03] Paolo Bouquet, Fausto Giunchiglia, Frank van Harmelen, Luciano Serafini, and Heiner Stuckenschmidt, *C-OWL: Contextualizing ontologies.*, International Semantic Web Conference (Dieter Fensel, Katia P. Sycara, and John Mylopoulos, eds.), Lecture Notes in Computer Science, vol. 2870, Springer, 2003, pp. 164–179.

[BHPS11] Piero A Bonatti, Aiden Hogan, Axel Polleres, and Luigi Sauro, *Robust and scalable linked data reasoning incorporating provenance and trust annotations*, Journal of Web Semantics: Science, Services and Agents on the World Wide Web **9** (2011), no. 2.

[BHS03] Franz Baader, Ian Horrocks, and Ulrike Sattler, *Description logics as ontology languages for the semantic web*, Festschrift in honor of Jörg Siekmann, Lecture Notes in Artificial Intelligence, Springer-Verlag, 2003, pp. 228–248.

[BHS12a] Loris Bozzato, Martin Homola, , and Luciano Serafini, *Context on the semantic web: Why and how*, Proceedings of the 4th International Workshop on Acquisition, Representation and Reasoning with Contextualized Knowledge (ARCOE-12), 2012.

[BHS12b] Loris Bozzato, Martin Homola, and Luciano Serafini, *Towards more effective tableaux reasoning for ckr*, Proceedings of the International Workshop on Description Logics (DL-12), 2012.

[BL95] Franz Baader and Armin Laux, *Terminological logics with modal operators*, Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-95), 1995.

[BM93] Saša Buvač and Ian A. Mason, *Propositional logic of context*, Proceedings of the Conference on Artificial Intelligence (AAAI-93), 1993.

[BS03] Alex Borgida and Luciano Serafini, *Distributed description logics: Assimilating information from peer sources*, Journal of Data Semantics **1** (2003).

[BSP11] Sotiris Batsakis, Kostas Stravoskoufos, and Euripides G. M. Petrakis, *Temporal Reasoning for Supporting Temporal Queries in OWL 2.0*, Proceedings of the International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (KES-11), 2011.

[BSS05] Paolo Bouquet, Luciano Serafini, and Heiko Stoermer, *Introducing context into rdf knowledge bases*, Proceedings of the Italian Semantic Web Workshop (SWAP-05), 2005.

[BTMS10] Jie Bao, Jiao Tao, Deborah L McGuinness, and Paul Smart, *Context representation for the semantic web*, Proceedings of Web Science Conference (2010).

[Buv96] Saša Buvač, *Quantificational logic of context*, Proceedings of the Conference on Artificial Intelligence (AAAI-96), 1996.

[BVSH09] Jie Bao, George Voutsadakis, Giora Slutzki, and Vasant Honavar, *Package-based description logics*, Modular Ontologies (Heiner Stuckenschmidt, Christine Parent, and Stefano Spaccapietra, eds.), 2009, pp. 349–371.

[CBHS05] Jeremy J. Carroll, Christian Bizer, Pat Hayes, and Patrick Stickler, *Named graphs*, Journal of Web Semantics **3** (2005), 247–267.

[CCT09] James Cheney, Laura Chiticariu, and Wang-Chiew Tan, *Provenance in databases: Why, how, and where*, Foundations and Trends in Databases **1** (2009), no. 4, 379–474.

[CDGL+07] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati, *Tractable reasoning and efficient query answering in description logics: The DL-Lite family*, Journal of Automated Reasoning **39** (2007), no. 3, 385–429.

[CDGL+11] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, Mariano Rodriguez-Muro, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo, *The mastro system for ontology-based data access*, Semantic Web Journal **2** (2011), no. 1, 43–53.

[CDGLR11] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati, *Actions and programs over description logic knowledge bases: A functional approach*, Knowing, Reasoning, and Acting: Essays in Honour of Hector Levesque (Gerhard Lakemeyer and Sheila A. McIlraith, eds.), College Publications, 2011.

[CGL+07] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati, *Eql-lite: Effective first-order query processing in description logics*, Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-07), 2007.

[CGP00] E. M. Clarke, O. Grumberg, and D. A. Peled, *Model checking*, The MIT Press, 2000.

[CGPS09] Bernardo Cuenca Grau, Bijan Parsia, and Evren Sirin, *Ontology integration using $\mathcal{E}$-connections*, Modular Ontologies (Heiner Stuckenschmidt, Christine Parent, and Stefano Spaccapietra, eds.), Springer-Verlag, Berlin, Heidelberg, 2009, pp. 293–320.

[CK07] Bernardo Cuenca Grau and Oliver Kutz, *Modular ontology languages revisited*, Proceedings of the Workshop on Semantic Web for Collaborative Knowledge Acquisition, 2007.

[CKS81] Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer, *Alternation*, Journal of the ACM **28** (1981), no. 1, 114–133.

[CT98] Jan Chomicki and David Toman, *Temporal logic in information systems*, Logics for Databases and Information Systems, 1998, pp. 31–70.

[CT05] Jan Chomicki and David Toman, *Temporal Databases*, Handbook of Temporal Reasoning in Artificial Intelligence (Foundations of Artificial Intelligence), Elsevier Science Inc., 2005, pp. 429–468.

[DBH07] Jos De Bruijn and Stijn Heymans, *Logical foundations of (e)rdf(s): complexity and reasoning*, Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference (Berlin, Heidelberg), ISWC'07/ASWC'07, Springer-Verlag, 2007.

[Dem06] Stéphane Demri, *LTL over integer periodicity constraints*, Theoretical Computer Science **360** (2006), 96–123.

[DLN+92] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, Andrea Schaerf, and Werner Nutt, *Adding epistemic operators to concept languages*, Proceedings of the International Conference on Principles of Knowledge Representation (KR-92), 1992.

[dSMF06] Paulo Pinheiro da Silva, Deborah L. McGuinness, and Richard Fikes, *A proof markup language for semantic web services*, Journal of Information Systems - Special issue: The semantic web and web services **31** (2006), no. 4, 381–395.

[DSSS09] Renata Dividino, Sergej Sizov, Steffen Staab, and Bernhard Schueler, *Querying for provenance, trust, uncertainty and other meta knowledge in rdf*, Journal of Web Semantics **7** (2009), 204–219.

[FG92] Marcelo Finger and Dov M. Gabbay, *Adding a temporal dimension to a logic system*, Journal of Logic, Language and Information **1** (1992), no. 3, 203–233.

[FvHA+08] Dieter Fensel, Frank van Harmelen, Bo Andersson, Paul Brennan, Hamish Cunningham, Emanuele Della Valle, Florian Fischer, Zhisheng Huang, Atanas Kiryakov, Tony Kyung il Lee, Lael School, Volker Tresp, Stefan Wesner, Michael Witbrock, and Ning Zhong, *Towards larkc: a platform for web-scale reasoning*, Proceedings of the IEEE International Conference on Semantic Computing (ICSC 2008), August 4-7, 2008, Santa Clara, CA, USA, 8 2008.

[GG01] Chiara Ghidini and Fausto Giunchiglia, *Local models semantics, or contextual reasoning=locality+compatibility*, Artificial Intelligence **127** (2001), no. 2, 221 – 259.

[GG11] Paul Groth and Yolanda Gil, *Editorial - using provenance in the semantic web*, Journal of Web Semantics: Science, Services and Agents on the World Wide Web **9** (2011), no. 2.

[GH08] Jennifer Golbeck and James Hendler, *A semantic web approach to the provenance challenge*, Concurrency and Computation: Practice and Experience **20** (2008), no. 5, 431–439.

[Ghi99] Chiara Ghidini, *Modelling (un)bounded beliefs*, Proceedings of the International and Interdisciplinary Conference on Modelling and Using Context (Context-99), 1999.

[GHLS08] Birte Glimm, Ian Horrocks, Carsten Lutz, and Uli Sattler, *Conjunctive query answering for the description logic SHIQ*, Journal of Artificial Intelligence Research, vol. 31, 2008, pp. 157–204.

[GHV07] Claudio Gutierrez, Carlos A. Hurtado, and Alejandro A. Vaisman, *Introducing Time into RDF*, IEEE Transactions on Knowledge and Data Engineering **19** (2007), no. 2, 207–218.

[Giu93] Fausto Giunchiglia, *Contextual reasoning*, Epistemologia **XVI** (1993), 345–364.

[GKT07] Todd J. Green, Gregory Karvounarakis, and Val Tannen, *Provenance semirings*, Proceedings of the Twenty-Sixth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS-07), 2007.

[GLR11] Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati, *Higher-order description logics for domain metamodeling*, Proceedings of the Conference on Artificial Intelligence (AAAI-11), 2011.

[GM03] Ramanathan V. Guha and John McCarthy, *Varieties of contexts*, Proceedings of the 4th International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT-03), 2003.

[GM05] Stephan Grimm and Boris Motik, *Closed world reasoning in the semantic web through epistemic operators*, Proceedings of the International Workshop on OWL: Experiences and Directions (OWLED-06), 2005.

[GMF04] Ramanathan V. Guha, Rob McCool, and Richard Fikes, *Contexts for the semantic web*, Proceedings of the International Semantic Web Conference (ISWC-04), 2004.

[GPS06] Bernardo Cuenca Grau, Bijan Parsia, and Evren Sirin, *Combining OWL ontologies using E-Connections*, Journal of Web Semantics: Science, Services and Agents on the World Wide Web **4** (2006), no. 1, 40–59.

[Gro07] Davide Grossi, *Desigining invisible handcuffs. formal investigations in institutions and organizations for multi-agent systems*, Ph.D. thesis, Utrecht University, 2007.

[Grü10] Michael Grüninger, *Ontologies for dates and duration*, Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR-10) (Fangzhen Lin, Ulrike Sattler, and Miroslaw Truszczynski, eds.), AAAI Press, 2010.

[GS94] Fausto Giunchiglia and Luciano Serafini, *Multilanguage hierarchical logics, or: How we can do without modal logics*, Artificial Intelligence **65** (1994), no. 1, 29 – 70.

[GS98] Chiara Ghidini and Luciano Serafini, *Distributed first order logics*, Proceedings of the Frontiers Of Combining Systems 2 (FroCoS-98), 1998.

[GSS92] Fausto Giunchiglia, Luciano Serafini, and Alex Simpson, *Hierarchical meta-logics: Intuitions, proof theory and semantics*, Meta-Programming in Logic, Lecture Notes in Computer Science, vol. 649, 1992, pp. 235–249.

[Guh91] Ramanathan V. Guha, *Contexts: a formalization and some applications*, Ph.D. thesis, Stanford University, 1991.

[GWW07] Krzysztof Goczyla, Wojciech Waloszek, and Aleksander Waloszek, *Contextualization of a DL knowledge base*, Proceedings of the International Workshop on Description Logics (DL-07), 2007.

[Haa06] Peter Haase, *Semantic technologies for distributed information systems*, Phdthesis, PhD thesis at the Universität Karlsruhe (TH), Fakultäut für Wirtschaftswissenschaften, 2006.

[HKS06] Ian Horrocks, Oliver Kutz, and Ulrike Sattler, *The Even More Irresistible SROIQ*, Proceedings of the International COnference on Principles of Knowledge Representation and Reasoning (KR-06), AAAI Press, 2006.

[HM04] Jan Hladik and Jörg Model, *Tableau systems for SHIO and SHIQ*, Proceedings of the International Workshop on Description Logics (DL-04), 2004.

[HP04] Jerry R. Hobbs and Feng Pan, *An ontology of time for the semantic web*, ACM Trans. Asian Lang. Inf. Process. **3** (2004), no. 1, 66–85.

[HPsH03] Ian Horrocks, Peter F. Patel-schneider, and Frank Van Harmelen, *From SHIQ and RDF to OWL: The making of a Web Ontology Language*, Journal of Web Semantics **1** (2003), 2003.

[HR06] Yoram Hirshfeld and Alexander Moshe Rabinovich, *An expressive temporal logic for real time*, Mathematical Foundations of Computer Science 2006, 31st International Symposium, MFCS 2006, 2006.

[HS05] Zhisheng Huang and Heiner Stuckenschmidt, *Reasoning with multi-version ontologies: A temporal logic approach*, Proceedings of the Intenational Semantic Web Conference (ISWC-05), 2005.

[HS12] Martin Homola and Luciano Serafini, *Contextualized knowledge repositories for the semantic web*, Journal of Web Semantics: Science, Services and Agents on the World Wide Web **12** (2012).

[JvOdB11] Michiel Hildebrand Jacco van Ossenbruggen and Victor de Boer, *Interactive vocabulary alignment*, Proceedings of the International Conference on Theory and Practice of Digital Libraries (TPDL 2011) (Berlin), September 2011.

[Kaz08] Yevgeny Kazakov, *RIQ and SROIQ are harder than SHOIQ*, Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR-08), 2008.

[KGH11] Ilianna Kollia, Birte Glimm, and Ian Horrocks, *Sparql query answering over owl ontologies*, Proceedings of the Extended Semantic Web Conferenc (ESWC-11), 2011, pp. 382–396.

[KGMW00] Christos T. Karamanolis, Dimitra Giannakopoulou, Jeff Magee, and Stuart M. Wheater, *Model checking of workflow schemas*, Proceedings of the IEEE International Enterprise Distributed Object Computing Conference (EDOC-00), 2000.

[KKS12] Yevgeny Kazakov, Markus Krötzsch, and František Simančík, *Elk reasoner: Architecture and evaluation*, Proceedings of the OWL Reasoner Evaluation Workshop (ORE-12), vol. 858, 2012.

[KLWZ03] Oliver Kutz, Carsten Lutz, Frank Wolter, and Michael Zakharyaschev, *$\mathcal{E}$-connections of description logics*, Proceedings of the International Workshop on Description Logics (DL-03), 2003.

[KLWZ04] Oliver Kutz, Carsten Lutz, Frank Wolter, and Michael Zakharyaschev, *E-connections of abstract description systems*, Artificial Intelligence **156** (2004), 1–73.

[KWZG03] Agi Kurucz, Frank Wolter, Michael Zakharyaschev, and Dov M. Gabbay, *Many-dimensional modal logics: Theory and applications*, Elsevier, 2003.

[Lan06] Martin Lange, *Model checking propositional dynamic logic with all extras*, Journal of Applied Logic **4** (2006), no. 1.

[Len98] Doug Lenat, *The dimensions of context space*, Tech. report, CYCORP, 1998.

[LS10] Carsten Lutz and Lutz Schröder, *Probabilistic description logics for subjective uncertainty*, Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR-10), 2010.

[LSW12] Carsten Lutz, Inanc Seylan, and Frank Wolter, *Mixing open and closed world assumption in ontology-based data access: Non-uniform data complexity*, Proceedings of the International Workshop on Description Logics (DL-12), 2012.

[Lut08] Carsten Lutz, *The complexity of conjunctive query answering in expressive description logics*, Proceedings of the International Joint Conference on Automated Reasoning (IJCAR-08), 2008.

[LWZ08] Carsten Lutz, Frank Wolter, and Michael Zakharyaschev, *Temporal description logics: A survey*, Proceedings of the International Symposium on Temporal Representation and Reasoning (TIME-08), 2008.

[M$^+$08] Luc Moreau et al., *Special issue: The first provenance challenge*, Concurrency and Computation : Practice and Experience **20** (2008).

[Mar99] Maarten Marx, *Complexity of products of modal logics*, Journal of Logic and Computation **9** (1999), no. 2, 197–214.

[MC09] Angelo Montanari and Jan Chomicki, *Time domain*, Encyclopedia of Database Systems, 2009, pp. 3103–3107.

[McC87] John McCarthy, *Generality in artificial intelligence*, Communications of the ACM **30** (1987), 1030–1035.

[McC93] John McCarthy, *Notes on formalizing context*, Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-93), 1993.

[MCF+] Luc Moreau, Ben Clifford, Juliana Freire, Joe Futrelle, Yolanda Gil, Paul Groth, Natalia Kwasnikowska, Simon Miles, Paolo Missier, Jim Myers, Beth Plale, Yogesh Simmhan, Eric Stephan, and Jan Van den Bussche, *The open provenance model — core specification (v1.1)*, Future Generation Computer Systems **27**.

[MDSC07] Deborah L. Mcguinness, Li Ding, Paulo Pinheiro Da Silva, and Cynthia Chang, *Pml2: A modular explanation interlingua*, Proceedings of the Workshop on Explanation-aware Computing (ExaCt-07), 2007.

[MGH+09] Boris Motik, Bernardo Cuenca Grau, Ian Horrocks, Zhe Wu, Achille Fokoue, and Carsten Lutz, *OWL 2 Web Ontology Language: Profiles*, W3C Recommendation, W3C, `http://www.w3.org/TR/owl2-profiles/`, October 2009.

[Mor11] Luc Moreau, *Provenance-based reproducibility in the semantic web*, Journal of Web Semantics: Science, Services and Agents on the World Wide Web **9** (2011), no. 2.

[Mot06] Boris Motik, *Reasoning in description logics using resolution and deductive databases*, Ph.D. thesis, University of Karlsruhe, January 2006.

[Mot12] Boris Motik, *Representing and querying validity time in RDF and OWL: A logic-based approach*, Journal of Web Semantics: Science, Services and Agents on the World Wide Web **12-13** (2012), 3–21.

[MV97] Maarten Marx and Yde Venema, *Multi dimensional modal logic*, Applied Logic Series, no. 4, Kluwer Academic Publishers, 1997.

[MWF+07] Simon Miles, Sylvia C. Wong, Weijian Fang, Paul Groth, Klaus-Peter Zauner, and Luc Moreau, *Provenance-based validation of e-science experiments*, Journal of Web Semantics **5** (2007).

[Nos03] Rolf Nossum, *A decidable multi-modal logic of context*, Journal of Applied Logic **1** (2003), no. 1-2, 119 – 133.

[OCE08] Magdalena Ortiz, Diego Calvanese, and Thomas Eiter, *Data complexity of query answering in expressive description logics via tableaux*, Journal of Automated Reasoning **41** (2008), 61–98.

[OD11] Martin J. O'Connor and Amar K. Das, *A method for representing and querying temporal information in OWL*, Biomedical Engineering Systems and Technologies (Selected Papers) (2011), 97–110.

[OG98] Hans Jürgen Ohlbach and Dov M. Gabbay, *Calendar logic*, Journal of Applied Non-Classical Logics **8** (1998), no. 4.

[PS08] Eric Prud'hommeaux and Andy Seaborne, *SPARQL query language for RDF*, Tech. report, W3C Recom.: `http://www.w3.org/TR/rdf-sparql-query/`, 2008.

[RA10] Riccardo Rosati and Alessandro Almatelli, *Improving query answering over dl-lite ontologies*, Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR-10), 2010.

[Rey10] Mark Reynolds, *The complexity of decision problems for linear temporal logics*, Journal of Studies in Logic **3** (2010), no. 1.

[Ros] Riccardo Rosati, *The limits of querying ontologies*, Proceedings of the Eleventh International Conference on Database Theory (ICDT-07), Springer-Verlag.

[SAA+08] Guus Schreiber, Alia Amin, Lora Aroyo, Mark van Assem, Victor de Boer, Lynda Hardman, Michiel Hildebrand, Borys Omelayenko, Jacco van Osenbruggen, Anna Tordai, Jan Wielemaker, and Bob Wielinga, *Semantic annotation and search of cultural-heritage collections: The Multimedian E-Culture demonstrator*, Journal of Web Semantics **6** (2008), no. 4, 243–249.

[San10] Katsuhiko Sano, *Axiomatizing hybrid products: How can we reason many-dimensionally in hybrid logic?*, Journal of Applied Logic **8** (2010), no. 4, 459 – 474.

[SB04] Luciano Serafini and Paolo Bouquet, *Comparing formal theories of context in AI*, Artificial Intelligence **155** (2004), no. 1-2, 41–67.

[SBPR07] Heiko Stoermer, Paolo Bouquet, Ignazio Palmisano, and Domenico Redavid, *A context-based architecture for rdf knowledge bases: Approach, implementation and preliminary results*, Proceedings of Web Reasoning and Rule Systems (RR-07), 2007.

[SMOD08] Ravi D. Shankar, Susana B. Martins, Martin J. O'Connor, and Amar K. Das, *An ontological approach to representing and reasoning with temporal constraints in clinical trial protocols*, Proceedings of the International Conference on Health Informatics (HEALTHINF-08), 2008.

[Spa08] Kent Spackman, *SNOMED CT style guide: Situations with explicit context.*, Tech. report, SNOMED CT, 2008.

[SPG05] Yogesh L. Simmhan, Beth Plale, and Dennis Gannon, *A survey of data provenance in e-science*, SIGMOD Rec. **34** (2005).

[SPG+07] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz, *Pellet: A practical owl-dl reasoner*, Journal of Web Semantics **5** (2007), no. 2, 51–53.

[SSH08] Satya S. Sahoo, Amit Sheth, and Cory Henson, *Semantic Provenance for eScience: Managing the Deluge of Scientific Data*, IEEE Internet Computing **12** (2008), no. 4.

[TCC+10] Giovanni Tummarello, Richard Cyganiak, Michele Catasta, Szymon Danielczyk, Renaud Delbru, and Stefan Decker, *Sig.ma: Live views on the web of data*, Journal of Web Semantics **8** (2010), no. 4, 355–364.

[TCR94] David Toman, Jan Chomicki, and David S. Rogers, *Datalog with integer periodicity constraints*, Proceedings of the International Symposium on Logic Programming (ILPS-94), 1994.

[tH05] Herman J. ter Horst, *Completeness, decidability and complexity of entailment for rdf schema and a semantic extension involving the owl vocabulary*, Journal of Web Semantics **3** (2005), 79–115.

[TH06] Dmitry Tsarkov and Ian Horrocks, *Fact++ description logic reasoner: System description*, In Proceedings of the International Joint Confnerence on Automated Reasoning (IJCAR 2006), Springer, 2006, pp. 292–297.

[THM$^+$08] Thanh Tran, Peter Haase, Boris Motik, Bernardo Cuenca Grau, and Ian Horrocks, *Metalevel information in ontology-based applications*, Proceedings of the Conference on Artificial Intelligence (AAAI-08), 2008.

[Tob01] Stephan Tobies, *Complexity results and practical algorithms for logics in knowledge representation*, Ph.D. thesis, LuFG Theoretical Computer Science, RWTH Aachen, 2001.

[Via09] Victor Vianu, *Automatic verification of database-driven systems: a new frontier*, Proceedings of the International Conference on Database Theory (ICDT-09), 2009.

[WZ99] Frank Wolter and Michael Zakharyaschev, *Multi-dimensional description logics*, Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-99) (San Francisco, CA, USA), 1999, pp. 104–109.

[WZ00] Frank Wolter and Michael Zakharyaschev, *Dynamic description logics*, Proceedings of Advances in Modal Logic (AiML-98), 2000.

[ZLPS12] Antoine Zimmermann, Nuno Lopes, Axel Polleres, and Umberto Straccia, *A general framework for representing, reasoning and querying with annotated Semantic Web data*, Journal of Web Semantics: Science, Services and Agents on the World Wide Web **11** (2012), 72–95.

ABSTRACT

This thesis offers novel insights into the question of how to represent and reason with contexts within the paradigm of Description Logic-based knowledge representation. We propose a generic logic-based framework, compatible with Description Logics, for modeling, studying and addressing a range of problems related to contextuality of knowledge, particularly in the Semantic Web environment. Our approach is inspired by John McCarthy's theory of formalizing contexts in AI, in which contexts are treated as formal objects over which one can quantify and express first-order properties. Our basic conceptual contribution is a reinterpretation of this theory on the grounds of two-dimensional possible world semantics, where one dimension represents a usual object domain and the other a domain of contexts. The notion of context is thus identified with that of Kripkean possible world. Further, the framework accounts for two (possibly) interacting languages — the object and the context language — for explicit modeling of their respective domains. We argue that this general setup brings a unifying and highly explicatory perspective on a number of diverse problems reflecting the studied phenomenon. In particular, we provide the following contributions to support our claim:

- We define a novel family of two-dimensional, two-sorted DLs of context, similar to product-like combinations of DLs with modal logics. We present results regarding their expressiveness, relationships to other known formalisms, and computational complexity of the basic decision problems, ranging from ExpTime- to 2ExpTime-completeness.

- We apply the framework to the problem of ontology integration, and introduce a novel task of metaknowledge-driven selection and querying of

data. We demonstrate the ease of the tasks under the proposed approach and report on a case study of aligning different versions of Wordnet ontologies.

- We apply the framework to the problem of formal verification of data provenance records, and propose a novel provenance specification logic, based on a combination of Propositional Dynamic Logic with ontology query languages. Our proposal is validated against the test queries of The First Provenance Challenge, and supported with an analysis of its computational properties.

- We apply the framework to the problem of reasoning with temporal data, and define a generic mechanism for constructing corresponding temporal query languages, based on combinations of linear temporal logics with ontology query languages. We elaborate on the practicality of our approach by enriching the query language and data annotations with additional temporal terms, and by proposing special restrictions that render temporal querying computationally cheap and relatively straightforward to implement.

SAMENVATTING

Dit proefschrift, getiteld "Redeneren met Context in Descriptie Logicas", laat nieuwe inzichten zien in het representeren van context en het redeneren met context, in het paradigma van Descriptie Logica gebaseerde kennis representatie. We presenteren een generiek logica-gebaseerd framework, welke overeenstemmend is met Descriptie Logica. Met dit framework kan men een breed scala aan problemen modelleren, bestuderen en oplossen gerelateerd aan de contextualiteit van kennis, met name op het gebied van de Semantic Web. Onze aanpak is geinspireerd door John McCarthy's theorie voor het formalizeren van context in Kunstmatige Intelligentie. Hierin wordt context gezien als formele objecten, waarover men first-order eigenschappen kan uitdrukken en quantificeren. Onze hoofdzakelijke conceptuele bijdrage is een re-interpretatie van deze theorie, op basis van de twee-dimensionaly 'possible-world' semantiek, waar een dimensie het gebruikelijke object domain representeerd, en de andere dimensie het domein van de context representeerd. We zien de notie van context daarom als dat van de Kripkean possible world. De frameworks houdt rekening met twee (mogelijke) interactie talen — een object en context taal — voor het expliciet modelleren van deze domeinen. We beargumenteren dat deze aanpak een verbindend en zeer verklarend perspectief bied voor het fenomeen dat we bestuderen. Deze bewering ondersteunen we door de volgende bijdragen:

- We definieren een nieuwe familie van twee-dimensionale, twee-gesorteerde Descriptie Logicas van context, welke overeenkomsten hebben met productie-achtige combinaties van Descriptie Logica met Modale Logica. We presenteren resultaten zoals hun expressiviteit, relaties met andere formalismes, en de computationele complexiteit van de basis besluitvorming problemen, varierend van EXPTIME- tot 2EXPTIME- completeness.

- We passen het framework toe op het probleem van ontologie integratie, en we introduceren een nieuwe taak van meta-kennisgedreven selectie en het querying van data. We demonstreren het gemak waarmee deze taken uitgevoerd kunnen worden, en presenteren een case study waarin verschillen versies van Wordnet ontologieen worden uitgelijnd

- We passen het framework toe op het problem van formele verificatie van herkomst informatie, en presenteren een nieuwe herkomst-informatie-specifeke logica, gebaseerd op een combinatie van propositionele logica met ontologie query talen. Ons voorstel is gevalideerd tegen de test queries van The First Provenance Challenge, and wordt ondersteund met een analyse van de computationele eigenschappen.

- We passen het framework toe op het probleem van redeneren met temporele data en we definieren een generiek mechanisme voor het creeeren van temporele query talen, welke gebaseerd is op combinaties van lineaire temporele logica. We gaan uitgebreid in op de praktische toepasbaarheid van onze aanpak, door het verrijken van de query taal en data annotaties met aanvullende termen, en door nieuwe speciale restricties voor te stellen welke temporele queries computationeel goedkoop maken en relatief eenvoudig te implementeren.

SIKS Dissertatiereeks

====
1998
====

1998-1 Johan van den Akker (CWI)
DEGAS - An Active, Temporal Database of Autonomous Objects

1998-2 Floris Wiesman (UM)
Information Retrieval by Graphically Browsing Meta-Information

1998-3 Ans Steuten (TUD)
A Contribution to the Linguistic Analysis of Business Conversations
within the Language/Action Perspective

1998-4 Dennis Breuker (UM)
Memory versus Search in Games

1998-5 E.W.Oskamp (RUL)
Computerondersteuning bij Straftoemeting

====
1999
====

1999-1 Mark Sloof (VU)
Physiology of Quality Change Modelling;
Automated modelling of Quality Change of Agricultural Products

1999-2 Rob Potharst (EUR)
Classification using decision trees and neural nets

1999-3 Don Beal (UM)
The Nature of Minimax Search

1999-4 Jacques Penders (UM)
The practical Art of Moving Physical Objects

1999-5 Aldo de Moor (KUB)
Empowering Communities: A Method for the Legitimate User-Driven
Specification of Network Information Systems

1999-6 Niek J.E. Wijngaards (VU)
Re-design of compositional systems

1999-7 David Spelt (UT)
Verification support for object database design

1999-8 Jacques H.J. Lenting (UM)
Informed Gambling: Conception and Analysis of a Multi-Agent
Mechanism for Discrete Reallocation.

====
2000
====

2000-1 Frank Niessink (VU)
Perspectives on Improving Software Maintenance

2000-2 Koen Holtman (TUE)
Prototyping of CMS Storage Management

2000-3 Carolien M.T. Metselaar (UVA)
Sociaal-organisatorische gevolgen van kennistechnologie;
een procesbenadering en actorperspectief.

2000-4 Geert de Haan (VU)
ETAG, A Formal Model of Competence Knowledge for User Interface Design

2000-5 Ruud van der Pol (UM)
Knowledge-based Query Formulation in Information Retrieval.

2000-6 Rogier van Eijk (UU)
Programming Languages for Agent Communication

2000-7 Niels Peek (UU)
Decision-theoretic Planning of Clinical Patient Management

2000-8 Veerle Coup (EUR)
Sensitivity Analyis of Decision-Theoretic Networks

2000-9 Florian Waas (CWI)
Principles of Probabilistic Query Optimization

2000-10 Niels Nes (CWI)
Image Database Management System Design Considerations,
Algorithms and Architecture

2000-11 Jonas Karlsson (CWI)
Scalable Distributed Data Structures for Database Management

====
2001
====

2001-1 Silja Renooij (UU)
Qualitative Approaches to Quantifying Probabilistic Networks

2001-2 Koen Hindriks (UU)
Agent Programming Languages: Programming with Mental Models

2001-3 Maarten van Someren (UvA)
Learning as problem solving

2001-4 Evgueni Smirnov (UM)

Conjunctive and Disjunctive Version Spaces with
Instance-Based Boundary Sets

2001-5 Jacco van Ossenbruggen (VU)
Processing Structured Hypermedia: A Matter of Style

2001-6 Martijn van Welie (VU)
Task-based User Interface Design

2001-7 Bastiaan Schonhage (VU)
Diva: Architectural Perspectives on Information Visualization

2001-8 Pascal van Eck (VU)
A Compositional Semantic Structure for Multi-Agent Systems Dynamics.

2001-9 Pieter Jan 't Hoen (RUL)
Towards Distributed Development of Large Object-Oriented Models,
Views of Packages as Classes

2001-10 Maarten Sierhuis (UvA)
Modeling and Simulating Work Practice
BRAHMS: a multiagent modeling and simulation language
for work practice analysis and design

2001-11 Tom M. van Engers (VUA)
Knowledge Management:
The Role of Mental Models in Business Systems Design

====
2002
====

2002-01 Nico Lassing (VU)
Architecture-Level Modifiability Analysis

2002-02 Roelof van Zwol (UT)
Modelling and searching web-based document collections

2002-03 Henk Ernst Blok (UT)
Database Optimization Aspects for Information Retrieval

2002-04 Juan Roberto Castelo Valdueza (UU)
The Discrete Acyclic Digraph Markov Model in Data Mining

2002-05 Radu Serban (VU)
The Private Cyberspace Modeling Electronic Environments
inhabited by Privacy-concerned Agents

2002-06 Laurens Mommers (UL)
Applied legal epistemology;
Building a knowledge-based ontology of the legal domain

2002-07 Peter Boncz (CWI)
Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications

2002-08 Jaap Gordijn (VU)

Value Based Requirements Engineering: Exploring Innovative
E-Commerce Ideas

2002-09 Willem-Jan van den Heuvel(KUB)
Integrating Modern Business Applications with Objectified Legacy Systems

2002-10 Brian Sheppard (UM)
Towards Perfect Play of Scrabble

2002-11 Wouter C.A. Wijngaards (VU)
Agent Based Modelling of Dynamics: Biological and Organisational Applications

2002-12 Albrecht Schmidt (Uva)
Processing XML in Database Systems

2002-13 Hongjing Wu (TUE)
A Reference Architecture for Adaptive Hypermedia Applications

2002-14 Wieke de Vries (UU)
Agent Interaction: Abstract Approaches to Modelling, Programming and
Verifying Multi-Agent Systems

2002-15 Rik Eshuis (UT)
Semantics and Verification of UML Activity Diagrams for Workflow Modelling

2002-16 Pieter van Langen (VU)
The Anatomy of Design: Foundations, Models and Applications

2002-17 Stefan Manegold (UVA)
Understanding, Modeling, and Improving Main-Memory Database Performance

====
2003
====

2003-01 Heiner Stuckenschmidt (VU)
Ontology-Based Information Sharing in Weakly Structured Environments

2003-02 Jan Broersen (VU)
Modal Action Logics for Reasoning About Reactive Systems

2003-03 Martijn Schuemie (TUD)
Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy

2003-04 Milan Petkovic (UT)
Content-Based Video Retrieval Supported by Database Technology

2003-05 Jos Lehmann (UVA)
Causation in Artificial Intelligence and Law - A modelling approach

2003-06 Boris van Schooten (UT)
Development and specification of virtual environments

2003-07 Machiel Jansen (UvA)
Formal Explorations of Knowledge Intensive Tasks

2003-08 Yongping Ran (UM)
Repair Based Scheduling

2003-09 Rens Kortmann (UM)
The resolution of visually guided behaviour

2003-10 Andreas Lincke (UvT)
Electronic Business Negotiation: Some experimental studies on the interaction between medium, innovation context and culture

2003-11 Simon Keizer (UT)
Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks

2003-12 Roeland Ordelman (UT)
Dutch speech recognition in multimedia information retrieval

2003-13 Jeroen Donkers (UM)
Nosce Hostem - Searching with Opponent Models

2003-14 Stijn Hoppenbrouwers (KUN)
Freezing Language: Conceptualisation Processes across ICT-Supported Organisations

2003-15 Mathijs de Weerdt (TUD)
Plan Merging in Multi-Agent Systems

2003-16 Menzo Windhouwer (CWI)
Feature Grammar Systems - Incremental Maintenance of Indexes to Digital Media Warehouses

2003-17 David Jansen (UT)
Extensions of Statecharts with Probability, Time, and Stochastic Timing

2003-18 Levente Kocsis (UM)
Learning Search Decisions

====
2004
====

2004-01 Virginia Dignum (UU)
A Model for Organizational Interaction: Based on Agents, Founded in Logic

2004-02 Lai Xu (UvT)
Monitoring Multi-party Contracts for E-business

2004-03 Perry Groot (VU)
A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving

2004-04 Chris van Aart (UVA)
Organizational Principles for Multi-Agent Architectures

2004-05 Viara Popova (EUR)
Knowledge discovery and monotonicity

2004-06 Bart-Jan Hommes (TUD)
The Evaluation of Business Process Modeling Techniques

2004-07 Elise Boltjes (UM)
Voorbeeldig onderwijs; voorbeeldgestuurd onderwijs, een opstap naar abstract denken, vooral voor meisjes

2004-08 Joop Verbeek(UM)
Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale politiële gegevensuitwisseling en digitale expertise

2004-09 Martin Caminada (VU)
For the Sake of the Argument; explorations into argument-based reasoning

2004-10 Suzanne Kabel (UVA)
Knowledge-rich indexing of learning-objects

2004-11 Michel Klein (VU)
Change Management for Distributed Ontologies

2004-12 The Duy Bui (UT)
Creating emotions and facial expressions for embodied agents

2004-13 Wojciech Jamroga (UT)
Using Multiple Models of Reality: On Agents who Know how to Play

2004-14 Paul Harrenstein (UU)
Logic in Conflict. Logical Explorations in Strategic Equilibrium

2004-15 Arno Knobbe (UU)
Multi-Relational Data Mining

2004-16 Federico Divina (VU)
Hybrid Genetic Relational Search for Inductive Learning

2004-17 Mark Winands (UM)
Informed Search in Complex Games

2004-18 Vania Bessa Machado (UvA)
Supporting the Construction of Qualitative Knowledge Models

2004-19 Thijs Westerveld (UT)
Using generative probabilistic models for multimedia retrieval

2004-20 Madelon Evers (Nyenrode)
Learning from Design: facilitating multidisciplinary design teams

====
2005
====

2005-01 Floor Verdenius (UVA)
Methodological Aspects of Designing Induction-Based Applications

2005-02 Erik van der Werf (UM))
AI techniques for the game of Go

2005-03 Franc Grootjen (RUN)
A Pragmatic Approach to the Conceptualisation of Language

2005-04 Nirvana Meratnia (UT)
Towards Database Support for Moving Object data

2005-05 Gabriel Infante-Lopez (UVA)
Two-Level Probabilistic Grammars for Natural Language Parsing

2005-06 Pieter Spronck (UM)
Adaptive Game AI

2005-07 Flavius Frasincar (TUE)
Hypermedia Presentation Generation for Semantic Web Information Systems

2005-08 Richard Vdovjak (TUE)
A Model-driven Approach for Building Distributed Ontology-based Web Applications

2005-09 Jeen Broekstra (VU)
Storage, Querying and Inferencing for Semantic Web Languages

2005-10 Anders Bouwer (UVA)
Explaining Behaviour: Using Qualitative Simulation in Interactive Learning Environments

2005-11 Elth Ogston (VU)
Agent Based Matchmaking and Clustering - A Decentralized Approach to Search

2005-12 Csaba Boer (EUR)
Distributed Simulation in Industry

2005-13 Fred Hamburg (UL)
Een Computermodel voor het Ondersteunen van Euthanasiebeslissingen

2005-14 Borys Omelayenko (VU)
Web-Service configuration on the Semantic Web; Exploring how semantics meets pragmatics

2005-15 Tibor Bosse (VU)
Analysis of the Dynamics of Cognitive Processes

2005-16 Joris Graaumans (UU)
Usability of XML Query Languages

2005-17 Boris Shishkov (TUD)
Software Specification Based on Re-usable Business Components

2005-18 Danielle Sent (UU)
Test-selection strategies for probabilistic networks

2005-19 Michel van Dartel (UM)
Situated Representation

2005-20 Cristina Coteanu (UL)
Cyber Consumer Law, State of the Art and Perspectives

2005-21 Wijnand Derks (UT)
Improving Concurrency and Recovery in Database Systems by
Exploiting Application Semantics

====
2006
====

2006-01 Samuil Angelov (TUE)
Foundations of B2B Electronic Contracting

2006-02 Cristina Chisalita (VU)
Contextual issues in the design and use of information technology in organizations

2006-03 Noor Christoph (UVA)
The role of metacognitive skills in learning to solve problems

2006-04 Marta Sabou (VU)
Building Web Service Ontologies

2006-05 Cees Pierik (UU)
Validation Techniques for Object-Oriented Proof Outlines

2006-06 Ziv Baida (VU)
Software-aided Service Bundling - Intelligent Methods & Tools
for Graphical Service Modeling

2006-07 Marko Smiljanic (UT)
XML schema matching – balancing efficiency and effectiveness by means of clustering

2006-08 Eelco Herder (UT)
Forward, Back and Home Again - Analyzing User Behavior on the Web

2006-09 Mohamed Wahdan (UM)
Automatic Formulation of the Auditor's Opinion

2006-10 Ronny Siebes (VU)
Semantic Routing in Peer-to-Peer Systems

2006-11 Joeri van Ruth (UT)
Flattening Queries over Nested Data Types

2006-12 Bert Bongers (VU)
Interactivation - Towards an e-cology of people, our technological environment,
and the arts

2006-13 Henk-Jan Lebbink (UU)
Dialogue and Decision Games for Information Exchanging Agents

2006-14 Johan Hoorn (VU)
Software Requirements: Update, Upgrade, Redesign - towards a
Theory of Requirements Change

2006-15 Rainer Malik (UU)
CONAN: Text Mining in the Biomedical Domain

2006-16 Carsten Riggelsen (UU)
Approximation Methods for Efficient Learning of Bayesian Networks

2006-17 Stacey Nagata (UU)

User Assistance for Multitasking with Interruptions on a Mobile Device

2006-18 Valentin Zhizhkun (UVA)
Graph transformation for Natural Language Processing

2006-19 Birna van Riemsdijk (UU)
Cognitive Agent Programming: A Semantic Approach

2006-20 Marina Velikova (UvT)
Monotone models for prediction in data mining

2006-21 Bas van Gils (RUN)
Aptness on the Web

2006-22 Paul de Vrieze (RUN)
Fundaments of Adaptive Personalisation

2006-23 Ion Juvina (UU)
Development of Cognitive Model for Navigating on the Web

2006-24 Laura Hollink (VU)
Semantic Annotation for Retrieval of Visual Resources

2006-25 Madalina Drugan (UU)
Conditional log-likelihood MDL and Evolutionary MCMC

2006-26 Vojkan Mihajlović (UT)
Score Region Algebra: A Flexible Framework for Structured Information Retrieval

2006-27 Stefano Bocconi (CWI)
Vox Populi: generating video documentaries from semantically annotated media repositories

2006-28 Borkur Sigurbjornsson (UVA)
Focused Information Access using XML Element Retrieval

====
2007
====

2007-01 Kees Leune (UvT)
Access Control and Service-Oriented Architectures

2007-02 Wouter Teepe (RUG)
Reconciling Information Exchange and Confidentiality: A Formal Approach

2007-03 Peter Mika (VU)
Social Networks and the Semantic Web

2007-04 Jurriaan van Diggelen (UU)
Achieving Semantic Interoperability in Multi-agent Systems: a dialogue-based approach

2007-05 Bart Schermer (UL)
Software Agents, Surveillance, and the Right to Privacy: a Legislative Framework
for Agent-enabled Surveillance

2007-06 Gilad Mishne (UVA)
Applied Text Analytics for Blogs

2007-07 Natasa Jovanovic' (UT)
To Whom It May Concern - Addressee Identification in Face-to-Face Meetings

2007-08 Mark Hoogendoorn (VU)
Modeling of Change in Multi-Agent Organizations

2007-09 David Mobach (VU)
Agent-Based Mediated Service Negotiation

2007-10 Huib Aldewereld (UU)
Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols

2007-11 Natalia Stash (TUE)
Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System

2007-12 Marcel van Gerven (RUN)
Bayesian Networks for Clinical Decision Support: A Rational Approach to Dynamic Decision-Making under Uncertainty

2007-13 Rutger Rienks (UT)
Meetings in Smart Environments; Implications of Progressing Technology

2007-14 Niek Bergboer (UM)
Context-Based Image Analysis

2007-15 Joyca Lacroix (UM)
NIM: a Situated Computational Memory Model

2007-16 Davide Grossi (UU)
Designing Invisible Handcuffs. Formal investigations in Institutions and Organizations for Multi-agent Systems

2007-17 Theodore Charitos (UU)
Reasoning with Dynamic Networks in Practice

2007-18 Bart Orriens (UvT)
On the development an management of adaptive business collaborations

2007-19 David Levy (UM)
Intimate relationships with artificial partners

2007-20 Slinger Jansen (UU)
Customer Configuration Updating in a Software Supply Network

2007-21 Karianne Vermaas (UU)
Fast diffusion and broadening use: A research on residential adoption and usage of broadband internet in the Netherlands between 2001 and 2005

2007-22 Zlatko Zlatev (UT)
Goal-oriented design of value and process models from patterns

2007-23 Peter Barna (TUE)
Specification of Application Logic in Web Information Systems

2007-24 Georgina Ramírez Camps (CWI)
Structural Features in XML Retrieval

2007-25 Joost Schalken (VU)
Empirical Investigations in Software Process Improvement

====
2008
====

2008-01 Katalin Boer-Sorbán (EUR)
Agent-Based Simulation of Financial Markets: A modular,continuous-time approach

2008-02 Alexei Sharpanskykh (VU)
On Computer-Aided Methods for Modeling and Analysis of Organizations

2008-03 Vera Hollink (UVA)
Optimizing hierarchical menus: a usage-based approach

2008-04 Ander de Keijzer (UT)
Management of Uncertain Data - towards unattended integration

2008-05 Bela Mutschler (UT)
Modeling and simulating causal dependencies on process-aware information systems from a cost perspective

2008-06 Arjen Hommersom (RUN)
On the Application of Formal Methods to Clinical Guidelines,
an Artificial Intelligence Perspective

2008-07 Peter van Rosmalen (OU)
Supporting the tutor in the design and support of adaptive e-learning

2008-08 Janneke Bolt (UU)
Bayesian Networks: Aspects of Approximate Inference

2008-09 Christof van Nimwegen (UU)
The paradox of the guided user: assistance can be counter-effective

2008-10 Wauter Bosma (UT)
Discourse oriented summarization

2008-11 Vera Kartseva (VU)
Designing Controls for Network Organizations: A Value-Based Approach

2008-12 Jozsef Farkas (RUN)
A Semiotically Oriented Cognitive Model of Knowledge Representation

2008-13 Caterina Carraciolo (UVA)
Topic Driven Access to Scientific Handbooks

2008-14 Arthur van Bunningen (UT)
Context-Aware Querying; Better Answers with Less Effort

2008-15 Martijn van Otterlo (UT)
The Logic of Adaptive Behavior: Knowledge Representation and Algorithms
for the Markov Decision Process Framework in First-Order Domains.

2008-16 Henriette van Vugt (VU)
Embodied agents from a user's perspective

2008-17 Martin Op 't Land (TUD)
Applying Architecture and Ontology to the Splitting and Allying of Enterprises

2008-18 Guido de Croon (UM)
Adaptive Active Vision

2008-19 Henning Rode (UT)
From Document to Entity Retrieval: Improving Precision and Performance of Focused Text Search

2008-20 Rex Arendsen (UVA)
Geen bericht, goed bericht. Een onderzoek naar de effecten van de introductie van elektronisch berichtenverkeer met de overheid op de administratieve lasten van bedrijven

2008-21 Krisztian Balog (UVA)
People Search in the Enterprise

2008-22 Henk Koning (UU)
Communication of IT-Architecture

2008-23 Stefan Visscher (UU)
Bayesian network models for the management of ventilator-associated pneumonia

2008-24 Zharko Aleksovski (VU)
Using background knowledge in ontology matching

2008-25 Geert Jonker (UU)
Efficient and Equitable Exchange in Air Traffic Management Plan Repair using Spender-signed Currency

2008-26 Marijn Huijbregts (UT)
Segmentation, Diarization and Speech Transcription: Surprise Data Unraveled

2008-27 Hubert Vogten (OU)
Design and Implementation Strategies for IMS Learning Design

2008-28 Ildiko Flesch (RUN)
On the Use of Independence Relations in Bayesian Networks

2008-29 Dennis Reidsma (UT)
Annotations and Subjective Machines - Of Annotators, Embodied Agents, Users, and Other Humans

2008-30 Wouter van Atteveldt (VU)
Semantic Network Analysis: Techniques for Extracting, Representing and Querying Media Content

2008-31 Loes Braun (UM)
Pro-Active Medical Information Retrieval

2008-32 Trung H. Bui (UT)
Toward Affective Dialogue Management using Partially Observable Markov Decision Processes

2008-33 Frank Terpstra (UVA)
Scientific Workflow Design; theoretical and practical issues

2008-34 Jeroen de Knijf (UU)
Studies in Frequent Tree Mining

2008-35 Ben Torben Nielsen (UvT)
Dendritic morphologies: function shapes structure

====
2009
====

2009-01 Rasa Jurgelenaite (RUN)
Symmetric Causal Independence Models

2009-02 Willem Robert van Hage (VU)
Evaluating Ontology-Alignment Techniques

2009-03 Hans Stol (UvT)
A Framework for Evidence-based Policy Making Using IT

2009-04 Josephine Nabukenya (RUN)
Improving the Quality of Organisational Policy Making using Collaboration Engineering

2009-05 Sietse Overbeek (RUN)
Bridging Supply and Demand for Knowledge Intensive Tasks - Based on Knowledge, Cognition, and Quality

2009-06 Muhammad Subianto (UU)
Understanding Classification

2009-07 Ronald Poppe (UT)
Discriminative Vision-Based Recovery and Recognition of Human Motion

2009-08 Volker Nannen (VU)
Evolutionary Agent-Based Policy Analysis in Dynamic Environments

2009-09 Benjamin Kanagwa (RUN)
Design, Discovery and Construction of Service-oriented Systems

2009-10 Jan Wielemaker (UVA)
Logic programming for knowledge-intensive interactive applications

2009-11 Alexander Boer (UVA)
Legal Theory, Sources of Law & the Semantic Web

2009-12 Peter Massuthe (TUE, Humboldt-Universitaet zu Berlin)
Operating Guidelines for Services

2009-13 Steven de Jong (UM)
Fairness in Multi-Agent Systems

2009-14 Maksym Korotkiy (VU)
From ontology-enabled services to service-enabled ontologies (making ontologies work

in e-science with ONTO-SOA)

2009-15 Rinke Hoekstra (UVA)
Ontology Representation - Design Patterns and Ontologies that Make Sense

2009-16 Fritz Reul (UvT)
New Architectures in Computer Chess

2009-17 Laurens van der Maaten (UvT)
Feature Extraction from Visual Data

2009-18 Fabian Groffen (CWI)
Armada, An Evolving Database System

2009-19 Valentin Robu (CWI)
Modeling Preferences, Strategic Reasoning and Collaboration in Agent-Mediated Electronic Markets

2009-20 Bob van der Vecht (UU)
Adjustable Autonomy: Controling Influences on Decision Making

2009-21 Stijn Vanderlooy (UM)
Ranking and Reliable Classification

2009-22 Pavel Serdyukov (UT)
Search For Expertise: Going beyond direct evidence

2009-23 Peter Hofgesang (VU)
Modelling Web Usage in a Changing Environment

2009-24 Annerieke Heuvelink (VUA)
Cognitive Models for Training Simulations

2009-25 Alex van Ballegooij (CWI)
RAM: Array Database Management through Relational Mapping

2009-26 Fernando Koch (UU)
An Agent-Based Model for the Development of Intelligent Mobile Services

2009-27 Christian Glahn (OU)
Contextual Support of social Engagement and Reflection on the Web

2009-28 Sander Evers (UT)
Sensor Data Management with Probabilistic Models

2009-29 Stanislav Pokraev (UT)
Model-Driven Semantic Integration of Service-Oriented Applications

2009-30 Marcin Zukowski (CWI)
Balancing vectorized query execution with bandwidth-optimized storage

2009-31 Sofiya Katrenko (UVA)
A Closer Look at Learning Relations from Text

2009-32 Rik Farenhorst (VU) and Remco de Boer (VU)
Architectural Knowledge Management: Supporting Architects and Auditors

2009-33 Khiet Truong (UT)
How Does Real Affect Affect Affect Recognition In Speech?

2009-34 Inge van de Weerd (UU)
Advancing in Software Product Management: An Incremental Method Engineering Approach

2009-35 Wouter Koelewijn (UL)
Privacy en Politiegegevens; Over geautomatiseerde normatieve informatie-uitwisseling

2009-36 Marco Kalz (OUN)
Placement Support for Learners in Learning Networks

2009-37 Hendrik Drachsler (OUN)
Navigation Support for Learners in Informal Learning Networks

2009-38 Riina Vuorikari (OU)
Tags and self-organisation: a metadata ecology for learning resources in a multilingual context

2009-39 Christian Stahl (TUE, Humboldt-Universitaet zu Berlin)
Service Substitution – A Behavioral Approach Based on Petri Nets

2009-40 Stephan Raaijmakers (UvT)
Multinomial Language Learning: Investigations into the Geometry of Language

2009-41 Igor Berezhnyy (UvT)
Digital Analysis of Paintings

2009-42 Toine Bogers
Recommender Systems for Social Bookmarking

2009-43 Virginia Nunes Leal Franqueira (UT)
Finding Multi-step Attacks in Computer Networks using Heuristic Search and Mobile Ambients

2009-44 Roberto Santana Tapia (UT)
Assessing Business-IT Alignment in Networked Organizations

2009-45 Jilles Vreeken (UU)
Making Pattern Mining Useful

2009-46 Loredana Afanasiev (UvA)
Querying XML: Benchmarks and Recursion

====
2010
====

2010-01 Matthijs van Leeuwen (UU)
Patterns that Matter

2010-02 Ingo Wassink (UT)
Work flows in Life Science

2010-03 Joost Geurts (CWI)
A Document Engineering Model and Processing Framework for Multimedia documents

2010-04 Olga Kulyk (UT)
Do You Know What I Know? Situational Awareness of Co-located Teams in Multidisplay Environments

2010-05 Claudia Hauff (UT)
Predicting the Effectiveness of Queries and Retrieval Systems

2010-06 Sander Bakkes (UvT)
Rapid Adaptation of Video Game AI

2010-07 Wim Fikkert (UT)
Gesture interaction at a Distance

2010-08 Krzysztof Siewicz (UL)
Towards an Improved Regulatory Framework of Free Software. Protecting user freedoms in a world of software communities and eGovernments

2010-09 Hugo Kielman (UL)
A Politiele gegevensverwerking en Privacy, Naar een effectieve waarborging

2010-10 Rebecca Ong (UL)
Mobile Communication and Protection of Children 2010-11 Adriaan Ter Mors (TUD)
The world according to MARP: Multi-Agent Route Planning

2010-12 Susan van den Braak (UU)
Sensemaking software for crime analysis

2010-13 Gianluigi Folino (RUN)
High Performance Data Mining using Bio-inspired techniques

2010-14 Sander van Splunter (VU)
Automated Web Service Reconfiguration

2010-15 Lianne Bodenstaff (UT)
Managing Dependency Relations in Inter-Organizational Models

2010-16 Sicco Verwer (TUD)
Efficient Identification of Timed Automata, theory and practice

2010-17 Spyros Kotoulas (VU)
Scalable Discovery of Networked Resources: Algorithms, Infrastructure, Applications

2010-18 Charlotte Gerritsen (VU)
Caught in the Act: Investigating Crime by Agent-Based Simulation

2010-19 Henriette Cramer (UvA)
People's Responses to Autonomous and Adaptive Systems

2010-20 Ivo Swartjes (UT)
Whose Story Is It Anyway? How Improv Informs Agency and Authorship of Emergent Narrative

2010-21 Harold van Heerde (UT)
Privacy-aware data management by means of data degradation

2010-22 Michiel Hildebrand (CWI)
End-user Support for Access to Heterogeneous Linked Data

2010-23 Bas Steunebrink (UU)
The Logical Structure of Emotions

2010-24 Dmytro Tykhonov
Designing Generic and Efficient Negotiation Strategies

2010-25 Zulfiqar Ali Memon (VU)
Modelling Human-Awareness for Ambient Agents: A Human Mindreading Perspective

2010-26 Ying Zhang (CWI)
XRPC: Efficient Distributed Query Processing on Heterogeneous XQuery Engines

2010-27 Marten Voulon (UL)
Automatisch contracteren

2010-28 Arne Koopman (UU)
Characteristic Relational Patterns

2010-29 Stratos Idreos(CWI)
Database Cracking: Towards Auto-tuning Database Kernels

2010-30 Marieke van Erp (UvT)
Accessing Natural History - Discoveries in data cleaning, structuring, and retrieval

2010-31 Victor de Boer (UVA)
Ontology Enrichment from Heterogeneous Sources on the Web

2010-32 Marcel Hiel (UvT)
An Adaptive Service Oriented Architecture: Automatically solving Interoperability Problems

2010-33 Robin Aly (UT)
Modeling Representation Uncertainty in Concept-Based Multimedia Retrieval

2010-34 Teduh Dirgahayu (UT)
Interaction Design in Service Compositions

2010-35 Dolf Trieschnigg (UT)
Proof of Concept: Concept-based Biomedical Information Retrieval

2010-36 Jose Janssen (OU)
Paving the Way for Lifelong Learning; Facilitating competence development through a learning path specification

2010-37 Niels Lohmann (TUE)
Correctness of services and their composition

2010-38 Dirk Fahland (TUE)
From Scenarios to components

2010-39 Ghazanfar Farooq Siddiqui (VU)
Integrative modeling of emotions in virtual agents

2010-40 Mark van Assem (VU)
Converting and Integrating Vocabularies for the Semantic Web

2010-41 Guillaume Chaslot (UM)
Monte-Carlo Tree Search

2010-42 Sybren de Kinderen (VU)
Needs-driven service bundling in a multi-supplier setting - the computational e3-service approach

2010-43 Peter van Kranenburg (UU)
A Computational Approach to Content-Based Retrieval of Folk Song Melodies

2010-44 Pieter Bellekens (TUE)
An Approach towards Context-sensitive and User-adapted Access to Heterogeneous Data Sources, Illustrated in the Television Domain

2010-45 Vasilios Andrikopoulos (UvT)
A theory and model for the evolution of software services

2010-46 Vincent Pijpers (VU)
e3alignment: Exploring Inter-Organizational Business-ICT Alignment

2010-47 Chen Li (UT)
Mining Process Model Variants: Challenges, Techniques, Examples

2010-48 Milan Lovric (EUR)
Behavioral Finance and Agent-Based Artificial Markets

2010-49 Jahn-Takeshi Saito (UM)
Solving difficult game positions

2010-50 Bouke Huurnink (UVA)
Search in Audiovisual Broadcast Archives

2010-51 Alia Khairia Amin (CWI)
Understanding and supporting information seeking tasks in multiple sources

2010-52 Peter-Paul van Maanen (VU)
Adaptive Support for Human-Computer Teams: Exploring the Use of Cognitive Models of Trust and Attention

2010-53 Edgar Meij (UVA)
Combining Concepts and Language Models for Information Access

====
2011
====

2011-01 Botond Cseke (RUN)
Variational Algorithms for Bayesian Inference in Latent Gaussian Models

2011-02 Nick Tinnemeier(UU)
Organizing Agent Organizations. Syntax and Operational Semantics of an Organization-Oriented Programming Language

2011-03 Jan Martijn van der Werf (TUE)
Compositional Design and Verification of Component-Based Information Systems

2011-04 Hado van Hasselt (UU)
Insights in Reinforcement Learning; Formal analysis and empirical evaluation of temporal-difference learning algorithms

2011-05 Base van der Raadt (VU)
Enterprise Architecture Coming of Age - Increasing the Performance of an Emerging Discipline.

2011-06 Yiwen Wang (TUE)
Semantically-Enhanced Recommendations in Cultural Heritage

2011-07 Yujia Cao (UT)
Multimodal Information Presentation for High Load Human Computer Interaction

2011-08 Nieske Vergunst (UU)
BDI-based Generation of Robust Task-Oriented Dialogues

2011-09 Tim de Jong (OU)
Contextualised Mobile Media for Learning

2011-10 Bart Bogaert (UvT)
Cloud Content Contention

2011-11 Dhaval Vyas (UT)
Designing for Awareness: An Experience-focused HCI Perspective

2011-12 Carmen Bratosin (TUE)
Grid Architecture for Distributed Process Mining

2011-13 Xiaoyu Mao (UvT)
Airport under Control. Multiagent Scheduling for Airport Ground Handling

2011-14 Milan Lovric (EUR)
Behavioral Finance and Agent-Based Artificial Markets

2011-15 Marijn Koolen (UvA)
The Meaning of Structure: the Value of Link Evidence for Information Retrieval

2011-16 Maarten Schadd (UM)
Selective Search in Games of Different Complexity

2011-17 Jiyin He (UVA)
Exploring Topic Structure: Coherence, Diversity and Relatedness

2011-18 Mark Ponsen (UM)
Strategic Decision-Making in complex games

2011-19 Ellen Rusman (OU)
The Mind ' s Eye on Personal Profiles

2011-20 Qing Gu (VU)
Guiding service-oriented software engineering - A view-based approach

2011-21 Linda Terlouw (TUD)
Modularization and Specification of Service-Oriented Systems

2011-22 Junte Zhang (UVA)
System Evaluation of Archival Description and Access

2011-23 Wouter Weerkamp (UVA)
Finding People and their Utterances in Social Media

2011-24 Herwin van Welbergen (UT)
Behavior Generation for Interpersonal Coordination with Virtual Humans On Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior

2011-25 Syed Waqar ul Qounain Jaffry (VU)
Analysis and Validation of Models for Trust Dynamics

2011-26 Matthijs Aart Pontier (VU)
Virtual Agents for Human Communication - Emotion Regulation and Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots

2011-27 Aniel Bhulai (VU)
Dynamic website optimization through autonomous management of design patterns

2011-28 Rianne Kaptein(UVA)
Effective Focused Retrieval by Exploiting Query Context and Document Structure

2011-29 Faisal Kamiran (TUE)
Discrimination-aware Classification

2011-30 Egon van den Broek (UT)
Affective Signal Processing (ASP): Unraveling the mystery of emotions

2011-31 Ludo Waltman (EUR)
Computational and Game-Theoretic Approaches for Modeling Bounded Rationality

2011-32 Nees-Jan van Eck (EUR)
Methodological Advances in Bibliometric Mapping of Science

2011-33 Tom van der Weide (UU)
Arguing to Motivate Decisions

2011-34 Paolo Turrini (UU)
Strategic Reasoning in Interdependence: Logical and Game-theoretical Investigations

2011-35 Maaike Harbers (UU)
Explaining Agent Behavior in Virtual Training

2011-36 Erik van der Spek (UU)
Experiments in serious game design: a cognitive approach

2011-37 Adriana Burlutiu (RUN)
Machine Learning for Pairwise Data, Applications for Preference Learning and Supervised Network Inference

2011-38 Nyree Lemmens (UM)
Bee-inspired Distributed Optimization

2011-39 Joost Westra (UU)
Organizing Adaptation using Agents in Serious Games

2011-40 Viktor Clerc (VU)
Architectural Knowledge Management in Global Software Development

2011-41 Luan Ibraimi (UT)
Cryptographically Enforced Distributed Data Access Control

2011-42 Michal Sindlar (UU)

Explaining Behavior through Mental State Attribution

2011-43 Henk van der Schuur (UU)
Process Improvement through Software Operation Knowledge

2011-44 Boris Reuderink (UT)
Robust Brain-Computer Interfaces

2011-45 Herman Stehouwer (UvT)
Statistical Language Models for Alternative Sequence Selection

2011-46 Beibei Hu (TUD)
Towards Contextualized Information Delivery: A Rule-based Architecture for the Domain of Mobile Police Work

2011-47 Azizi Bin Ab Aziz(VU)
Exploring Computational Models for Intelligent Support of Persons with Depression

2011-48 Mark Ter Maat (UT)
Response Selection and Turn-taking for a Sensitive Artificial Listening Agent

2011-49 Andreea Niculescu (UT)
Conversational interfaces for task-oriented spoken dialogues:
design aspects influencing interaction quality

====
2012
====

2012-01 Terry Kakeeto (UvT)
Relationship Marketing for SMEs in Uganda

2012-02 Muhammad Umair(VU)
Adaptivity, emotion, and Rationality in Human and Ambient Agent Models

2012-03 Adam Vanya (VU)
Supporting Architecture Evolution by Mining Software Repositories

2012-04 Jurriaan Souer (UU)
Development of Content Management System-based Web Applications

2012-05 Marijn Plomp (UU)
Maturing Interorganisational Information Systems

2012-06 Wolfgang Reinhardt (OU)
Awareness Support for Knowledge Workers in Research Networks

2012-07 Rianne van Lambalgen (VU)
When the Going Gets Tough: Exploring Agent-based Models of Human Performance under Demanding Conditions

2012-08 Gerben de Vries (UVA)
Kernel Methods for Vessel Trajectories

2012-09 Ricardo Neisse (UT)
Trust and Privacy Management Support for Context-Aware Service Platforms

2012-10 David Smits (TUE)
Towards a Generic Distributed Adaptive Hypermedia Environment

2012-11 J.C.B. Rantham Prabhakara (TUE)
Process Mining in the Large: Preprocessing, Discovery, and Diagnostics

2012-12 Kees van der Sluijs (TUE)
Model Driven Design and Data Integration in Semantic Web Information Systems

2012-13 Suleman Shahid (UvT)
Fun and Face: Exploring non-verbal expressions of emotion during playful interactions

2012-14 Evgeny Knutov(TUE)
Generic Adaptation Framework for Unifying Adaptive Web-based Systems

2012-15 Natalie van der Wal (VU)
Social Agents. Agent-Based Modelling of Integrated Internal and Social Dynamics of Cognitive and Affective Processes

2012-16 Fiemke Both (VU)
Helping people by understanding them - Ambient Agents supporting task execution and depression treatment

2012-17 Amal Elgammal (UvT)
Towards a Comprehensive Framework for Business Process Compliance

2012-18 Eltjo Poort (VU)
Improving Solution Architecting Practices

2012-19 Helen Schonenberg (TUE)
What's Next? Operational Support for Business Process Execution

2012-20 Ali Bahramisharif (RUN)
Covert Visual Spatial Attention, a Robust Paradigm for Brain-Computer Interfacing

2012-21 Roberto Cornacchia (TUD)
Querying Sparse Matrices for Information Retrieval

2012-22 Thijs Vis (UvT)
Intelligence, politie en veiligheidsdienst: verenigbare grootheden?

2012-23 Christian Muehl (UT)
Toward Affective Brain-Computer Interfaces: Exploring the Neurophysiology of Affect during Human Media Interaction

====
2013
====

2013-01 Viorel Milea (EUR)
News Analytics for Financial Decision Support

2013-02 Erietta Liarou (CWI)
MonetDB/DataCell: Leveraging the Column-store Database Technology for Efficient and Scalable Stream Processing