Context Reasoning for Role-Based Models

Dissertation

zur Erlangung des akademischen Grades
Doktoringenieur (Dr.-Ing.)

vorgelegt an der
Technischen Universität Dresden
Fakultät Informatik

eingereicht von
Dipl.-Math. Stephan Böhme
geboren am 23. April 1987 in Dresden

verteidigt am 29. Oktober 2017

Gutachter:
Prof. Dr.-Ing. Franz Baader
Technische Universität Dresden

Dr. Jeff Z. Pan
University of Aberdeen

Dresden, im Juni 2017

Abstract

In a modern world software systems are literally everywhere. These should cope with very complex scenarios including the ability of context-awareness and self-adaptability. The concept of roles provide the means to model such complex, context-dependent systems. In role-based systems, the relational and context-dependent properties of objects are transferred into the roles that the object plays in a certain context. However, even if the domain can be expressed in a well-structured and modular way, role-based models can still be hard to comprehend due to the sophisticated semantics of roles, contexts and different constraints. Hence, unintended implications or inconsistencies may be overlooked. A feasible logical formalism is required here. In this setting Description Logics (DLs) fit very well as a starting point for further considerations since as a decidable fragment of first-order logic they have both an underlying formal semantics and decidable reasoning problems. DLs are a well-understood family of knowledge representation formalisms which allow to represent application domains in a well-structured way by DL-concepts, i.e. unary predicates, and DL-roles, i.e. binary predicates. However, classical DLs lack expressive power to formalise contextual knowledge which is crucial for formalising role-based systems.

We investigate a novel family of contextualised description logics that is capable of expressing contextual knowledge and preserves decidability even in the presence of rigid DL-roles, i.e. relational structures that are context-independent. For these contextualised description logics we thoroughly analyse the complexity of the consistency problem. Furthermore, we present a mapping algorithm that allows for an automated translation from a formal role-based model, namely a Compartment Role Object Model (CROM), into a contextualised DL ontology. We prove the semantical correctness and provide ideas how features extending CROM can be expressed in our contextualised DLs. As final step for a completely automated analysis of role-based models, we investigate a practical reasoning algorithm and implement the first reasoner that can process contextual ontologies.

Contents

Chapter 1

Introduction

Nowadays, we are literally everywhere surrounded by software systems. Current developments indicate a continuing growth in the future. Not only the amount of systems increases, but also the requirements and expectations users impose on current software steadily rise. Modern software systems should cope with very complex scenarios. This includes the ability of context-awareness and self-adaptability. For example, a robot in a smart factory should recognise when a human co-worker approaches and switch to a different, human-friendly working mode accordingly. Similarly, software in autonomous cars or in the area of smart homes needs to adapt to various situations of which some are not even stated explicitly. Furthermore, software must be easily maintainable and, when necessary, changes on the system should be realised without much down time which, for example, in a smart factory is very costly.

1.1 Role-Based Systems

In order to achieve all these goals, the concept of *roles* is very promising. First introduced by Bachman [BD77], roles appeared over the last decades in several fields of computer science. Most prominent is the *role-based access control* [FKC03; AF11; SCF+96], albeit it is only a special application for roles with a narrow scope. Roles are also introduced, for example, in data modelling [Hal06], conceptual modelling [Ste00; Gui05; Ste07] and programming languages [BBT06; Her07; BGE07].

The relational or context-dependent properties and behaviour of objects are transferred into the roles that object plays in a certain context. This paradigm also supports Dijkstra's separation of concerns [Dij82] which simplifies development and maintenance of such systems. Due to the use of roles, *role-based systems* can model application domains cleaner and more structured, since ontologically different entities are modelled by different concepts.

Let us consider, for example, the concepts of Person and Customer. With an object-oriented approach of inheritance as a specialisation relation, we could model Customer as a subclass of Person, as not every person is a customer. On the other hand, if we restrict our domain to a business context and add the concept of a Company, the inheritance relation would flip and we also have Company as subclass of Customer. This conflict can be resolved by recognising Person and Company as context-independent basic concepts, so-called *natural types* and Customer as a role a person or company can play in a business context. Here, it becomes also apparent that the concept of a *context* is closely related to a role.

While role-based modelling provides the means to handle and model complex and context-dependent domains in a well-structured and modular way, the process can still be tedious, hard and error-prone. Due to the sophisticated semantics of roles, contexts and many different kinds of constraints, unintended implications or even inconsistencies can easily be hidden within such a model. Since it is nearly impossible to uncover all inferences, it becomes imperative for domain analysts to reason on role-based models to find such implicit knowledge. Here, a feasible logical formalism is needed.

1.2 Description Logics

Description Logics (DLs) [BCM+07] are a well-known formalism for knowledge representation. They possess formal semantics and allow to define a variety of reasoning problems.

The basic building blocks in description logics are so-called *concept names* and *role names*. Concept names denote sets of domain elements. For example, the concept names Person or Bank denote the sets of all persons or banks in a domain. Relational structures are represented by so-called DL *role names*, which are essentially binary relations on the domain. The term "role" originates from the early knowledge representation system KL-ONE [WS92] and has only little in common with roles of role-based systems except that it reflects the relational property of a role. A person which is related to a bank via a DL role $\mathsf{customer}$ could be seen as someone playing the role of a customer in the context of a bank. Besides that, DL roles are merely binary relations. With the help of *concept* and *role constructors*, complex concepts and roles can be defined. Which constructors are allowed depends on the specific DL. Complex concepts can be used as descriptions and to classify domain elements, e.g. the complex concept

$$\mathsf{NFL\_Player} \sqcap \mathsf{Healthy} \sqcap \exists.\mathsf{wins}(\mathsf{NFL\_Game}) \tag{1.1}$$

describes the set of all healthy NFL players who win NFL games.

With the help of concepts, we can express our knowledge about a domain through *DL axioms*. General knowledge is phrased via *general concept inclusions (GCIs)*, which state that one concept is a sub-concept of another. For example the GCI

$$\mathsf{NFL\_Player} \sqcap \mathsf{Healthy} \sqcap \exists.\mathsf{wins}(\mathsf{NFL\_Game}) \sqsubseteq \mathsf{Happy\_NFL\_Player} \tag{1.2}$$

states that a healthy NFL player who wins NFL games is a happy NFL player. Conversely, it does not say anything on whether every happy NFL player is healthy or wins games. Facts about a domain can be expressed via *concept* and *role assertions*. To express facts, we also introduce *individual names* which denote single domain elements. As an example consider the following axioms:

$$(\mathsf{NFL\_Player} \sqcap \mathsf{Healthy})(\mathit{AaronRodgers}) \tag{1.3}$$

$$\mathsf{NFL\_Game}(\mathit{SuperBowlXLV}), \tag{1.4}$$

$$\mathsf{wins}(\mathit{AaronRodgers}, \mathit{SuperBowlXLV}) \tag{1.5}$$

The first two concept assertions (1.3) and (1.4) state that Aaron Rodgers is a healthy NFL player and that Super Bowl XLV is an NFL game, while (1.5) expresses that he won Super Bowl XLV. So he is also a happy NFL player, even if not stated explicitly. A *DL knowledge base* is a set of such axioms.

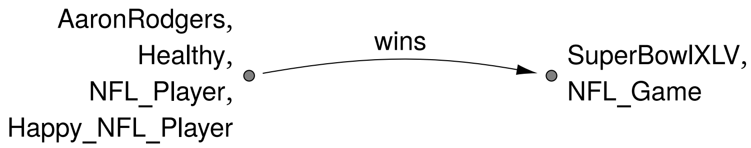


Figure 1.1: Interpretation that models axioms (1.2) to (1.6).

The semantics of DLs are defined in a model-theoretic way and capture exactly the above mentioned intentions. An interpretation $\mathcal{I}$ consists of a domain and an interpretation function $\cdot^{\mathcal{I}}$ which maps concept, role and individual names, respectively, to subsets, binary relations and elements of the domain. From there, it is exactly defined how complex concepts must be interpreted. For example, $(A \sqcap B)^{\mathcal{I}}$ is the intersection of $A^{\mathcal{I}}$ and $B^{\mathcal{I}}$. The concept name NFL_Player itself has no meaning and an interpretation must make sure that $\mathsf{NFL\_Player}^{\mathcal{I}}$ actually is the set of all NFL players.

Now, the most interesting reasoning problems are the *consistency problem* and the *entailment problem*. A knowledge base is consistent if there exists some interpretation that *models* the knowledge base, i.e. an interpretation that fulfils all the axioms. An axiom is *entailed* by a knowledge base if every model of the knowledge base also models that axiom. For example, the following axiom is entailed by (1.2) to (1.5):

$$\mathsf{Happy\_NFL\_Player}(\mathit{AaronRodgers}). \tag{1.6}$$

Figure 1.1 depicts an interpretation which is a model of axioms (1.2) to (1.6).

However, classical DLs lack expressive power to formalise that some individuals satisfy certain concepts and relate to other individuals depending on a *specific context* which is needed to reason on role-based systems.

1.3 Contextualised Description Logics

To overcome that deficiency in expressiveness of classical DLs, often two-dimensional DLs are employed [KG10; KG11b; KG11a; KG16]. This approach uses one DL $\mathcal{L}_{\mathsf{M}}$ as the *meta logic* to represent the contexts and their relationships to each other, and combines it with the *object logic* $\mathcal{L}_{\mathsf{O}}$ that captures the relational structure within each context. Moreover, while some pieces of information depend on the context, other pieces of information are shared throughout all contexts. For instance, the name of a person typically stays the same independent of the actual context. Expressing this context-independent information requires that some concepts and roles are designated to be *rigid*, i.e. they are required to be interpreted the same in all contexts. Unfortunately, if rigid roles are admitted, reasoning in the above mentioned two-dimensional DLs of context turns out to be undecidable; see [KG10].

We propose and investigate a family of two-dimensional context DLs $\mathcal{L}_{\mathsf{M}}[\![\mathcal{L}_{\mathsf{O}}]\!]$ that meets the above requirements, but is a restricted form of the ones defined in [KG10] in the sense that we limit the interaction of $\mathcal{L}_{\mathsf{M}}$ and $\mathcal{L}_{\mathsf{O}}$. More precisely, in our family of context DLs the meta logic can refer to the internal structure of each context, but not vice versa. That means that information is viewed in a top-down manner, i.e. information of different contexts is strictly capsuled and can only be accessed from the meta level. Hence, we cannot express, for

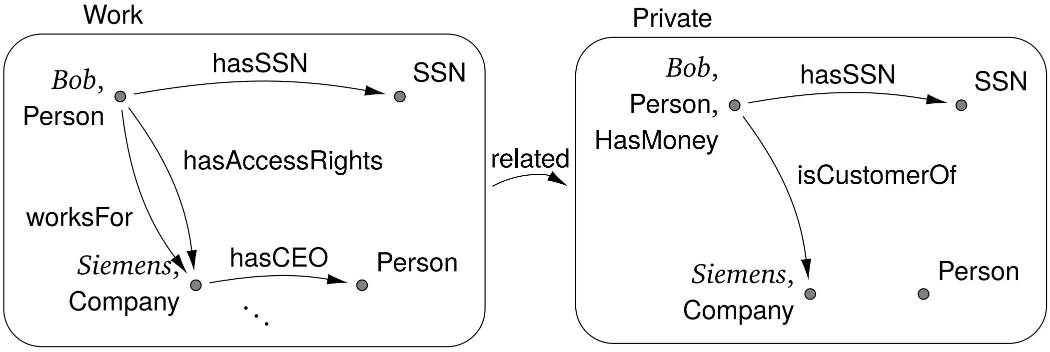


Figure 1.2: Nested interpretation that models of Axioms (1.7)– (1.13)

instance, that everybody who is employed by a company has a certain property in the context of private life. We show that reasoning in $\mathcal{L}_{\mathsf{M}}[\![\mathcal{L}_{\mathsf{O}}]\!]$ stays decidable with such a restriction, even in the presence of rigid roles. In some sense this restriction is similar to what is done in [BGL08; BGL12; Lip14] to obtain a decidable temporalised DL with rigid roles.

To provide a better intuition on how our formalism works, we examine the following example. Consider these axioms:

$$\top \sqsubseteq [\![\exists \mathsf{worksFor}.\{\mathit{Siemens}\} \sqsubseteq \exists \mathsf{hasAccessRights}.\{\mathit{Siemens}\}]\!] \tag{1.7}$$

$$\mathsf{Work} \sqsubseteq [\![\mathsf{worksFor}(\mathit{Bob}, \mathit{Siemens})]\!] \tag{1.8}$$

$$[\![(\exists \mathsf{worksFor}.\top)(\mathit{Bob})]\!] \sqsubseteq \exists \mathsf{related}.(\mathsf{Private} \sqcap [\![\mathsf{HasMoney}(\mathit{Bob})]\!]) \tag{1.9}$$

$$\top \sqsubseteq [\![\exists \mathsf{isCustomerOf}.\top \sqsubseteq \mathsf{HasMoney}]\!] \tag{1.10}$$

$$\mathsf{Private} \sqsubseteq [\![\mathsf{isCustomerOf}(\mathit{Bob}, \mathit{Siemens})]\!] \tag{1.11}$$

$$\mathsf{Private} \sqcap \mathsf{Work} \sqsubseteq \bot \tag{1.12}$$

$$\neg\mathsf{Work} \sqsubseteq [\![\exists \mathsf{worksFor}.\top \sqsubseteq \bot]\!] \tag{1.13}$$

The ‘outside’ or meta GCIs like $\top \sqsubseteq [\![\ldots]\!]$ express knowledge about the meta dimension whereas the axioms inside $[\![\ldots]\!]$ refer to knowledge in the object level. The complex meta concept $[\![\alpha]\!]$ describes all contexts in which the object axiom α holds. In detail, the first axiom states that in all contexts somebody who works for Siemens also has access rights to certain data. The second axiom states that Bob is an employee of Siemens in any work context. Furthermore, Axioms (1.9) and (1.10) say intuitively that if Bob has a job, he will earn money, which he can spend as a customer. Axiom (1.11) formalises that Bob is a customer of Siemens in any private context. Moreover, Axiom (1.12) ensures that the private contexts are disjoint from the work contexts. Finally, Axiom (1.13) states that the worksFor relation only exists in work contexts.

1.4 An Ontology Generator

Besides the capability of context description logics to formalise role-based models, it is rather hard for domain analysts—who in general are not experts in DLs—to grasp the precise semantics of the ontology, and to define the contextualised ontology in a way that all entities and constraints appearing in the role-based model are mapped correctly. Therefore, it would be ideal to have an algorithm that translates role-based models into context DL knowledge bases. In this thesis, we present exactly such a mapping.

First of all, we have to decide how to represent role-based models. Here, we focus on two essential properties of the modelling language. It is very important that the role-based model is already equipped with a formal semantics. Otherwise, we cannot ensure that the intended meaning of every model is correctly translated into the ontology. Furthermore, the modelling language needs to be expressive enough to express all concepts needed in the role-based model. The *Compartment Role Object Model (CROM)*[KLG+14; KBG+15] emerged to be a capable candidate that meets exactly the above mentioned requirements.

Next, there is some freedom on how to express roles and role-playing in an ontology. While it is important to consider the ontological nature of roles such as identity or rigidity, we also have to consider practical reasons. Whether some constraints of a role-based model can be expressed in an ontology highly depends on how roles and other predicates are translated.

On the one hand, we can and will prove the semantical correctness of our translation from role-based models into an $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$ ontology. On the other hand, for practical use there must exist some implementation of the mapping. We based our implementation on the reference implementation for CROM[1], which in turn can be used by FRaMED[2], a graphical editor allowing the specification of role-based models. In the end, our implemented mapping produces an ontology which is specially formatted in the Web Ontology Language (OWL). This leads to the last open part in the overall workflow.

1.5 A Reasoner for Contextualised Description Logics

A contextualised DL capable of formalising role-based models and an automated mapping from role-based models into an ontology still helps only very little in practice, if there es no DL reasoner available which can process such ontologies. Usually DL reasoners use OWL as language for the input ontology. But OWL in general does not have the syntactical means to express contextualised DL axioms. However, OWL enables us to annotate axioms which we use to encode $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$-axioms.

Although the reasoning tasks in DLs have a high complexity, DLs have been successfully introduced as a formalism for knowledge representation. One reason for the success of DLs is the availability of highly optimised reasoners which makes drawing logical inferences feasible. A black-box approach for deciding consistency of an $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$-ontology could benefit from these optimised reasoners. The main idea is to divide the consistency problem into separate subproblems each of which can be processed by a standard DL reasoner. Taking into account the special form of the axioms of the generated ontologies further optimisations are possible.

1.6 Outline of the Thesis

In the following, we give a short outline how the thesis is structured.

[1] `https://github.com/Eden-06/CROM`
[2] `https://github.com/leondart/FRaMED`

In the first section of Chapter 2, we present the basic definitions of description logics which we will use throughout the thesis. We define the syntax and semantics of concepts, axioms and knowledge bases and state specific DLs together with the respective complexities of the consistency problem. We then present an ontological overview of the notion of a role and introduce a syntactical variant of the Compartment Role Object Model, the modelling language we use for role-based modelling.

Chapter 3 starts with an discussion about the requirements for a logical formalism in order to be feasible in our setting, and we then introduce the contextualised description logic $\mathcal{L}_{\mathsf{M}}[\![\mathcal{L}_{\mathsf{O}}]\!]$. We again start with the definition of the syntax and semantics. For the complexity analysis of the consistency problem, the upper bounds are investigated first. We consider the case of the lightweight description logic $\mathcal{EL}$ in a separate section, which at the same time yields the lower bounds of the complexity for more expressive contextualised DLs. For the complexity analysis, we consider three different settings which result in different complexity results. In the first setting, we do not allow any rigid names. In the second setting, we assume the presence of rigid concepts, but forbid any rigid roles. In the last and complexity-wise hardest setting, we allow both rigid concept and rigid role names. At last, we introduce an extension of $\mathcal{L}_{\mathsf{M}}[\![\mathcal{L}_{\mathsf{O}}]\!]$, and show that the consistency problem becomes undecidable in the presence of rigid roles. The main results of this chapter, namely the complexity results excluding the description logic $\mathcal{SHOIQ}$, have already been published in [BL15b; BL15a; BL15c]:

- Stephan Böhme and Marcel Lippmann: 'Decidable Description Logics of Context with Rigid Roles'. In *Proc. of the 10th Int. Symp. on Frontiers of Combining Systems (FroCoS 2015)*, Wroclaw, Poland. Edited by Carsten Lutz and Silvio Ranise. Volume 9322. Lecture Notes in Artificial Intelligence. Springer-Verlag, Sept. 2015, pages 17–32
- Stephan Böhme and Marcel Lippmann: 'Decidable Contextualized DLs with Rigid Roles'. In *Proc. of the 28th Int. Workshop on Description Logics (DL 2015)*, Athens, Greece. Edited by Diego Calvanese and Boris Konev. Volume 1350. CEUR Workshop Proceedings. `CEUR-WS.org`, June 2015, pages 92–95
- Stephan Böhme and Marcel Lippmann: 'Description Logics of Context with Rigid Roles Revisited'. LTCS-Report 15-04. See `http://lat.inf.tu-dresden.de/research/reports.html`. Chair of Automata Theory, TU Dresden, 2015

In Chapter 4, we present the mapping from role-based models into $\mathcal{L}_{\mathsf{M}}[\![\mathcal{L}_{\mathsf{O}}]\!]$-ontologies. We explain in detail how the different assumptions of a CROM are translated and proof that the mapping algorithm preserves the semantics of the role-based model. We conclude the chapter with some thoughts on how to express constraints for role-based models that go beyond CROM. Both the formal role-based modelling language CROM and the mapping algorithm have already been published in [KBG+15; BK17]:

- Thomas Kühn, Stephan Böhme, Sebastian Götz, and Uwe Aßmann: 'A combined formal model for relational context-dependent roles'. In *Proc. of the 8th ACM SIGPLAN Int. Conf. on Software Language Engineering (SLE 2015)*, Pittsburgh, PA, USA. edited by Richard F. Paige, Davide Di Ruscio, and Markus Völter. ACM, Oct. 2015, pages 113–124
- Stephan Böhme and Thomas Kühn: 'Reasoning on Context-Dependent Domain Models'. In *Proc. of the 7th Joint Int. Conf. on Semantic Technology (JIST 2017)*, Gold Coast, QLD,

Australia. Edited by Zhe Wang, Anni-Yasmin Turhan, Kewen Wang, and Xiaowang Zhang. Volume 10675. Lecture Notes in Computer Science. Springer-Verlag, Nov. 2017, pages 69–85

Finally, in Chapter 5, we present JConHT, our reasoner for $\mathcal{L}_{\mathsf{M}}[\![\mathcal{L}_{\mathsf{O}}]\!]$. To implement a decision procedure for the consistency problem, we have to adapt some ideas of Chapter 3, which are discussed in Section 5.1. Section 5.2 covers the analysis of the contextualised hypertableau algorithm. We finish the chapter with some notes on the implementation and an evaluation of our implementation. Both the contextualised hypertableau algorithm and a system description of JConHT are also published in [BK17].

Chapter 2

Preliminaries

In this chapter, we will introduce the preliminaries which are necessary for the rest of the thesis. In Section 2.1, we introduce *description logics (DLs)* as a well-established logical formalism for knowledge representation and show the specific notations used in this thesis to emphasise parameters which are important later on. A short overview of the ontological foundations of roles followed by the presentation of the formal role-based modelling language CROM is given in Section 2.2.

2.1 Description Logics

Description logics are a family of knowledge representation formalisms. As already outlined in the introduction, DLs allow to represent application domains in a well-structured way. In this section, we present the notations, definitions and known results which are used in this thesis. For a more thorough introduction into description logics we refer the reader to [BCM+07].

2.1.1 Description Logic Concepts

As shown in Section 1.2, DL concepts describe sets of elements. Concepts are build from *concept names*, *role names* and *individual names* using concept and role constructors. Note that in the following definitions we refer to the triple $\mathsf{N} := (\mathsf{N_C}, \mathsf{N_R}, \mathsf{N_I})$ explicitly although it is usually left implicit in standard definitions. This turns out to be useful in Chapter 3 as we need to distinguish between different DLs and symbols used in the meta level and the object level. Sometimes we omit N, however, if they are irrelevant or clear from the context.

Definition 2.1 (Syntax of roles over N and concepts over N). *Let $\mathsf{N_C}$, $\mathsf{N_R}$, $\mathsf{N_I}$ be countably infinite, pairwise disjoint sets of* concept names*,* role names*, and* individual names*. Then, the triple $\mathsf{N} := (\mathsf{N_C}, \mathsf{N_R}, \mathsf{N_I})$ is a* signature*. A* role *r* over N *is either a role name, i.e. $r \in \mathsf{N_R}$, or it is of the form s^- with $s \in \mathsf{N_R}$ (*inverse role*).*

The set of concepts over N *is the smallest set such that*

- *for all $A \in \mathsf{N_C}$, then A is a concept over N (*atomic concept*), and*
- *if C and D are concepts over N, r is a role over N and $a \in \mathsf{N_I}$, then $\neg C$ (*negation*), $C \sqcap D$ (*conjunction*), $\exists r.C$ (*existential restriction*), $\{a\}$ (*nominal*) and $\geqslant_n r.C$ (*at-least restriction*) are concepts over* N. ◇

Non-atomic concepts are also called *complex concepts*. As usual in description logics, we use the following abbreviations:

- $C \sqcup D$ (*disjunction*) for $\neg(\neg C \sqcap \neg D)$,
- $\top$ (*top*) for $A \sqcup \neg A$ where $A \in \mathsf{N_C}$ is arbitrary but fixed,
- $\bot$ (*bottom*) for $\neg\top$,
- $\forall r.C$ (*value restriction*) for $\neg(\exists r.\neg C)$, and
- $\leqslant_n r.C$ (*at-most restriction*) for $\neg(\geqslant_{n+1} r.C)$.

The semantics of description logic concepts are defined in a model-theoretic way using the notion of interpretations.

Definition 2.2 (N-interpretation, Semantics of concepts over N). *Let* $\mathsf{N} := (\mathsf{N_C}, \mathsf{N_R}, \mathsf{N_I})$ *be the signature. Then, an* N-interpretation $\mathcal{I}$ *is a pair* $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ *where the* domain $\Delta^{\mathcal{I}}$ *is a non-empty set and the* interpretation function $\cdot^{\mathcal{I}}$ *maps*

- *every concept name* $A \in \mathsf{N_C}$ *to a set* $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$,
- *every role name* $r \in \mathsf{N_R}$ *to a binary relation* $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, *and*
- *every individual name* $a \in \mathsf{N_I}$ *to an element* $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ *such that different individual names are mapped to different elements, i.e. for* $a, b \in \mathsf{N_I}$ *it holds that* $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ *if* $a \neq b$.

This function is extended to inverse roles and complex concepts as follows:

- $(s^-)^{\mathcal{I}} := \left\{(d,c) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \,\middle|\, (c,d) \in s^-\right\}$,
- $(\neg C)^{\mathcal{I}} := \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$,
- $(C \sqcap D)^{\mathcal{I}} := C^{\mathcal{I}} \cap D^{\mathcal{I}}$,
- $(\exists r.C)^{\mathcal{I}} := \{d \in \Delta^{\mathcal{I}} \mid \text{there is an } e \in C^{\mathcal{I}} \text{ with } (d,e) \in r^{\mathcal{I}}\}$,
- $\{a\}^{\mathcal{I}} := \{a^{\mathcal{I}}\}$, *and*
- $(\geqslant_n r.C)^{\mathcal{I}} := \{d \in \Delta^{\mathcal{I}} \mid \sharp\{e \in C^{\mathcal{I}} \mid (d,e) \in r^{\mathcal{I}}\} \geq n\}$.

where $\sharp S$ *denotes the cardinality of the set* S. ◇

For any $x \in \mathsf{N_C} \cup \mathsf{N_R} \cup \mathsf{N_I}$, $x^{\mathcal{I}}$ is called the *extension of* x. Note that in the above definition of an interpretation, we adopt the so-called *unique name assumption* stating that every individual name is interpreted as a distinct element. By doing so, we emphasize that an individual name is meant to be the identity of an individual, rather than just a tag as it is usually used in the context of semantic web.

Now, we can look at a first example using the notions just defined.

Example 2.3. *Consider the following complex concept* C:

$$\begin{aligned}&\mathsf{NFL\_Team} \sqcap \neg\mathsf{AFC} \sqcap \geqslant_1 \mathsf{playsFor}^-.(\exists \mathsf{position}.\mathsf{Quarterback})\\ &\quad \sqcap \exists \mathsf{coaches}^-.\{\mathit{MikeMcCarthy}\}\end{aligned}$$

It describes all NFL teams which are not in the AFC, have at least one person playing for them at quarterback position and are coached by Mike McCarthy.

Figure 2.1 depicts an interpretation in which the Green Bay Packers are in the extension of the above concept. ◇

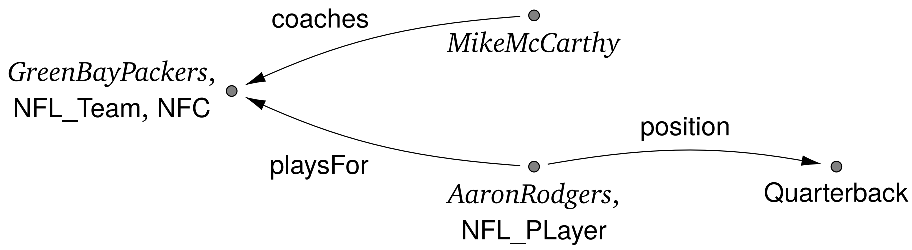


Figure 2.1: An Interpretation $\mathcal{I}$ such that $GreenBayPackers^{\mathcal{I}} \in C^{\mathcal{I}}$ and $\mathcal{I} \models \mathcal{O}$ with C from Example 2.3 and $\mathcal{O}$ from Example 2.6.

2.1.2 Boolean Knowledge Bases

With the notion of concepts at hand, we can formulate *axioms* to capture domain knowledge in a so-called *Boolean knowledge base (BKB)*. Each BKB consists of a Boolean combination of certain axioms and an RBox which states the general knowledge about roles.

Definition 2.4 (Syntax of axioms over N and BKBs over N). *Let $\mathsf{N} := (\mathsf{N_C}, \mathsf{N_R}, \mathsf{N_I})$ be the signature. Then, if C and D are concepts over N, r and s are roles over N, and $\{a, b\} \subseteq \mathsf{N_I}$, then*

- $C \sqsubseteq D$ *(*general concept inclusion, GCI*),*
- $C(a)$ *(*concept assertion*),*
- $r(a, b)$ *(*role assertion*),*
- $r \sqsubseteq s$ *(*role inclusion*), and*
- $\mathsf{trans}(r)$ *(*transitivity axiom*)*

are axioms over N. *Moreover, an* RBox $\mathcal{R}$ over N *is a finite set of role inclusions over* N *and transitivity axioms over* N. *A* Boolean axiom formula over N *is defined inductively as follows:*

- *every GCI over N is a Boolean axiom formula over N,*
- *every concept and role assertion over N is a Boolean axiom formula over N,*
- *if $\mathcal{B}_1$, $\mathcal{B}_2$ are Boolean axiom formulas over N, then so are $\neg\mathcal{B}_1$ (axiom negation) and $\mathcal{B}_1 \wedge \mathcal{B}_2$ (axiom conjunction), and*
- *nothing else is a Boolean axiom formula over N.*

Finally, a Boolean knowledge base (BKB) over N *is a pair $\mathfrak{B} = (\mathcal{B}, \mathcal{R})$, where $\mathcal{B}$ is a Boolean axiom formula over N and $\mathcal{R}$ is an RBox over N. An* ontology over N *is an BKB over N, where only axiom conjunction and no axiom negation is allowed in the Boolean axiom formula.* $\diamondsuit$

Again as usual in description logics, we use $C \equiv D$ (*concept equivalence*) as abbreviation for $(C \sqsubseteq D) \wedge (D \sqsubseteq C)$ and $\mathcal{B}_1 \vee \mathcal{B}_2$ (*axiom disjunction*) as abbreviation for $\neg(\neg\mathcal{B}_1 \wedge \neg\mathcal{B}_2)$. Often an ontology $\mathcal{O} = (\mathcal{B}, \mathcal{R})$ is written as a triple $\mathcal{O} = (\mathcal{T}, \mathcal{A}, \mathcal{R})$ where $\mathcal{T}$ (*TBox*) is the set of all GCIs occurring in $\mathcal{B}$ and $\mathcal{A}$ (*ABox*) is the set of all assertion axioms occurring in $\mathcal{B}$.

Definition 2.5 (Semantics of axioms over N, BKBs over N). *An* N*-interpretation* $\mathcal{I}$ *is a model of*

- *the GCI* $C \sqsubseteq D$ *over* N *if* $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$,
- *the concept assertion* $C(a)$ *over* N *if* $a^{\mathcal{I}} \in C^{\mathcal{I}}$,
- *the role assertion* $r(a,b)$ *over* N *if* $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$,
- *the role inclusion* $r \sqsubseteq s$ *over* N *if* $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$, *and*
- *the transitivity axiom* $\mathsf{trans}(r)$ *over* N *if* $r^{\mathcal{I}} = (r^{\mathcal{I}})^{+}$, *where* $\cdot^{+}$ *denotes the transitive closure of a binary relation.*

This is extended to Boolean axiom formulas over N *inductively as follows:*

- $\mathcal{I}$ *is a model of* $\neg\mathcal{B}_1$ *if it is not a model of* $\mathcal{B}_1$, *and*
- $\mathcal{I}$ *is a model of* $\mathcal{B}_1 \wedge \mathcal{B}_2$ *if it is a model of both* $\mathcal{B}_1$ *and* $\mathcal{B}_2$.

We write $\mathcal{I} \models \alpha$ *and* $\mathcal{I} \models \mathcal{B}$ *if* $\mathcal{I}$ *is a model of the axiom* α *over* N *or* $\mathcal{I}$ *is a model of the Boolean axiom formula* $\mathcal{B}$, *respectively. Furthermore,* $\mathcal{I}$ *is a model of an RBox* $\mathcal{R}$ *over* N *(written* $\mathcal{I} \models \mathcal{R}$*) if it is a model of each axiom in* $\mathcal{R}$.

Finally, $\mathcal{I}$ *is a model of the BKB* $\mathfrak{B} = (\mathcal{B}, \mathcal{R})$ *over* N *(written* $\mathcal{I} \models \mathfrak{B}$*) if it is a model of both* $\mathcal{B}$ *and* $\mathcal{R}$. *We call* $\mathfrak{B}$ consistent *if it has a model. The* consistency problem *is the problem of deciding whether a given BKB is consistent.* ◇

Note that besides the consistency problem there are several other reasoning tasks for description logics. The entailment problem, for instance, is the problem of deciding, given a BKB $\mathfrak{B}$ and an axiom β, whether $\mathfrak{B}$ *entails* β, i.e. whether all models of $\mathfrak{B}$ are also models of β. The consistency problem, however, is fundamental in the sense that most other standard reasoning tasks can be polynomially reduced to it in the presence of axiom negation. For example, the entailment problem can be reduced to the *in*consistency problem: $\mathfrak{B} = (\mathcal{B}, \mathcal{R})$ entails β iff $(\mathcal{B} \wedge \neg\beta, \mathcal{R})$ is inconsistent. If we consider only an ontology $\mathfrak{O} = (\mathcal{O}, \mathcal{R})$ without any axiom negations, we can still simulate most of the negated axioms $\neg\beta$. Let us for example consider the GCI $\beta = C \sqsubseteq D$. Here we can check $(\mathcal{O} \wedge (C \sqcap \neg D)(x), \mathcal{R})$ with $x \in \mathsf{N_I}$ not occurring in $\mathcal{O}$ for inconsistency to decide the entailment problem. Hence, we focus in this thesis only on the consistency problem.

To show an example of an ontology, we continue contentwise with American football.

Example 2.6. *Consider the following ontology* $\mathcal{O} = (\mathcal{B}, \emptyset)$ *with* $\mathcal{B} =$

$$
\begin{aligned}
&\mathsf{NFC}(\mathit{GreenBayPackers}) \quad \wedge \\
&\mathsf{playsFor}(\mathit{AaronRodgers}, \mathit{GreenBayPackers}) \quad \wedge \\
&\exists \mathsf{playsFor}.\mathsf{NFL\_Team} \sqsubseteq \mathsf{NFL\_Player} \quad \wedge \\
&\mathsf{NFL\_Team} \equiv \mathsf{NFC} \sqcup \mathsf{AFC} \quad \wedge \\
&\mathsf{NFC} \sqcap \mathsf{AFC} \sqsubseteq \bot.
\end{aligned}
$$

The first two axioms assert that the Green Bay Packers are in the NFC and that Aaron Rodgers plays for Green Bay. The third axiom states that everybody who plays for an NFL team is an

NFL player. Finally, the last two axioms define NFL teams as a disjoint union of the NFC and the AFC.

The ontology $\mathcal{O}$ is consistent and Figure 2.1 depicts a model of $\mathcal{O}$. Note here, that it is not coincidentally that Aaron Rodgers is in the extension of NFL_Player*, since $\mathcal{O}$ entails*

$$\mathsf{NFL\_Player}(\mathit{AaronRodgers}). \qquad \diamondsuit$$

2.1.3 Specific Description Logics

The specific description logics differ in the available concept and role constructors to formulate concepts and axioms, and also in the available axioms in a knowledge base.

The prototypical description logic is $\mathcal{ALC}$, the *attributive language with complement*. There are no inverse roles and only negation, conjunction and existential restriction are allowed as concept constructors. Furthermore, only GCIs, concept and role assertions are allowed as axioms. Hence, only role names are roles in $\mathcal{ALC}$ and an RBox in $\mathcal{ALC}$ is always the empty set. The DL $\mathcal{ALC}$ is the smallest propositionally closed DL [SS91]. Adding a letter to $\mathcal{ALC}$ stands for certain constructors or axioms that are additionally allowed. For example, $\mathcal{ALCI}$ additionally allows inverse roles in complex concepts. By the naming convention of DLs, specific letters denote a concept or role constructor or a type of axioms that is allowed in that DL:

- $\mathcal{O}$ means nominals,
- $\mathcal{I}$ means inverse roles,
- $\mathcal{Q}$ means at-least restrictions,
- $\mathcal{H}$ means role inclusions, and
- $\mathcal{S}$ means transitivity axioms.

$\mathcal{ALC}$ with additional transitivity axioms is called $\mathcal{S}$ instead of $\mathcal{ALCS}$, due to its connection to the modal logic S4. Thus, $\mathcal{SHOIQ}$, for example, is the DL that allows all constructors and axioms which are introduced above. Besides extensions of $\mathcal{ALC}$, there also exist many sublogics of $\mathcal{ALC}$ of which we only consider $\mathcal{EL}$ in this thesis. The sub-Boolean description logic $\mathcal{EL}$ is the fragment of $\mathcal{ALC}$ where only conjunction, existential restriction, and the top concept (which cannot be expressed as an abbreviation anymore due to the lack of negation) are admitted.

If necessary, we clarify the specific DL used by prefixing the specific DL name, e.g. an $\mathcal{ALCOIQ}$-concept can contain all concept constructors defined in Def. 2.1 and an $\mathcal{ALCH}$-RBox can only contain role inclusions, but no transitivity axioms.

In [HST00], it is shown for $\mathcal{SHQ}$ that allowing arbitrary roles in number restrictions leads to undecidability of the consistency problem. Decidability can be regained by restricting roles used in number restrictions to simple roles. To define what a *simple* role is, for a given BKB $\mathfrak{B} = (\mathcal{B}, \mathcal{R})$, we introduce $\sqsubseteq^*_{\mathcal{R}}$ as the transitive-reflexive closure of $\sqsubseteq$ on $\mathcal{R} \cup \{\mathsf{Inv}(r) \sqsubseteq \mathsf{Inv}(s) \mid r \sqsubseteq s \in \mathcal{R}\}$ where $\mathsf{Inv}(r)$ is defined as

$$\mathsf{Inv}(r) := \begin{cases} r^- & \text{if } r \in \mathsf{N_R}, \text{ and} \\ s & \text{if } r \text{ is an inverse role with } r = s^- \end{cases}$$

and $r \equiv_{\mathcal{R}} s$ as abbreviation for $r \sqsubseteq^*_{\mathcal{R}} s$ and $s \sqsubseteq^*_{\mathcal{R}} r$. A role r is *transitive w.r.t.* $\mathcal{R}$ if for some s with $r \equiv_{\mathcal{R}} s$, we have $\mathsf{trans}(s) \in \mathcal{R}$ or $\mathsf{trans}(\mathsf{Inv}(s)) \in \mathcal{R}$. A role r is called *simple w.r.t.* $\mathcal{R}$ if it is neither transitive nor has any transitive sub-role, i.e. there is no s such that $s \sqsubseteq^*_{\mathcal{R}} r$ and s is transitive w.r.t. $\mathcal{R}$.

In the rest of this thesis, we make this restriction to the syntax of $\mathcal{SHQ}$ and all its extensions. This restriction is also the reason why there are no Boolean combinations of role inclusions and transitivity axioms allowed in an RBox $\mathcal{R}$ over N in the above definition. Otherwise, the notion of a simple role w.r.t. $\mathcal{R}$ involves reasoning. Consider, for instance, the Boolean combination of axioms $(\mathsf{trans}(r) \lor \mathsf{trans}(s)) \land r \sqsubseteq s$. It should be clear that s is not simple, but this is no longer a pure syntactic check.

The complexity of the consistency problem for DL ontologies is well-investigated. Is is ExpTime-complete for any DL between $\mathcal{ALC}$ and $\mathcal{SHOQ}$ and NExpTime-complete for $\mathcal{SHOIQ}$. The lower bound for $\mathcal{ALC}$ was shown in [Sch91], the upper bound for $\mathcal{SHOQ}$ in [Tob01]. For $\mathcal{SHOIQ}$ the lower and upper bound were proven in [Tob00] and [Tob01], respectively. While for BKBs the complexity class stays the same, this is much less explored. It is in ExpTime for $\mathcal{SHOQ}$ [Lip14] and it remains in NExpTime for $\mathcal{SHOIQ}$ as a consequence of Theorem 2 in [Pra05].

For $\mathcal{EL}$-ontologies, the consistency problem is trivial since no contradictions can be expressed and, thus, every $\mathcal{EL}$-ontology is consistent. On the other hand, we do not consider $\mathcal{EL}$-BKBs as it seems very unnatural to admit axiom negation while denying concept negation at the same time.

2.2 Role-Based Modelling

In this section we present the essentials of role-based modelling needed in this thesis. By all means this is not a thorough introduction to role-based modelling and we assume that the reader is already familiar with the basic concepts. After discussing some ontological foundations of roles, we introduce the *Compartment Role Object Model (CROM)* as a modelling language with well-defined formal semantics. We mentioned the importance of formal semantics of the modelling language already in the introduction and will pick up the argument in Section 3.1 again.

To avoid the confusion with the notations, from now on we differentiate *rôles* as in rôle-based systems and *roles* as used in description logics whenever we feel it is necessary. Otherwise, we drop that distinction if it is clear from the context.

2.2.1 Ontological Foundation of Rôles

The word *Role* originated from the French word *Rôle* which referred to a form of rolled parchment on which the lines were written that an actor had to memorise. Since then, a role is a function assumed or part played by a person or thing in a certain situation.

Roles have been introduced in computer science already in 1977 by Bachman et al. He defined a role as a behaviour pattern which may be assumed by entities of different kinds. Since an entity can concurrently play several roles, the set of played roles characterise that entity.

Guarino approaches roles form an ontological point of view. He, among others, developed *OntoClean* [GW09], a methodology for analysing ontologies. He considers several domain-independent *metaproperties*, i.e. properties which describe *classes*. Here, a class is merely a set of instances, i.e. domain elements in a possible world, and a class itself can be an instance of a *metapredicate* such as *Role*. One metaproperty that is important for roles is *rigidity*. A class is *rigid* if all entities which are instances of that class are necessarily instances of that class in every possible world. For example, every instance of Person will always be a person, independent of the context or the time. But an instance of Student can cease to be a student. Another metaproperty is *dependence*. A class is dependent if each instance of it implies the existence of some other entity. The class Student can only have instances if there are also instances of Teacher. We here omit further metaproperties and their implications for metapredicates since they are not relevant in our setting and refer the interested reader to [GW00a; GW00b; GW00c; WG01].

The metapredicate *Role* implies non-rigidity and dependence. Thus, a class which is a role such as Student must not be rigid but dependent. That roles are dependent can be argued in two directions. First, one can say that each role needs its co-role, e.g. a student depends on a teacher or an orchestra musician depends on a conductor. On the other hand, a role always depends on a context, e.g. a student only exists in a school or university and an orchestra musician only exists in an orchestra. Anyhow, an instance of a role depends on the existence of another entity.

Besides the analysis of Guarino, Steimann [Ste00] introduces 15 features, depicted in Table 2.1, to classify roles. These features are completed by Kühn et al. in [KLG+14] by 11 additional features since Steimann neglected any features concerning role constraints or the contextual nature of roles. These features show how diverse roles can be seen, and that there is not one definition of what a role is. Based on these features, Kühn et al. [KLG+14] propose a so-called *feature model* to classify role-based modelling languages, i.e. several features organised in a tree shape from which a domain expert can select the features he needs. These features include, for example, whether role constraints such as role implication exist or whether the same role can be played several times by an entity.

The Compartment Role Object Model [KBG+15], which we focus on in this thesis, is one instance of this feature model.

2.2.2 A Formal Role-Based Modelling Language

We will present here a syntactical variant of the *Compartment Role Object Model* published in [KBG+15]. This variant is semantically equivalent to the CROM proposed in [KBG+15] but we introduce it here a bit different for easier explanation of the mapping to description logics.

Type and Instance Level

In CROM, we can model rôle-based systems via different kinds of predicates: *natural types*, *rôle types*, *compartment types* and *relationship types*. These differ in the the above mentioned metaproperties rigidity and dependence.

Table 2.1: Classifying features for roles [Ste00; KLG+14]

1.	Roles have properties and behaviour.
2.	Roles depend on relationships.
3.	Objects may play different roles simultaneously.
4.	Objects may play the same role (type) several times.
5.	Objects may acquire and abandon roles dynamically.
6.	The sequence of role acquisition and removal may be restricted.
7.	Unrelated objects can play the same role.
8.	Roles can play roles.
9.	Roles can be transferred between objects.
10.	The state of an object can be role-specific.
11.	Features of an object can be role-specific.
12.	Roles restrict access.
13.	Different roles may share structure and behavior.
14.	An object and its roles share identity.
15.	An object and its roles have different identities.
16.	Relationships between roles can be constrained.
17.	There may be constraints between relationships.
18.	Roles can be grouped and constrained together.
19.	Roles depend on Compartments.
20.	Compartments have properties and behaviors.
21.	A Role can be part of several Compartments.
22.	Compartments may play roles like objects.
23.	Compartments may play roles which are part of themselves.
24.	Compartments can contain other compartments.
25.	Different compartments may share structure and behavior.
26.	Compartments have their own identity.

- Natural types, e.g. Person or Table, are rigid and independent. Instances of natural types, called *naturals*, are instances of that type until they cease to exist. A table is always a table, independent of its function.
- Rôle types, e.g. Student, DiningTable or WorkDesk, are non-rigid and dependent. Instances of rôle types, called *rôles*, may be played by some entity in some context but not in another. But there always must be a context in which that rôle is played. A table might be used as a dining table in the context of a family celebration, i.e. the table plays the role of a dining table, whereas the same table usually is used as work desk.
- Compartment types, e.g. University or FamilyCelebration, are rigid and dependent. Intuitively, compartment types are objectified contexts. As long as an instance of a compartment type, called *compartment*, exists it is of that type. But unlike naturals a compartment depends on other entities, i.e. the rôles that are played within that compartment.
- Relationship types, e.g. supervise, are non-rigid and dependent. But in contrast to rôle types, which are unary predicates, relationship types are binary predicates. Hence, an instance of a relationship type depends on the existence of the two entities that are interrelated. An instance of *supervise* needs a professor who supervises and a student who is supervised.

Besides these different types, we can restrict which entities are allowed to play which rôles by a fills-relation which assigns each rigid type, i.e. natural types and compartment types, the set of rôle types, so that an instance of the rigid type can only play rôles of rôle types the rigid types fills. For example, besides tables also picnic blankets could play the rôle of a dining table in some contexts. But not only natural types can play rôles. For example, persons can play the rôle of an NFL player in the context of an NFL team, e.g. Aaron Rodgers is the quarterback in the context of the Green Bay Packers. The Green Bay Packers themselves as compartment can now play the rôle of a Super Bowl contender in the context of Super Bowl XLV.

In a CROM, a rôle type is also explicitly assigned to a single compartment type, in which it can be played. This is implemented through parts. Last, to each relationship type a pair of rôle types is assigned, which are the domain and the range of the relation. Here it is asserted that both rôle types are part of the same compartment type and that the same rôle type is not the domain and the range.

Definition 2.7 (Compartment Role Object Model). *Let* N_{NT}*,* N_{RT}*,* N_{CT} *and* $\mathsf{N}_{\mathsf{RST}}$ *be finite and mutually disjoint sets of Natural Types, Role Types, Compartment Types, and Relationship Types, respectively. The tuple* $\Sigma = (\mathsf{N}_{\mathsf{NT}}, \mathsf{N}_{\mathsf{RT}}, \mathsf{N}_{\mathsf{CT}}, \mathsf{N}_{\mathsf{RST}})$ *is the* vocabulary*. A* Compartment Role Object Model $\mathcal{M}$ over Σ (Σ-CROM) *is a tuple* $\mathcal{M} = (\mathsf{fills}, \mathsf{parts}, \mathsf{rel})$ *where*

1. $\mathsf{fills} \subseteq (\mathsf{N}_{\mathsf{NT}} \cup \mathsf{N}_{\mathsf{CT}}) \times \mathsf{N}_{\mathsf{RT}}$ *is a right-total binary relation,*
2. $\mathsf{parts} : \mathsf{N}_{\mathsf{CT}} \to \mathfrak{P}$ *is a bijection where* $\mathfrak{P}$ *is an arbitrary but fixed partition of the set* N_{RT}*, and*
3. $\mathsf{rel} : \mathsf{N}_{\mathsf{RST}} \to \mathcal{S}$ *is a bijection where* $\mathcal{S} \subseteq \bigcup_{P \in \mathfrak{P}} P \times P$ *is an irreflexive binary relation.* $\diamondsuit$

In the above definition, fills specifies which rigid type is allowed to play which role type and parts expresses in which compartment type a certain role type can be played. Finally, rel defines the domain and range of relationship types. In the rest of this thesis, we use the following phrases:

- *T fills RT* if $(T, RT) \in \mathsf{fills}$,
- *RT participates in CT* if $RT \in \mathsf{parts}(CT)$,
- *RST participates in CT* if $\mathsf{rel}(RST) = (RT_1, RT_2)$ with $\{RT_1, RT_2\} \subseteq \mathsf{parts}(CT)$,
- *RT_1 and RT_2 are related via RST* if $(RT_1, RT_2) = \mathsf{rel}(RST)$, and
- *RT_1 is the domain of RST* *(*$\mathsf{dom}(RST)$*) and RT_2 is the range of RST (*$\mathsf{ran}(RST)$*)* if $(RT_1, RT_2) \in \mathsf{rel}(RST)$.

Note here, that the above definition of Σ-CROM always ensures *well-formedness* as defined in Definition 1 of [KBG+15]. This is unproblematic since reasoning about role-based models does not include checking well-formedness as this is a pure syntactical check.

We will now start to introduce an example which we will use throughout this chapter to explain some interesting aspects.

Example 2.8. *We consider a banking application. In the context of a bank we have consultants and customers where either persons or companies can be customers of a bank, but only persons* *can be consultants. A customer can own savings or checking accounts. These rôles can be attained by any physical entity which is an account. Accounts can, moreover, be the source or the target in the context of a transaction. Transactions in turn can have the function of a money transfer in the context of a bank and customers can issue such money transfers.*

Hence, we have the following vocabulary $\Sigma_{\mathsf{Bank}} = (\mathsf{N}_{\mathsf{NT,Bank}}, \mathsf{N}_{\mathsf{RT,Bank}}, \mathsf{N}_{\mathsf{CT,Bank}}, \mathsf{N}_{\mathsf{RST,Bank}})$ *with*

$$\begin{aligned}
\mathsf{N}_{\mathsf{NT,Bank}} &:= \{\mathsf{Person}, \mathsf{Company}, \mathsf{Account}\},\\
\mathsf{N}_{\mathsf{RT,Bank}} &:= \{\mathsf{Consultant}, \mathsf{Customer}, \mathsf{CheckingAccount}, \mathsf{SavingsAccount}, \mathsf{MoneyTransfer},\\
&\quad\ \mathsf{Source}, \mathsf{Target}\},\\
\mathsf{N}_{\mathsf{CT,Bank}} &:= \{\mathsf{Bank}, \mathsf{Transaction}\},\ \textit{and}\\
\mathsf{N}_{\mathsf{RST,Bank}} &:= \{\mathsf{advises}, \mathsf{own_ca}, \mathsf{own_sa}, \mathsf{issues}, \mathsf{trans}\}.
\end{aligned}$$

Furthermore, fills*,* parts*, and* rel *are defined as follows:*

$$\begin{aligned}
\mathsf{fills} &:= \{(\mathsf{Person}, \mathsf{Consultant}), (\mathsf{Person}, \mathsf{Customer}), (\mathsf{Company}, \mathsf{Customer}),\\
&\quad\ (\mathsf{Account}, \mathsf{SavingsAccount}), (\mathsf{Account}, \mathsf{CheckingAccount}),\\
&\quad\ (\mathsf{Account}, \mathsf{Source}), (\mathsf{Account}, \mathsf{Target}), (\mathsf{Transaction}, \mathsf{MoneyTransfer})\}\\
\mathsf{parts}(\mathsf{Bank}) &:= \{\mathsf{Consultant}, \mathsf{Customer}, \mathsf{CheckingAccount}, \mathsf{SavingsAccount},\\
&\quad\ \mathsf{MoneyTransfer}\}\\
\mathsf{parts}(\mathsf{Transaction}) &:= \{\mathsf{Source}, \mathsf{Target}\}\\
\mathsf{rel}(\mathsf{advises}) &:= (\mathsf{Consultant}, \mathsf{Customer})\\
\mathsf{rel}(\mathsf{own_ca}) &:= (\mathsf{Customer}, \mathsf{CheckingAccount})\\
\mathsf{rel}(\mathsf{own_sa}) &:= (\mathsf{Customer}, \mathsf{SavingsAccount})\\
\mathsf{rel}(\mathsf{issues}) &:= (\mathsf{Customer}, \mathsf{MoneyTransfer})\\
\mathsf{rel}(\mathsf{trans}) &:= (\mathsf{Source}, \mathsf{Target})
\end{aligned}$$

Hence, for example Person *fills* Customer*, and* Customer *and* advises *participate in* Bank. Customer *and* SavingsAccount *are related via* own_sa *since the domain of* own_sa *is* Customer *and the range of* own_sa *is* SavingsAccount.

Figure 2.2 depicts the whole example in graphical notation including some constraints which we will introduce in the next sections. ◇

In our definitions, we omit to precisely define the graphical notation as they are not relevant for reasoning on CROMs and refer to [KBG+15].

Next, we introduce instances of role-based models. An instance is based on a non-empty domain, where each element is of exactly one type, i.e. a natural type, a rôle type or a compartment type. Objects playing rôle in a compartment are collected in a ternary relation plays and the relation of two rôles via a relationship type in a compartment is stored in links.

Definition 2.9 (Compartment Role Object Instance, Satisfiability). *Let* $\Sigma = (\mathsf{N}_{\mathsf{NT}}, \mathsf{N}_{\mathsf{RT}}, \mathsf{N}_{\mathsf{CT}}, \mathsf{N}_{\mathsf{RST}})$ *be a vocabulary. Then, a* Compartment Role Object Instance $\mathcal{I}$ over Σ (Σ-CROI) *is a tuple* $\mathcal{I} = (\Gamma^{\mathcal{I}}, \mathsf{type}, \mathsf{plays}, \mathsf{links})$*, where*

- $\Gamma^{\mathcal{I}}$ *is a non-empty domain, and*

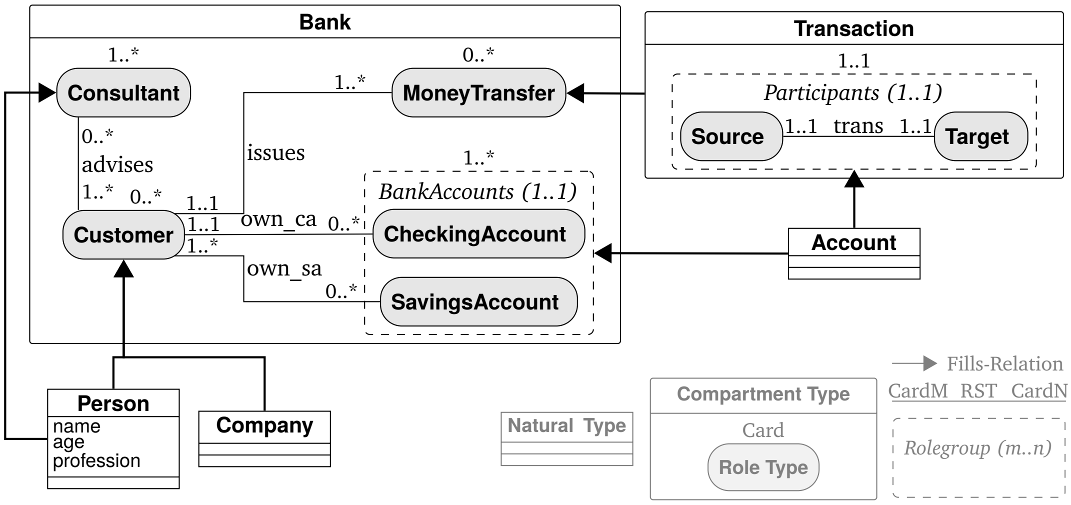


Figure 2.2: Graphical notation of a CROM for an banking application

- $\mathsf{type} : \Gamma^{\mathcal{I}} \to \mathsf{N_{NT}} \cup \mathsf{N_{RT}} \cup \mathsf{N_{CT}}$ *is a total function.*

Based on the type*-function, we can partition the domain into the set* $\mathsf{N}^{\mathcal{I}}$ *of* naturals, *i.e. all instances of any natural type, the set* $\mathsf{R}^{\mathcal{I}}$ *of* rôles, *i.e. all instances of any rôle type, and the set* $\mathsf{C}^{\mathcal{I}}$ *of* compartments, *i.e. all instances of any compartment type. Furthermore, the set* $\mathsf{O}^{\mathcal{I}}$ *of* objects *denote all domain elements that are eligible to play a rôle, i.e. all naturals and compartments.*

$$\begin{aligned}
\mathsf{N}^{\mathcal{I}} &:= \{d \in \Gamma^{\mathcal{I}} \mid \mathsf{type}(d) \in \mathsf{N_{NT}}\} \\
\mathsf{R}^{\mathcal{I}} &:= \{d \in \Gamma^{\mathcal{I}} \mid \mathsf{type}(d) \in \mathsf{N_{RT}}\} \\
\mathsf{C}^{\mathcal{I}} &:= \{d \in \Gamma^{\mathcal{I}} \mid \mathsf{type}(d) \in \mathsf{N_{CT}}\} \\
\mathsf{O}^{\mathcal{I}} &:= \mathsf{N}^{\mathcal{I}} \cup \mathsf{C}^{\mathcal{I}}
\end{aligned}$$

Now, plays *and* links *are defined as follows:*

- $\mathsf{plays} \subseteq \mathsf{O}^{\mathcal{I}} \times \mathsf{C}^{\mathcal{I}} \times \mathsf{R}^{\mathcal{I}}$ *is a ternary relation, and*
- $\mathsf{links} : (\mathsf{N_{RST}} \times \mathsf{C}^{\mathcal{I}}) \to \mathcal{P}(\mathsf{R}^{\mathcal{I}} \times \mathsf{R}^{\mathcal{I}})$ *is a total function.*

Furthermore, the set $T^{\mathcal{I}}$ *of all elements of type* $T \in (\mathsf{N_{NT}} \cup \mathsf{N_{RT}} \cup \mathsf{N_{CT}})$*, the set* $\mathsf{O}^{\mathcal{I},c}$ *of all objects playing a role in c, the set* $\mathsf{O}^{\mathcal{I},c,RT}$ *of all objects playing an RT-role in c, and the set* $\mathsf{R}^{\mathcal{I},c}$ *of all roles played in c are defined as follows:*

$$\begin{aligned}
T^{\mathcal{I}} &:= \{d \in \Gamma \mid \mathsf{type}(d) = T\}, \\
\mathsf{O}^{\mathcal{I},c} &:= \{o \in \mathsf{O}^{\mathcal{I}} \mid \textit{there is some } r \textit{ with } (o,c,r) \in \mathsf{plays}\}, \textit{ and} \\
\mathsf{O}^{\mathcal{I},c,RT} &:= \{o \in \mathsf{O}^{\mathcal{I}} \mid \textit{there is some } r \textit{ with } (o,c,r) \in \mathsf{plays} \textit{ and } r \in RT^{\mathcal{I}}\} \\
\mathsf{R}^{\mathcal{I},c} &:= \{r \in \mathsf{R}^{\mathcal{I}} \mid \textit{there is some } o \textit{ with } (o,c,r) \in \mathsf{plays}\}.
\end{aligned}$$

A Σ*-CROI* $\mathcal{I}$ satisfies *a* Σ*-CROM* $\mathcal{M}$*, denoted by* $\mathcal{I} \models \mathcal{M}$*, if it has the following properties:*

1. *The* plays*-relation respects* fills*, i.e. for each tuple* $(o,c,r) \in \mathsf{plays}$ *the type of o fills the type of r:*

$$\{(o,r) \mid (o,\cdot,r) \in \mathsf{plays}\} \subseteq \{(o,r) \mid \textit{there exists } (T,RT) \in \mathsf{fills} \textit{ s.t. } o \in T^{\mathcal{I}},\ r \in RT^{\mathcal{I}}\}.$$

2. *The* plays*-relation respects* parts*, i.e. for each tuple* $(o,c,r) \in \mathsf{plays}$ *the type of r participates in the type of c:*

$$\begin{aligned}&\{(c,r) \mid (\cdot,c,r) \in \mathsf{plays}\}\\ &\quad\subseteq \{(c,r) \mid \textit{there exists } CT \in \mathsf{N_{CT}},\ RT \in \mathsf{N_{RT}} \textit{ s.t. } c \in CT^{\mathcal{I}},\ r \in RT^{\mathcal{I}},\ RT \in \mathsf{parts}(CT)\}.\end{aligned}$$

3. *Each object can only play one role of each role type in each compartment:*

$$\{(o,c,r),(o,c,r')\} \subseteq \mathsf{plays} \quad \textit{implies} \quad \mathsf{type}(r) \neq \mathsf{type}(r').$$

4. *Each role is played by exactly one object in exactly one compartment:*

$$|\{(o,c) \mid (o,c,r) \in \mathsf{plays}\}| = 1 \quad \textit{for all } r \in \mathsf{R}^{\mathcal{I}}.$$

5. *Roles occurring in the image of* links *are played in the associated compartment, i.e. for each* $(r_1,r_2) \in \mathsf{links}(RST,c)$ *there exists objects that play* r_1 *and* r_2 *in c:*

$$\{r_1 \mid (r_1,\cdot) \in \mathsf{links}(\cdot,c)\} \cup \{r_2 \mid (\cdot,r_2) \in \mathsf{links}(\cdot,c)\} \subseteq \{r \mid (\cdot,c,r) \in \mathsf{plays}\}.$$

6. *The* links*-function respects* rel*, i.e. for each* $(r_1,r_2) \in \mathsf{links}(RST,\cdot)$ *the types of* r_1 *and* r_2 *are related via RST:*

$$(r_1,r_2) \in \mathsf{links}(RST,\cdot) \quad \textit{implies} \quad \mathsf{rel}(RST) = (\mathsf{type}(r_1),\mathsf{type}(r_2)).$$

A Σ*-CROM* $\mathcal{M}$ *is* satisfiable *if there exists any* Σ*-CROI* $\mathcal{I}$ *such that* $\mathcal{I} \models \mathcal{M}$. ◇

We say that r is an RT-role and c is a CT-compartment if, respectively, $\mathsf{type}(r) = RT \in \mathsf{N_{RT}}$ and $\mathsf{type}(c) = CT \in \mathsf{N_{CT}}$. Furthermore, *o plays r in c* and o is the *player* of r if $(o,c,r) \in \mathsf{plays}$, and r_1 *is linked to* r_2 *via RST in c* if $(r_1,r_2) \in \mathsf{links}(RST,c)$.

Before we investigate how the information about a Σ-CROM can be encoded in a description logic ontology, we have to discuss the main reasoning tasks for role-based models. The arguably most apparent question is, given a Σ-CROM $\mathcal{M}$ and a Σ-CROI $\mathcal{I}$, whether $\mathcal{I}$ is compliant with $\mathcal{M}$. But as this task rather belongs to the area of model checking, we will not focus on that problem in this thesis. Instead, given a Σ-CROM $\mathcal{M}$, it is more interesting whether there exists any Σ-CROI that is compliant with $\mathcal{M}$. Additionally, we often want to know for a specific Σ-CROM $\mathcal{M}$ whether there exists a compliant Σ-CROI that fulfills certain assertions, e.g. that a role of a certain type is played. To express this assertional knowledge, we introduce a so-called Σ-CROA, a finite set of assertions which should additionally be satisfied by a Σ-CROI.

Definition 2.10 (Σ-Compartment Role Object Assertions). *Let $\Sigma = (\mathsf{N_{NT}}, \mathsf{N_{RT}}, \mathsf{N_{CT}}, \mathsf{N_{RST}})$ be a vocabulary and let $\mathsf{N_{M\text{-}IND}}$ and $\mathsf{N_{O\text{-}IND}}$ be two non-empty, disjoint sets of meta and object individual names disjoint from Σ. A* Compartment Role Object Assertion over Σ *is of the form*

- *$T(c)$ with $T \in \mathsf{N_{CT}}$ and $c \in \mathsf{N_{M\text{-}IND}}$ (*meta type assertion*),*
- *$T(a,c)$ with $T \in \mathsf{N_{NT}} \cup \mathsf{N_{CT}} \cup \mathsf{N_{RT}}$, $a \in \mathsf{N_{O\text{-}IND}}$ and $c \in \mathsf{N_{M\text{-}IND}}$ (*object type assertion*),*
- *$\mathsf{play\_assert}(a_1, c, a_2)$ with $a_1, a_2 \in \mathsf{N_{O\text{-}IND}}$ and $c \in \mathsf{N_{M\text{-}IND}}$ (*plays assertion*), or*
- *$\mathsf{link\_assert}(RST, c, a_1, a_2)$ with $RST \in \mathsf{N_{RST}}$ and $a_1, a_2 \in \mathsf{N_{O\text{-}IND}}$ and $c \in \mathsf{N_{M\text{-}IND}}$ (*link assertion*).*

A set of Compartment Role Object Assertions $\mathcal{A}$ over Σ (Σ-CROA) *is a finite set of such assertions. We extend the Σ-CROI $\mathcal{I}$ to additionally map individual names to domain elements, e.g. $a \in \mathsf{N_{O\text{-}IND}}$ and $c \in \mathsf{N_{M\text{-}IND}}$ are mapped to a domain elements $a^{\mathcal{I}} \in \Gamma^{\mathcal{I}}$ and $c^{\mathcal{I}} \in \Gamma^{\mathcal{I}}$. A Σ-CROI $\mathcal{I}$* satisfies an assertion *α, denoted by $\mathcal{I} \models \alpha$, if the following conditions hold:*

- *if $\alpha = T(c)$, then $c^{\mathcal{I}} \in T^{\mathcal{I}}$,*
- *if $\alpha = T(a,c)$, then $a^{\mathcal{I}} \in T^{\mathcal{I}} \cap (\mathsf{O}^{\mathcal{I}} \cup \mathsf{R}^{\mathcal{I},c})$,*
- *if $\alpha = \mathsf{play\_assert}(a_1, c, a_2,)$, then $(a_1^{\mathcal{I}}, c^{\mathcal{I}}, a_2^{\mathcal{I}}) \in \mathsf{plays}$, and*
- *if $\alpha = \mathsf{link\_assert}(RST, c, a_1, a_2)$, then there exist $r_1, r_2 \in \mathsf{R}^{\mathcal{I}}$ with $(a_1, c, r_1) \in \mathsf{plays}$, $(a_2, c, r_2) \in \mathsf{plays}$ and $(r_1^{\mathcal{I}}, r_2^{\mathcal{I}}) \in \mathsf{links}(RST, c^{\mathcal{I}})$.*

A Σ-CROI $\mathcal{I}$ satisfies *$\mathcal{A}$, denoted by $\mathcal{I} \models \mathcal{A}$ if it satisfies all assertions in $\mathcal{A}$.* ◇

Note here that the link assertion asserts for two objects that they play roles which are related via RST, and not that the objects themselves are related. Moreover, without any assertions there always exists a trivial CROI that satisfies $\mathcal{M}$ with the singleton set $\Gamma = \{o\}$ where the type of o is some natural type, and plays and links are empty sets. Therefore, we introduce in the next section further constraints.

Constraint Level

When modelling a domain of interest, not only the type of an object defines whether that object is allowed to play a certain role. In [KBG+15] additional constraints were introduced. These can be divided into four groups.

Role constraints are the first category of constraints which state, for example, that roles mutually exclude each other or playing one role implies playing another role. More general these constraints are formalised with so-called *role groups*. These consist of a set of role types (or again role groups), a lower and an upper bound. An object fulfills a role group if it plays at least the lower and at most the upper bound of roles from the set of role types.

Definition 2.11 (Syntax of role groups). *Let $\mathsf{N_{RT}}$ be a set of role types. The set of* role groups over $\mathsf{N_{RT}}$ *is the smallest such that*

- *if $RT \in \mathsf{N_{RT}}$, then RT is an (*atomic*) role group, and*
- *if $A_1, \ldots, A_n$ are role groups, $k, \ell \in \mathbb{N}$, then $(\{A_1, \ldots, A_n\}, k, \ell)$ is a (*complex*) role group.*

Atoms *of a role group A are defined as:*

$$\mathsf{atom}(A) := \begin{cases} \{RT\} & \text{if } A = RT \in \mathsf{N}_{\mathsf{RT}} \\ \bigcup_{i=1}^{n} \mathsf{atom}(A_i) & \text{if } A = (\{A_1, \ldots, A_n\}, k, \ell). \end{cases}$$

Role groups that occur within other role groups are called nested. ♢

The semantics of a role group are based on a Σ-CROI and are locally evaluated for each domain element and each compartment. The interpretation function $\cdot^{\mathcal{I},c,o}$ calculates recursively whether an object fulfills the role group.

Definition 2.12 (Semantics of role groups). *Given a Σ-CROI $\mathcal{I}$, the semantics of a role group A is defined for an object $o \in \mathsf{O}^{\mathcal{I}}$ in $c \in \mathsf{C}^{\mathcal{I}}$ as follows:*

$$A^{\mathcal{I},c,o} := \begin{cases} 1 & \text{if } A = RT \in \mathsf{N}_{\mathsf{RT}} \text{ and } o \text{ plays an } RT\text{-role in } c, \text{ or} \\ & \text{if } A = (\{B_1, \ldots, B_n\}, k, \ell) \text{ and } k \leq \sum_{i=1}^{n} B_i^{\mathcal{I},c,o} \leq \ell, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

If $A^{\mathcal{I},c,o} = 1$, we say that o fulfills A in c. ♢

Basic role constraints, for example as defined in [RG98], i.e. role implication, role equivalence and role prohibition, can be expressed with role groups as well as much more complex ones. In fact, any propositional formula can be emulated with role groups.

Proposition 2.13. *Let φ be some propositional formula. Then, there exists a role group A_φ such that φ is satisfiable if and only if A_φ can be fulfilled.*

Proof. We define A_φ inductively as follows:

$$\begin{array}{lll} \text{if } \varphi = p & \text{then} & A_\varphi := RT_p, \\ \text{if } \varphi = \neg\psi & \text{then} & A_\varphi := (\{A_\psi\}, 0, 0), \\ \text{if } \varphi = \psi_1 \wedge \psi_2 & \text{then} & A_\varphi := (\{A_{\psi_1}, A_{\psi_2}\}, 2, 2), \text{ and} \\ \text{if } \varphi = \psi_1 \vee \psi_2 & \text{then} & A_\varphi := (\{A_{\psi_1}, A_{\psi_2}\}, 1, 2). \end{array}$$

Next, we establish a 1-to-1-relation between a valuation ρ for φ and a Σ-CROI $\mathcal{I}$. For every propositional variable P_i occurring in φ, we introduce a role type RT_i and assume that o plays an RT_i-role iff $\rho(P_i) = \mathsf{true}$. By induction, it follows that $\rho(\varphi) = \mathsf{true}$ iff o fulfills A_φ. □

The next category of constraints are *occurrence constraints*. These state how often a role type or role group must at least or at most be played in a compartment. Therefore, we introduce the notion of a *cardinality*, a pair $(k, \ell) \in \mathbb{N} \times \mathbb{N}_\infty$ with $k \leq \ell$. We usually denote cardinalities by $(k..\ell)$. Since role types are also atomic role groups, it suffices to specify occurrence constraints for role groups. Similar to multiplicities specified for associations in UML class diagrams, we specify *cardinality constraints* for relationship types. They express how often a role of certain type must be related via a relationship type to some other role type.

Last but not least, the category of *intra-relationship type constraints* imposes constraints on the players of roles which are related via a relationship type. For example, stating that the relationship type isAncestorOf between the role types Parent and Child is transitive assures the existence of a respective link between a grandparent and a grandchild. Note here, that the transitivity is evaluated over the players and not the roles themselves.

Definition 2.14 (Constraint set). *Let* $\Sigma = (\mathsf{N_{NT}}, \mathsf{N_{RT}}, \mathsf{N_{CT}}, \mathsf{N_{RST}})$ *be a vocabulary, let* $\mathbb{RG}$ *be the set of role groups over* $\mathsf{N_{RT}}$ *and let* $\mathsf{Card} := \mathbb{N} \times \mathbb{N}_\infty$ *be the set of cardinalities. Then, a* Σ-Compartment Role Object Constraint Set (Σ-CROC) $\mathcal{C}$ *is a tuple* $\mathcal{C} = (\mathsf{occur}, \mathsf{card}, \mathsf{intra})$ *where*

- $\mathsf{occur} : \mathsf{N_{CT}} \to \mathcal{P}(\mathsf{Card} \times \mathbb{RG})$,
- $\mathsf{card} : \mathsf{N_{RST}} \to \mathsf{Card} \times \mathsf{Card}$*, and*
- $\mathsf{intra} : \mathsf{N_{RST}} \to \mathcal{P}(\mathcal{E})$ *with* $\mathcal{E}$ *being a set of functions of the form* $e : \mathcal{P}(A \times B) \to \{\mathsf{true}, \mathsf{false}\}$ *for arbitrary sets A, B.*

are total functions. The set of all non-nested role groups that appear in occur *is the set of* top-level role groups $\mathbb{RG}^\top$. *A* Σ*-CROC* $\mathcal{C}$ *is* compliant *to a* Σ*-CROM if all atoms of a role group that is in the occurrence constraints of a compartment type participate in that compartment type. Before we define satisfiability, we introduce the following auxiliary functions:*

$$\begin{aligned}
\mathsf{succ}(RST, r, c) &:= \{r' \in \mathsf{R}^\mathcal{I} \mid (r, r') \in \mathsf{links}(RST, c)\},\\
\mathsf{pred}(RST, r, c) &:= \{r' \in \mathsf{R}^\mathcal{I} \mid (r', r) \in \mathsf{links}(RST, c)\}, \text{ and}\\
\mathsf{links}^*(RST, c) &:= \{(o_1, o_2) \mid (r_1, r_2) \in \mathsf{links}(RST, c) \text{ and } (o_1, c, r_1), (o_2, c, r_2) \in \mathsf{plays}\}.
\end{aligned}$$

A Σ*-CROI* $\mathcal{I}$ satisfies $\mathcal{C}$*, denoted by* $\mathcal{I} \models \mathcal{C}$*, if it has the following properties:*

1. *All occurrence constraints are respected, i.e. if* $(k..\ell, A) \in \mathsf{occur}(CT)$*, then in every CT-compartment there must exist at least k and at most* ℓ *objects that fulfill role group A:*

$$((k,\ell),A) \in \mathsf{occur}(CT) \quad \text{implies} \quad CT^\mathcal{I} \subseteq \left\{ c \in \mathsf{C}^\mathcal{I} \;\middle|\; k \leq \sum\nolimits_{o \in \mathsf{O}^{\mathcal{I},c}} A^{\mathcal{I},c,o} \leq \ell \right\}$$

2. *All top-level rôle groups must be satisfied, i.e. if an object o plays an RT-role and RT is an atom of a top-level rôle group A, then o must fulfill A:*

$$(o, c, r) \in \mathsf{plays},\ r \in RT^\mathcal{I} \text{ and } RT \in \mathsf{atom}(A) \quad \text{implies} \quad A^{\mathcal{I},c,o} = 1, \qquad \text{for all } o \in \mathsf{O}^\mathcal{I},\ A \in \mathbb{RG}^\top.$$

3. *All cardinality constraints are respected, i.e. every rôle that is played in a compartment c and whose type is either the domain or the range of a relationship type RST with* $\mathsf{card}(RST) = (i..j, k..\ell)$ *must have at least k and at most* ℓ *RST-successors in c or at least i and at most j RST-predecessors in c, respectively:*

$$(\cdot, c, r) \in \mathsf{plays},\ r \in {RT_1}^\mathcal{I},\ \mathsf{rel}(RST) = (RT_1, \cdot) \text{ and } \mathsf{card}(RST) = (\cdot, k..\ell) \\ \text{implies} \quad k \leq |\mathsf{succ}(RST, r, c)| \leq \ell$$

$$(\cdot, c, r) \in \mathsf{plays},\ r \in {RT_2}^\mathcal{I},\ \mathsf{rel}(RST) = (\cdot, RT_2) \text{ and } \mathsf{card}(RST) = (i..j, \cdot) \\ \text{implies} \quad i \leq |\mathsf{pred}(RST, r, c)| \leq j$$

4. *All intra-relationship type constraints are respected, i.e. every function* $\mathsf{f} \in \mathsf{intra}(RST)$*, evaluated over the players of the roles related via RST, must return* true*:*

$$\mathsf{f} \in \mathsf{intra}(RST) \quad \textit{implies} \quad \mathsf{f}(\mathsf{links}^*(RST, c)) = \mathsf{true}$$
$$\textit{for all } c \textit{ in } CT^{\mathcal{I}} \textit{ s.t. } RST \textit{ participates in } CT. \qquad \diamondsuit$$

The definition of satisfying a constraint model in [KBG+15] is, neglecting ε-roles, exactly reflected in the above definition. With constraints being formally introduced, we can complete the CROM for the banking application of Example 2.8 and Figure 2.2.

Example 2.15. *We first define the complex rôle groups that occur in our example. The rôle group* Participants *in the context of an transaction ensures that a single account cannot be both the source and the target of an transaction. The rôle group* BankAccounts *states the same for savings and checking accounts in the context of a bank. A single account cannot attain both rôles simultaneously.*

$$\mathsf{Participants} := (\{\mathsf{Source}, \mathsf{Target}\}, 1, 1)$$
$$\mathsf{BankAccounts} := (\{\mathsf{SavingsAccount}, \mathsf{CheckingAccount}\}, 1, 1)$$

We next analyse the occurrence constraints. In our example a bank can only exists if it contains at least on consultant and one bank account. A transaction has exactly one source or target[1]*.*

$$\mathsf{occur}(\mathsf{Bank}) := \{(1..\infty, \mathsf{Consultant}), (1..\infty, \mathsf{BankAccounts})\}$$
$$\mathsf{occur}(\mathsf{Transaction}) := \{(1..1, \mathsf{Participants})\}$$

For the appearing cardinality constraints we assume the following in the context of a bank. A consultant must advise at least one customer but not every customer necessarily needs a consultant. A customer does not need to have any bank account, but every bank account needs to have an owner. A checking account has exactly one owner, a savings account could have several. A money transfer is issued by exactly one customer and each customer issues at least one money transfer. In the context of a transaction, there is a one-to-one connection between source and target.

$$\mathsf{card}(\mathsf{advises}) := (0..\infty, 1..\infty)$$
$$\mathsf{card}(\mathsf{own\_ca}) := (1..1, 0..\infty)$$
$$\mathsf{card}(\mathsf{own\_sa}) := (1..\infty, 0..\infty)$$
$$\mathsf{card}(\mathsf{issues}) := (1..1, 1..\infty)$$
$$\mathsf{card}(\mathsf{trans}) := (1..1, 1..1)$$

The intra-relationship type constraints are empty since we do not have such constraints in our example. $\diamondsuit$

At last we combine all parts of the role-based system into one *constrained* Σ*-compartment Object Role Model*.

[1]Note here, that this small modelling error is made on purpose. We will discuss its logical implications later. Actually a transaction has exactly one source *and* one target, and, hence, exactly two participants.

Definition 2.16 (Constrained Σ-Compartment Role Object Model). *Let* $\Sigma = (\mathsf{N}_{\mathsf{NT}}, \mathsf{N}_{\mathsf{RT}}, \mathsf{N}_{\mathsf{CT}}, \mathsf{N}_{\mathsf{RST}})$ *be a vocabulary, let* $\mathcal{M}$ *be a* Σ*-CROM, let* $\mathcal{A}$ *be a* Σ*-CROA and let* $\mathcal{C}$ *be a* Σ*-CROC. Then, a* Constrained Σ-Compartment Role Object Model Σ-CCROM *is the tuple* $\mathcal{K} = (\mathcal{M}, \mathcal{A}, \mathcal{C})$.

The satisfiability problem *for* Σ*-CCROMs is the problem of deciding for a given* Σ*-CCROM* $\mathcal{K} = (\mathcal{M}, \mathcal{A}, \mathcal{C})$ *whether there exists a* Σ*-CROI that satisfies* $\mathcal{M}$*,* $\mathcal{A}$ *and* $\mathcal{C}$*.* $\diamondsuit$

Chapter 3

The Contextualised Description Logic $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$

In the previous chapter we introduced description logics as a logical formalism for knowledge representation as well as the Compartment Role Object Model as a modelling language for role based models with a formal semantics. Now, the next step in our overall workflow is to investigate whether DLs are sufficient to express CROM and, if necessary, develop an extension of DLs which possesses the required expressive power.

Hence, we start this chapter with discussing the requirements for a feasible logical formalism in Section 3.1. Then, after defining the syntax and semantics of contextualised DLs in Section 3.2, we analyse the computational complexity of the consistency problem for contextualised DLs in Section 3.3. Particular consideration is needed in the case of $\mathcal{EL}$, which is discussed in Section 3.4. At last, we show in Section 3.5 that introducing an additional concept constructor directly leads to undecidability.

3.1 Requirements for Logical Formalism

First we have to decide how to represent role-based models. Although nearly every large software project starts with modelling the application domain in UML and, hence, UML is the de facto standard as modelling language, it has several drawbacks. First of all, it has no formally defined semantics [FEL+98]. While UML meta-models capture the precise syntax of concepts used for modelling, they do little for answering questions of how to interpret non-trivial UML diagrams. For example, if one wants to map UML class diagrams to a logical formalism to be able to reason about these diagrams, one takes certain assumptions about the intended meaning of the occurring elements. Then a domain expert modelling some application has to trust that the logician constructed the mapping with the same semantics in mind. Nevertheless, over the last years several approaches for formal frameworks to reason on UML arose [Eva98; CCD+02; SMS+03; SSJ+04; BCD05; SBH+08; AN10], from which we can adapt some ideas, e.g. how to model attributes of a UML class with DL roles and how multiplicities of associations are modelled in [CCD+02].

The second major drawback is that UML lacks expressive power to model context-dependent domains. There also exists some work on extending UML in that direction [SB05], but here the semantics are even more ambiguous. Therefore, instead of trying to formalise and reason on semantically ambiguous UML diagrams that try to capture role-based models, we focus on role-based models that are modelled in CROM which we introduced in Chapter 2. CROM overcomes both these deficits since it has both a well-defined, formal semantics and the means to formulate the necessary concepts accordingly.

Next, we investigate the requirements for the logical formalism. In order to be feasible with the workflow outlined in the introduction, we need a formalism that has the expressive power to capture the information in CROM on one hand but preserves decidability of the needed reasoning tasks on the other. As shown in the preliminaries description logics serve as a very powerful formalism in knowledge representation. Still, the main drawback might be the expressive restrictions of 'classical' description logics as explained in the introduction. Standard DLs cannot formalise contextual knowledge in a proper way which is crucial for role-based systems. In the recent years many different approaches and extensions of DLs have been proposed [BGH+03; BGH+04; BAF+06; BVS+09; BKP12; CP14; CP17]. However, many were tailored to different goals, for example to support context-specific reuse of ontologies or enable probabilistic reasoning. Additionally, these approaches have a quite different intuition of what a context exactly is. In most cases it is simply a finite set of names. Serafini et al. [SH12] defines contexts by a set of attribute-value declarations, one for each *dimension*, e.g. time, topic or location to name a few. There is only one context for each dimensional vector whereas in CROM there can exist many instances of one compartment type. Furthermore, except a coverage relation, one can hardly express any other knowledge about the contexts such as the relational structure between them. Hence, these approaches seem not appropriate to model CROMs.

A far more promising formalism is the description logic of context $\mathcal{ALC}_{\mathcal{ALC}}$ [KG10; KG16]. Klarman et al. follow the ideas of McCarthy's formalisation of contexts [McC87; McC93] where contexts are formal objects, that have properties and can be described, and that are organised in a relational structure. This results in a very expressive, two-dimensional description logic with strong interactions between the object and the context level allowing to express information within a context that is valid in some other context. While transcending object knowledge through contexts is very important in the general application of contextualised knowledge, it is complexity-wise very costly, especially in the presence of rigid roles where the consistency problem becomes undecidable.

By restricting that interaction to a top-down view of contexts, we can retain decidability. From the meta or context level, we can impose axioms that must hold within a certain context, but on the object level, i.e. within a context, we cannot 'look outside'. When formalising CROM compartments, this is exactly the needed expressiveness. For each compartment type we want to specify the constraints which must hold within a compartment of that type. This analysis leads us to the construction of a new contextualised description logic $\mathcal{L}_{\mathsf{M}}[\![\mathcal{L}_{\mathsf{O}}]\!]$. In Section 3.5, we show that $\mathcal{ALC}[\![\mathcal{ALC}]\!]$ is indeed a sublogic of $\mathcal{ALC}_{\mathcal{ALC}}$. In Chapter 4, we will present how a CROM can be represented in $\mathcal{L}_{\mathsf{M}}[\![\mathcal{L}_{\mathsf{O}}]\!]$.

3.2 Syntax and Semantics of the Contextualised Description Logic $\mathcal{L}_{\mathsf{M}}[\![\mathcal{L}_{\mathsf{O}}]\!]$

Our approach for contextualised description logics is similar to temporal description logics as investigated in [LWZ08] in which a temporal logic for the time dimension and a description logic for the object dimension are combined in order to express information valid at a certain time. We combine two, possibly different, description logics: one DL $\mathcal{L}_{\mathsf{M}}$ for the *context* or *meta dimension* for knowledge *about* contexts and one DL $\mathcal{L}_{\mathsf{O}}$ for the *object dimension* for knowledge *within* contexts. Considering the semantics, $\mathcal{L}_{\mathsf{M}}[\![\mathcal{L}_{\mathsf{O}}]\!]$ can also be seen as a restricted variant of the context description logic proposed by Klarman et al. [KG16].

The contextualised DL $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$ is a two-dimensional and two-sorted description logic. Syntactically, we start with a description logic $\mathcal{L}_\mathsf{M}$ for the meta level to describe the relational structure between contexts. We add one meta concept constructor that allows to refer to the inner structure of each context. More precisely, we allow an $\mathcal{L}_\mathsf{O}$-axiom to constitute a meta concept which denotes the set of contexts in which that axiom holds.

Throughout the rest of this thesis, let $\mathsf{M} = (\mathsf{M_C}, \mathsf{M_R}, \mathsf{M_I})$ and $\mathsf{O} = (\mathsf{O_C}, \mathsf{O_R}, \mathsf{O_I})$ denote the signatures for $\mathcal{L}_\mathsf{M}$ and $\mathcal{L}_\mathsf{O}$, respectively. Thus, we call $\mathsf{M_C}$, $\mathsf{M_R}$, $\mathsf{M_I}$, $\mathsf{O_C}$, $\mathsf{O_R}$ and $\mathsf{O_I}$, respectively, the set of *meta concept*, *role* and *individual names* and *object concept*, *role* and *individual names*.

Definition 3.1 (Syntax of $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$). *A* concept of the object logic $\mathcal{L}_\mathsf{O}$ (o-concept) *is an $\mathcal{L}_\mathsf{O}$-concept over* O*. An* o-axiom *is an $\mathcal{L}_\mathsf{O}$-GCI over* O*, an $\mathcal{L}_\mathsf{O}$-concept assertion over* O*, or an $\mathcal{L}_\mathsf{O}$-role assertion over* O*.*

The set of concepts of the meta logic $\mathcal{L}_\mathsf{M}$ (m-concepts) *is the smallest set such that*

- *for all $A \in \mathsf{M_C}$, A is a* basic *meta concept,*
- *for all o-axioms α, $[\![\alpha]\!]$ is a* referring *meta concept, and*
- *all complex concepts that can be built with the concept constructors allowed in $\mathcal{L}_\mathsf{M}$ are meta concepts.*

A meta general concept inclusion (m-GCI) *is a GCI $C \sqsubseteq D$ where C and D are m-concepts, a* meta concept assertion *is a concept assertion $C(s)$ where C is an m-concept and $s \in \mathsf{M_I}$, and a* meta role assertion *is simply an $\mathcal{L}_\mathsf{M}$-role assertion over* M*. An* m-axiom *is an m-GCI, a meta concept assertion or a meta role assertion.*

As for DLs, a Boolean m-axiom formula *is inductively defined as follows:*

- *every m-axiom is a Boolean m-axiom formula,*
- *if $\mathcal{B}_1, \mathcal{B}_2$ are Boolean m-axiom formulas, then so are $\neg\mathcal{B}_1$ and $\mathcal{B}_1 \wedge \mathcal{B}_2$, and*
- *nothing else is a Boolean m-axiom formula.*

Finally, a Boolean $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$-knowledge base ($\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$-BKB) *is a triple $\mathfrak{B} = (\mathcal{B}, \mathcal{R}_\mathsf{O}, \mathcal{R}_\mathsf{M})$, where $\mathcal{R}_\mathsf{O}$ is an $\mathcal{L}_\mathsf{O}$-RBox over* O*, $\mathcal{R}_\mathsf{M}$ an $\mathcal{L}_\mathsf{M}$-RBox over* M*, and $\mathcal{B}$ is a Boolean m-axiom formula. An* $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$-ontology *is an $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$-BKB, where only axiom conjunction and no axiom negation is allowed in the Boolean m-axiom formula.* ◇

Essentially, m-GCIs and meta concept assertions are $\mathcal{L}_\mathsf{M}$-GCIs over M and $\mathcal{L}_\mathsf{M}$-concept assertions over M in which additionally referring m-concepts are admitted. For the same reasons as mentioned in the end of Section 2.1, role inclusions over O and transitivity axioms over O are not allowed to constitute m-concepts. However, we fix an RBox $\mathcal{R}_\mathsf{O}$ over O that contains such o-axioms and holds in *all* contexts. The same applies to role inclusions over M and transitivity axioms over M, which are only allowed to occur in a RBox $\mathcal{R}_\mathsf{M}$ over M. Again, we use the usual abbreviations (for disjunctions etc.) for m-concepts and Boolean m-axiom formulas.

The semantics of $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$ is defined by the notion of *nested interpretations*. These consist of O-interpretations for the specific contexts and an M-interpretation for the relational structure between them. We assume that all contexts speak about the same non-empty domain (*constant domain assumption*). We show later that this is no real restriction.

As argued earlier, in some situations it is desired that concepts or roles in the object logic are interpreted the same in all contexts. Therefore we introduce *rigid names*. Let $\mathsf{O}_{\mathsf{Crig}} \subseteq \mathsf{O}_{\mathsf{C}}$ be the set of *rigid object concept names* and $\mathsf{O}_{\mathsf{Rrig}} \subseteq \mathsf{O}_{\mathsf{R}}$ be the set of *rigid object role names*. Often, we refer to $\mathsf{O}_{\mathsf{Crig}}$and $\mathsf{O}_{\mathsf{Rrig}}$ simply as *rigid concepts* and *rigid roles*, as there is no such notion on the meta level. We set $\mathsf{O}_{\mathsf{Cflex}} := \mathsf{O}_{\mathsf{C}} \setminus \mathsf{O}_{\mathsf{Crig}}$ and $\mathsf{O}_{\mathsf{Rflex}} := \mathsf{O}_{\mathsf{R}} \setminus \mathsf{O}_{\mathsf{Rrig}}$ and call these concept names and role names *flexible*. Moreover, following the argument for the UNA, we assume that the identity of an individual is context-independent and, thus, individual names of the object logic are always interpreted the same in all contexts (*rigid individual assumption*).

Definition 3.2 (Nested interpretation). *A* nested interpretation *is a tuple* $\mathcal{J} = (\mathbb{C}, \cdot^{\mathcal{J}}, \Delta^{\mathcal{J}}, (\cdot^{\mathcal{I}_c})_{c\in\mathbb{C}})$*, where* $\mathbb{C}$ *is a non-empty set (called set of* contexts*) and* $(\mathbb{C}, \cdot^{\mathcal{J}})$ *is an* M*-interpretation. Moreover, for every* $c \in \mathbb{C}$, $\mathcal{I}_c := (\Delta^{\mathcal{J}}, \cdot^{\mathcal{I}_c})$ *is an* O*-interpretation such that for all* $c, c' \in \mathbb{C}$*, we have that* $x^{\mathcal{I}_c} = x^{\mathcal{I}_{c'}}$ *for every* $x \in \mathsf{O}_{\mathsf{I}} \cup \mathsf{O}_{\mathsf{Crig}} \cup \mathsf{O}_{\mathsf{Rrig}}$. ◇

We are now ready to define the semantics of $\mathcal{L}_{\mathsf{M}}[\![\mathcal{L}_{\mathsf{O}}]\!]$.

Definition 3.3 (Semantics of $\mathcal{L}_{\mathsf{M}}[\![\mathcal{L}_{\mathsf{O}}]\!]$). *Let* $\mathcal{J} = (\mathbb{C}, \cdot^{\mathcal{J}}, \Delta^{\mathcal{J}}, (\cdot^{\mathcal{I}_c})_{c\in\mathbb{C}})$ *be a nested interpretation. The mapping* $\cdot^{\mathcal{J}}$ *is extended to referring meta concepts as follows:*

$$[\![\alpha]\!]^{\mathcal{J}} := \{c \in \mathbb{C} \mid \mathcal{I}_c \models \alpha\}.$$

Moreover, $\mathcal{J}$ *is a model of the m-axiom* β *if* $(\mathbb{C}, \cdot^{\mathcal{J}})$ *is a model of* β*. This is extended to Boolean m-axiom formulas inductively as follows:*

- $\mathcal{J}$ *is a model of* $\neg\mathcal{B}_1$ *if it is not a model of* $\mathcal{B}_1$*, and*
- $\mathcal{J}$ *is a model of* $\mathcal{B}_1 \wedge \mathcal{B}_2$ *if it is a model of both* $\mathcal{B}_1$ *and* $\mathcal{B}_2$*.*

We write $\mathcal{J} \models \mathcal{B}$ *if* $\mathcal{J}$ *is a model of the Boolean m-axiom formula* $\mathcal{B}$*. Furthermore,* $\mathcal{J}$ *is a model of* $\mathcal{R}_{\mathsf{M}}$ *(written* $\mathcal{J} \models \mathcal{R}_{\mathsf{M}}$*) if* $(\mathbb{C}, \cdot^{\mathcal{J}})$ *is a model of* $\mathcal{R}_{\mathsf{M}}$*, and* $\mathcal{J}$ *is a model of* $\mathcal{R}_{\mathsf{O}}$ *(written* $\mathcal{J} \models \mathcal{R}_{\mathsf{O}}$*) if* $\mathcal{I}_c$ *is a model of* $\mathcal{R}_{\mathsf{O}}$ *for all* $c \in \mathbb{C}$*.*

Finally, $\mathcal{J}$ *is a model of the* $\mathcal{L}_{\mathsf{M}}[\![\mathcal{L}_{\mathsf{O}}]\!]$*-BKB* $\mathfrak{B} = (\mathcal{B}, \mathcal{R}_{\mathsf{O}}, \mathcal{R}_{\mathsf{M}})$ *(written* $\mathcal{J} \models \mathfrak{B}$*) if* $\mathcal{J}$ *is a model of* $\mathcal{B}$*,* $\mathcal{R}_{\mathsf{O}}$*, and* $\mathcal{R}_{\mathsf{M}}$*. We call* $\mathfrak{B}$ consistent *if it has a model.*

The consistency problem in $\mathcal{L}_{\mathsf{M}}[\![\mathcal{L}_{\mathsf{O}}]\!]$ *is the problem of deciding whether a given* $\mathcal{L}_{\mathsf{M}}[\![\mathcal{L}_{\mathsf{O}}]\!]$*-BKB is consistent.* ◇

Note here, that in the above definition we do not need to consider o-axioms separately since on the object level the semantics are defined as for the DL $\mathcal{L}_{\mathsf{O}}$.

Now with these notions, we can refine the examples of Section 2.1.

Example 3.4. *Considering again Example 2.3 and Example 2.6, we recognise that from an ontological point of view, the occurring concepts are not modelled well. There is a qualitative difference between the concepts of a* Quarterback *and an* NFL_Team*. In fact,* NFL_Team *actually is a context within the* Quaterback *position makes sense. Analogous to the concept* C *of Ex. 2.3, we can model the meta concept* C' *as follows:*

$$\mathsf{NFL\_Team} \sqcap \neg\mathsf{AFC} \sqcap \neg[\![\exists\mathsf{plays.Quarterback} \sqsubseteq \bot]\!] \sqcap [\![\mathsf{Coach}(\mathit{MikeMcCarthy})]\!].$$

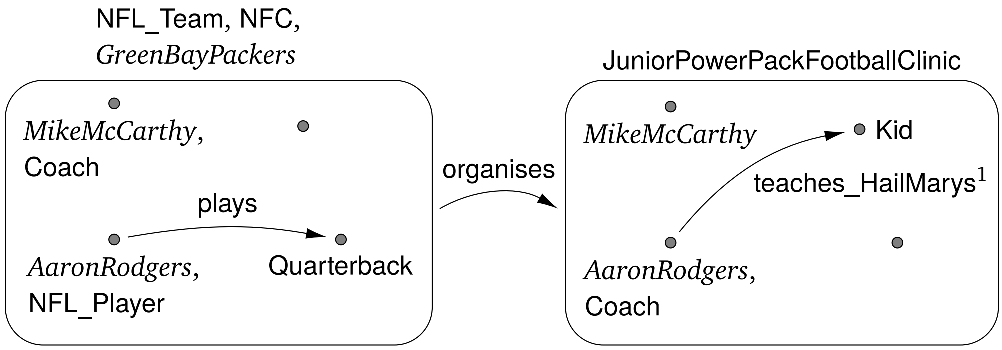


Figure 3.1: A Nested interpretation $\mathcal{J}$ such that $\mathit{GreenBayPackers}^\mathcal{J} \in C'^\mathcal{J}$ and $\mathcal{J} \models \mathcal{O}'$.

It describes the context of an NFL team that is not in the AFC, has someone playing quarterback and has Mike McCarthy as coach. Here, $\exists\mathsf{plays.Quarterback} \sqsubseteq \bot$ *is an o-axiom which builds the m-concept* $[\![\exists\mathsf{plays.Quarterback} \sqsubseteq \bot]\!]$.

In reference to the ontology $\mathcal{O}$ of Ex. 2.6 the contextualised ontology $\mathcal{O}'$ could look like the following:

$$\begin{gathered}\mathsf{NFC}(\mathit{GreenBayPackers}) \quad \wedge \\ [\![(\exists\mathsf{plays}.\top)(\mathit{AaronRodgers})]\!](\mathit{GreenBayPackers}) \quad \wedge \\ \mathsf{NFL\_Team} \sqsubseteq [\![\exists\mathsf{plays}.\top \sqsubseteq \mathsf{NFL\_Player}]\!] \quad \wedge \\ \mathsf{NFL\_Team} \equiv \mathsf{NFC} \sqcup \mathsf{AFC} \quad \wedge \\ \mathsf{NFC} \sqcap \mathsf{AFC} \sqsubseteq \bot.\end{gathered}$$

The first meta concept assertion states that the context of the Green Bay Packers belongs to the meta concept NFC. The second meta concept assertion states that the Green Bay Packers are a context in which Aaron Rodgers plays something. Furthermore, people who play something within the context of an NFL team are NFL players, and last, the NFL is a disjoint union of the NFC and the AFC. As before, we can entail that Aaron Rodgers is an NFL player, at least in the context of the Packers, i.e. $\mathcal{O}'$ entails

$$[\![\mathsf{NFL\_Player}(\mathit{AaronRodgers})]\!](\mathit{GreenBayPackers}).$$

A nested interpretation $\mathcal{J}$, such that Green Bay Packers are in the extension of C', and $\mathcal{J}$ is a model of $\mathcal{O}'$, is depicted in Figure 3.1. In order to show the relational structure of contexts we added the context of the 'Junior Football Clinic' in $\mathcal{J}$ though it has nothing to do with neither C' nor $\mathcal{O}'$. However, if we would model this with standard DLs it is not clear anymore in which context Aaron Rodgers is a player and when he is a coach. ♢

To argue that the constant domain assumption is no serious restriction, we show that consistency with varying domains can be polynomially reduced to consistency with constant domains. Here, we can adapt the ideas of [GKW+03] and [LWZ08]. Let $\mathfrak{B}$ be an $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$-BKB and let $\mathcal{J}_v$ denote a nested interpretation with varying domains, i.e. $\mathcal{J}_v = (\mathbb{C}, \cdot^\mathcal{J}, (\Delta^{\mathcal{I}_c})_{c\in\mathbb{C}}, (\cdot^{\mathcal{I}_c})_{c\in\mathbb{C}})$,

[1]A Hail Mary pass is a very long forward pass in American football, made in desperation with only a small chance of success.

where $(\Delta^{\mathcal{I}_c})_{c\in\mathbb{C}}$ is a set of, possibly overlapping, object domains. We introduce a fresh object concept name E that expresses the existence of an element in an object domain. W.l.o.g. we assume that all o-GCIs are of the form $\top \sqsubseteq C$. We obtain C_E from the o-concept C by replacing every o-subconcept $\exists r.D$ with $\exists r.(D \sqcap E)$ and every o-subconcept $\geqslant_n r.D$ with $\geqslant_n r.(D \sqcap E)$. We obtain $\mathfrak{B}_E$ from $\mathfrak{B}$ by replacing every occurrence of an o-axiom α in $\mathfrak{B}$ with α_E, where α_E is defined as follows:

$$\alpha_E := \begin{cases} E \sqsubseteq C_E & \text{if } \alpha = \top \sqsubseteq C, \\ (C_E \sqcap E)(a) & \text{if } \alpha = C(a), \text{ and} \\ \alpha & \text{otherwise.} \end{cases}$$

Proposition 3.5. *$\mathfrak{B}$ is consistent w.r.t. varying domains if and only if $\mathfrak{B}_E$ is consistent w.r.t. constant domains and $\mathfrak{B}_E$ is of size polynomial in the size of $\mathfrak{B}$.*

Proof. For the 'only if' direction, let $\mathcal{J}_v = (\mathbb{C}, \cdot^{\mathcal{J}}, (\Delta^{\mathcal{I}_c})_{c\in\mathbb{C}}, (\cdot^{\mathcal{I}_c})_{c\in\mathbb{C}})$ be a model of $\mathfrak{B}$ with varying domains. We construct the nested interpretation $\mathcal{J} = (\mathbb{C}, \cdot^{\mathcal{J}}, \Delta^{\mathcal{J}}, (\cdot^{\mathcal{I}_c})_{c\in\mathbb{C}})$ with a constant domain from $\mathcal{J}_v$ such that $\Delta^{\mathcal{J}} := \bigcup_{c\in\mathbb{C}} \Delta^{\mathcal{I}_c}$ and $E^{\mathcal{I}_c} := \Delta^{\mathcal{I}_c}$. Let $\mathcal{I}'_c = (\Delta^{\mathcal{I}_c}, \cdot^{\mathcal{I}_c})$ and $\mathcal{I}_c = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{I}_c})$.

<u>Claim:</u> $\{c \in \mathbb{C} \mid \mathcal{I}'_c \models \alpha\} = \{c \in \mathbb{C} \mid \mathcal{I}_c \models \alpha_E\}$

<u>Proof:</u> *We prove the claim by showing $\mathcal{I}'_c \models \alpha$ iff $\mathcal{I}_c \models \alpha_E$ for the different cases of α:*

$\alpha = \top \sqsubseteq C$: $\mathcal{I}'_c \models \top \sqsubseteq C$ *iff* $\Delta^{\mathcal{I}_c} = C^{\mathcal{I}_c}$ *iff*[2] $E^{\mathcal{I}_c} = C_E^{\mathcal{I}_c}$ *iff* $\mathcal{I}_c \models E \sqsubseteq (C_E \sqcap E) = \alpha_E$.

$\alpha = C(a)$: $\mathcal{I}'_c \models C(a)$ *iff* $a^{\mathcal{I}_c} \in C^{\mathcal{I}_c}$. *Since* $E^{\mathcal{I}_c} = \Delta^{\mathcal{I}_c}$, *we also have* $a^{\mathcal{I}_c} \in E^{\mathcal{I}_c}$ *and, hence,* $\mathcal{I}_c \models (C \sqcap E)(a)$.

$\alpha = r(a, b)$: *We have* $\alpha = \alpha_E$ *and, since* $a^{\mathcal{I}_c}, b^{\mathcal{I}_c} \in \Delta^{\mathcal{I}_c}$, $\mathcal{I}'_c \models \alpha$ *iff* $\mathcal{I}_c \models \alpha_E$. ⁄

Since we do not change the meta interpretation and $[\![\alpha]\!]^{\mathcal{J}_v} = [\![\alpha_E]\!]^{\mathcal{J}}$, we get that $\mathcal{J} \models \mathfrak{B}_E$.

For the 'if' direction, let $\mathcal{J} = (\mathbb{C}, \cdot^{\mathcal{J}}, \Delta^{\mathcal{J}}, (\cdot^{\mathcal{I}_c})_{c\in\mathbb{C}})$ be a model of $\mathfrak{B}_E$. Then, we define $\mathcal{J}_v = (\mathbb{C}, \cdot^{\mathcal{J}}, (\Delta^{\mathcal{I}'_c})_{c\in\mathbb{C}}, (\cdot^{\mathcal{I}'_c})_{c\in\mathbb{C}})$ with $\Delta^{\mathcal{I}'_c} := E^{\mathcal{I}_c}$ and $A^{\mathcal{I}'_c} := A^{\mathcal{I}_c} \cap E^{\mathcal{I}_c}$. By similar arguments as above and due to the restriction in existential and at-least restrictions for r-successors being in E, we have $[\![\alpha]\!]^{\mathcal{J}_v} = [\![\alpha_E]\!]^{\mathcal{J}}$ and therefore $\mathcal{J}_v \models \mathfrak{B}$. □

The above proposition, however, does not imply that $\mathfrak{B}$ is consistent w.r.t. constant domains if and only if it is consistent w.r.t. varying domains. For example, with a rigid concept name $A \in \mathsf{O}_{\mathsf{Crig}}$ the BKB $\mathfrak{B} = ([\![\top \sqsubseteq A]\!] \sqcap \exists t.[\![A \sqsubseteq \exists r.\neg A]\!])(c)$ is consistent w.r.t. varying domains, but not w.r.t. constant domains. Nonetheless, $\mathfrak{B}_E$ is consistent w.r.t. constant domains and checking consistency of $\mathfrak{B}$ w.r.t. varying domains can be done by considering $\mathfrak{B}_E$. Thus, for the rest of the thesis, we concentrate on constant domains.

3.3 Complexity of the Consistency Problem in $\mathcal{L}_{\mathsf{M}}[\![\mathcal{L}_{\mathsf{O}}]\!]$

After we introduced the syntax and semantics of our contextualised description logics, in this section, we investigate the computational complexity of the consistency problem. Therefore, we consider three different settings dependent on whether rigid concepts and rigid roles are

[2]Exchanging C with C_E can be done here, as the concept E is equivalent to $\top$ for $\mathcal{I}'_c$.

Table 3.1: The complexity results for the consistency problem in $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$

<table>
<tr><th colspan="2">$\mathcal{L}_\mathsf{O}$ / $\mathcal{L}_\mathsf{M}$</th><th>$\mathcal{EL}$</th><th>$\mathcal{ALC}$</th><th>$\mathcal{SHOQ}$</th><th>$\mathcal{SHOIQ}$</th></tr>
<tr><td rowspan="4">Setting (i)</td><td>$\mathcal{EL}$</td><td>constant</td><td colspan="2" rowspan="3">Th. 3.31
ExpTime-complete
Th. 3.17</td><td rowspan="4">Th. 3.32
NExpTime-complete
Th. 3.18</td></tr>
<tr><td>$\mathcal{ALC}$</td><td rowspan="2">Th. 3.23</td></tr>
<tr><td>$\mathcal{SHOQ}$</td></tr>
<tr><td>$\mathcal{SHOIQ}$</td><td colspan="3">Th. 3.24</td></tr>
<tr><td rowspan="4">Setting (ii)</td><td>$\mathcal{EL}$</td><td>constant</td><td colspan="2" rowspan="4">Th. 3.34
NExpTime-complete
Th. 3.21</td><td rowspan="4">NExpTime-hard and in N2ExpTime
Th. 3.20</td></tr>
<tr><td>$\mathcal{ALC}$</td><td rowspan="3">Th. 3.28</td></tr>
<tr><td>$\mathcal{SHOQ}$</td></tr>
<tr><td>$\mathcal{SHOIQ}$</td></tr>
<tr><td rowspan="4">Setting (iii)</td><td>$\mathcal{EL}$</td><td>constant</td><td colspan="2" rowspan="4">Th. 3.33
2ExpTime-complete
Th. 3.19</td><td rowspan="4">2ExpTime-hard and in N2ExpTime
Th. 3.20</td></tr>
<tr><td>$\mathcal{ALC}$</td><td rowspan="3">NExpTime-complete
Th. 3.29</td></tr>
<tr><td>$\mathcal{SHOQ}$</td></tr>
<tr><td>$\mathcal{SHOIQ}$</td></tr>
</table>

Settings: (i) No rigid names are allowed, i.e. $\mathsf{O}_{\mathsf{Crig}} = \mathsf{O}_{\mathsf{Rrig}} = \emptyset$. (ii) Only rigid concepts are allowed, i.e. $\mathsf{O}_{\mathsf{Crig}} \neq \emptyset$ and $\mathsf{O}_{\mathsf{Rrig}} = \emptyset$. (iii) Rigid roles are allowed, i.e. $\mathsf{O}_{\mathsf{Rrig}} \neq \emptyset$.

admitted. In Setting (i) no rigid names are allowed at all. In Setting (ii) rigid concepts are allowed, but no rigid roles. Setting (iii) then allows rigid roles. There, it is not necessary to distinguish whether rigid concepts are admitted or not, since rigid concepts can be emulated via rigid roles. For this, one simply replaces the rigid concept A by $\exists r_A.\top$ where r_A is a rigid role which does not occur in the original knowledge base.

Our results for the complexity of the consistency problem are listed in Table 3.1. Here, it is worth noting that if no rigid names are admitted, the complexity class does not increase compared to the consistency problem for the classical DL, i.e. for any $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$ up to $\mathcal{SHOQ}[\![\mathcal{SHOQ}]\!]$ the consistency problem is ExpTime-complete, if $\mathcal{SHOIQ}$ is involved, it is NExpTime-complete. Furthermore, in the presence of rigid roles we can retain decidability. This distinguishes our approach from the semantically similar context DL introduced by Klarman et al. [KG10; KG16], where the consistency problem becomes undecidable when rigid roles occur in the ontology.

Since the lower bounds already hold for the case of $\mathcal{EL}$, we handle them in Section 3.4. For the upper bounds, let in the following $\mathfrak{B} = (\mathcal{B}, \mathcal{R}_\mathsf{O}, \mathcal{R}_\mathsf{M})$ be an $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$-BKB. We proceed similar to what was done for $\mathcal{ALC}$-LTL in [BGL08; BGL12] (and $\mathcal{SHOQ}$-LTL in [Lip14]) and reduce the consistency problem to two separate decision problems.

For the first decision problem, we consider the so-called *outer abstraction*, which is the $\mathcal{L}_\mathsf{M}$-BKB over M obtained by replacing each referring m-concept of the form $[\![\alpha]\!]$ occurring in $\mathcal{B}$ by a fresh concept name such that there is a 1–1 relationship between them.

Definition 3.6 (Outer abstraction). *Let $\mathfrak{B} = (\mathcal{B}, \mathcal{R}_\mathsf{O}, \mathcal{R}_\mathsf{M})$ be an $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$-BKB. Let* b *be the bijection mapping every referring m-concept of the form $[\![\alpha]\!]$ occurring in $\mathcal{B}$ to the so-called* abstracted concept name $A_{[\![\alpha]\!]} \in \mathsf{M_C}$, *where we assume w.l.o.g. that $A_{[\![\alpha]\!]}$ does not occur in $\mathcal{B}$.*

1. *The $\mathcal{L}_\mathsf{M}$-concept C^b over M is obtained from the m-concept C by replacing every occurrence of $[\![\alpha]\!]$ by $\mathsf{b}([\![\alpha]\!])$.*

2. *The Boolean $\mathcal{L}_\mathsf{M}$-axiom formula $\mathcal{B}^\mathsf{b}$ over M is obtained from $\mathcal{B}$ by replacing every m-concept C occurring in $\mathcal{B}$ with C^b. We call the $\mathcal{L}_\mathsf{M}$-BKB $\mathfrak{B}^\mathsf{b} = (\mathcal{B}^\mathsf{b}, \mathcal{R}_\mathsf{M})$ the* outer abstraction of $\mathfrak{B}$.

3. *Given $\mathcal{J} = (\mathbb{C}, \cdot^\mathcal{J}, \Delta^\mathcal{J}, (\cdot^{\mathcal{I}_c})_{c\in\mathbb{C}})$, its* outer abstraction *is the* M*-interpretation $\mathcal{J}^\mathsf{b} = (\mathbb{C}, \cdot^{\mathcal{J}^\mathsf{b}})$ where*
   - *for every $x \in \mathsf{M_R} \cup \mathsf{M_I} \cup (\mathsf{M_C} \setminus \mathsf{ran}(\mathsf{b}))$, we have $x^{\mathcal{J}^\mathsf{b}} = x^\mathcal{J}$, and*
   - *for every $A \in \mathsf{ran}(\mathsf{b})$, we have $A^{\mathcal{J}^\mathsf{b}} = (\mathsf{b}^{-1}(A))^\mathcal{J}$,*

   where $\mathsf{ran}(\mathsf{b})$ *denotes the image of* b. ♢

For simplicity, for $\mathfrak{B}' = (\mathcal{B}', \mathcal{R}_\mathsf{O}, \mathcal{R}_\mathsf{M})$, where $\mathcal{B}'$ is a subformula of $\mathcal{B}$, we denote by $(\mathfrak{B}')^\mathsf{b}$ the outer abstraction of $\mathfrak{B}'$ that is obtained by restricting b to the m-concepts occurring in $\mathcal{B}'$. Now let us consider the following small example.

Example 3.7. *Let $\mathfrak{B}_{ex} = (\mathcal{B}_{ex}, \emptyset, \emptyset)$ with $\mathcal{B}_{ex} := C \sqsubseteq ([\![A \sqsubseteq \bot]\!]) \ \wedge \ (C \sqcap [\![A(a)]\!])(c)$ be an $\mathcal{ALC}[\![\mathcal{ALC}]\!]$-BKB. Then,* b *maps $[\![A \sqsubseteq \bot]\!]$ to $A_{[\![A\sqsubseteq\bot]\!]}$ and $[\![A(a)]\!]$ to $A_{[\![A(a)]\!]}$. Thus, we have that*

$$\mathfrak{B}^\mathsf{b}_{ex} := \Big(C \sqsubseteq (A_{[\![A\sqsubseteq\bot]\!]}) \ \wedge \ (C \sqcap A_{[\![A(a)]\!]})(c), \ \emptyset\Big)$$

is the outer abstraction of $\mathfrak{B}_{ex}$. ♢

The following lemma makes the relationship between $\mathfrak{B}$ and its outer abstraction $\mathfrak{B}^\mathsf{b}$ explicit. It is proved by induction on the structure of $\mathcal{B}$.

Lemma 3.8. *Let $\mathcal{J}$ be a nested interpretation such that $\mathcal{J}$ is a model of $\mathcal{R}_\mathsf{O}$. Then, $\mathcal{J}$ is a model of $\mathfrak{B}$ iff $\mathcal{J}^\mathsf{b}$ is a model of $\mathfrak{B}^\mathsf{b}$.*

Proof. Since $r^\mathcal{J} = r^{\mathcal{J}^\mathsf{b}}$ for all $\mathcal{L}_\mathsf{M}$-role r over M, we have that $\mathcal{J}$ is a model of $\mathcal{R}_\mathsf{M}$ iff $\mathcal{J}^\mathsf{b}$ is a model of $\mathcal{R}_\mathsf{M}$. Thus, it is only left to show that for any m-axiom γ occurring in $\mathcal{B}$, it holds that $\mathcal{J} \models \gamma$ iff $\mathcal{J}^\mathsf{b} \models \gamma^\mathsf{b}$.

Claim: *For any $x \in \mathbb{C}$ it holds that $x \in C^\mathcal{J}$ iff $x \in (C^\mathsf{b})^{\mathcal{J}^\mathsf{b}}$.*

Proof: *We prove the claim by induction on the structure of C:*

$C = A \in \mathsf{M_C} \setminus \mathsf{ran}(\mathsf{b})$: $x \in A^{\mathcal{J}}$ *iff* $x \in (A^\mathsf{b})^{\mathcal{J}^\mathsf{b}}$ *by definition of* $\mathcal{J}^\mathsf{b}$ *and since* $A = A^\mathsf{b}$

$C = [\![\alpha]\!]$: $x \in [\![\alpha]\!]^{\mathcal{J}}$ *iff* $x \in (A_{[\![\alpha]\!]})^{\mathcal{J}^\mathsf{b}}$ *iff* $x \in ([\![\alpha]\!]^\mathsf{b})^{\mathcal{J}^\mathsf{b}}$

$C = \neg D$: $x \in (\neg D)^{\mathcal{J}}$ *iff* $x \notin D^{\mathcal{J}}$ *iff, by induction hypothesis,* $x \notin (D^\mathsf{b})^{\mathcal{J}^\mathsf{b}}$ *iff* $x \in (\neg D^\mathsf{b})^{\mathcal{J}^\mathsf{b}}$ *iff* $x \in ((\neg D)^\mathsf{b})^{\mathcal{J}^\mathsf{b}}$

$C = D \sqcap E$: $x \in (D \sqcap E)^{\mathcal{J}}$ *iff* $x \in D^{\mathcal{J}}$ *and* $x \in E^{\mathcal{J}}$ *iff, by induction hypothesis,* $x \in (D^\mathsf{b})^{\mathcal{J}^\mathsf{b}}$ *and* $x \in (E^\mathsf{b})^{\mathcal{J}^\mathsf{b}}$ *iff* $x \in (D^\mathsf{b} \sqcap E^\mathsf{b})^{\mathcal{J}^\mathsf{b}}$ *iff* $x \in ((D \sqcap E)^\mathsf{b})^{\mathcal{J}^\mathsf{b}}$

$C = \exists r.D$: $x \in (\exists r.D)^{\mathcal{J}}$ *iff there exists* $y \in \mathbb{C}$ *s. t.* $(x,y) \in r^{\mathcal{J}}$ *and* $y \in D^{\mathcal{J}}$ *iff there exists* $y \in \mathbb{C}$ *s. t.* $(x,y) \in r^{\mathcal{J}^\mathsf{b}}$ *and* $y \in (D^\mathsf{b})^{\mathcal{J}^\mathsf{b}}$ *iff* $x \in (\exists r.D^\mathsf{b})^{\mathcal{J}^\mathsf{b}}$ *iff* $x \in ((\exists r.D)^\mathsf{b})^{\mathcal{J}^\mathsf{b}}$

$C = \{a\}$: $x \in \{a\}^{\mathcal{J}}$ *iff* $x \in (\{a\}^\mathsf{b})^{\mathcal{J}^\mathsf{b}}$ *by definition of* $\mathcal{J}^\mathsf{b}$ *and since* $\{a\} = \{a\}^\mathsf{b}$

$C = \geqslant_n r.D$: $x \in (\geqslant_n r.D)^{\mathcal{J}}$ *iff there are at least n elements* $y \in \mathbb{C}$ *s.t.* $(x,y) \in r^{\mathcal{J}}$ *and* $y \in D^{\mathcal{J}}$ *iff there are at least n elements* $y \in \mathbb{C}$ *s.t.* $(x,y) \in r^{\mathcal{J}^\mathsf{b}}$ *and* $y \in (D^\mathsf{b})^{\mathcal{J}^\mathsf{b}}$ *iff* $x \in (\geqslant_n r.D^\mathsf{b})^{\mathcal{J}^\mathsf{b}}$ *iff* $x \in ((\geqslant_n r.D)^\mathsf{b})^{\mathcal{J}^\mathsf{b}}$ ⁄

- If γ is of the form $C \sqsubseteq D$, we have that $\mathcal{J} \models C \sqsubseteq D$ iff $x \in C^{\mathcal{J}}$ implies $x \in D^{\mathcal{J}}$ iff (by claim) $x \in (C^\mathsf{b})^{\mathcal{J}^\mathsf{b}}$ implies $x \in (D^\mathsf{b})^{\mathcal{J}^\mathsf{b}}$ iff $\mathcal{J}^\mathsf{b} \models C^\mathsf{b} \sqsubseteq D^\mathsf{b}$.
- If γ is of the form $C(a)$, we have that $\mathcal{J} \models C(a)$ iff $a^{\mathcal{J}} \in C^{\mathcal{J}}$ iff (by claim) $a^{\mathcal{J}^\mathsf{b}} \in (C^\mathsf{b})^{\mathcal{J}^\mathsf{b}}$ iff $\mathcal{J}^\mathsf{b} \models C^\mathsf{b}(a)$.
- If γ is of the form $r(a,b)$, we have that $\mathcal{J} \models r(a,b)$ iff $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in r^{\mathcal{J}}$ iff $(a^{\mathcal{J}^\mathsf{b}}, b^{\mathcal{J}^\mathsf{b}}) \in r^{\mathcal{J}^\mathsf{b}}$ iff $\mathcal{J}^\mathsf{b} \models r(a,b)$.
- If $\mathcal{B}$ is of the form $\neg\mathcal{B}_1$, we have that $\mathcal{J} \models \mathcal{B}$ iff not $\mathcal{J} \models \mathcal{B}_1$ iff not $\mathcal{J}^\mathsf{b} \models \mathcal{B}^\mathsf{b}{}_1$ iff $\mathcal{J}^\mathsf{b} \models \mathcal{B}^\mathsf{b}$.
- If $\mathcal{B}$ is of the form $\mathcal{B}_1 \wedge \mathcal{B}_2$, we have that $\mathcal{J} \models \mathcal{B}$ iff $\mathcal{J} \models \mathcal{B}_1$ and $\mathcal{B}_2$ iff $\mathcal{J}^\mathsf{b} \models \mathcal{B}^\mathsf{b}{}_1$ and $\mathcal{J}^\mathsf{b} \models \mathcal{B}^\mathsf{b}{}_2$ iff $\mathcal{J}^\mathsf{b} \models \mathcal{B}^\mathsf{b}$.

Since $\mathcal{J} \models \mathcal{R}_\mathsf{O}$, $\mathcal{J} \models \mathcal{R}_\mathsf{M}$ iff $\mathcal{J}^\mathsf{b} \models \mathcal{R}_\mathsf{M}$ and $\mathcal{J} \models \mathcal{B}$ iff $\mathcal{J}^\mathsf{b} \models \mathcal{B}^\mathsf{b}$, we have $\mathcal{J} \models \mathfrak{B}$ iff $\mathcal{J}^\mathsf{b} \models \mathfrak{B}^\mathsf{b}$. □

Note that this lemma yields that consistency of $\mathfrak{B}$ implies consistency of $\mathfrak{B}^\mathsf{b}$. Thus, the consistency of $\mathfrak{B}^\mathsf{b}$ is a necessary condition for the consistency of $\mathfrak{B}$. However, it is not sufficient since the converse does not hold as the following example shows.

Example 3.9. *Consider again $\mathfrak{B}_{ex}$ of Example 3.7. First, we observe that $\mathfrak{B}_{ex}$ is inconsistent since the meta individual c belongs to the meta concepts C, $[\![A \sqsubseteq \bot]\!]$ and $[\![A(a)]\!]$. Thus, in any model of $\mathfrak{B}_{ex}$ the* O*-interpretation $\mathcal{I}_c$ must model both o-axioms $A \sqsubseteq \bot$ and $A(a)$ which is obviously not possible.*

However, there exists an M*-interpretation $\mathcal{H} = (\Delta^{\mathcal{H}}, \cdot^{\mathcal{H}})$ such that $\Delta^{\mathcal{H}} = \{e\}$, $d^{\mathcal{H}} = e$, and $C^{\mathcal{H}} = A^{\mathcal{H}}_{[\![A \sqsubseteq \bot]\!]} = A^{\mathcal{H}}_{[\![A(a)]\!]} = \{e\}$ which is a model of $\mathfrak{B}^\mathsf{b}_{ex}$. Hence, $\mathfrak{B}^\mathsf{b}_{ex}$ is consistent while $\mathfrak{B}_{ex}$ is not. Note that this does not contradicts Lemma 3.8 since there does not exist any nested interpretation $\mathcal{J}$ such that $\mathcal{H} = \mathcal{J}^\mathsf{b}$.* ◇

The above example illustrates that there exist implicit restrictions on the interpretation of the meta level as certain combinations of concept names in $\mathsf{ran}(\mathsf{b})$ are not allowed. Therefore, we need to ensure that these are not treated independently. For expressing such a restriction on the model $\mathcal{H}$ of $\mathfrak{B}^{\mathsf{b}}$, we adapt a notion of [BGL08; BGL12]. It is also worth noting that this problem occurs also in much less expressive DLs such as $\mathcal{EL}^{\bot}$ (i.e. $\mathcal{EL}$ extended with the bottom concept).

Definition 3.10 ($\mathcal{U}$-type, N-interpretation (weakly) respects $(\mathcal{U}, \mathcal{Y})$). *Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be an* N*-interpretation, let $\mathcal{U} \subseteq \mathsf{N_C}$ and let $\mathcal{Y} \subseteq \mathcal{P}(\mathcal{U})$, where $\mathcal{P}(S)$ denotes the power set of S. The $\mathcal{U}$-*type of $d \in \Delta^{\mathcal{I}}$ in $\mathcal{I}$ *is defined as* $\mathsf{type}^{\mathcal{I}}_{\mathcal{U}}(d) := \{A \in \mathcal{U} \mid d \in A^{\mathcal{I}}\}$. *The interpretation $\mathcal{I}$* respects $(\mathcal{U}, \mathcal{Y})$ *if $\mathcal{Z} = \mathcal{Y}$, where*

$$\mathcal{Z} := \{Y \subseteq \mathcal{U} \mid \textit{there is some } d \in \Delta^{\mathcal{I}} \textit{ with } \mathsf{type}^{\mathcal{I}}_{\mathcal{U}}(d) = Y\}$$

It weakly respects $(\mathcal{U}, \mathcal{Y})$ *if $\mathcal{Z} \subseteq \mathcal{Y}$.* ◇

For $\mathcal{U} = \mathsf{ran(b)}$, the $\mathcal{U}$-type of a context c, i.e. an element $c \in \mathbb{C}$ in a nested interpretation, is the set of all abstracted concept names of which c is an instance. In other words, it describes all o-axioms which hold in that context. The $\mathsf{ran(b)}$-type of c is also called its *restricted type*. The second decision problem that we use for deciding consistency is needed to make sure that such a set of abstracted concept names is admissible in the following sense.

Definition 3.11 (Admissibility). *Let $\mathcal{X} = \{X_1, \ldots, X_k\} \subseteq \mathcal{P}(\mathsf{ran(b)})$. We call $\mathcal{X}$* admissible *if there exist* O*-interpretations $\mathcal{I}_1 = (\Delta, \cdot^{\mathcal{I}_1}), \ldots, \mathcal{I}_k = (\Delta, \cdot^{\mathcal{I}_k})$ such that*

- *$x^{\mathcal{I}_i} = x^{\mathcal{I}_j}$ for all $x \in \mathsf{O_I} \cup \mathsf{O_{Crig}} \cup \mathsf{O_{Rrig}}$ and all $i, j \in \{1, \ldots, k\}$, and*
- *every $\mathcal{I}_i$, $1 \leq i \leq k$, is a model of the $\mathcal{L}_{\mathsf{O}}$-BKB $\mathfrak{B}_{X_i} = (\mathcal{B}_{X_i}, \mathcal{R}_{\mathsf{O}})$ over* O *where*

$$\mathcal{B}_{X_i} := \bigwedge_{\mathsf{b}([\![\alpha]\!]) \in X_i} \alpha \;\wedge \bigwedge_{\mathsf{b}([\![\alpha]\!]) \in \mathsf{ran(b)} \setminus X_i} \neg\alpha.$$

◇

Note that any subset $\mathcal{X}' \subseteq \mathcal{X}$ is admissible if $\mathcal{X}$ is admissible. Intuitively, the sets X_i in an admissible set $\mathcal{X}$ consist of referring meta concepts such that the corresponding o-axioms 'fit together'. Consider again Example 3.9. Clearly, the set $\{A_{[\![A \sqsubseteq \bot]\!]}, A_{[\![A(a)]\!]}\} \in \mathcal{P}(\mathsf{ran(b)})$ *cannot* be contained in any admissible set $\mathcal{X}$.

The next definition captures the above mentioned restriction on the model $\mathcal{H}$ of $\mathfrak{B}^{\mathsf{b}}$. The set of restricted types that occur in the meta interpretation must be a subset of some admissible set.

Definition 3.12 (Outer consistency). *Let $\mathcal{X} \subseteq \mathcal{P}(\mathsf{ran(b)})$. We call the $\mathcal{L}_{\mathsf{M}}$-BKB $\mathfrak{B}^{\mathsf{b}}$ over* M outer consistent w.r.t. $\mathcal{X}$ *if there exists a model of $\mathfrak{B}^{\mathsf{b}}$ that weakly respects* $(\mathsf{ran(b)}, \mathcal{X})$. ◇

The next two lemmas show that the consistency problem in $\mathcal{L}_{\mathsf{M}}[\![\mathcal{L}_{\mathsf{O}}]\!]$ can be decided by checking whether there is an admissible set $\mathcal{X}$ such that outer abstraction of the given $\mathcal{L}_{\mathsf{M}}[\![\mathcal{L}_{\mathsf{O}}]\!]$-BKB is outer consistent w.r.t. $\mathcal{X}$.

Lemma 3.13. *For every* M*-interpretation $\mathcal{H} = (\Delta^{\mathcal{H}}, \cdot^{\mathcal{H}})$, the following two statements are equivalent:*

1. *There exists a model $\mathcal{J}$ of $\mathfrak{B}$ with $\mathcal{J}^{\mathsf{b}} = \mathcal{H}$.*
2. *$\mathcal{H}$ is a model of $\mathfrak{B}^{\mathsf{b}}$ and the set $\{X_d \mid d \in \Delta^{\mathcal{H}}\}$ is admissible, where X_d is defined as $X_d := \mathsf{type}^{\mathcal{H}}_{\mathsf{ran(b)}}(d)$.*

Proof. (1 ⇒ 2): Let $\mathcal{J} = (\mathbb{C}, \cdot^{\mathcal{J}}, \Delta^{\mathcal{J}}, (\cdot^{\mathcal{I}_c})_{c\in\mathbb{C}})$ be a model of $\mathfrak{B}$ with $\mathcal{J}^\mathsf{b} = \mathcal{H}$. Since $\mathcal{J}^\mathsf{b} = \mathcal{H}$, we have that $\mathbb{C} = \Delta^{\mathcal{H}}$. By Lemma 3.8, we have that $\mathcal{H}$ is a model of $\mathfrak{B}^\mathsf{b}$. Moreover, since b is a bijection between referring meta concepts of the form $[\![\alpha]\!]$ occurring in $\mathfrak{B}$ and abstracted concept names of $\mathsf{M_C}$, we have that $\mathsf{ran}(\mathsf{b})$ is finite, and thus also that the set $\mathcal{X} := \{X_d \mid d \in \Delta^{\mathcal{H}}\} \subseteq \mathcal{P}(\mathsf{ran}(\mathsf{b}))$ is finite. Let $\mathcal{X} = \{Y_1, \ldots, Y_k\}$. Since $\mathbb{C} = \Delta^{\mathcal{H}}$, there exists an index function $\nu\colon \mathbb{C} \to \{1, \ldots, k\}$ such that $X_c = Y_{\nu(c)}$ for every $c \in \mathbb{C}$, i.e.

$$\begin{aligned} Y_{\nu(c)} &= \big\{\mathsf{b}([\![\alpha]\!]) \mid [\![\alpha]\!] \text{ occurs in } \mathfrak{B} \text{ and } c \in \mathsf{b}([\![\alpha]\!])^{\mathcal{H}}\big\} \\ &= \big\{\mathsf{b}([\![\alpha]\!]) \mid [\![\alpha]\!] \text{ occurs in } \mathfrak{B} \text{ and } \mathcal{I}_c \models \alpha\big\}. \end{aligned}$$

Conversely, for every $\mu \in \{1, \ldots, k\}$, there is an element $c \in \mathbb{C}$ such that $\nu(c) = \mu$. The O-interpretations for showing admissibility of $\mathcal{X}$ are obtained as follows. Take $c_1, \ldots, c_k \in \mathbb{C}$ such that $\nu(c_1) = 1, \ldots, \nu(c_k) = k$. Now, for every i, $1 \le i \le k$, we define the O-interpretation $\mathcal{G}_i := (\Delta, \cdot^{\mathcal{I}_{c_i}})$. Clearly, we have that $\mathcal{G}_i \models \mathcal{B}_{Y_i}$ and since $\mathcal{J} \models \mathcal{R}_\mathsf{O}$, we have that $\mathcal{G}_i \models \mathfrak{B}_{Y_i}$. Moreover, the definition of a nested interpretation yields that $x^{\mathcal{G}_i} = x^{\mathcal{G}_j}$ for all $x \in \mathsf{O_I} \cup \mathsf{O_{Crig}} \cup \mathsf{O_{Rrig}}$ and all $i, j \in \{1, \ldots, k\}$. Hence, the O-interpretations $\mathcal{G}_1, \ldots, \mathcal{G}_k$ attest admissibility of $\mathcal{X}$.

(2 ⇒ 1): Assume that $\mathcal{H} = (\Delta^{\mathcal{H}}, \cdot^{\mathcal{H}})$ is a model of $\mathfrak{B}^\mathsf{b}$ and that the set $\mathcal{X} := \{X_d \mid d \in \Delta^{\mathcal{H}}\}$ is admissible. Again, since $\mathsf{ran}(\mathsf{b})$ is finite, we have that $\mathcal{X} \subseteq \mathcal{P}(\mathsf{ran}(\mathsf{b}))$ is finite. Let $\mathcal{X} = \{Y_1, \ldots, Y_k\}$. Since $\mathcal{X}$ is admissible, there are O-interpretations $\mathcal{G}_1 = (\Delta^{\mathcal{G}}, \cdot^{\mathcal{G}_1}), \ldots, \mathcal{G}_k = (\Delta^{\mathcal{G}}, \cdot^{\mathcal{G}_k})$ such that $\mathcal{G}_i \models \mathfrak{B}_{Y_i}$ and $x^{\mathcal{G}_i} = x^{\mathcal{G}_j}$ for all $x \in \mathsf{O_I} \cup \mathsf{O_{Crig}} \cup \mathsf{O_{Rrig}}$ and all $i, j \in \{1, \ldots, k\}$. Furthermore, there exists an index function $\nu\colon \Delta^{\mathcal{H}} \to \{1, \ldots, k\}$ such that $Y_{\nu(d)} = X_d$ for every $d \in \Delta^{\mathcal{H}}$. We define a nested interpretation $\mathcal{J} = (\mathbb{C}, \cdot^{\mathcal{J}}, \Delta^{\mathcal{J}}, (\cdot^{\mathcal{I}_c})_{c\in\mathbb{C}})$ as follows:

- $\mathbb{C} := \Delta^{\mathcal{H}}$,
- $x^{\mathcal{J}} := x^{\mathcal{H}}$ for every $x \in \mathsf{M_C} \cup \mathsf{M_R} \cup \mathsf{M_I}$,
- $\Delta^{\mathcal{J}} := \Delta^{\mathcal{G}}$, and
- $x^{\mathcal{I}_c} := x^{\mathcal{G}_{\nu(c)}}$ for every $x \in \mathsf{O_C} \cup \mathsf{O_R} \cup \mathsf{O_I}$ and every $c \in \mathbb{C}$.

By construction of $\mathcal{J}$, we have that $x^{\mathcal{J}^\mathsf{b}} = x^{\mathcal{H}}$ for every $x \in \mathsf{M_R} \cup \mathsf{M_I} \cup (\mathsf{M_C} \setminus \mathsf{ran}(\mathsf{b}))$. Let $A \in \mathsf{ran}(\mathsf{b})$, and let $\mathsf{b}^{-1}(A) = [\![\alpha]\!]$. We have for every $d \in \Delta^{\mathcal{H}} = \mathbb{C}$ that $d \in A^{\mathcal{J}^\mathsf{b}}$ iff $d \in (\mathsf{b}^{-1}(A))^{\mathcal{J}}$ iff $d \in [\![\alpha]\!]^{\mathcal{J}}$ iff $\mathcal{I}_d \models \alpha$ iff $\mathcal{G}_{\nu(d)} \models \alpha$ iff $\mathsf{b}([\![\alpha]\!]) = A \in Y_{\nu(d)}$ (since $\mathcal{G}_{\nu(d)} \models \mathcal{B}_{Y_{\nu(d)}}$) iff $A \in X_d$ iff $d \in A^{\mathcal{H}}$. Hence, we have $\mathcal{J}^\mathsf{b} = \mathcal{H}$. Since $\mathcal{H}$ is a model of $\mathfrak{B}^\mathsf{b}$ and, by construction of $\mathcal{J}$, $\mathcal{J}$ is a model of $\mathcal{R}_\mathsf{O}$, we have by Lemma 3.8 that $\mathcal{J}$ is a model of $\mathfrak{B}$. □

The following lemma is a direct consequence of the previous one. It states that we can split the consistency check into two subtasks.

Lemma 3.14. *The $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$-BKB $\mathfrak{B}$ is consistent iff there is a set $\mathcal{X} \subseteq \mathcal{P}(\mathsf{ran}(\mathsf{b}))$ such that*

1. *$\mathcal{X}$ is admissible, and*
2. *$\mathfrak{B}^\mathsf{b}$ is outer consistent w.r.t. $\mathcal{X}$.*

Proof. (⟹): Let $\mathcal{J}$ be a model of $\mathfrak{B}$, and let $\mathcal{J}^\mathsf{b} = (\mathbb{C}, \cdot^{\mathcal{J}^\mathsf{b}})$. By Lemma 3.13, we have that $\mathcal{J}^\mathsf{b}$ is a model of $\mathfrak{B}^\mathsf{b}$, and the set $\mathcal{X} := \{\mathsf{type}^{\mathcal{J}^\mathsf{b}}_{\mathsf{ran}(\mathsf{b})}(c) \mid c \in \mathbb{C}\}$ is admissible. By construction, $\mathcal{J}^\mathsf{b}$ weakly respects $(\mathsf{ran}(\mathsf{b}), \mathcal{X})$, and hence $\mathfrak{B}^\mathsf{b}$ is outer consistent w.r.t. $\mathcal{X}$.

($\Longleftarrow$): Let $\mathcal{X} = \{X_1, \ldots, X_k\} \subseteq \mathcal{P}(\mathsf{ran}(b))$ such that $\mathcal{X}$ is admissible and $\mathfrak{B}^\mathsf{b}$ is outer consistent w.r.t. $\mathcal{X}$. Hence there is a model $\mathcal{H} = (\Delta^\mathcal{H}, \cdot^\mathcal{H})$ of $\mathfrak{B}^\mathsf{b}$ that weakly respects $(\mathsf{ran}(\mathsf{b}), \mathcal{X})$. We define $\mathcal{Z} := \{\mathsf{type}^\mathcal{H}_{\mathsf{ran}(\mathsf{b})}(c) \mid c \in \Delta^\mathcal{H}\}$. Since $\mathcal{H}$ weakly respects $(\mathsf{ran}(\mathsf{b}), \mathcal{X})$, we have that $\mathcal{Z} \subseteq \mathcal{X}$. Since $\mathcal{X}$ is admissible, this yields admissibility of $\mathcal{Z}$. Lemma 3.13 now yields consistency of $\mathfrak{B}$. □

Before we can analyse the complexity of the consistency problem in $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$, we need to state two complexity results for the consistency problem of $\mathcal{SHOQ}$-BKBs and $\mathcal{SHOIQ}$-BKBs. For the former we follow the idea of [Lip14]. The latter is an adaptation of a proof of [Pra05].

Lemma 3.15. *Deciding whether a $\mathcal{SHOQ}$-BKB $\mathfrak{B}^\mathsf{b}$ is outer consistent w.r.t. $\mathcal{X}$ can be done in time exponential in the size of $\mathfrak{B}^\mathsf{b}$ and linear in size of $\mathcal{X}$.*

Proof. It is enough to show that deciding whether $\mathfrak{B}^\mathsf{b}$ has a model that weakly respects $(\mathsf{ran}(\mathsf{b}), \mathcal{X})$ can be done in time exponential in the size of $\mathfrak{B}^\mathsf{b}$ and linear in the size of $\mathcal{X}$.

Here, we will adapt the ideas of [Lip14]. It is not hard to see that we can adapt the notion of a quasimodel respecting a pair $(\mathsf{ran}(\mathsf{b}), \mathcal{X})$ of [Lip14] to a quasimodel *weakly* respecting $(\mathsf{ran}(\mathsf{b}), \mathcal{X})$. Condition (i) in Definition 3.25 of [Lip14] states that for every $X \in \mathcal{X}$ there must exist a concept type that, restricted to $\mathsf{ran}(\mathsf{b})$, coincides with X. Hence, dropping Condition (i) yields that the quasimodel weakly respects $(\mathsf{ran}(\mathsf{b}), \mathcal{X})$. Then, the proof of Lemma 3.26 of [Lip14] can be adapted such that our claim follows. This is done by dropping the check whether Condition (i) is satisfied in Step 4 of the algorithm of [Lip14].

The only condition for which $\mathcal{X}$ is relevant, is Condition (h) which is checked in Step 2. Clearly, this check can be done in time linear in the size of $\mathcal{X}$. □

Lemma 3.16. *Deciding whether a $\mathcal{SHOIQ}$-BKB $\mathfrak{B}^\mathsf{b}$ is outer consistent w.r.t. $\mathcal{X}$ can be non-deterministically done in time exponential in the size of $\mathfrak{B}^\mathsf{b}$ and linear in size of $\mathcal{X}$.*

Proof. In [Pra05], it is shown the deciding satisfiability of a formula φ of the two-variable fragment of first-order logic with counting quantifiers $\mathcal{C}^2$ can be done non-deterministically in time exponential in the size of φ. There, Theorem 2 of [Pra05] can be adapted is such a way that the set Π of 1-types that occur in a model can be restricted.

Since $\mathcal{SHOIQ}$-BKBs are also $\mathcal{C}^2$-formulas and $\mathcal{X}$ is a set of 1-types, the claim follows. □

3.3.1 Consistency without rigid names

In this section, we consider the case where neither rigid concept names nor rigid role names are allowed.

Theorem 3.17. *The consistency problem in $\mathcal{SHOQ}[\![\mathcal{SHOQ}]\!]$ is in* ExpTime *if* $\mathsf{O}_\mathsf{Crig} = \mathsf{O}_\mathsf{Rrig} = \emptyset$.

Proof. Let $\mathfrak{B}$ be a $\mathcal{SHOQ}[\![\mathcal{SHOQ}]\!]$-BKB and $\mathfrak{B}^\mathsf{b}$ its outer abstraction. We can decide consistency of $\mathfrak{B}$ using Lemma 3.14. We define $\mathcal{X} := \{X \subseteq \mathsf{ran}(\mathsf{b}) \mid \mathfrak{B}_X \text{ is consistent}\}$ where $\mathfrak{B}_X$ is defined as in Definition 3.11. We first show that $\mathcal{X} = \{X_1, \ldots, X_k\}$ is admissible. Let $\mathcal{I}_i$ be a model of $\mathfrak{B}_{X_i}$, which exists since $\mathfrak{B}_{X_i}$ is consistent. Since $\mathsf{N_I}$ is countably infinite, interpretations must respect the UNA and due to the Löwenheim-Skolem theorem, we can assume that all models $\mathcal{I}_i$, $1 \leq i \leq k$, have a countably infinite domain. Thus, w.l.o.g. we can assume that all models have the same domain Δ. Furthermore, we can assume that individual names are interpreted the same. Since $\mathsf{O}_\mathsf{Crig} = \mathsf{O}_\mathsf{Rrig} = \emptyset$, the set $\mathcal{X}$ fulfills all conditions of

Definition 3.11 for admissibility. Thus, if $\mathfrak{B}^\mathsf{b}$ is outer consistent w.r.t. $\mathcal{X}$, then we have by Lemma 3.14 that $\mathfrak{B}$ is consistent.

Conversely, assume that $\mathfrak{B}$ is consistent. Then, by Lemma 3.14, there is an admissible set $\mathcal{X}' \subseteq \mathcal{P}(\mathsf{ran}(\mathsf{b}))$ and $\mathfrak{B}^\mathsf{b}$ is outer consistent w.r.t. $\mathcal{X}'$. Since $\mathcal{X}$ is the maximal admissible subset of $\mathcal{P}(\mathsf{ran}(\mathsf{b}))$, we have $\mathcal{X}' \subseteq \mathcal{X}$. If $\mathfrak{B}^\mathsf{b}$ is outer consistent w.r.t. $\mathcal{X}'$, it is also outer consistent w.r.t. $\mathcal{X}$. Hence, $\mathfrak{B}$ is consistent iff $\mathfrak{B}^\mathsf{b}$ is outer consistent w.r.t. $\mathcal{X}$, which yields a decision procedure for the consistency problem in $\mathcal{SHOQ}[\![\mathcal{SHOQ}]\!]$.

It remains to analyze the complexity. There are exponentially many $\mathcal{X} \in \mathcal{P}(\mathsf{ran}(\mathsf{b}))$, but each $\mathcal{SHOQ}$-BKB $\mathfrak{B}_X$ can be constructed in time polynomial in the size of $\mathfrak{B}$. We can decide consistency of $\mathfrak{B}_X$ in time exponential [Lip14]. Thus, the set $\mathcal{X}$ can be constructed in time exponential in the size of $\mathfrak{B}$ and it is of exponential size. Due to Lemma 3.15, deciding whether $\mathfrak{B}^\mathsf{b}$ is outer consistent w.r.t. $\mathcal{X}$ can be done in time exponential in the size of $\mathfrak{B}^\mathsf{b}$ and linear in the size of $\mathcal{X}$. Thus, overall we can decide the consistency problem in exponential time. □

Theorem 3.18. *If $\mathsf{O}_{\mathsf{Crig}} = \mathsf{O}_{\mathsf{Rrig}} = \emptyset$, the consistency problem in $\mathcal{SHOIQ}[\![\mathcal{SHOIQ}]\!]$ is in* NEXPTIME.

Proof. Let $\mathfrak{B}$ be a $\mathcal{SHOIQ}[\![\mathcal{SHOIQ}]\!]$-BKB and $\mathfrak{B}^\mathsf{b}$ is outer abstraction. Analogous to the proof of Theorem 3.17, we construct the maximal admissible subset $\mathcal{X}$ of $\mathcal{P}(\mathsf{ran}(\mathsf{b}))$. Again, $\mathfrak{B}$ is consistent iff $\mathfrak{B}^\mathsf{b}$ is outer consistent w.r.t. $\mathcal{X}$.

In contrast to the proof of Theorem 3.17, we can decide consistency of $\mathfrak{B}_X$ non-deterministically in time exponential in the size of $\mathfrak{B}_X$. Thus, the set $\mathcal{X}$ can be constructed non-deterministically in time exponential in the size of $\mathfrak{B}$. By Lemma 3.16, deciding whether $\mathfrak{B}^\mathsf{b}$ is outer consistent w.r.t. $\mathcal{X}$ can be non-deterministically done in time exponential in the size of $\mathfrak{B}^\mathsf{b}$ and linear in the size of $\mathcal{X}$. Overall we can decide the consistency problem non-deterministically in exponential time. □

As the lower bounds already hold for $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$ involving $\mathcal{EL}$, we prove them separately in Section 3.4. Anticipating the lower bounds shown in Theorems 3.23, 3.31, 3.24 and 3.32 we obtain EXPTIME-completeness for the consistency problem in $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$ for $\mathcal{L}_\mathsf{M}$ and $\mathcal{L}_\mathsf{O}$ being DLs between $\mathcal{EL}$ and $\mathcal{SHOQ}$, excluding $\mathcal{EL}[\![\mathcal{EL}]\!]$, and NEXPTIME-completeness for $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$ where either $\mathcal{L}_\mathsf{M}$ or $\mathcal{L}_\mathsf{O}$ is $\mathcal{SHOIQ}$, if $\mathsf{O}_{\mathsf{Crig}} = \mathsf{O}_{\mathsf{Rrig}} = \emptyset$.

3.3.2 Consistency with rigid role names

In this section, we consider the case where rigid role names are present.

Theorem 3.19. *The consistency problem in $\mathcal{SHOIQ}[\![\mathcal{SHOQ}]\!]$ is in* 2EXPTIME *if $\mathsf{O}_{\mathsf{Rrig}} \neq \emptyset$.*

Proof. Let $\mathfrak{B} = (\mathcal{B}, \mathcal{R}_\mathsf{O}, \mathcal{R}_\mathsf{M})$ be a $\mathcal{SHOIQ}[\![\mathcal{SHOQ}]\!]$-BKB and $\mathfrak{B}^\mathsf{b} = (\mathcal{B}^\mathsf{b}, \mathcal{R}_\mathsf{M})$ its outer abstraction. We can decide consistency of $\mathfrak{B}$ using Lemma 3.14. For that, we enumerate all sets $\mathcal{X} \subseteq \mathcal{P}(\mathsf{ran}(\mathsf{b}))$, which can be done in time doubly exponential in $\mathfrak{B}$. For each of these sets $\mathcal{X} = \{X_1, \ldots, X_k\}$, we check whether $\mathfrak{B}^\mathsf{b}$ is outer consistent w.r.t. $\mathcal{X}$, which can be done non-deterministically in time exponential in the size of $\mathfrak{B}^\mathsf{b}$ and linear in the size of $\mathcal{X}$.

Then, we check $\mathcal{X}$ for admissibility using the renaming technique of [BGL08; BGL12]. For every i, $1 \leq i \leq k$, every flexible concept name A occurring in $\mathcal{B}^\mathsf{b}$, and every flexible role name r occurring in $\mathcal{B}^\mathsf{b}$ or $\mathcal{R}_\mathsf{O}$, we introduce copies $A^{(i)}$ and $r^{(i)}$. The $\mathcal{SHOQ}$-

BKB $\mathfrak{B}^{(i)}_{X_i} = (\mathcal{B}^{(i)}_{X_i}, \mathcal{R}_\mathsf{O}{}^{(i)})$ over O is obtained from $\mathfrak{B}_{X_i}$ (see Definition 3.11) by replacing every occurrence of a flexible name x by $x^{(i)}$. We define

$$\mathfrak{B}_\mathcal{X} := \Big(\bigwedge\nolimits_{1\le i\le k} \mathcal{B}^{(i)}_{X_i}, \bigcup\nolimits_{1\le i\le k} \mathcal{R}_\mathsf{O}{}^{(i)} \Big).$$

<u>*Claim:*</u> *$\mathcal{X}$ is admissible iff $\mathfrak{B}_\mathcal{X}$ is consistent.*

<u>*Proof:*</u> *We prove the claim similarly to what was done in [Lip14].*

If $\mathcal{X}$ is admissible, then there exist O*-interpretations $\mathcal{G}_1 = (\Delta^\mathcal{G}, \cdot^{\mathcal{G}_1})$, ..., $\mathcal{G}_k = (\Delta^\mathcal{G}, \cdot^{\mathcal{G}_k})$. We define the* O*-interpretation $\mathcal{G} = (\Delta^\mathcal{G}, \cdot^\mathcal{G})$ as follows:*

$$\begin{aligned} x^\mathcal{G} &:= x^{\mathcal{G}_1} \textit{ for all } x \in \mathsf{O}_\mathsf{Crig} \cup \mathsf{O}_\mathsf{Rrig} \cup \mathsf{O}_\mathsf{I} \\ (x^{(i)})^\mathcal{G} &:= x^{\mathcal{G}_i} \textit{ for all } A \in (\mathsf{O}_\mathsf{C} \setminus \mathsf{O}_\mathsf{Crig}) \cup (\mathsf{O}_\mathsf{R} \setminus \mathsf{O}_\mathsf{Rrig}) \end{aligned}$$

It is not hard to verify that $\mathcal{G}$ is a model of $\mathfrak{B}_\mathcal{X}$.

If $\mathfrak{B}_\mathcal{X}$ is consistent, then there exists a model $\mathcal{G} = (\Delta^\mathcal{G}, \cdot^\mathcal{G})$. Analogously, we define the O*-interpretations $\mathcal{G}_1 = (\Delta^\mathcal{G}, \cdot^{\mathcal{G}_1})$, ..., $\mathcal{G}_k = (\Delta^\mathcal{G}, \cdot^{\mathcal{G}_k})$ as follows:*

$$x^{\mathcal{G}_i} := \begin{cases} x^\mathcal{G} & \textit{for all } x \in \mathsf{O}_\mathsf{Crig} \cup \mathsf{O}_\mathsf{Rrig} \cup \mathsf{O}_\mathsf{I} \\ (x^{(i)})^\mathcal{G} & \textit{for all } A \in (\mathsf{O}_\mathsf{C} \setminus \mathsf{O}_\mathsf{Crig}) \cup (\mathsf{O}_\mathsf{R} \setminus \mathsf{O}_\mathsf{Rrig}) \end{cases}$$

Again, it is not hard to verify that $\mathcal{X}$ is admissible. /

Note that $\mathfrak{B}_\mathcal{X}$ is of size at most exponential in $\mathfrak{B}$ and can be constructed in exponential time. Moreover, consistency of $\mathfrak{B}_\mathcal{X}$ can be decided in time exponential in the size of $\mathfrak{B}_\mathcal{X}$ [Lip14], and thus in time doubly exponential in the size of $\mathfrak{B}$. □

Theorem 3.20. *The consistency problem in $\mathcal{SHOIQ}[\![\mathcal{SHOIQ}]\!]$ is in* N2ExpTime *if* $\mathsf{O}_\mathsf{Rrig} \neq \emptyset$.

Proof. Let $\mathfrak{B} = (\mathcal{B}, \mathcal{R}_\mathsf{O}, \mathcal{R}_\mathsf{M})$ be a $\mathcal{SHOIQ}[\![\mathcal{SHOIQ}]\!]$-BKB and $\mathfrak{B}^\mathsf{b} = (\mathcal{B}^\mathsf{b}, \mathcal{R}_\mathsf{M})$ its outer abstraction. We proceed the same as in the proof of Theorem 3.19. Enumerating all sets $\mathcal{X} \subseteq \mathcal{P}(\mathsf{ran(b)})$ can still be done in time doubly exponential in the size of $\mathfrak{B}$, but checking for each $\mathcal{X}$ whether $\mathfrak{B}^\mathsf{b}$ is outer consistent w.r.t. $\mathcal{X}$ can be done non-deterministically in time exponential in the size of $\mathfrak{B}^\mathsf{b}$ and linear in the size of $\mathcal{X}$.

To check $\mathcal{X}$ for admissibility, we again use the renaming technique of [BGL08; BGL12], and by the same arguments as above, $\mathcal{X}$ is admissible iff $\mathfrak{B}_\mathcal{X}$ is consistent. Again, $\mathfrak{B}_\mathcal{X}$ is of size at most exponential in $\mathfrak{B}$ and can be constructed in exponential time, but consistency of $\mathfrak{B}_\mathcal{X}$ can be decided non-deterministically in time exponential in the size of $\mathfrak{B}_\mathcal{X}$, and thus non-deterministically in time doubly exponential in the size of $\mathfrak{B}$. □

Together with the lower bound shown in Theorem 3.33, we obtain 2ExpTime-completeness for the consistency problem in any $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$ between $\mathcal{EL}[\![\mathcal{ALC}]\!]$ and $\mathcal{SHOIQ}[\![\mathcal{SHOQ}]\!]$ if $\mathsf{O}_\mathsf{Rrig} \neq \emptyset$. The consistency problem in $\mathcal{L}_\mathsf{M}[\![\mathcal{SHOIQ}]\!]$ for $\mathcal{L}_\mathsf{M}$ between $\mathcal{EL}$ and $\mathcal{SHOIQ}$ is 2ExpTime-hard and in N2ExpTime, if $\mathsf{O}_\mathsf{Rrig} \neq \emptyset$.

3.3.3 Consistency with only rigid concept names

In this section, we consider the case where rigid concept are present, but rigid role names are not allowed.

Theorem 3.21. *The consistency problem in $\mathcal{SHOIQ}[\![\mathcal{SHOQ}]\!]$ is in* NEXPTIME *if* $\mathsf{O}_{\mathsf{Crig}} \neq \emptyset$ *and* $\mathsf{O}_{\mathsf{Rrig}} = \emptyset$.

Proof. Let $\mathfrak{B} = (\mathcal{B}, \mathcal{R}_{\mathsf{O}}, \mathcal{R}_{\mathsf{M}})$ be a $\mathcal{SHOIQ}[\![\mathcal{SHOQ}]\!]$-BKB and $\mathfrak{B}^{\mathsf{b}} = (\mathcal{B}^{\mathsf{b}}, \mathcal{R}_{\mathsf{M}})$ its outer abstraction. We can decide consistency of $\mathfrak{B}$ using Lemma 3.14. We first non-deterministically guess the set $\mathcal{X} = \{X_1, \ldots, X_k\} \subseteq \mathcal{P}(\mathsf{ran}(\mathsf{b}))$, which is of size at most exponential in $\mathfrak{B}$. Due to Lemma 3.16 we can check whether $\mathfrak{B}^{\mathsf{b}}$ is outer consistent w.r.t. $\mathcal{X}$ non-deterministically in time exponential in the size of $\mathfrak{B}^{\mathsf{b}}$ and linear in the size of $\mathcal{X}$.

It remains to check $\mathcal{X}$ for admissibility. Here we follow the ideas of [BGL08; BGL12; Lip14]. Instead of checking the consistency of $\mathfrak{B}_{\mathcal{X}}$ directly, which would yield a double-exponential time bound, we reduce it to k separate consistency checks each of which can be decided in time exponential in the size of $\mathcal{B}$. Therefore, for each individual name a in $\mathfrak{B}$ we guess the set of rigid concepts of which a is an instance. More precisely, let $\mathsf{O}_{\mathsf{Crig}}(\mathcal{B}) \subseteq \mathsf{O}_{\mathsf{Crig}}$ and $\mathsf{O}_{\mathsf{I}}(\mathcal{B}) \subseteq \mathsf{O}_{\mathsf{I}}$ be the sets of all rigid concept names and individual names occurring in $\mathcal{B}$, respectively. We non-deterministically guess a set $\mathcal{Y} \subseteq \mathcal{P}(\mathsf{O}_{\mathsf{Crig}}(\mathcal{B}))$ and a mapping $\kappa\colon \mathsf{O}_{\mathsf{I}}(\mathcal{B}) \to \mathcal{Y}$, which also can be done in time exponential in the size of $\mathfrak{B}$.

<u>Claim:</u> *$\mathcal{X}$ is admissible iff $\widehat{\mathfrak{B}}_{X_i}$ has a model that respects $(\mathsf{O}_{\mathsf{Crig}}(\mathcal{B}), \mathcal{Y})$, for all $1 \leq i \leq k$, where $\widehat{\mathfrak{B}}_{X_i}$ is defined as:*

$$\widehat{\mathfrak{B}}_{X_i} := \left(\mathcal{B}_{X_i} \wedge \bigwedge_{a \in \mathsf{O}_{\mathsf{I}}(\mathcal{B})} \left(\bigsqcap_{A \in \kappa(a)} A \sqcap \bigsqcap_{A \in \mathsf{O}_{\mathsf{Crig}}(\mathcal{B}) \setminus \kappa(a)} \neg A \right)(a),\ \mathcal{R}_{\mathsf{O}} \right).$$

<u>Proof:</u> *If $\mathcal{X}$ is admissible, then there exists an O-interpretation $\mathcal{G}_i = (\Delta^{\mathcal{G}}, \cdot^{\mathcal{G}_i})$ such that $\mathcal{G}_i$ is a model of $(\mathcal{B}_{X_i}, \mathcal{R}_{\mathsf{O}})$. Let $\mathcal{Y}$ be the set of all $\mathsf{O}_{\mathsf{Crig}}$-types in $\mathcal{G}_i$, i.e. $\mathcal{Y} = \{\mathsf{type}^{\mathcal{G}_i}_{\mathsf{O}_{\mathsf{Crig}}}(d) \mid d \in \Delta^{\mathcal{G}_i}\}$ and let κ map d to its type, i.e. $\kappa(d) = \mathsf{type}^{\mathcal{G}_i}_{\mathsf{O}_{\mathsf{Crig}}}(d)$. It is easy to see that $\mathcal{G}_i$ is a model of $\widehat{\mathfrak{B}}_{X_i}$, and by construction, $\mathcal{G}_i$ respects $(\mathsf{O}_{\mathsf{Crig}}(\mathcal{B}), \mathcal{Y})$.*

Now, assume that there exist O-interpretations $\mathcal{G}_1 = (\Delta^{\mathcal{G}_1}, \cdot^{\mathcal{G}_1})$, $\ldots$, $\mathcal{G}_k = (\Delta^{\mathcal{G}_k}, \cdot^{\mathcal{G}_k})$ s.t. $\mathcal{G}_i \models \widehat{\mathfrak{B}}_{X_i}$ and $\mathcal{G}_i$ respects $(\mathsf{O}_{\mathsf{Crig}}(\mathcal{B}), \mathcal{Y})$, for all $1 \leq i \leq k$. By the same arguments as in the proof of Theorem 3.17, we can assume w.l.o.g. that all models have a countably infinite domain. Since a disjoint union of $\mathcal{G}_i$ with itself is again a model of $\widehat{\mathfrak{B}}_{X_i}$, we can also assume that for each $Y \in \mathcal{Y}$ there are infinitely many elements d in $C_Y{}^{\mathcal{G}_i}$ where

$$C_Y := \bigsqcap_{A \in Y} A \sqcap \bigsqcap_{A \in \mathsf{O}_{\mathsf{Crig}} \setminus Y} \neg A.$$

Hence, we can partition the domain $\Delta^{\mathcal{G}_i}$ into

$$\Delta^{\mathcal{G}_i} = \bigcup_{Y \in \mathcal{Y}} P_i(Y) \quad \text{with} \quad P_i(Y) := \{d \in \Delta^{\mathcal{G}_i} \mid d \in C_Y\}.$$

Because of the above assumptions and since $\mathcal{G}_i \models \bigwedge_{a \in \mathsf{O}_{\mathsf{I}}(\mathcal{B})} C_{\kappa(a)}(a)$, there exist bijections $\pi_i : \Delta^{\mathcal{G}_1} \to \Delta^{\mathcal{G}_i}$, $2 \leq i \leq k$, such that

- *$\pi_i(P_1(Y)) = P_i(Y)$, for every $Y \in \mathcal{Y}$, and*
- *$\pi_i(a^{\mathcal{G}_1}) = a^{\mathcal{G}_i}$ for every $a \in \mathsf{O}_{\mathsf{I}}(\mathcal{B})$.*

Hence, we can assume that all $\mathcal{G}_i$ have the same domain and interpret individual names and rigid concept names in the same way. ⁄

The $\mathcal{SHOQ}$-BKB $\widehat{\mathfrak{B}}_{X_i}$ is of size polynomial in the size of $\mathcal{B}$ and can be constructed in time exponential in the size of $\mathcal{B}$. We can check if $\widehat{\mathfrak{B}}_{X_i}$ has a model that respects $(\mathsf{O}_{\mathsf{Crig}}(\mathcal{B}), \mathcal{Y})$ in time exponential in the size of $\widehat{\mathfrak{B}}_{X_i}$ [Lip14], and thus in time exponential in the size of $\mathfrak{B}$. □

Together with the lower bounds shown in Theorem 3.28 and 3.34, we obtain NExpTime-completeness for the consistency problem in $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$ for $\mathcal{L}_\mathsf{M}$ and $\mathcal{L}_\mathsf{O}$ being DLs between $\mathcal{EL}$ and $\mathcal{SHOIQ}$, excluding $\mathcal{EL}[\![\mathcal{EL}]\!]$, if $\mathsf{O}_{\mathsf{Crig}} \neq \emptyset$ and $\mathsf{O}_{\mathsf{Rrig}} = \emptyset$.

Summing up the results, we obtain the following corollary.

Corollary 3.22. *For $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$ between $\mathcal{EL}[\![\mathcal{ALC}]\!]$ and $\mathcal{SHOQ}[\![\mathcal{SHOQ}]\!]$, the consistency problem in $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$ is*

- ExpTime-*complete if* $\mathsf{O}_{\mathsf{Crig}} = \emptyset$ *and* $\mathsf{O}_{\mathsf{Rrig}} = \emptyset$,
- NExpTime-*complete if* $\mathsf{O}_{\mathsf{Crig}} \neq \emptyset$ *and* $\mathsf{O}_{\mathsf{Rrig}} = \emptyset$, *and*
- 2ExpTime-*complete if* $\mathsf{O}_{\mathsf{Rrig}} \neq \emptyset$.

For $\mathcal{SHOIQ}[\![\mathcal{L}_\mathsf{O}]\!]$ with $\mathcal{L}_\mathsf{O}$ between $\mathcal{ALC}$ and $\mathcal{SHOQ}$, the consistency problem in $\mathcal{SHOIQ}[\![\mathcal{L}_\mathsf{O}]\!]$ is

- NExpTime-*complete if* $\mathsf{O}_{\mathsf{Rrig}} = \emptyset$, *and*
- 2ExpTime-*complete if* $\mathsf{O}_{\mathsf{Rrig}} \neq \emptyset$.

For $\mathcal{L}_\mathsf{M}[\![\mathcal{SHOIQ}]\!]$ with $\mathcal{L}_\mathsf{M}$ between $\mathcal{EL}$ and $\mathcal{SHOIQ}$, the consistency problem in $\mathcal{L}_\mathsf{M}[\![\mathcal{SHOIQ}]\!]$ is

- NExpTime-*complete if* $\mathsf{O}_{\mathsf{Crig}} = \emptyset$ *and* $\mathsf{O}_{\mathsf{Rrig}} = \emptyset$,
- NExpTime-*hard and in* N2ExpTime *if* $\mathsf{O}_{\mathsf{Crig}} \neq \emptyset$ *and* $\mathsf{O}_{\mathsf{Rrig}} = \emptyset$, *and*
- 2ExpTime-*hard and in* N2ExpTime *if* $\mathsf{O}_{\mathsf{Rrig}} \neq \emptyset$.

3.4 Contextualised Description Logics Involving $\mathcal{EL}$

In this section, we give some complexity results for context DLs $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$ where $\mathcal{L}_\mathsf{M}$ or $\mathcal{L}_\mathsf{O}$ is $\mathcal{EL}$. In Section 3.4.1, we consider $\mathcal{L}_\mathsf{M}[\![\mathcal{EL}]\!]$ where $\mathcal{L}_\mathsf{M}$ is between $\mathcal{ALC}$ and $\mathcal{SHOIQ}$. Then, in Section 3.4.2, we consider the remaining context DLs $\mathcal{EL}[\![\mathcal{L}_\mathsf{O}]\!]$ where $\mathcal{L}_\mathsf{O}$ is between $\mathcal{ALC}$ and $\mathcal{SHOIQ}$.

For $\mathcal{EL}$ as $\mathcal{L}_\mathsf{M}$, instead of considering $\mathcal{EL}[\![\mathcal{L}_\mathsf{O}]\!]$-BKBs, we allow only $\mathcal{EL}[\![\mathcal{L}_\mathsf{O}]\!]$-ontologies, i.e. *conjunctions* of m-axioms. It seems unnatural to allow axiom negation for a logic which does not have concept negation. Furthermore, this would miss any practical motivation. For $\mathcal{EL}[\![\mathcal{EL}]\!]$-ontologies the consistency problem becomes trivial since all $\mathcal{EL}[\![\mathcal{EL}]\!]$-ontologies are consistent, as $\mathcal{EL}$ lacks to express contradictions. This restriction, however, does not yield a better complexity in the cases of $\mathcal{EL}[\![\mathcal{L}_\mathsf{O}]\!]$, where $\mathcal{L}_\mathsf{O}$ is between $\mathcal{ALC}$ and $\mathcal{SHOQ}$.

3.4.1 The Contextualised Description Logics $\mathcal{L}_\mathsf{M}[\![\mathcal{EL}]\!]$

In this section, we consider $\mathcal{L}_\mathsf{M}[\![\mathcal{EL}]\!]$ where $\mathcal{L}_\mathsf{M}$ is between $\mathcal{ALC}$ and $\mathcal{SHOIQ}$. We start with the lower bounds for $\mathcal{ALC}[\![\mathcal{EL}]\!]$ and $\mathcal{SHOIQ}[\![\mathcal{EL}]\!]$ in the case without rigid names.

Theorem 3.23. *The consistency problem in $\mathcal{ALC}[\![\mathcal{EL}]\!]$ is* ExpTime*-hard if no rigid names are allowed, i.e.* $\mathsf{O}_\mathsf{Crig} = \mathsf{O}_\mathsf{Rrig} = \emptyset$.

Proof. Deciding whether a given conjunction of $\mathcal{ALC}$-axioms $\mathcal{B}$ is consistent is ExpTime-hard [Sch91]. Obviously, $\mathcal{B}$ is also an $\mathcal{ALC}[\![\mathcal{EL}]\!]$-BKB. □

Theorem 3.24. *The consistency problem in $\mathcal{SHOIQ}[\![\mathcal{EL}]\!]$ is* NExpTime*-hard if no rigid names are allowed, i.e.* $\mathsf{O}_\mathsf{Crig} = \mathsf{O}_\mathsf{Rrig} = \emptyset$.

Proof. Deciding whether a given conjunction of $\mathcal{ALCOIQ}$-axioms $\mathcal{B}$ is consistent is NExpTime-complete [Tob00]. Obviously, $\mathcal{B}$ is also an $\mathcal{SHOIQ}[\![\mathcal{EL}]\!]$-BKB. □

For the cases of rigid names, the lower bounds of NExpTime are obtained by a careful reduction of the satisfiability problem in the temporalised DL $\mathcal{EL}$-LTL [BT15b; BT15a], which is a fragment of $\mathcal{ALC}$-LTL introduced in [BGL08; BGL12]. For the sake of completeness, we recall the basic definitions of $\mathcal{L}$-LTL here, where $\mathcal{L}$ is a DL.

Definition 3.25 (Syntax of $\mathcal{L}$-LTL). $\mathcal{L}$-LTL-formulas over O *are defined by induction:*

- *if α is an $\mathcal{L}$-axiom over O, then α is an $\mathcal{L}$-LTL-formula, and*
- *if ϕ, ψ are $\mathcal{L}$-LTL-formulas over O, then so are $\neg\phi$ (negation), $\phi \wedge \psi$ (conjunction), $\phi \mathsf{U} \psi$ (until), $\mathsf{X}\phi$ (next), and*
- *nothing else is an $\mathcal{L}$-LTL-formula.* ♢

As usual in temporal logics, we use the following abbreviations:

- $\phi \vee \psi$ (disjunction) for $\neg(\neg\phi \wedge \neg\psi)$,
- true (tautology) for $A(a) \vee \neg A(a)$ where $A \in \mathsf{O}_\mathsf{C}$ is arbitrary but fixed,
- $\Diamond\phi$ (eventually) for $\mathsf{true}\,\mathsf{U}\,\phi$, and
- $\Box\phi$ (always) for $\neg\Diamond\neg\phi$.

The semantics of $\mathcal{L}$-LTL is based on DL-LTL-structures. These are sequences of O-interpretations over the same non-empty domain that additionally respect rigid names and the rigid individual assumption.

Definition 3.26 (DL-LTL-structure). *A* DL-LTL-structure over O *is a sequence* $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$ *of* O*-interpretations* $(\Delta, \cdot^{\mathcal{I}_i})$ *such that* $x^{\mathcal{I}_i} = x^{\mathcal{I}_j}$ *holds for all* $x \in \mathsf{O}_\mathsf{Crig} \cup \mathsf{O}_\mathsf{Rrig} \cup \mathsf{O}_\mathsf{I}$, $i, j > 0$. ♢

We are now ready to define the semantics of $\mathcal{L}$-LTL.

Definition 3.27 (Semantics of $\mathcal{L}$-LTL). *The validity of an $\mathcal{L}$-LTL-formula ϕ in a DL-LTL-structure $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$ at time $i \geq 0$, denoted by $\mathfrak{I}, i \models \varphi$, is defined inductively:*

$$
\begin{array}{lll}
\mathfrak{I}, i \models \alpha & \textit{iff} & \mathcal{I}_i \models \alpha \text{ where } \alpha \text{ is an } \mathcal{ALC}\text{-axiom over } \mathsf{O}, \\
\mathfrak{I}, i \models \phi \wedge \psi & \textit{iff} & \mathfrak{I}, i \models \phi \text{ and } \mathfrak{I}, i \models \psi, \\
\mathfrak{I}, i \models \neg\phi & \textit{iff} & \text{not } \mathfrak{I}, i \models \phi, \\
\mathfrak{I}, i \models \mathsf{X}\phi & \textit{iff} & \mathfrak{I}, i+1 \models \phi, \\
\mathfrak{I}, i \models \phi \,\mathsf{U}\, \psi & \textit{iff} & \text{there is } k \geq i \text{ such that } \mathfrak{I}, k \models \psi \text{ and} \\
 & & \mathfrak{I}, j \models \phi \text{ for all } j \text{ with } i \leq j < k.
\end{array}
$$

We call an $\mathcal{L}$-LTL-structure $\mathfrak{I}$ a model of ϕ *if $\mathfrak{I}, 0 \models \phi$. The* satisfiability problem in $\mathcal{L}$-LTL *is the question whether a given $\mathcal{L}$-LTL-formula ϕ has a model.* ♢

In [BT15b; BT15a], it is shown that the satisfiability problem in $\mathcal{EL}$-LTL is NExpTime-hard as soon as rigid concept names are present. We reduce the satisfiability problem in $\mathcal{EL}$-LTL to the consistency problem in $\mathcal{ALC}[\![\mathcal{EL}]\!]$ to obtain the lower bounds of NExpTime, where we use the fact that the lower bounds of [BT15b; BT15a] already hold for a syntactically restricted fragment of $\mathcal{EL}$-LTL.

Theorem 3.28. *The consistency problem in $\mathcal{ALC}[\![\mathcal{EL}]\!]$ is* NExpTime*-hard if $\mathsf{O}_{\mathsf{Crig}} \neq \emptyset$ and $\mathsf{O}_{\mathsf{Rrig}} = \emptyset$.*

Proof. In fact, the lower bounds hold for $\mathcal{EL}$-LTL-formulas of the form $\Box\phi$ where ϕ is an $\mathcal{EL}$-LTL-formula that contains only X as temporal operator [BT15a].

Let $\Box\phi$ be such an $\mathcal{EL}$-LTL-formula over O. Now, we obtain the m-concept C_ϕ from ϕ by replacing $\mathcal{EL}$-axioms α by $[\![\alpha]\!]$, $\wedge$ by $\sqcap$, and subformulas of the form $\mathsf{X}\psi$ by $\forall t.C_\psi \sqcap \exists t.C_\psi$, where $t \in \mathsf{M}_{\mathsf{R}}$ is arbitrary but fixed.

Claim: *$\Box\phi$ is satisfiable iff $\mathcal{B} = \top \sqsubseteq C_\phi \sqcap \exists t.\top$ is consistent.*

Proof: *First, assume that $\Box\phi$ is satisfiable. Take any DL-LTL-structure $\mathfrak{I} = (\Delta, \cdot^{\mathcal{I}_i})_{i \geq 0}$ with $\mathfrak{I}, 0 \models \Box\phi$. We define the nested interpretation $\mathcal{J} = (\mathbb{C}, \cdot^{\mathcal{J}}, \Delta^{\mathcal{J}}, (\cdot^{\mathcal{I}_c})_{c \in \mathbb{C}})$ as follows:*

$$
\begin{aligned}
\mathbb{C} &:= \{c_i \mid i \geq 0\}, \\
\Delta^{\mathcal{J}} &:= \Delta, \\
\cdot^{\mathcal{I}_{c_i}} &:= \cdot^{\mathcal{I}_i}, \\
t^{\mathcal{J}} &:= \{(c_i, c_{i+1}) \mid i \geq 0\}.
\end{aligned}
$$

We now show that for every $i \geq 0$, we have $\mathfrak{I}, i \models \phi$ iff $c_i \in C_\phi^{\mathcal{J}}$ by induction on the structure of ϕ:

$$
\begin{array}{ll}
\phi = \alpha: & \mathfrak{I}, i \models \phi \text{ iff } \mathcal{I}_i \models \alpha \text{ iff } \mathcal{I}_{c_i} \models \alpha \text{ iff } c_i \in [\![\alpha]\!]^{\mathcal{J}} = C_\phi^{\mathcal{J}}, \\
\phi = \neg\psi: & \mathfrak{I}, i \models \phi \text{ iff } \mathfrak{I}, i \not\models \psi \text{ iff } c_i \notin C_\psi^{\mathcal{J}} \text{ iff } c_i \in (\neg C_\psi)^{\mathcal{J}} = C_\phi^{\mathcal{J}}, \\
\phi = \psi_1 \wedge \psi_2: & \mathfrak{I}, i \models \phi \text{ iff } \mathfrak{I}, i \models \psi_1 \text{ and } \mathfrak{I}, i \models \psi_2 \text{ iff } c_i \in C_{\psi_1}^{\mathcal{J}} \text{ and } c_i \in C_{\psi_2}^{\mathcal{J}} \\
 & \qquad \text{iff } c_i \in (C_{\psi_1} \sqcap C_{\psi_2})^{\mathcal{J}} = C_\phi^{\mathcal{J}}, \text{ and} \\
\phi = \mathsf{X}\psi: & \mathfrak{I}, i \models \phi \text{ iff } \mathfrak{I}, i+1 \models \psi \text{ iff } c_{i+1} \in C_\psi^{\mathcal{J}} \text{ iff } c_i \in (\forall t.C_\psi \sqcap \exists t.C_\psi)^{\mathcal{J}} = C_\phi^{\mathcal{J}},
\end{array}
$$

where α is an $\mathcal{EL}$-axiom over O. It follows that $\mathfrak{I}, 0 \models \Box\phi$ iff $\mathcal{J} \models \top \sqsubseteq C_\phi$. Furthermore, since $(c_i, c_{i+1}) \in t^{\mathcal{J}}$, we have $c_i \in (\exists t.\top)^{\mathcal{J}}$. Thus, $\mathcal{J} \models \top \sqsubseteq \exists t.\top$.

For the 'if' direction, take any nested interpretation $\mathcal{J} = (\mathbb{C}, \cdot^{\mathcal{J}}, \Delta^{\mathcal{J}}, (\cdot^{\mathcal{I}_c})_{c \in \mathbb{C}})$ that is a model of $\top \sqsubseteq C_\phi \sqcap \exists t.\top$. Let P be an infinite path $\mathsf{P} = c_0 c_1 \ldots$ with $c_i \in \mathbb{C}$ and $(c_i, c_{i+1}) \in t^{\mathcal{J}}$ for *every $i \geq 0$. Such a path exists, because $\mathcal{J} \models \top \sqsubseteq \exists t.\top$. We define the nested interpretation $\mathcal{J}_{\mathsf{P}} := (\{c_i \mid i \geq 0\}, \cdot^{\mathcal{J}_{\mathsf{P}}}, \Delta^{\mathcal{J}}, (\cdot^{\mathcal{I}_{c_i}})_{i \geq 0})$ where $\cdot^{\mathcal{J}_{\mathsf{P}}}$ is the restriction of $\cdot^{\mathcal{J}}$ to the domain $\{c_i \mid i \geq 0\}$.*

By construction we have that $\mathcal{J}_{\mathsf{P}} \models \top \sqsubseteq \exists t.\top$. We show by a simple case distinction that $\mathcal{J}_{\mathsf{P}} \models \top \sqsubseteq C_\phi$. If ϕ does not contain any X-operator, then no meta role names occur in C_ϕ and the restriction on the set of worlds preserves the entailment relation. Otherwise, C_ϕ is of the form $C'_\phi \sqcap \forall t.C_\psi \sqcap \exists t.C_\psi$ where C'_ϕ is the possibly empty conjunction contained in C_ϕ that does not contain meta role names. Hence, $\mathcal{J}_{\mathsf{P}} \models C'_\phi$. Furthermore, since $\mathcal{J}_{\mathsf{P}} \models \top \sqsubseteq \exists t.\top$, $\mathcal{J}_{\mathsf{P}} \models \top \sqsubseteq C_\psi$ (by induction), and there is only one t-successor, we have $\mathcal{J}_{\mathsf{P}} \models \top \sqsubseteq C_\phi$. Hence, $\mathcal{J}_{\mathsf{P}} \models \top \sqsubseteq C_\phi \sqcap \exists t.\top$.

We define the DL-LTL-structure $\mathfrak{I}$ over O as $\mathfrak{I} := (\Delta^{\mathcal{J}}, \cdot^{\mathcal{I}_i})_{i \geq 0}$ where $\cdot^{\mathcal{I}_i} := \cdot^{\mathcal{I}_{c_i}}$. Again, we show that for every $i \geq 0$, that we have $c_i \in C_\phi^{\mathcal{J}_{\mathsf{P}}}$ iff $\mathfrak{I}, i \models \phi$ by induction on the structure of ϕ:

$\phi = \alpha:$ $\quad c_i \in C_\phi^{\mathcal{J}_{\mathsf{P}}} = [\![\alpha]\!]^{\mathcal{J}_{\mathsf{P}}}$ *iff* $\mathcal{I}_{c_i} \models \alpha$ *iff* $\mathcal{I}_i \models \alpha$ *iff* $\mathfrak{I}, i \models \phi$,

$\phi = \neg\psi:$ $\quad c_i \in C_\phi^{\mathcal{J}_{\mathsf{P}}} = (\neg C_\psi)^{\mathcal{J}_{\mathsf{P}}}$ *iff* $c_i \notin C_\psi^{\mathcal{J}_{\mathsf{P}}}$ *iff* $\mathfrak{I}, i \not\models \psi$ *iff* $\mathfrak{I}, i \models \phi$,

$\phi = \psi_1 \wedge \psi_2:$ $\quad c_i \in C_\phi^{\mathcal{J}_{\mathsf{P}}} = (C_{\psi_1} \sqcap C_{\psi_2})^{\mathcal{J}_{\mathsf{P}}}$ *iff* $c_i \in C_{\psi_1}^{\mathcal{J}_{\mathsf{P}}}$ *and* $c_i \in C_{\psi_1}^{\mathcal{J}_{\mathsf{P}}}$ *iff* $\mathfrak{I}, i \models \psi_1$ *and* $\mathfrak{I}, i \models \psi_2$ *iff* $\mathfrak{I}, i \models \phi$, *and*

$\phi = \mathsf{X}\psi:$ $\quad c_i \in C_\phi^{\mathcal{J}_{\mathsf{P}}} = (\forall t.C_\psi \sqcap \exists t.C_\psi)^{\mathcal{J}_{\mathsf{P}}}$ *iff* $c_{i+1} \in C_\psi^{\mathcal{J}_{\mathsf{P}}}$ *iff* $\mathfrak{I}, i+1 \models \psi$ *iff* $\mathfrak{I}, i \models \phi$,

where α is an $\mathcal{EL}$-axiom over O. It follows that $\mathcal{J}_P \models \top \sqsubseteq C_\phi$ iff $\mathfrak{I}, 0 \models \Box\phi$. ⁄

This claim yields the lower bound of NExpTime for the consistency problem in $\mathcal{ALC}[\![\mathcal{EL}]\!]$ if $\mathsf{O}_{\mathsf{Crig}} \neq \emptyset$. □

Next, we prove the upper bound of NExpTime for the consistency problem in the case of rigid names.

Theorem 3.29. *The consistency problem in $\mathcal{SHOIQ}[\![\mathcal{EL}]\!]$ is in* NExpTime *if $\mathsf{O}_{\mathsf{Rrig}} \neq \emptyset$.*

Proof. Let $\mathfrak{B} = (\mathcal{B}, \emptyset, \mathcal{R}_{\mathsf{M}})$ be a $\mathcal{SHOIQ}[\![\mathcal{EL}]\!]$-BKB and $\mathfrak{B}^{\mathsf{b}} = (\mathcal{B}^{\mathsf{b}}, \emptyset)$ its outer abstraction. We again use Lemma 3.14 to decide consistency of $\mathfrak{B}$. First, we non-deterministically guess a set $\mathcal{X} \subseteq \mathcal{P}(\mathsf{ran}(\mathsf{b}))$. By Lemma 3.16, we can decide outer consistency of $\mathcal{B}^{\mathsf{b}}$ w.r.t. $\mathcal{X}$ non-deterministically in time exponential in the size of $\mathcal{B}^{\mathsf{b}}$ and linear in the size of $\mathcal{X}$.

To check $\mathcal{X}$ for admissibility, we construct the $\mathcal{EL}$-BKB $\mathcal{B}_{\mathcal{X}}$ over O as in the proof of Theorem 3.19. This actually is a conjunction of $\mathcal{EL}$-literals over O, i.e. a conjunction of (negated) $\mathcal{EL}$-axioms over O. The following claim shows that consistency of $\mathcal{B}_{\mathcal{X}}$ can be reduced to consistency of a conjunction of $\mathcal{ELO}^\perp$-axioms over O, where $\mathcal{ELO}^\perp$ is the extension of $\mathcal{EL}$ with nominals and the bottom concept.

<u>Claim:</u> *For every conjunction of $\mathcal{EL}$-literals $\mathcal{B}$ over O, there exists an equisatisfiable conjunction $\mathcal{B}'$ of $\mathcal{ELO}^\perp$-axioms over O which is of size polynomial in the size of $\mathcal{B}$.*

<u>Proof:</u> *Let $\mathcal{B}$ be a conjunction of $\mathcal{EL}$-literals over O, i.e.*

$$\mathcal{B} = \alpha_1 \wedge \cdots \wedge \alpha_n \wedge \neg\beta_1 \wedge \cdots \wedge \neg\beta_m$$

where α_i, $1 \leq i \leq n$, β_j, $1 \leq j \leq m$ are $\mathcal{EL}$-axioms over O. We define $\mathcal{B}'$ as follows:

$$\mathcal{B}' = \alpha_1 \wedge \cdots \wedge \alpha_n \wedge \gamma_1 \wedge \cdots \wedge \gamma_m,$$

where

$$\gamma_i := \begin{cases} C(a_i) \wedge D'(a_i) \wedge D \sqcap D' \sqsubseteq \bot & \text{if } \beta_i = C \sqsubseteq D, \\ A'(a) \wedge A \sqcap A' \sqsubseteq \bot & \text{if } \beta_i = A(a), \text{ and} \\ \{a\} \sqcap \exists r.\{b\} \sqsubseteq \bot & \text{if } \beta_i = r(a,b) \end{cases}$$

with A', D' being fresh concept names and a_i being fresh individual names. It is easy to see that if an O*-interpretation $\mathcal{I}$ is a model of $\neg\beta_1 \wedge \cdots \wedge \neg\beta_m$, there exists an extension of $\mathcal{I}$ that is a model of $\gamma_1 \wedge \cdots \wedge \gamma_m$. Conversely, if an* O*-interpretation $\mathcal{I}'$ is a model of $\gamma_1 \wedge \cdots \wedge \gamma_m$, it is also a model of $\neg\beta_1 \wedge \cdots \wedge \neg\beta_m$. Hence $\mathcal{B}$ and $\mathcal{B}'$ are equisatisfiable. Clearly, $\mathcal{B}'$ is of size polynomial in the size of $\mathcal{B}$.* ⁄

Since $\mathcal{B}_\mathcal{X}$ is at most exponential in $\mathcal{B}$ and the fact that the consistency of conjunctions of $\mathcal{ELO}^\bot$-axioms can be decided in polynomial time [BBL05], we can check whether $\mathcal{B}_\mathcal{X}$ is consistent in time polynomial in the size of $\mathcal{B}_\mathcal{X}$ and, thus, in time exponential in the size of $\mathcal{B}$.

Overall, this yields the claimed upper bound. □

Summming up the results of this section, we obtain the following corollary.

Corollary 3.30. *For all $\mathcal{L}_\mathsf{M}$ between $\mathcal{ALC}$ and $\mathcal{SHOIQ}$, the consistency problem in $\mathcal{L}_\mathsf{M}[\![\mathcal{EL}]\!]$ is*

- ExpTime*-complete if $\mathsf{O}_\mathsf{Crig} = \emptyset$, $\mathsf{O}_\mathsf{Rrig} = \emptyset$ and $\mathcal{L}_\mathsf{M}$ is between $\mathcal{ALC}$ and $\mathcal{SHOQ}$, and*
- NExpTime*-complete otherwise.*

3.4.2 The Contextualised Description Logics $\mathcal{EL}[\![\mathcal{L}_\mathsf{O}]\!]$

In this section, we consider $\mathcal{EL}[\![\mathcal{L}_\mathsf{O}]\!]$ where $\mathcal{L}_\mathsf{O}$ is between $\mathcal{ALC}$ and $\mathcal{SHOQ}$. First, we show the lower bounds for the case without rigid names.

Theorem 3.31. *The consistency problem in $\mathcal{EL}[\![\mathcal{ALC}]\!]$ is* ExpTime*-hard if no rigid names are allowed, i.e. $\mathsf{O}_\mathsf{Crig} = \mathsf{O}_\mathsf{Rrig} = \emptyset$.*

Proof. Deciding whether a given conjunction $\mathcal{B} = \alpha_1 \wedge \cdots \wedge \alpha_n$ of $\mathcal{ALC}$-axioms is consistent is ExpTime-hard[Sch91]. Obviously, $\mathcal{B}$ is consistent iff the $\mathcal{EL}[\![\mathcal{ALC}]\!]$-BKB $([\![\alpha_1]\!] \sqcap \cdots \sqcap [\![\alpha_n]\!])(c)$ is consistent, where $c \in \mathsf{M}_\mathsf{I}$. □

Theorem 3.32. *The consistency problem in $\mathcal{EL}[\![\mathcal{SHOIQ}]\!]$ is* NExpTime*-hard if no rigid names are allowed, i.e. $\mathsf{O}_\mathsf{Crig} = \mathsf{O}_\mathsf{Rrig} = \emptyset$.*

Proof. Deciding whether a given conjunction $\mathcal{B} = \alpha_1 \wedge \cdots \wedge \alpha_n$ of $\mathcal{ALCOIQ}$-axioms is consistent is NExpTime-complete [Tob00]. Obviously, $\mathcal{B}$ is consistent iff the $\mathcal{EL}[\![\mathcal{SHOIQ}]\!]$-BKB $([\![\alpha_1]\!] \sqcap \cdots \sqcap [\![\alpha_n]\!])(c)$ is consistent, where $c \in \mathsf{M}_\mathsf{I}$. □

For the case of rigid role names, we have lower bounds of 2ExpTime.

Theorem 3.33. *The consistency problem in $\mathcal{EL}[\![\mathcal{ALC}]\!]$ is* 2ExpTime*-hard if $\mathsf{O}_\mathsf{Rrig} \neq \emptyset$.*

Proof. To show the lower bound, we adapt the proof ideas of [BGL08; BGL12], and reduce the word problem for exponentially space-bounded alternating Turing machines (i.e. is a given word w accepted by the machine M) to the consistency problem in $\mathcal{EL}[\![\mathcal{ALC}]\!]$ with rigid roles, i.e. $\mathsf{O}_{\mathsf{Rrig}} \neq \emptyset$. In [BGL08; BGL12], a reduction was provided to show 2ExpTime-hardness for the temporalised DL $\mathcal{ALC}$-LTL in the presence of rigid roles. Here, we mimic the properties of the time dimension that are important for the reduction using a role name $t \in \mathsf{M}_{\mathsf{R}}$.

Our $\mathcal{EL}[\![\mathcal{ALC}]\!]$-BKB is the conjunction of the $\mathcal{EL}[\![\mathcal{ALC}]\!]$-BKBs introduced below. First, we ensure that we never have a *last* time point:

$$\top \sqsubseteq \exists t.\top$$

Note that in the corresponding model, we do not enforce a t-chain since cycles are not prohibited. This, however, is not important in the reduction.

The $\mathcal{ALC}$-LTL-formula obtained in the reduction of [BGL08; BGL12] is a conjunction of $\mathcal{ALC}$-LTL-formulas of the form $\Box\phi$, where ϕ is an $\mathcal{ALC}$-LTL-formula. This makes sure that ϕ holds in all (temporal) worlds. For the cases where ϕ is an $\mathcal{ALC}$-axiom, we can simply express this by:

$$\top \sqsubseteq [\![\phi]\!]$$

This captures all except for two conjuncts of the $\mathcal{ALC}$-LTL-formula of the reduction of [BGL08; BGL12]. There, a k-bit binary counter using concept names $A'_0, \ldots, A'_{k-1}$ was attached to the individual name a, which is incremented along the temporal dimension. We can express something similar in $\mathcal{EL}[\![\mathcal{ALC}]\!]$. Instead of incrementing the counter values along a sequence of t-successors, we have to go backwards since $\mathcal{EL}$ does allow for branching but does not allow for value restrictions, i.e. we cannot make sure that all t-successors behave the same. More precisely, if the counter value n is attached to a in context c, the value $n+1$ (modulo 2^k-1) must be attached to a in *all* of c's t-predecessors. First, we ensure which bits must be flipped:

$$\bigwedge_{i<k}\Big(\exists t.\big([\![A'_0(a)]\!] \sqcap \cdots \sqcap [\![A'_{i-1}(a)]\!] \sqcap [\![A'_i(a)]\!]\big) \sqsubseteq [\![(\neg A'_i)(a)]\!]\Big)$$
$$\bigwedge_{i<k}\Big(\exists t.\big([\![A'_0(a)]\!] \sqcap \ldots \sqcap [\![A'_{i-1}(a)]\!] \sqcap [\![(\neg A'_i)(a)]\!]\big) \sqsubseteq [\![A'_i(a)]\!]\Big)$$

Next, we ensure that all other bits stay the same:

$$\bigwedge_{0<i<k}\ \bigwedge_{j<i}\Big(\exists t.\big([\![(\neg A'_j)(a)]\!] \sqcap [\![A'_i(a)]\!]\big) \sqsubseteq [\![A'_i(a)]\!]\Big)$$
$$\bigwedge_{0<i<k}\ \bigwedge_{j<i}\Big(\exists t.\big([\![(\neg A'_j)(a)]\!] \sqcap [\![(\neg A'_i)(a)]\!]\big) \sqsubseteq [\![(\neg A'_i)(a)]\!]\Big)$$

Note that due to the first m-axiom above, we enforce that every context has a t-successor. By the other m-axioms, we make sure that we enforce a t-chain of length 2^k. As in [BGL08; BGL12], it is not necessary to initialize the counter. Since we decrement the counter along the t-chain (modulo 2^k-1), every value between 0 and 2^k-1 is reached.

The conjunction of all the $\mathcal{EL}[\![\mathcal{ALC}]\!]$-BKBs above yields an $\mathcal{EL}[\![\mathcal{ALC}]\!]$-BKB $\mathcal{B}$ that is consistent iff the given word w is accepted by the machine M. □

Finally, we obtain a lower bound of NExpTime in the case of rigid concept names only.

Theorem 3.34. *The consistency problem in $\mathcal{EL}[\![\mathcal{ALC}]\!]$ is* NExpTime*-hard if* $\mathsf{O}_{\mathsf{Crig}} \neq \emptyset$ *and* $\mathsf{O}_{\mathsf{Rrig}} = \emptyset$.

Proof. To show the lower bound, we again adapt the proof ideas of [BGL08; BGL12], and reduce an exponentially bounded version of the domino problem to the consistency problem in $\mathcal{EL}[\![\mathcal{ALC}]\!]$ with rigid concepts, i.e. $\mathsf{O}_{\mathsf{Crig}} \neq \emptyset$ and $\mathsf{O}_{\mathsf{Rrig}} = \emptyset$. In [BGL08; BGL12], a reduction was provided to show NExpTime-hardness for the temporalised DL $\mathcal{ALC}$-LTL in the presence of rigid concepts. As in the proof of Theorem 3.33, we mimic the properties of the time dimension that are important for the reduction using a role name $t \in \mathsf{M}_{\mathsf{R}}$.
Our $\mathcal{EL}[\![\mathcal{ALC}]\!]$-BKB is the conjunction of the $\mathcal{EL}[\![\mathcal{ALC}]\!]$-BKBs introduced below. We proceed in a similar way as in the proof of Theorem 3.33. First, we ensure that we never have a *last* time point:

$$\top \sqsubseteq \exists t.\top$$

Note that in the corresponding model, we do not enforce a t-chain since cycles are not prohibited. As in the reduction in the proof of Theorem 3.33, this is not important in the reduction here.

Next, note that since the $\Box$-operator distributes over conjunction, most of the conjuncts of the $\mathcal{ALC}$-LTL-formula of the reduction of [BGL08; BGL12] can be rewritten as conjunctions of $\mathcal{ALC}$-LTL-formulas of the form $\Box\alpha$, where α is an $\mathcal{ALC}$-axiom. As already argued in the proof of Theorem 3.33, this can equivalently be expressed by $\top \sqsubseteq [\![\alpha]\!]$.

In [BGL08; BGL12], a $(2n+2)$-bit binary counter is employed using concept names $Z_0, \ldots, Z_{2n+1}$. This counter is attached to an individual name a, which is incremented along the temporal dimension. This can be expressed in $\mathcal{EL}[\![\mathcal{ALC}]\!]$ as shown in the proof of Theorem 3.33:

$$\bigwedge_{i<2n+2} \Big(\exists t.\big([\![Z_0(a)]\!] \sqcap \ldots \sqcap [\![Z_{i-1}(a)]\!] \sqcap [\![Z_i(a)]\!]\big) \sqsubseteq [\![(\neg Z_i)(a)]\!]\Big)$$

$$\bigwedge_{i<2n+2} \Big(\exists t.\big([\![Z_0(a)]\!] \sqcap \ldots \sqcap [\![Z_{i-1}(a)]\!] \sqcap [\![(\neg Z_i)(a)]\!]\big) \sqsubseteq [\![Z_i(a)]\!]\Big)$$

$$\bigwedge_{0<i<2n+2} \bigwedge_{j<i} \Big(\exists t.\big([\![(\neg Z_j)(a)]\!] \sqcap [\![Z_i(a)]\!]\big) \sqsubseteq [\![Z_i(a)]\!]\Big)$$

$$\bigwedge_{0<i<2n+2} \bigwedge_{j<i} \Big(\exists t.\big([\![(\neg Z_j)(a)]\!] \sqcap [\![(\neg Z_i)(a)]\!]\big) \sqsubseteq [\![(\neg Z_i)(a)]\!]\Big)$$

Note that due to the first m-axiom above, we enforce that every context has a t-successor. By the other m-axioms, we make sure that we enforce a t-chain of length 2^{2n+2}. As in [BGL08; BGL12], it is not necessary to initialize the counter. Since we decrement the counter along the t-chain (modulo 2^{2n+1}), every value between 0 and 2^{2n+1} is reached.

In [BGL08; BGL12], an $\mathcal{ALC}$-LTL-formula is used to express that the value of the counter is shared by all domain elements belonging to the current (temporal) world. This is expressed using a disjunction, which we can simulate as follows:

$$\bigwedge_{0 \leq i \leq 2n+1} \Big([\![Z_i(a)]\!] \sqsubseteq [\![\top \sqsubseteq Z_i]\!] \;\wedge\; [\![(\neg Z_i)(a)]\!] \sqsubseteq [\![Z_i \sqsubseteq \bot]\!]\Big)$$

Table 3.2: Classification of different two-dimensional temporal and contextual description logics ([LWZ08; BGL12; KG10])

Temporal or contextual operators		in front of/around axioms: yes	in front of/around axioms: no
inside concepts	yes	temporal $\mathrm{LTL}_{\mathcal{ALC}}$, $\mathcal{ALC}_{\mathcal{ALC}}$, $\mathcal{L}_{\mathsf{M}}[\![\mathcal{L}_{\mathsf{O}}]\!]^{+}$	$\mathrm{LTL}_{\mathcal{ALC}}$, $\emptyset$
	no	$\mathcal{ALC}$-LTL, $\mathcal{L}_{\mathsf{M}}[\![\mathcal{L}_{\mathsf{O}}]\!]$	$\mathcal{ALC}$

Next, there is a concept name N, which is required to be non-empty in every (temporal) world. We express this using a role name $r \in \mathsf{O}_{\mathsf{R}}$:

$$\top \sqsubseteq [\![(\exists r.N)(a)]\!]$$

It is only left to express the following $\mathcal{ALC}$-LTL-formula of [BGL08; BGL12] that states that every world gets one domino type:

$$\Box\Big(\bigvee_{d \in D} (\top \sqsubseteq d')\Big)$$

For readability, let $D = \{d_1, \ldots, d_k\}$. We use non-convexity of $\mathcal{ALC}$ as follows to express this:

$$\top \sqsubseteq \big[\!\big[(d_1' \sqcup \cdots \sqcup d_k')(a)\big]\!\big] \;\wedge \bigwedge_{1 \le i \le k} \Big([\![d_i'(a)]\!] \sqsubseteq [\![\top \sqsubseteq d_i']\!]\Big)$$

The conjunction of all the $\mathcal{EL}[\![\mathcal{ALC}]\!]$-BKBs above yields an $\mathcal{EL}[\![\mathcal{ALC}]\!]$-BKB $\mathcal{B}$ that is consistent iff the exponentially bounded version of the domino problem has a solution. □

3.5 Adding Contextualised Concepts

In this section we discuss a possible extension to our contextualised description logic. We start with a comparison to temporal description logics. Both DL-LTL-structures and nested DL interpretations use a *possible worlds semantics*. Single time points or contexts are represented in a meta dimension and for each such a meta element, or possible world, there exists one DL interpretation on the object level. Important for both the expressivity and the complexity of reasoning problems is how these two dimensions can interact. Syntactically, it is a question where these *meta* operators, i.e. temporal or contextual, can be used.

Table 3.2 gives an overview over some two-dimensional DLs. This is not a complete overview, but it illustrates some common properties about the complexity of the consistency problem. Note first that neither having meta operators in front of axioms nor having meta operators inside concepts is strictly more expressive than the other. Meta operators in front of axioms can handle general knowledge that holds in some or all worlds. On the other hand, meta operators inside object concepts allow to access the extension of concepts in other worlds. We illustrate this by an example taken from [LWZ08].

$$\Diamond\Box(\mathsf{European\_country} \sqsubseteq \mathsf{EU\_country})$$
$$\mathsf{Independent\_country} \sqsubseteq \Box\mathsf{Independent\_country}$$

The first temporalised GCI states that eventually, i.e. at some time point t in the future, all European countries will forever be EU members, i.e. in every time point after t. The second axiom says that the extension of the concept $\mathsf{Independent\_country}$ does not decrease.

In $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$, we only have the contextual operator $[\![\cdot]\!]$ around axioms. We can express that certain axioms must hold in some worlds, but we cannot access the extension of a concept from another world, i.e. we cannot express the set of domain elements which belong to some concept in another context. The idea is to overcome this lack of expressive power by introducing a new contextual operator inside object concepts.

Before extending $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$, we analyse some general behaviour of two-dimensional DLs. The first common property that we want to focus on is the computational complexity if rigid roles are present. If meta operators are allowed within object concepts, the consistency problem becomes undecidable. This holds for all logics in the first row of Table 3.2. We prove this negative result for $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]^+$ below. If meta operators are only allowed in front of axioms, we may retain decidability, but at the cost that the consistency problem is one exponential harder. If no rigid names are allowed, the complexity of the consistency problem increases for logics with meta operators both in front of axioms and in object concepts: from ExpTime-completeness for $\mathcal{ALC}$ to ExpSpace-completeness for temporal $\mathrm{LTL}_{\mathcal{ALC}}$ TBoxes and to 2ExpTime-completeness for $\mathcal{ALC}_{\mathcal{ALC}}$ knowledge bases. If only one kind of meta operators is allowed, the complexity class stays the same, i.e. the consistency problem is ExpTime-complete in $\mathcal{ALC}[\![\mathcal{ALC}]\!]$, $\mathcal{ALC}$-LTL, and $\mathrm{LTL}_{\mathcal{ALC}}$.

A setting in which only rigid concepts, but no rigid roles are allowed, is only interesting if meta operators are not allowed inside object concepts. Otherwise, rigid concepts can easily be simulated. In $\mathrm{LTL}_{\mathcal{ALC}}$, this can be done by adding $C \sqsubseteq \Box C$ and $\neg C \sqsubseteq \Box\neg C$ to the TBox. We show below how rigid concepts can be simulated in $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]^+$. The contextualised description logic $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]^+$ is an extension of $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$ in which we additionally allow contextualised object concepts. Therefore, we update the definition of o-concepts from Def. 3.1.

Definition 3.35 (Object concepts of $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]^+$). *The set of* concepts of the object logic $\mathcal{L}_\mathsf{O}$ (o-concepts) *is the smallest set such that*

- *for all $A \in \mathsf{O_C}$, A is an object concept,*
- *if D is an object concept, $\mathbf{C} \in \mathsf{M_C}$, $\mathbf{r} \in \mathsf{M_R}$, then $\Diamond_{\mathbf{r},\mathbf{C}}D$ is also an object concept, and*
- *all complex concepts that can be built with the concept constructors allowed in $\mathcal{L}_\mathsf{O}$ are object concepts.*

Furthermore, for a nested interpretation $\mathcal{J} = (\mathbb{C}, \cdot^\mathcal{J}, \Delta^\mathcal{J}, (\cdot^{\mathcal{I}_c})_{c\in\mathbb{C}})$ the mapping $\cdot^{\mathcal{I}_c}$ is extended to $\Diamond_{\mathbf{r},\mathbf{C}}D$ as follows: $(\Diamond_{\mathbf{r},\mathbf{C}}D)^{\mathcal{I}_c} := \{d \in \Delta^\mathcal{J} \mid$ there is some $c' \in \mathbf{C}^\mathcal{J}$ s.t. $(c,c') \in \mathbf{r}^\mathcal{J}$ and $d \in D^{\mathcal{I}_{c'}}\}$. ◊

Following customs of modal logic, we use $\Box_{\mathbf{r},\mathbf{C}}D$ as an abbreviation for $\neg\Diamond_{\mathbf{r},\mathbf{C}}(\neg D)$. Intuitively, in a context c the concept $\Diamond_{\mathbf{r},\mathbf{C}}D$ denotes the set of all object domain elements that belong to the concept D in some other context c' which belongs to the meta concept **C** and is related to c via **r**. An object domain element is in the extension of concept $\Box_{\mathbf{r},\mathbf{C}}D$ in context c, if it belongs to D in all contexts c' that belong to the meta concept **C** and are related to c via **r**.

Thus, within a context we can talk about object elements that belong to some object concept in some other context. This is somehow similar to $\mathsf{X}C$ in $\mathrm{LTL}_{\mathcal{ALC}}$, which denotes the set of all elements that are in C *in the next time point*.

Example 3.36. *Going back to Example 3.4, the following meta concept assertion states that someone who plays quarterback for the Green Bay Packers must work as coach in a junior training camp that is organised by Green Bay:*

$$\llbracket \exists \mathsf{plays.Quarterback} \sqsubseteq \Diamond_{\mathsf{organises,JuniorFootballClinic}}\mathsf{Coach} \rrbracket(\mathit{GreenBayPackers}).$$

Note here, that Green Bay Packers and the junior training camp are two different contexts and that this cannot be expressed in $\mathcal{L}_{\mathsf{M}}\llbracket\mathcal{L}_{\mathsf{O}}\rrbracket$. $\Diamond$

The contextualised description logic $\mathcal{ALC}\llbracket\mathcal{ALC}\rrbracket^{+}$ without rigid names is a syntactical variant of $\mathcal{ALC}_{\mathcal{ALC}}$ [KG10; KG16]. Consequently, the consistency problem in $\mathcal{ALC}\llbracket\mathcal{ALC}\rrbracket^{+}$ has the same complexity.

Theorem 3.37. *The consistency problem in* $\mathcal{ALC}\llbracket\mathcal{ALC}\rrbracket^{+}$ *is* 2EXPTIME*-complete if* $\mathsf{O}_{\mathsf{Crig}} = \emptyset$ *and* $\mathsf{O}_{\mathsf{Rrig}} = \emptyset$.

Proof (Sketch). We can prove the theorem by a mutual reduction of an $\mathcal{ALC}_{\mathcal{ALC}}$ and an $\mathcal{ALC}\llbracket\mathcal{ALC}\rrbracket^{+}$ knowledge base. Without introducing the complete syntax of $\mathcal{ALC}_{\mathcal{ALC}}$, we show how to map an $\mathcal{ALC}_{\mathcal{ALC}}$ ontology into $\mathcal{ALC}\llbracket\mathcal{ALC}\rrbracket^{+}$.

Table 3.3 shows in the upper part the two special constructors for object concepts available in $\mathcal{ALC}_{\mathcal{ALC}}$. The middle part provides the syntax and semantics of *object formulas* which in turn constitute the *object ontology axioms*, shown in lower part. The rightmost column defines the mapping τ which translates terms from $\mathcal{ALC}_{\mathcal{ALC}}$ to $\mathcal{ALC}\llbracket\mathcal{ALC}\rrbracket^{+}$. The following example shows how an object ontology axiom is mapped.

$$\mathbf{C} : \langle\mathbf{C}\rangle_{\mathbf{r}}(\langle\mathbf{C}\rangle_{\mathbf{r}}D \sqsubseteq A) \qquad \rightsquigarrow \qquad C \sqsubseteq \exists r.\big(C \sqcap \big\llbracket \Diamond_{\mathbf{r},\mathbf{C}}D \sqsubseteq A\big\rrbracket\big)$$

An $\mathcal{ALC}_{\mathcal{ALC}}$ ontology $\mathcal{K} = (\mathcal{C}, \mathcal{O})$ consists of a context ontology $\mathcal{C}$, which is in fact a standard $\mathcal{ALC}$ ontology, and of an object ontology. Given $\mathcal{K}$, let us define the $\mathcal{ALC}\llbracket\mathcal{ALC}\rrbracket$ ontology $\mathcal{B}_{\mathcal{K}} := \mathcal{C} \wedge \tau(\mathcal{O})$. It is easy to verify that a nested interpretation $\mathcal{J}$ is a model of $\mathcal{K}$ if and only if it is a model of $\mathcal{B}_{\mathcal{K}}$.

Conversely, for an $\mathcal{ALC}\llbracket\mathcal{ALC}\rrbracket$ ontology $\mathcal{B}$, we take the outer abstraction $\mathcal{B}^{\mathsf{b}}$ as context ontology and for each $\llbracket\alpha\rrbracket$ in $\mathcal{B}$, we add $(\mathbf{A}_{\llbracket\alpha\rrbracket} : \alpha)$ and $(\neg\mathbf{A}_{\llbracket\alpha\rrbracket} : \neg\alpha)$ to the object ontology $\mathcal{O}_{\mathcal{B}}$. Again, it is easy to show that $\mathcal{J}$ models $\mathcal{B}$ iff $\mathcal{J}$ models $\mathcal{K}_{\mathcal{B}} = (\mathcal{B}^{\mathsf{b}}, \mathcal{O}_{\mathcal{B}})$. $\square$

The more interesting setting with rigid roles behaves much worse. One can easily show that the consistency problem becomes undecidable.

Theorem 3.38. *The consistency problem in* $\mathcal{EL}\llbracket\mathcal{ALC}\rrbracket^{+}$ *is undecidable if* $\mathsf{O}_{\mathsf{Rrig}} \neq \emptyset$.

Proof. Similar to the idea of [LWZ08], we proof the claim by reduction of a well-known undecidable problem, namely the *domino problem* [Ber66]: given a triple $\mathcal{D} = (D, H, V)$ with

Table 3.3: Syntax and semantics of $\mathcal{ALC}_{\mathcal{ALC}}$, and the mapping τ to $\mathcal{ALC}[\![\mathcal{ALC}]\!]^+$

Syntax	Semantics	mapping $\tau(x)$
$\langle\mathbf{C}\rangle_{\mathbf{r}}D$	$\cdot^{\mathcal{I}_c} = \{d \in \Delta \mid \text{there is } c' \text{ s.t. } (c,c') \in \mathbf{r}^{\mathcal{J}},\ c' \in C^{\mathcal{J}},\ d \in D^{\mathcal{I}_{c'}}\}$	$\Diamond_{\mathbf{r},\mathbf{C}}D$
$[\mathbf{C}]_{\mathbf{r}}D$	$\cdot^{\mathcal{I}_c} = \{d \in \Delta \mid (c,c') \in \mathbf{r}^{\mathcal{J}} \text{ and } c' \in C^{\mathcal{J}} \text{ imply } d \in D^{\mathcal{I}_{c'}}\}$	$\Box_{\mathbf{r},\mathbf{C}}D$
$B \sqsubseteq D$	$\mathcal{I}_c \models B \sqsubseteq D$ iff $B^{\mathcal{I}_c} \subseteq D^{\mathcal{I}_c}$	$[\![B \sqsubseteq D]\!]$
$D(a)$	$\mathcal{I}_c \models D(a)$ iff $a^{\mathcal{I}_c} \in D^{\mathcal{I}_c}$	$[\![D(a)]\!]$
$s(a,b)$	$\mathcal{I}_c \models s(a,b)$ iff $(a^{\mathcal{I}_c}, b^{\mathcal{I}_c}) \in s^{\mathcal{I}_c}$	$[\![s(a,b)]\!]$
$\neg\varphi$	$\mathcal{I}_c \models \neg\varphi$ iff $\mathcal{I}_c \not\models \varphi$	$\neg\tau(\varphi)$
$\varphi \wedge \psi$	$\mathcal{I}_c \models \varphi \wedge \psi$ iff $\mathcal{I}_c \models \varphi$ and $\mathcal{I}_c \models \psi$	$\tau(\varphi) \sqcap \tau(\psi)$
$\langle\mathbf{C}\rangle_{\mathbf{r}}\varphi$	$\mathcal{I}_c \models \langle\mathbf{C}\rangle_{\mathbf{r}}\varphi$ iff there is $c' \in \mathbf{C}^{\mathcal{J}}$ s.t. $(c,c') \in \mathbf{r}^{\mathcal{J}}$, $\mathcal{I}_{c'} \models \varphi$	$\exists r.(C \sqcap \tau(\varphi))$
$[\mathbf{C}]_{\mathbf{r}}\varphi$	$\mathcal{I}_c \models [\mathbf{C}]_{\mathbf{r}}\varphi$ iff every $c' \in \mathbf{C}^{\mathcal{J}}$, $(c,c') \in \mathbf{r}^{\mathcal{J}}$ implies $\mathcal{I}_{c'} \models \varphi$	$\forall r.(\neg C \sqcup \tau(\varphi))$
$\mathbf{a} : \varphi$	$\mathcal{J} \models \mathbf{a} : \varphi$ iff $\mathcal{I}_c \models \varphi$ with $c = \mathbf{a}^{\mathcal{J}}$	$(\tau(\varphi))(a)$
$\mathbf{C} : \varphi$	$\mathcal{J} \models \mathbf{C} : \varphi$ iff $\mathcal{I}_c \models \varphi$ for every c with $c \in \mathbf{C}^{\mathcal{J}}$	$C \sqsubseteq \tau(\varphi)$

a set of domino types $D = \{d_1, \ldots, d_n\}$, a horizontal compatibility relation $H \subseteq D \times D$ and a vertical compatibility relation $V \subseteq D \times D$, decide whether there exists a solution to cover the $\mathbb{N} \times \mathbb{N}$-grid with these domino types respecting the compatibility relations, i.e. does there exist a *tiling* $\mathrm{t} : \mathbb{N} \times \mathbb{N} \to D$ s.t. $(\mathrm{t}(i,j), \mathrm{t}(i+1,j)) \in H$ and $(\mathrm{t}(i,j), \mathrm{t}(i,j+1)) \in V$?

We encode this problem in $\mathcal{EL}[\![\mathcal{ALC}]\!]^+$ with a single rigid role $v \in \mathsf{O}_{\mathsf{Rrig}}$. Let $\mathcal{B}_{\mathcal{D}}$ be the conjunction of the following meta axioms. Each context represents a *column* of the grid. Using a meta role $h \in \mathsf{M}_{\mathsf{R}}$ and a rigid object role $v \in \mathsf{O}_{\mathsf{Rrig}}$, we ensure the existence of a grid:

$$\top \sqsubseteq \exists h.\top \tag{α_1}$$

$$\top \sqsubseteq [\![\top \sqsubseteq \exists v.\top]\!] \tag{α_2}$$

Let $A_0, \ldots, A_n \in \mathsf{O}_{\mathsf{C}}$ be object concept names representing the given domino types. To ensure that a single domino type is assigned to each grid position, we use

$$\top \sqsubseteq \left[\!\!\left[\top \sqsubseteq (A_1 \sqcup \cdots \sqcup A_n) \sqcap \bigsqcap_{1 \le i < j \le n} \neg(A_i \sqcap A_j) \right]\!\!\right]. \tag{α_3}$$

To enforce the compatibility relations, we use

$$\top \sqsubseteq \bigsqcap_{i=1}^{n} \left[\!\!\left[A_i \sqsubseteq \forall v.(\bigsqcup_{(d_i,d_j) \in V} A_j) \right]\!\!\right], \text{ and} \tag{α_4}$$

$$\top \sqsubseteq \bigsqcap_{i=1}^{n} \left[\!\!\left[A_i \sqsubseteq \bigsqcup_{(d_i,d_j) \in H} \Box_{\mathbf{h},\top} A_j \right]\!\!\right]. \tag{α_5}$$

<u>*Claim:*</u> *$\mathcal{B}_{\mathcal{D}}$ is consistent iff $\mathcal{D}$ has a solution.*

Proof: *Assume that* t *is a solution for* $\mathcal{D}$. *Then, based on* t *we define the nested interpretation* $\mathcal{J}_t = (\mathbb{N}, \cdot^{\mathcal{J}_t}, \mathbb{N}, (\cdot^{\mathcal{I}_x})_{x\in\mathbb{N}})$ *with*

$$h^{\mathcal{J}_t} := \{(k, k+1) \mid k \geq 0\}$$
$$v^{\mathcal{I}_x} := \{(l, l+1) \mid l \geq 0\} \qquad \text{for all } n \geq 0$$
$$A_i^{\mathcal{I}_x} := \{y \in \mathbb{N} \mid \mathrm{t}(x,y) = d_i\}$$

By definition, $\mathcal{J}_t$ *models* α_1 *and* α_2. *For each object domain element* $y \in \mathbb{N}$ *and each* $\mathcal{I}_x$, $x \geq 0$, *we have that* $y \in A_i^{\mathcal{I}_x}$ *and* $y \notin A_j^{\mathcal{I}_x}$, $1 \leq j \leq n, i \neq j$, *for* $\mathrm{t}(x,y) = d_i$. *Hence,* $\mathcal{J}_t \models \alpha_3$. *By definition,* $y+1$ *is the only* v*-successor of* y. *If* $y \in A_i^{\mathcal{I}_x}$ *we know that* $y + 1 \in \big(\bigsqcup_{(d_i,d_j)\in V} A_j\big)^{\mathcal{I}_x}$ *because* t *is a solution and* $(\mathrm{t}(x,y), \mathrm{t}(x,y+1)) \in V$. *Analogously,* $x+1$ *is the only* h*-successor of* x *and if* $y \in A_i^{\mathcal{I}_x}$, *we know that* $y \in \big(\bigsqcup_{(d_i,d_j)\in H} A_j\big)^{\mathcal{I}_{x+1}}$ *because* t *is a solution and* $(\mathrm{t}(x,y), \mathrm{t}(x+1,y)) \in H$. *Hence,* $\mathcal{J}_t \models \alpha_4 \wedge \alpha_5$. *Thus,* $\mathcal{B}_\mathcal{D}$ *is consistent.*

Assume that $\mathcal{J} = (\mathbb{C}, \cdot^{\mathcal{J}}, \Delta^{\mathcal{J}}, (\cdot^{\mathcal{I}_c})_{c\in\mathbb{C}})$ *is a model of* $\mathcal{B}_\mathcal{D}$. *Let* P_M *and* $\mathsf{P}_{\mathsf{O}^\mathcal{I}}$ *be infinite paths* $\mathsf{P}_\mathsf{M} = c_0 c_1 \ldots$ *and* $\mathsf{P}_{\mathsf{O}^\mathcal{I}} = o_0 o_1 \ldots$ *with* $c_i \in \mathbb{C}$, $(c_i, c_{i+1}) \in h^{\mathcal{J}}$, $o_i \in \Delta^{\mathcal{J}}$ *and* $(o_i, o_{i+1}) \in v^{\mathcal{I}_c}$ *for some* $c \in \mathbb{C}$. *Such paths exists, because* $\mathcal{J} \models \alpha_1$, $\mathcal{J} \models \alpha_2$ *and* v *is a rigid role. We define the nested interpretation* $\mathcal{J}_\mathsf{P} := (\{c_i \mid i \geq 0\}, \cdot^{\mathcal{J}_\mathsf{P}}, \{o_i \mid i \geq 0\}, (\cdot^{\mathcal{I}_{\mathsf{P},c_i}})_{i\geq 0})$, *where* $\cdot^{\mathcal{J}_\mathsf{P}}$ *is the restriction of* $\cdot^{\mathcal{J}}$ *to the domain* $c_i \mid i \geq 0$, *and* $\cdot^{\mathcal{I}_{\mathsf{P},c_i}}$ *is the restriction of* $\cdot^{\mathcal{I}_{c_i}}$ *to the domain* $o_i \mid i \geq 0$.

By construction, $\mathcal{J}_\mathsf{P}$ *is a model of* α_1 *and* α_2. *Since* α_3 *to* α_5 *do not contain any existential or at-least restrictions, the restriction of the meta and the object domain preserves the entailment relation, and* $\mathcal{J}_\mathsf{P} \models \mathcal{B}_\mathcal{D}$. *We define the tiling* t *as follows:*

$$t(x,y) = d_i \quad \text{if } o_y \in A_i^{\mathcal{I}_{\mathsf{P},c_x}}.$$

The tiling t *is a total function and well-defined, due to* α_3. *Let* $o_j \in A_{k_1}^{\mathcal{I}_{\mathsf{P},c_i}}$, $o_{j+1} \in A_{k_2}^{\mathcal{I}_{\mathsf{P},c_i}}$ *and* $o_j \in A_{k_3}^{\mathcal{I}_{\mathsf{P},c_{i+1}}}$. *Thus, we have* $\mathrm{t}(i,j) = d_{k_1}$, $\mathrm{t}(i,j+1) = d_{k_2}$ *and* $\mathrm{t}(i+1,j) = d_{k_3}$. *By* α_4 *we know that* $o_j \in \forall v.\big(\bigsqcup_{(d_{k_1},d_j)\in V} A_j\big)^{\mathcal{I}_{\mathsf{P},c_i}}$ *and, hence,* $(d_{k_1}, d_{k_2}) \in V$. *Analogously by* α_5, *we get* $(d_{k_1}, d_{k_3}) \in H$. *Thus,* $\mathcal{D}$ *has a solution.* ⁄

Thus, deciding whether $\mathcal{B}_\mathcal{D}$ is consistent is undecidable. □

In Section 3.3, we discussed three different settings depending on whether rigid concept and role names are admitted. We already obtained the complexity results for $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]^+$ for Setting (i), i.e. no rigid names are allowed, and for Setting (iii), i.e. rigid roles are allowed. With some assumptions, in $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]^+$, however, Setting (ii) that allows rigid concept names but no rigid roles coincides with Setting (i), since rigid concepts can be simulated. Hence, we obtain the following result:

Theorem 3.39. *The consistency problem for* $\mathcal{SHOI}[\![\mathcal{SHOI}]\!]^+$*-ontologies is* 2EXPTIME*-complete if* $\mathsf{O}_{\mathsf{Rrig}} = \emptyset$.

Proof. First, we have to simulate the universal role u. With u, we can simulate rigid concepts and, thus, reduce the consistency problem to the case without rigid names.

Let $\mathfrak{O} = (\mathcal{O}, \mathcal{R}_\mathsf{O}, \mathcal{R}_\mathsf{M})$ be an $\mathcal{SHOI}[\![\mathcal{SHOI}]\!]^+$-ontology. We first have to simulate a universal role u on the meta level. We can assume w.l.o.g. that $u \in \mathsf{M_R}$ does not occur in $\mathfrak{O}$. We obtain $\mathfrak{O}_u$ from $\mathfrak{O}$ by adding the following axioms to $\mathfrak{O}$:

- $u^- \sqsubseteq u$,

- $r \sqsubseteq u$, for all $r \in \mathsf{M_R}$ occurring in $\mathfrak{O}$,
- $u(a, b)$ for all $a, b \in \mathsf{M_I}$ occurring in $\mathfrak{O}$.

Note that $\mathfrak{O}_u$ is of size polynomial in the size of $\mathfrak{O}$. Assume there exists an unnamed context in a model of $\mathfrak{O}$, i.e. $c \in \mathbb{C}$ such that there is no $a \in \mathsf{M_I}$ occurring in $\mathfrak{O}$ with $a^{\mathcal{J}} = c$, and c is not connected to any named context by some path. Then, we can always remove that unnamed context from the interpretation and still have a model. Hence, we can assume that every model of $\mathfrak{O}_u$ is connected and that every context can be reached by any other context via a path of u-edges. Note that we restrict $\mathfrak{O}$ to $\mathcal{SHOI}[\![\mathcal{SHOI}]\!]^+$-ontologies, since in an $\mathcal{SHOI}[\![\mathcal{SHOI}]\!]^+$-BKB, a negated meta GCI can enforce the existence of an unnamed context that is not necessarily connected to the rest of the model.

With the help of u and contextualised concepts, we can express that $A \in \mathsf{O_C}$ is rigid by the following two axioms:

$$\top \sqsubseteq \left[\!\left[A \sqsubseteq \Box_{\mathbf{u},\top} A \right]\!\right]$$
$$\top \sqsubseteq \left[\!\left[\neg A \sqsubseteq \Box_{\mathbf{u},\top} \neg A \right]\!\right]$$

Due to these axioms, the extension of A in a context c are exactly these elements which are in A in every context c' that is related to c via u. Thus, the extension of A is equal in every context, i.e. A is rigid.

In [KG16], Theorem 6 states that deciding the consistency problem in $\mathcal{SHOI}_{\mathcal{SHOI}}$ without rigid names is 2ExpTime-complete. We can again use the same translation as shown in the proof of Theorem 3.37 and, hence, obtain 2ExpTime-completeness for $\mathcal{SHOI}[\![\mathcal{SHOI}]\!]^+$-ontologies with rigid concepts. □

To sum up, we showed that adding contextualised concepts dramatically increases the complexity. In the presence of rigid roles, the consistency problem for even less expressive $\mathcal{EL}[\![\mathcal{ALC}]\!]^+$-ontologies becomes undecidable. Without rigid roles for any contextualised DL between $\mathcal{ALC}[\![\mathcal{ALC}]\!]^+$ and $\mathcal{SHOI}[\![\mathcal{SHOI}]\!]^+$ deciding the consistency of an ontology is 2ExpTime-complete.

Chapter 4

A Mapping from Role-Based Models to Description Logic Ontologies

In the last chapter we introduced a family of description logics which is capable of expressing contextualised knowledge. This provides the needed expressiveness to be able to reason on role-based models. But due to the rather elaborate semantics both of role-based models and of contextualised description logics it will be tedious and error-prone to manually construct the DL ontology which exactly captures a role-based model, especially since the domain analyst who generates the role-based model is in general not an expert in DLs. Hence, an automated mapping from role-based models into DL would be desirable. Therefore, a formal representation of role-based systems with a well-defined semantics is necessary which is the case since we use role-based systems formalised in CROM.

In this chapter we will present a mapping algorithm from such role-based models into contextualised DLs in order to automate this step. In Section 4.1, we present the mapping algorithm along with the proof that the mapping preserves the semantics. Then, in Section 4.2 we discuss possible features of role-based systems that go beyond CROM but still can be expressed in an ontology.

4.1 Representing Role-Based Models

We would like to emphasise here that checking well-formedness of a Σ-CROM $\mathcal{M}$ and compliance of a constraint set with $\mathcal{M}$ is not considered here, because these are purely syntactical checks and no reasoning is necessary. Furthermore, checking whether a given Σ-CROI satisfies $\mathcal{M}$ is also not the considered task as it rather belongs to the area of model checking and is not in the scope of this thesis. More interesting is whether there exists such a Σ-CROI at all. Then, we can additionally test if certain axioms are entailed or whether specific role types can be played since these questions can be reduced to the satisfiability problem. Thus, the main objective is, given a Σ-CCROM $\mathcal{K}$, to construct an $\mathcal{ALC}[\![\mathcal{SHOIQ}]\!]$-ontology $\mathfrak{B}_{\mathcal{K}}$ such that $\mathfrak{B}_{\mathcal{K}}$ is consistent iff $\mathcal{K}$ is satisfiable. Which contextualised description logic is exactly necessary depends also on the constraints occurring in $\mathcal{K}$ and will be discussed in Section 4.1.5 in more detail.

The general idea is to model compartment types as concepts on the meta level, and objects playing a rôle, the relationship types as well as all the constraints within a compartment type on the object level. For this, we introduce o-concepts for natural types and rôle types and a special object role plays. The fills relation is transformed into corresponding domain and range axioms for plays.

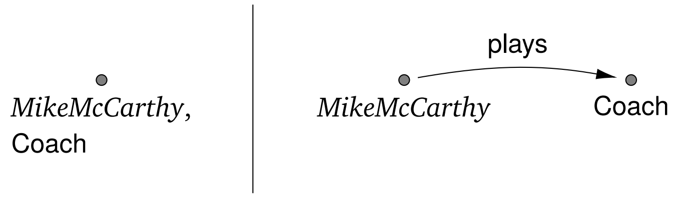


Figure 4.1: Possibilities to formalise objects playing rôles.

Here, we made a first design decision on how to express that a rôle is played within a compartment. There are two possibilities, depicted in Figure 4.1, which we already showed in Example 3.4. On the one hand, we can introduce an o-concept Coach $\in \mathsf{N}_{\mathsf{RT}}$ and elements playing a Coach-rôle are in the extension of *RT* (such as Mike McCarthy playing the rôle of a coach in the left side of Figure 4.1). Here, an object o and a rôle r with $(o, \cdot, r) \in \mathsf{plays}$ would be mapped to a single element d in the object interpretation for that compartment. This variant is close to the ontological nature of rôles where an entity is in the extension of a rôle, seen as unary predicate, if this entity plays that rôle. On the other hand, we can introduce the o-concept Coach as well but disjoint from Person and we additionally introduce an object role plays and elements playing an Coach-rôle have a plays-successor that is in the extension of Coach (such as Mike playing the rôle of a coach in the left side of Figure 4.1). This is closer to the semantics of Σ-CROIs, but introduces new object domain elements for each rôle that is played. Still, we chose the latter variant, since later we need to count the number of rôles to assure occurrence and cardinality constraints. In DLs, this can be done via qualified number restrictions.

Generally, in a Σ-CROM, compartments are also allowed to play rôles in other compartments. Within these other compartments a compartment playing a rôle does not behave differently than a natural. Hence, we have compartment types both as meta concept names, i.e. as contexts in which other objects play rôles, and as object concept types, i.e. as objects that play rôles in a context. In a Σ-CROM there is a one-to-one relationship, since the compartment as rôle-player and as context is the same object. In our formalism, we cannot establish that tight connection, but we can assure the existence of contexts of a certain compartment type via a meta role nested if objects of that type play rôles. Since we cannot restrict the number of existing contexts in a nested DL interpretation, this is sufficient for the satisfiability problem. To distinguish between compartment types as contexts and as objects, we call the former simply *contexts* and the latter *o-compartments* and introduce a copy $\mathsf{N}_{\mathsf{CT}'}$ of all compartment types N_{CT}.

Relationship types are intuitively modelled as object roles. Here, it might be more natural to span these between the played rôles instead of the players. But due to the one-to-one correspondence between players and played rôles, we can also construct the relationship types between the players. In doing so, we can avoid the use of role value maps, which would cause the consistency problem to become undecidable [Sch89], to formalise intra relationship type constraints. Even so, we can only support such constraints that are expressible in the underlying DL.

Rôle groups are handled like rôles with an additional axiom stating that "playing" a rôle group is equivalent to fulfilling the constraints specified in that rôle group. Furthermore, if an object plays an atom of a top-level rôle group, that object must fulfill the rôle group. For the occurrence constraints we introduce a fresh individual name $\mathsf{counter}$ and a object role counts and enforce that each played role or fulfilled role group is connected to this counter. Thereby we can use qualified number restrictions to encode the occurrence constraints. For cardinality constraints, we also utilise number restrictions.

To sum up, we consider the object signature $\mathsf{O} = (\mathsf{O_C}, \mathsf{O_R}, \mathsf{O_I})$ and the meta signature $\mathsf{M} = (\mathsf{M_C}, \mathsf{M_R}, \mathsf{M_I})$ such that

- $\mathsf{N_{CT}} \subseteq \mathsf{M_C}$, since every compartment type is a meta concept,
- $\mathsf{nested} \in \mathsf{M_R}$, to assure the existence of compartments that play rôles,
- $\mathsf{N_{NT}} \cup \mathsf{N_{CT'}} \subseteq \mathsf{O_{Crig}}$, since every natural type and every o-compartment type are rigid object concepts,
- $\mathsf{N_{RT}} \subseteq \mathsf{O_{Cflex}}$, since every rôle type is a flexible object concept
- $\mathsf{plays} \in \mathsf{O_{Rflex}}$, to express the plays-relation,
- $\mathsf{N_{RST}} \subseteq \mathsf{O_{Rflex}}$, since every relationship type is an object role,
- $\mathsf{counter} \in \mathsf{O_I}$ and $\mathsf{counts} \in \mathsf{O_{Rflex}}$, to express the occurrence constraints, and
- $\mathsf{N_{M\text{-}IND}} \in \mathsf{M_I}$ and $\mathsf{N_{O\text{-}IND}} \in \mathsf{O_I}$, to interpret individual names on their respective level.

Additionally, we introduce the following object concept names:

- $A_{\mathsf{O}} \in \mathsf{O_{Crig}}$ for all objects eligible of playing rôles, i.e. naturals and o-compartments,
- $A_{\mathsf{RT}} \in \mathsf{O_{Cflex}}$ for all rôles, and
- $A_{\mathsf{RG}} \in \mathsf{O_{Cflex}}$ for all instances of rôle groups since we will consider rôle groups of the constraint set similar to rôles.

4.1.1 A Mapping for the Vocabulary Σ

In the next sections, we present the mapping from rôle-based systems into contextualised description logics in detail. At first, we express general knowledge about occurring types which is independent of the specific Σ-CROM.

1. Every context belongs to exactly one compartment type.

$$\top \sqsubseteq \bigsqcup_{CT \in \mathsf{N_{CT}}} CT \tag{4.1}$$

$$CT_1 \sqcap CT_2 \sqsubseteq \bot \qquad \text{for all } CT_1, CT_2 \in \mathsf{N_{CT}},\ CT_1 \neq CT_2 \tag{4.2}$$

2. In every context, every natural or o-compartment and every rôle belongs to exactly one type.

$$\top \sqsubseteq \left[\!\!\left[A_{\mathsf{O}} \equiv \bigsqcup_{NT \in \mathsf{N_{NT}}} NT \sqcup \bigsqcup_{CT' \in \mathsf{N_{CT'}}} CT' \right]\!\!\right] \tag{4.3}$$

$$\top \sqsubseteq \left[\!\!\left[A_{\mathsf{RT}} \equiv \bigsqcup_{RT \in \mathsf{N_{RT}}} RT \right]\!\!\right] \tag{4.4}$$

$$\top \sqsubseteq \bigsqcap_{\substack{T_1, T_2 \in \mathsf{N}_{\mathsf{NT}} \cup \mathsf{N}_{\mathsf{CT}'} \cup \mathsf{N}_{\mathsf{RT}},\\ T_1 \neq T_2}} [\![T_1 \sqcap T_2 \sqsubseteq \bot]\!] \tag{4.5}$$

3. On the object level, an element can either be a rôle, a natural or o-compartment, an instance of a rôle group or the individual counter.

$$\begin{aligned} \top \sqsubseteq [\![A_{\mathsf{O}} \sqcap A_{\mathsf{RT}} \sqsubseteq \bot]\!] \sqcap [\![A_{\mathsf{O}} \sqcap A_{\mathsf{RG}} \sqsubseteq \bot]\!] \sqcap [\![A_{\mathsf{RT}} \sqcap A_{\mathsf{RG}} \sqsubseteq \bot]\!] \\ \sqcap [\![\neg(A_{\mathsf{O}} \sqcup A_{\mathsf{RT}} \sqcup A_{\mathsf{RG}})(\mathsf{counter})]\!] \end{aligned} \tag{4.6}$$

4. Every natural or o-compartment can play at most one RT-rôle in each context and each rôle must be played by some object.

$$\top \sqsubseteq \bigsqcap_{RT \in \mathsf{N}_{\mathsf{RT}}} [\![A_{\mathsf{O}} \sqsubseteq \leqslant_1 \mathsf{plays}.RT]\!] \tag{4.7}$$

$$\top \sqsubseteq \left[\!\!\left[A_{\mathsf{RT}} \sqsubseteq \geqslant_1 \mathsf{plays}^-.\top \sqcap \leqslant_1 \mathsf{plays}^-.\top \right]\!\!\right] \tag{4.8}$$

5. We formalise a general domain and range restriction for plays. Only naturals or o-compartments can play something, and only rôles or instances of rôle groups can be played.

$$\top \sqsubseteq [\![\exists \mathsf{plays}.\top \sqsubseteq A_{\mathsf{O}}]\!] \tag{4.9}$$

$$\top \sqsubseteq [\![\top \sqsubseteq \forall \mathsf{plays}.(A_{\mathsf{RT}} \sqcup A_{\mathsf{RG}})]\!] \tag{4.10}$$

6. Finally, if an o-compartment plays a rôle in some context, the o-compartment must also exist as context.

$$\neg\left[\!\!\left[CT' \sqcap \exists \mathsf{plays}.\top \sqsubseteq \bot \right]\!\!\right] \sqsubseteq \exists \mathsf{nested}.CT \qquad \text{for all } CT' \in \mathsf{N}_{\mathsf{CT}'} \tag{4.11}$$

4.1.2 A Mapping for the Σ-CROM $\mathcal{M}$

With the general knowledge about the vocabulary Σ being set up, we can look into the translation of the specifications for a given Σ-CROM $\mathcal{M} = (\mathsf{fills}, \mathsf{parts}, \mathsf{rel})$.

1. The fills-relation specifies which natural or compartment types are allowed to play which rôle types. Hence, elements that play a certain rôle type can only be naturals or o-compartments of types which fill that rôle type.

$$\top \sqsubseteq \bigsqcap_{RT \in \mathsf{N}_{\mathsf{RT}}} \left[\!\!\left[\exists \mathsf{plays}.RT \sqsubseteq \left(\bigsqcup\nolimits_{(T,RT) \in \mathsf{fills}} T \right) \right]\!\!\right] \tag{4.12}$$

Note here, that Equation (4.12) is sufficient in the sense that in conjunction with Equations (4.4) and (4.10), it entails that all plays-successors of naturals or o-compartments of a specific type are either instances of a rôle type that are filled by that type or instances of a rôle group:

$$\top \sqsubseteq \bigsqcap_{T \in \mathsf{N_{NT}} \cup \mathsf{N_{CT'}}} \left[\!\!\left[T \sqsubseteq \forall \mathsf{plays}.\left(A_{\mathsf{RG}} \sqcup \bigsqcup\nolimits_{(T,RT) \in \mathsf{fills}} RT \right) \right]\!\!\right].$$

2. Since in a satisfying Σ-CROI the plays-relation respects parts, only RT-rôles with $RT \in \mathsf{parts}(CT)$ exist in a CT-context.

$$CT \sqsubseteq \left[\!\!\left[A_{\mathsf{RT}} \sqsubseteq \bigsqcup_{RT \in \mathsf{parts}(CT)} RT \right]\!\!\right] \qquad \text{for all } CT \in \mathsf{N_{CT}} \tag{4.13}$$

3. Analogous to fills restricting the domain and range of plays, the rel-function restricts them for each relationship type.

$$\begin{aligned} \top \sqsubseteq [\![\exists RST.\top \sqsubseteq \exists \mathsf{plays}.RT_1]\!] \sqcap [\![\top \sqsubseteq \forall RST.(\exists \mathsf{plays}.RT_2)]\!] \\ \text{for all } RST \in \mathsf{N_{RST}} \text{ and } \mathsf{rel}(RST) = (RT_1, RT_2) \end{aligned} \tag{4.14}$$

Note here, that due to equations (4.1), (4.4), (4.5) and (4.13) and the fact that parts' codomain is a partition of $\mathsf{N_{RT}}$, in any context that is not in CT there are no rôles of a type the participates in CT, e.g. the following axiom is entailed for all $CT \in \mathsf{N_{CT}}$:

$$\neg CT \sqsubseteq \left[\!\!\left[\bigsqcup_{RT \in \mathsf{parts}(CT)} RT \sqsubseteq \bot \right]\!\!\right]$$

4.1.3 A Mapping for the Σ-CROA $\mathcal{A}$

Next, let $\mathcal{A}$ be a Σ-CROA. We can translate the different Compartment Role Object Assertions into meta assertions in $\mathcal{L}_{\mathsf{M}}[\![\mathcal{L}_{\mathsf{O}}]\!]$.

1. For any meta type assertion of the form $CT(c) \in \mathcal{A}$ with $CT \in \mathsf{N_{CT}}$ and $c \in \mathsf{N_{M\text{-}IND}}$ the individual a must be a context of type CT.

$$CT(c) \tag{4.15}$$

2. For any object type assertion $T(a,c) \in \mathcal{A}$ with $T \in \mathsf{N_{NT}} \cup \mathsf{N_{CT}} \cup \mathsf{N_{RT}}$, $a \in \mathsf{N_{O\text{-}IND}}$ and $c \in \mathsf{N_{M\text{-}IND}}$ the individual a is a natural, an o-compartment or rôle that belongs to the concept T in context c.

$$[\![T(a)]\!](c) \tag{4.16}$$

3. A plays assertion $\mathsf{play\_assert}(a_1, c, a_2) \in \mathcal{A}$ states that a_1 plays a_2 in c.

$$[\![\mathsf{plays}(a_1, a_2)]\!](c) \tag{4.17}$$

4. A links assertion $\mathsf{link\_assert}(RST, c, a_1, a_2) \in \mathcal{A}$ states that a_1 and a_2 play rôles which are related in c via RST. Due to the way we modelled relationship types, this is simply the following axiom:

$$[\![RST(a_1, a_2)]\!](c) \tag{4.18}$$

4.1.4 A Mapping for the Σ-CROC $\mathcal{C}$

Let $\mathcal{C}$ be a Σ-CROC, let $\mathbb{RG}(\mathcal{C})$ denote the set of all complex rôle groups occurring in $\mathcal{C}$ and let $\mathbb{RG}^{\top}(\mathcal{C}) \subseteq \mathbb{RG}(\mathcal{C})$ denote the set of all complex top-level rôle groups.

1. Analogous to rôles, rôle groups are disjoint, every instance of a rôle group must be played by some object and every object can either fulfill or not fulfill a rôle group.

$$\top \sqsubseteq \left[\!\!\left[A_{\mathsf{RG}} \equiv \bigsqcup_{RG \in \mathbb{RG}(\mathcal{C})} RG \right]\!\!\right] \sqcap \bigsqcap_{\substack{RG_1, RG_2 \in \mathbb{RG}(\mathcal{C}),\\ RG_1 \neq RG_2}} [\![RG_1 \sqcap RG_2 \sqsubseteq \bot]\!] \tag{4.19}$$

$$\top \sqsubseteq \left[\!\!\left[A_{\mathsf{RG}} \sqsubseteq \geqslant_1 \mathsf{plays}^{-}.\top \sqcap \leqslant_1 \mathsf{plays}^{-}.\top \right]\!\!\right] \tag{4.20}$$

$$\top \sqsubseteq \bigsqcap_{RG \in \mathbb{RG}(\mathcal{C})} [\![A_{\mathsf{O}} \sqsubseteq \leqslant_1 \mathsf{plays}.RG]\!] \tag{4.21}$$

2. Complex rôle groups are treated like abstract rôles. An object can "play" an instance of a rôle group. This is equivalent to fulfilling that role group. So, the object must "play" the required number of containing rôle groups. Furthermore, if an object plays a rôle whose type is an atom of a top-level rôle group, the object must also fulfill that rôle group.

$$\top \sqsubseteq \bigsqcap_{\substack{RG \in \mathbb{RG}(\mathcal{C}),\\ RG = (\{A_1, \ldots, A_n\}, k, \ell)}} \begin{aligned}[t] [\![\exists \mathsf{plays}.RG \equiv (\geqslant_k \mathsf{plays}.(A_1 \sqcup \cdots \sqcup A_n))\\ \sqcap (\leqslant_\ell \mathsf{plays}.(A_1 \sqcup \cdots \sqcup A_n))]\!]\end{aligned} \tag{4.22}$$

$$\top \sqsubseteq \bigsqcap_{RG \in \mathbb{RG}^{\top}(\mathcal{C})} \left[\!\!\left[\exists \mathsf{plays}.\left(\bigsqcup\nolimits_{RT \in \mathsf{atom}(RG)} RT \right) \sqsubseteq \exists \mathsf{plays}.RG \right]\!\!\right] \tag{4.23}$$

3. Since for the occurrence constraints we only consider objects that play any rôle in this compartment, we assure that fulfilling a rôle group implies playing some rôle.

$$\top \sqsubseteq [\![\exists \mathsf{plays}.A_{\mathsf{RG}} \sqsubseteq \exists \mathsf{plays}.A_{\mathsf{RT}}]\!] \tag{4.24}$$

4. To capture the occurrence constraints we use an object individual name counter and introduce an counts-role from that individual to all rôles and rôle group instances. Thus, the occurrence constraints can be enforced with the help of qualified number restrictions.

$$\top \sqsubseteq \left[\!\!\left[A_{\mathsf{RT}} \sqcup A_{\mathsf{RG}} \sqsubseteq \geqslant_1 \mathsf{counts}^{-}.\{\mathsf{counter}\} \sqcap \leqslant_1 \mathsf{counts}^{-}.\{\mathsf{counter}\} \right]\!\!\right] \tag{4.25}$$

$$\begin{gathered} CT \sqsubseteq [\![(\geqslant_k \mathsf{counts}.RG)(\mathsf{counter})]\!] \sqcap [\![(\leqslant_\ell \mathsf{counts}.RG)(\mathsf{counter})]\!] \\ \text{for all } (k..\ell, RG) \in \mathsf{occur}(CT), \text{ for all } CT \in \mathsf{N_{CT}} \end{gathered} \tag{4.26}$$

5. Cardinality constraints restrict the number of rôles that are related to rôle via a relationship type. Rôles whose type is the domain or range of a relationship type *RST* for which there exists a cardinality constraint must have the correct amount of *RST* successors or predecessors, respectively.

$$\top \sqsubseteq \bigsqcap_{\substack{RST \in \mathsf{N}_{\mathsf{RST}},\\ \mathsf{rel}(RST)=(RT_1,RT_2),\\ \mathsf{card}(RST)=(i..j,k..\ell)}} \llbracket \exists \mathsf{plays}.RT_1 \sqsubseteq \geqslant_k RST.\top \sqcap \leqslant_\ell RST.\top \rrbracket \sqcap \left\llbracket \exists \mathsf{plays}.RT_2 \sqsubseteq \geqslant_i RST^-.\top \sqcap \leqslant_j RST^-.\top \right\rrbracket \tag{4.27}$$

6. As mentioned above, because we spanned the relationship types between the players and not between the rôles, we can model at least some of the intra-relationship type constraints, exactly those that are expressible in the object logic. In $\mathcal{SHOIQ}$, these would be transitivity and symmetry. Thus, we add the following role axioms to the object RBox $\mathcal{R}_\mathsf{O}$:

$$\mathsf{trans}(RST) \qquad \text{for all } \mathsf{trans} \in \mathsf{intra}(RST) \tag{4.28}$$

$$RST \sqsubseteq RST^- \qquad \text{for all } \mathsf{symm} \in \mathsf{intra}(RST) \tag{4.29}$$

where trans and symm are functions that return true if and only if the binary relation, respectively, is transitive or symmetric. But it is important to note, due to the restriction that only simple roles are allowed to appear in number restrictions, we cannot impose any cardinality constraints on the relationship type *RST* if *RST* is supposed to be transitive.

4.1.5 Semantic Integrity of Mapping Algorithm

Before we establish the desired relation between the CROM and the ontology, we do a short analysis of the required expressiveness of the meta and the object logic. To state the disjoint union axioms we need at least $\mathcal{ALC}$. As we do not need anything more on the meta level, we can fix $\mathcal{L}_\mathsf{M}$ to be $\mathcal{ALC}$. On the object level, we need qualified number restrictions and inverse roles to assure that every rôle is played exactly once. If we have a single occurrence constraint, we have to add nominals. Transitivity axioms or role hierarchies are only required if some intra-relationship type constraint specifies a transitive or symmetric relationship type, respectively. Table 4.1 shows a summary of the required DLs.

Anyhow, from a practical point of view not only the required DLs are relevant but also the complexity of the reasoning. Besides the specific DL it is also important which names in the ontology need to be rigid. We need rigid concept names for mapping natural types and o-compartment types. Whether we also need rigid role names depends on the constraints we want to model. Rigid roles are necessary to model attributes of natural types, e.g. the name or date of birth of a person. Complexity-wise it is important to only map details of the role-based model which might have logical implications. Since in the current version of CROM we do not have any constraints based on attributes of naturals, we will not map attributes and, hence, do not need rigid roles. In Section 4.2, we analyse some constraints beyond CROM which we can model in $\mathcal{L}_\mathsf{M}\llbracket\mathcal{L}_\mathsf{O}\rrbracket$ and therefore we also need rigid roles. Table 4.2 shows an overview of the resulting complexities. Note here, that these are all worst-case complexities

Table 4.1: Summary of required DLs for $\mathcal{L}_\mathsf{M}$ and $\mathcal{L}_\mathsf{O}$.

$\mathcal{L}_\mathsf{O}$	minimal	$\mathcal{ALCIQ}$
	with occurrence constraints	$\mathcal{ALCOIQ}$
	with intra-relationship type constraints	$\mathcal{SHOIQ}$
$\mathcal{L}_\mathsf{M}$		$\mathcal{ALC}$

Table 4.2: Overview of complexities for reasoning on CROM models with (I): No attribute-based constraints (only rigid concepts), and (II): With attribute-based constraints (with rigid roles).

	minimal ($\mathcal{ALC}[\![\mathcal{ALCIQ}]\!]$)	with occ. constr. ($\mathcal{ALC}[\![\mathcal{ALCOIQ}]\!]$)	with in.-rel. constr. ($\mathcal{ALC}[\![\mathcal{SHOIQ}]\!]$)
(I)	NExpTime	NExpTime-hard and in N2ExpTime	
(II)	2ExpTime	2ExpTime-hard and in N2ExpTime	

and that the actual reasoning effort might vary depending on the used constraints even if the different cases, i.e. with and without intra-relationship type constraints, are in the same complexity class.

Given a Σ-CCROM $\mathcal{K}$, we obtain the $\mathcal{ALC}[\![\mathcal{SHOIQ}]\!]$ ontology $\mathfrak{B}_\mathcal{K} = (\mathcal{B}_\mathcal{K}, \emptyset, \mathcal{R}_\mathsf{O})$ where $\mathcal{B}_\mathcal{K}$ is the conjunction of all meta axioms from (4.1) to (4.27) and $\mathcal{R}_\mathsf{O}$ is the set of all role axioms (4.28) and (4.29).

Theorem 4.1. *Let $\mathcal{K} = (\mathcal{M}, \mathcal{A}, \mathcal{C})$ be a Σ-CCROM. Then, $\mathcal{K}$ is satisfiable iff $\mathfrak{B}_\mathcal{K}$ is consistent.*

Proof. Assume that $\mathcal{K}$ is satisfiable and let $\mathcal{I}_\mathcal{K}$ denote a Σ-CROI that satisfies $\mathcal{M}$, $\mathcal{A}$ and $\mathcal{C}$. We define the set $\Delta_{\mathsf{N}_\mathsf{RT},c}$ of all rôles played in c, the set $\Delta_{\mathbb{RG},c}$ of all rôle group instances fulfilled in c, and the nested interpretation $\mathcal{J} = (\mathbb{C}, \cdot^\mathcal{J}, \Delta^\mathcal{J}, (\cdot^{\mathcal{I}_c})_{c\in\mathbb{C}})$ as follows:

$$
\begin{aligned}
\Delta_{\mathsf{N}_\mathsf{RT},c} &:= \{r \in \mathsf{R}^{\mathcal{I}_\mathcal{K}} \mid (\cdot, c, r) \in \mathsf{plays}\} \qquad \text{for all } c \in \mathsf{C}^{\mathcal{I}_\mathcal{K}} \\
\Delta_{\mathbb{RG},c} &:= \{(x, c, y) \in \mathbb{RG} \times \mathsf{C}^{\mathcal{I}_\mathcal{K}} \times \mathsf{O}^{\mathcal{I}_\mathcal{K}} \mid x^{\mathcal{I}_\mathcal{K}, c, y} = 1\} \qquad \text{for all } c \in \mathsf{C}^{\mathcal{I}_\mathcal{K}} \\
\mathbb{C} &:= \mathsf{C}^{\mathcal{I}_\mathcal{K}} \\
CT^\mathcal{J} &:= CT^{\mathcal{I}_\mathcal{K}} \qquad \text{for all } CT \in \mathsf{N}_\mathsf{CT} \\
\mathsf{nested}^\mathcal{J} &:= \{(c, o) \mid o, c \in \mathsf{C}^{\mathcal{I}_\mathcal{K}}, (o, c, \cdot) \in \mathsf{plays}\} \\
\Delta^\mathcal{J} &:= \mathsf{N}^{\mathcal{I}_\mathcal{K}} \cup \mathsf{C}^{\mathcal{I}_\mathcal{K}} \cup \mathsf{R}^{\mathcal{I}_\mathcal{K}} \cup \bigcup_{c \in \mathsf{C}^{\mathcal{I}_\mathcal{K}}} \Delta_{\mathbb{RG},c} \cup \{d_\mathsf{counter}\} \\
NT^{\mathcal{I}_c} &:= NT^{\mathcal{I}_\mathcal{K}} \qquad \text{for all } NT \in \mathsf{N}_\mathsf{NT} \\
CT'^{\mathcal{I}_c} &:= CT^{\mathcal{I}_\mathcal{K}} \qquad \text{for all } CT \in \mathsf{N}_\mathsf{CT} \\
RT^{\mathcal{I}_c} &:= RT^{\mathcal{I}_\mathcal{K}} \cap \Delta_{\mathsf{N}_\mathsf{RT},c} \qquad \text{for all } RT \in \mathsf{N}_\mathsf{RT} \\
RG^{\mathcal{I}_c} &:= \{(RG, c, y) \mid RG^{\mathcal{I}_\mathcal{K}, c, y} = 1\} \qquad \text{for all } RG \in \mathbb{RG}(\mathcal{C}) \\
{A_\mathsf{RT}}^{\mathcal{I}_c} &:= \Delta_{\mathsf{N}_\mathsf{RT},c}
\end{aligned}
$$

$$\begin{aligned}
A_{\mathsf{O}}{}^{\mathcal{I}_c} &:= \mathsf{O}^{\mathcal{I}_\mathcal{K}}\\
A_{\mathsf{RG}}{}^{\mathcal{I}_c} &:= \Delta_{\mathbb{RG},c}\\
RST^{\mathcal{I}_c} &:= \{(o_1,o_2) \mid \text{ there are } r_1,\, r_2 \text{ s.t. } (o_1,c,r_1),(o_2,c,r_2) \in \mathsf{plays} \text{ and}\\
&\qquad\qquad\qquad (r_1,r_2) \in \mathsf{links}(RST,c)\}\\
\mathsf{plays}^{\mathcal{I}_c} &:= \{(o,r) \mid (o,c,r) \in \mathsf{plays}\}\\
&\qquad\qquad\qquad \cup \{(o,rg) \mid rg = (x,c,o) \in \Delta_{\mathbb{RG},c} \text{ with } x^{\mathcal{I}_\mathcal{K},c,o} = 1\}\\
\mathsf{counts}^{\mathcal{I}_c} &:= \{(d_{\mathsf{counter}},y) \mid y \in \Delta_{\mathsf{N_{RT}},c} \cup \Delta_{\mathbb{RG},c}\}\\
\mathsf{counter}^{\mathcal{I}_c} &:= d_{\mathsf{counter}}\\
c^{\mathcal{J}} &:= c^{\mathcal{I}_\mathcal{K}} \qquad \text{for } c \in \mathsf{N_{M\text{-}IND}}\\
a^{\mathcal{I}_c} &:= a^{\mathcal{I}_\mathcal{K}} \qquad \text{for } a \in \mathsf{N_{O\text{-}IND}}
\end{aligned}$$

It is straight forward to show that $\mathcal{J}$ is a model of $\mathfrak{B}_\mathcal{K}$. Note, that $\mathcal{J}$ respects the rigid names since all NT, CT', A_{O} and individual names are interpreted the same in every world c. Axioms (4.1), (4.2), (4.3), (4.5), (4.6), (4.9), (4.10), (4.15) to (4.21), (4.25) are modelled by construction of $\mathcal{J}$. Since $\Delta_{\mathsf{N_{RT}},c} \subseteq \bigcup_{RT\in\mathsf{N_{RT}}} RT^{\mathcal{I}_\mathcal{K}} = \mathsf{R}^{\mathcal{I}_\mathcal{K}}$, $\mathcal{J}$ models (4.4). Axioms (4.7) and (4.8) are modelled due to 3. and 4. of Definition 2.9. Assume that $c_1 \in (\neg[\![CT' \sqcap \exists\mathsf{plays}.\top \sqsubseteq \bot]\!])^{\mathcal{J}}$. Hence, $\mathcal{I}_{c_1} \not\models CT' \sqcap \exists\mathsf{plays}.\top \sqsubseteq \bot$. Thus, there exists $c_2 \in (CT' \sqcap \exists\mathsf{plays}.\top)^{\mathcal{I}_{c_1}}$. Therefore, $(c_2,c_1,\cdot) \in \mathsf{plays}$, $(c_1,c_2) \in \mathsf{nested}^{\mathcal{J}}$ and $c_2 \in CT^{\mathcal{J}}$. Overall, $\mathcal{J}$ models Axiom (4.11). Since plays respects fills and parts, and links respects rel, (4.12), (4.13) and (4.14) are satisfied. Axioms (4.22) to (4.24) are satisfied due to the semantics of rôle groups and the fact that all top-level rôle groups are satisfied. $\mathcal{I}_\mathcal{K}$ respecting occurrence and cardinality constraints ensures that $\mathcal{J}$ models (4.26) and (4.27). Finally, the respective intra-relationship constraints imply the satisfaction of $\mathcal{R}_{\mathsf{O}}$.

Conversely, let $\mathcal{J} = (\mathbb{C}, \cdot^{\mathcal{J}}, \Delta^{\mathcal{J}}, (\cdot^{\mathcal{I}_c})_{c\in\mathbb{C}})$ denote a model of $\mathcal{B}_\mathcal{K}$. W.l.o.g. we can assume that all o-compartments that exist also play some rôles. Otherwise we could simply delete them, and still have a model. Let $\Delta_{\mathsf{N_{NT}}} \subseteq \Delta^{\mathcal{J}}$ and $\Delta_{\mathsf{N_{CT}}} \subseteq \Delta^{\mathcal{J}}$ denote, respectively, the set of all naturals and the set of all o-compartments, i.e.

$$\begin{aligned}
\Delta_{\mathsf{N_{NT}}} &:= \bigcup_{NT\in\mathsf{N_{NT}}} NT^{\mathcal{I}_{\hat{c}}}, \text{ and}\\
\Delta_{\mathsf{N_{CT}}} &:= \bigcup_{CT'\in\mathsf{N_{CT'}}} CT'^{\mathcal{I}_{\hat{c}}}
\end{aligned}$$

for some $\hat{c} \in \mathbb{C}$. Due to (4.11), there exists a mapping $\mu : \Delta_{\mathsf{N_{NT}}} \cup \Delta_{\mathsf{N_{CT}}} \to \Delta_{\mathsf{N_{NT}}} \cup \mathbb{C}$ which maps o-compartments to contexts of respective type, i.e.

$$\mu(o) := \begin{cases} o & \text{if } o \in \Delta_{\mathsf{N_{NT}}}, \text{ and}\\ c \text{ such that } c \in CT^{\mathcal{J}} & \text{if } o \in CT'^{\mathcal{I}_{\hat{c}}}. \end{cases}$$

W.l.o.g, we can assume that there exist sufficiently many contexts $c \in \mathbb{C}$ to assure that μ preserves the occurrence and cardinality constraints, since we could introduce copies of c if necessary. We define the Σ-CROI $\mathcal{I}$ as follows:

$$\Gamma^{\mathcal{I}} := \mathbb{C}^{\mathcal{J}} \cup \Delta_{\mathsf{N_{NT}}} \cup \bigcup_{c\in\mathbb{C}} A_{\mathsf{RT}}{}^{\mathcal{I}_c}$$

$$\mathsf{type}(d) := \begin{cases} T & \text{if } T \in \mathsf{N}_{\mathsf{CT}} \text{ and } d \in T^{\mathcal{J}} \\ T & \text{if } T \in \mathsf{N}_{\mathsf{NT}} \cup \mathsf{N}_{\mathsf{RT}} \text{ and } d \in T^{\mathcal{I}_c} \text{ for some } c \end{cases}$$

$$(\mu(o), c, r) \in \mathsf{plays} \quad \text{iff} \quad (o, r) \in \mathsf{plays}^{\mathcal{I}_c}$$

$$(r_1, r_2) \in \mathsf{links}(RST, c) \quad \text{iff} \quad (o_1, r_1), (o_2, r_2) \in \mathsf{plays}^{\mathcal{I}_c} \text{ and } (o_1, o_2) \in RST^{\mathcal{I}_c}$$

$\mathcal{I}$ is well-defined and analogous to above one can go through the axioms step by step and show that $\mathcal{I}$ satisfies $\mathcal{K}$, e.g. Axiom (4.13) ensures that plays respects parts in $\mathcal{I}$. □

Let us consider once again Example 2.8 and Figure 2.2 to analyse some implications which can be drawn due to the mapping into description logics.

Example 4.2. *Instead of writing down all axioms of the respective ontology $\mathcal{O}_{\mathsf{Bank}}$, we will rather point out some interesting inferences. At first we have a look at the* Bank *compartment type, the rôle group* BankAccounts*, the relationship types* own_ca *and* own_sa *and the rôle type* Customer.

Omitting general axioms, e.g. stating that every role or rolegroup instance is played and connected to counter*, the following axioms are contained in the ontology, among others:*

$$\mathsf{Bank} \sqsubseteq [\![(\geqslant_1 \mathsf{counts}.\mathsf{BankAccounts})(\mathsf{counter})]\!] \tag{4.30}$$

$$\begin{aligned} \top \sqsubseteq [\![\exists \mathsf{plays}.\mathsf{BankAccounts} \equiv\ &\geqslant_1 \mathsf{plays}.(\mathsf{CheckingAccount} \sqcup \mathsf{SavingsAccount}) \\ &\sqcap \leqslant_1 \mathsf{plays}.(\mathsf{CheckingAccount} \sqcup \mathsf{SavingsAccount})]\!] \end{aligned} \tag{4.31}$$

$$\top \sqsubseteq \left[\!\!\left[\mathsf{SavingsAccount} \sqsubseteq \geqslant_1 \mathsf{own\_sa}^{-}.\top\right]\!\!\right] \tag{4.32}$$

$$\top \sqsubseteq \left[\!\!\left[\mathsf{CheckingAccount} \sqsubseteq \geqslant_1 \mathsf{own\_ca}^{-}.\top\right]\!\!\right] \tag{4.33}$$

$$\top \sqsubseteq [\![\exists \mathsf{own\_sa}.\top \sqsubseteq \mathsf{Customer}]\!] \tag{4.34}$$

$$\top \sqsubseteq [\![\exists \mathsf{own\_ca}.\top \sqsubseteq \mathsf{Customer}]\!] \tag{4.35}$$

In any interpretation $\mathcal{J}$ that satisfies $\mathcal{O}_{\mathsf{Bank}}$ with $c \in \mathsf{Bank}^{\mathcal{J}}$, equations (4.30) *and* (4.31) *entail the existence of an element in* $\mathsf{CheckingAccount}^{\mathcal{I}_c}$ *or* $\mathsf{SavingsAccount}^{\mathcal{I}_c}$*. Due to* (4.32) *and* (4.33)*, there must be some element "owning a CA or SA", which must be in* $\mathsf{Customer}^{\mathcal{I}_c}$ *(*(4.34) *and* (4.35)*). Hence, the following axiom can be entailed:*

$$\mathsf{Bank} \sqsubseteq [\![(\geqslant_1 \mathsf{counts}.\mathsf{Customer})(\mathsf{counter})]\!], \tag{4.36}$$

stating that the occurrence constraint for Customer *is essentially* $1..*$*. For the* Transaction *compartment type we have:*

$$\mathsf{Transaction} \sqsubseteq [\![(\geqslant_1 \mathsf{counts}.\mathsf{Participants} \sqcap \leqslant_1 \mathsf{counts}.\mathsf{Participants})(\mathsf{counter})]\!] \tag{4.37}$$

$$\begin{aligned} \top \sqsubseteq [\![\exists \mathsf{plays}.\mathsf{Participants} \equiv\ &\geqslant_1 \mathsf{plays}.(\mathsf{Source} \sqcup \mathsf{Target}) \\ &\sqcap \leqslant_1 \mathsf{plays}.(\mathsf{Source} \sqcup \mathsf{Target})]\!] \end{aligned} \tag{4.38}$$

$$\begin{aligned} \top \sqsubseteq\ &[\![\mathsf{Source} \sqsubseteq \geqslant_1 \mathsf{trans}.\top \sqcap \leqslant_1 \mathsf{trans}.\top]\!] \\ &\sqcap \left[\!\!\left[\mathsf{Target} \sqsubseteq \geqslant_1 \mathsf{trans}^{-}.\top \sqcap \leqslant_1 \mathsf{trans}^{-}.\top\right]\!\!\right] \end{aligned} \tag{4.39}$$

$$\top \sqsubseteq [\![\exists \mathsf{trans}.\top \sqsubseteq \mathsf{Source}]\!] \sqcap \left[\!\!\left[\exists \mathsf{trans}^{-}.\top \sqsubseteq \mathsf{Target}\right]\!\!\right] \tag{4.40}$$

Assume that there would be some $d \in \mathsf{Transaction}^{\mathcal{J}}$*. By* (4.37) *and* (4.38)*, we know that there exists exactly one element in* $\mathsf{Source}^{\mathcal{I}_d}$ *or* $\mathsf{Target}^{\mathcal{I}_d}$*. But due to* (4.39) *and* (4.40)*, there must also be an element in* $\mathsf{Target}^{\mathcal{I}_d}$ *or* $\mathsf{Source}^{\mathcal{I}_d}$*, respectively. This results in two elements in* $\mathsf{Participants}^{\mathcal{I}_d}$ *which contradicts* (4.37)*. Thus,*

$$\mathsf{Transaction} \sqsubseteq \bot \tag{4.41}$$

is entailed. Actually, the occurrence constraints for Participants *should be* 2..2*. Back to the* Bank *compartment, we also have*

$$\top \sqsubseteq [\![\mathsf{Customer} \sqsubseteq \geqslant_1 \mathsf{issues}.\top]\!] \tag{4.42}$$

$$\top \sqsubseteq \left[\!\!\left[\exists \mathsf{issues}^{-}.\top \sqsubseteq \mathsf{MoneyTransfer}\right]\!\!\right] \tag{4.43}$$

$$\top \sqsubseteq \left[\!\!\left[\exists \mathsf{plays}.\mathsf{MoneyTransfer} \sqsubseteq \mathsf{Transaction}'\right]\!\!\right] \tag{4.44}$$

$$\neg\left[\!\!\left[\mathsf{Transaction}' \sqcap \exists \mathsf{plays}.\top \sqsubseteq \bot\right]\!\!\right] \sqsubseteq \exists \mathsf{nested}.\mathsf{Transaction} \tag{4.45}$$

Due to (4.36)*,* (4.42) *and* (4.43)*, there must be some element in* $\mathsf{MoneyTransfer}^{\mathcal{I}_c}$ *which must be* played *by an element in* $\mathsf{Transaction}'^{\mathcal{I}_c}$*. Thus, by* (4.45) *we have to have some context* c_2 *connected to* c *via* nested *with* $c_2 \in \mathsf{Transaction}$*. But that contradicts* (4.41)*.*

Overall, the banking example is indeed inconsistent, due to a small modelling error in some occurrence constraints within one compartment type that makes the whole domain model unsatisfiable. ♢

Besides checking satisfiability in general, we can also address the following other questions a domain analyst would be interested in:

(Q1) Is a specific compartment type CT instantiable, i.e. does there exist some Σ-CROI $\mathcal{I}$ s.t. there exists a compartment c with $c \in CT^{\mathcal{I}}$?

(Q2) Is the rôle type RT playable, i.e. does there exist some Σ-CROI $\mathcal{I}$ s.t. there is some $r \in RT^{\mathcal{I}}$?

(Q3) Can two rôles be linked via the relationship type RST, i.e. does there exist some Σ-CROI $\mathcal{I}$ s.t. there are some $r_1, r_2 \in \mathsf{R}^{\mathcal{I}}$, $c \in \mathsf{C}^{\mathcal{I}}$ with $\mathsf{links}(RST, c) = (r_1, r_2)$?

(Q4) Can more precise constraints be entailed, i.e. do there exist some cardinalities in $\mathcal{C}$ which can never be reached?

(Q5) Is some partial knowledge about an instance satisfiable, i.e. does there exist some Σ-CROI $\mathcal{I}$ based on this partial knowledge?

To answer all these questions, we utilise Σ-CROAs as introduced earlier. For (Q1), we add the meta type assertion

$$CT(a)$$

to $\mathcal{A}$ and check $\mathfrak{B}_{(\mathcal{M},\mathcal{A},\mathcal{C})}$ for consistency. To answer (Q2), an object type assertion of the form

$$RT(a, c),$$

added to $\mathcal{A}$, is sufficient. W.l.o.g. we assume that a and c do not occur in $\mathfrak{B}_{(\mathcal{M},\mathcal{A},\mathcal{C})}$ before. Due to the other axioms already specified, the rôle must also be played. Similarly, for (Q3) we add the link assertion

$$\mathsf{link\_assert}(RST, c, a_1, a_2).$$

To test whether constraints are sharp, we assert the opposite and check for inconsistency. Here, we will directly add axioms to $\mathfrak{B}_{(\mathcal{M},\mathcal{A},\mathcal{C})}$. If, for example, there must exist at least n roles of type RT in CT, we add

$$CT \sqsubseteq [\![(\leqslant_n \mathsf{counts}.RT)(\mathsf{counter})]\!] \wedge CT(a)$$

to $\mathfrak{B}_{(\mathcal{M},\mathcal{A},\mathcal{C})}$ which states that there must also exist at most n RT-roles. Then, inconsistency of $\mathfrak{B}_{(\mathcal{M},\mathcal{A},\mathcal{C})}$ would imply that there must indeed exist $n+1$ RT-roles in CT.

For Q5, we assume that we know certain facts about the domain. This could be the existence of a compartment, a role or a link between two roles. These can all be formulated via a Σ-CROA, with individual names adjusted accordingly. If adding all respective assertions still preserves consistency of $\mathfrak{B}_{(\mathcal{M},\mathcal{A},\mathcal{C})}$, then there exists some Σ-CROI that satisfies $\mathcal{M}$, $\mathcal{A}$ and $\mathcal{C}$ and respects these facts.

4.2 Going beyond Σ-CROMs

In [KLG+14], Kühn et al. present a feature model for role-based modelling languages of which the above defined CROM is one instance. To support other variants of role-based models, the same basic ideas of our mapping can be applied. However, a detailed analysis which features can easily be supported, is necessary. The first feature that we will discuss is inheritance.

Let us assume, that additionally to a CROM we also have an irreflexive, asymmetric, functional inheritance relation $\prec_{NT}$ over the natural types. Inheritance is intuitively captured as a GCI:

$$\top \sqsubseteq [\![NT_a \sqsubseteq NT_b]\!] \text{ for all } NT_a \prec_{NT} NT_b. \tag{4.46}$$

Then, obviously we have to adjust axiom (4.5) since natural types do not need to be disjoint anymore. This implies that if some natural type NT fills a role type RT, then every subtype of NT also fills RT. Similarly, inheritance for compartment types could be handled. Here, one has to make sure what exactly the intended semantics of compartment inheritance is supposed to be. In our setting, a compartment subtype would be a specialization in the sense that all axioms and constraints that hold for the supertype also hold for the subtype but there could exist additional axioms or constraints in the subtype.

Some features in the feature model handle behaviour and dynamic aspects of role-based models. These are not relevant in our case, as they have no influence on satisfiability. The same holds for features about the ontological identity of roles and compartments. All other features can easily be supported since only little changes to the mapping are required. These include, among others, deep rôles, i.e. rôles which are allowed to play rôles, or whether a rôle type can be played several times by one object in the same compartment.

Two kinds of constraints that, in our opinion, might be important in practice, were not considered until now: Constraints based on attributes of players and temporal constraints. The first ones are quite easy to express in $\mathcal{L}_{\mathsf{M}}[\![\mathcal{L}_{\mathsf{O}}]\!]$. Besides some technical changes on the above axioms, for each attribute att_i of a natural type or o-compartment, we introduce the rigid role $\mathsf{att}_i \in \mathsf{O}_{\mathsf{Rrig}}$, since we assume that attributes of a rigid type are also rigid. Here, it might even be worth considering a description logic with concrete domains [Lut02b; Lut02a] as the object logic. We did not investigate these extensions of DLs, but we conjecture that all results of Chapter 3 can be transferred without much effort.

By utilising rigid types which are modelled with more detail, we can also introduce more fine-grained constraints. For example, the fills-relation could be specified to not only require a player of a certain rôle type to be of some specific natural type but rather to demand that certain attributes must exist. To be allowed to “play” the rôle of an employee, a person must have a tax ID. Going a step further, we could also restrict rôle-playing based on the values of the attributes. In the context of USA, a person can only play president if he is a “natural born citizen” and at least 35 years old:

$$\mathsf{USA} \sqsubseteq [\![\exists \mathsf{plays}.\mathsf{President} \sqsubseteq \mathsf{Person} \sqcap \exists \mathsf{natural\_born\_citizen}.\{\mathsf{true}\} \sqcap \exists \mathsf{age} \geq_{35}]\!].$$

Last but not least we can express additional complex constraints with arbitrary $\mathcal{L}_{\mathsf{M}}[\![\mathcal{L}_{\mathsf{O}}]\!]$ axioms. For example,

$$[\![\exists RST.(\exists RST.\top) \sqsubseteq \bot]\!] \sqsubseteq [\![\exists RST.\top \sqsubseteq \bot]\!]$$

states that if there are no chains of length two for an relationship type RST, then there will not be any RST at all in that particular compartment.

Chapter 5

JConHT – A $\mathcal{SHOIQ}[\![\mathcal{SHOIQ}]\!]$ Reasoner

In the last chapter, we presented a mapping for role-based models into contextualised description logic ontologies. This step can be automated which helps the automated processing and investigation of role-based models. But to obtain a full automation we need a reasoner that is capable of handling these ontologies, and can check the consistency automatically.

Since the translation for role-based models produces $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$-ontologies, i.e. conjunctions of meta axioms, we will only consider $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$-ontologies. Throughout this chapter, let $\mathfrak{O} = (\mathcal{O}, \mathcal{R}_\mathsf{M}, \mathcal{R}_\mathsf{O})$ denote a $\mathcal{SHOIQ}[\![\mathcal{SHOIQ}]\!]$-ontology and let b denote the bijection as in Definition 3.6. Hence, $\mathcal{O}^\mathsf{b}$ denotes the outer abstraction of $\mathcal{O}$. We also remind the reader that the *restricted type* is the $\mathsf{ran}(\mathsf{b})$-type (see Definition 3.10) of some element and thus a subset of $\mathsf{ran}(b)$.

In this chapter, we present a practical algorithm to check consistency. Furthermore, we implemented this algorithm and give details on the design of the implementation. Since internally, this algorithm only calls standard DL consistency checks, we can reuse existing, highly optimised DL reasoners.

5.1 A Black-Box Approach

In Section 3.3, we showed that we can reduce the consistency problem of $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$ to two separate decision problems. Now, the general idea for an implementation is to use performant reasoners as black-boxes for these subtasks. In this section we discuss how the two subtasks, namely admissibility of a set $\mathcal{X}$ and outer consistency w.r.t. $\mathcal{X}$, can be reduced to standard reasoning tasks.

5.1.1 Admissibility

In the definition of admissibility (Definition 3.11), where we define $\mathcal{B}_{X_i}$, we require negated o-axioms. Negated axioms, especially negated GCIs, are usually not supported by classical description logic reasoners. Therefore, we introduce the notion of *weakly negated axioms*.

Definition 5.1 (Weakly negated axioms). *Let α be an axiom over N, then the* weakly negated axiom *α^* is defined as follows:*

- *if $\alpha = C \sqsubseteq D$, then $\alpha^* := (C \sqcap \neg D)(x)$ where x is a fresh variable,*
- *if $\alpha = C(a)$, then $\alpha^* := (\neg C)(a)$, and*
- *for all other α, $\alpha^* := \neg\alpha$.* $\diamondsuit$

Note here, that for concept assertions the negation of the axiom $\neg\alpha$ and weakly negated axiom α^* are semantically equal, but not syntactical since $\neg\alpha$ uses axiom negation, while α^* only requires concept negation. Furthermore, in the presence of nominals we could rewrite a negated role assertion of the form $\neg r(a,b)$ as $(\neg\exists r.\{b\})(a)$. But as OWL reasoners in general support negated role assertions, we keep $\neg r(a,b)$. Moreover, this definition reflects that $\alpha \wedge \alpha^*$ is inconsistent, but not that $\mathcal{I} \models \alpha \vee \alpha^*$ for all interpretations $\mathcal{I}$.

Lemma 5.2. *Let α be an axiom, and let $\mathcal{I}$ be an interpretation. Then,*

1. *$\alpha \wedge \alpha^*$ is inconsistent, and*
2. *if $\mathcal{I} \not\models \alpha$, then there exists an extension $\mathcal{I}'$ of $\mathcal{I}$ such that $\mathcal{I}' \models \alpha^*$.*

Proof. Let $\alpha = C \sqsubseteq D$ and $\alpha^* = (C \sqcup \neg D)(x_{\mathsf{new}})$. Assume that $\mathcal{I} \models \alpha$ and $\mathcal{I} \models C(x_{\mathsf{new}})$. Then, we know that ${x_{\mathsf{new}}}^{\mathcal{I}} \in C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. Thus we have ${x_{\mathsf{new}}}^{\mathcal{I}} \in D^{\mathcal{I}}$ and clearly ${x_{\mathsf{new}}}^{\mathcal{I}} \notin (C \sqcap \neg D)^{\mathcal{I}}$. Hence, $C \sqsubseteq D \wedge (C \sqcap \neg D)(x_{\mathsf{new}})$ is inconsistent.

If $\mathcal{I} \not\models C \sqsubseteq D$, then there exists $d \in \Delta^{\mathcal{I}}$ such that $d \in C^{\mathcal{I}} \cap (\neg D)^{\mathcal{I}}$. Then, the interpretation $\mathcal{I}'$, obtained from $\mathcal{I}$ by setting ${x_{\mathsf{new}}}^{\mathcal{I}'} := d$, models α^*.

For the other axioms, the weakly negated axiom is defined as the normally negated axiom and the claim follows directly from the definition of $\models$ (see Definition 2.5). □

To check admissibility of a set of types, we distinguish whether or not rigid names are present. In the latter case, i.e. $\mathsf{O}_{\mathsf{Crig}} = \mathsf{O}_{\mathsf{Rrig}} = \emptyset$, the first condition of Definition 3.11 is always fulfilled. W.l.o.g., we can interpret the individual names in the same way in every interpretation. Therefore, we can check each $\mathfrak{B}_{X_i}$ separately. We show that it is sufficient to consider the weakly negated axioms.

Lemma 5.3. *If no rigid names are present, i.e. $\mathsf{O}_{\mathsf{Crig}} = \mathsf{O}_{\mathsf{Rrig}} = \emptyset$, the set $\mathcal{X} = \{X_1, \ldots, X_k\}$ of restricted types is admissible iff $\mathfrak{O}_{X_i} = (\mathcal{O}_{X_i}, \mathcal{R}_{\mathsf{O}})$ is consistent for all $1 \leq i \leq k$, where $\mathcal{O}_{X_i}$ is defined as*

$$\mathcal{O}_{X_i} := \bigwedge_{\mathsf{b}([\![\alpha]\!]) \in X_i} \alpha \;\wedge \bigwedge_{\mathsf{b}([\![\alpha]\!]) \in \mathsf{ran}(\mathsf{b}) \setminus X_i} \alpha^*.$$

Proof. Assume that $\mathcal{X}$ is admissible. Then there exists O-interpretations $\mathcal{I}_1 = (\Delta, \cdot^{\mathcal{I}_1})$, …, $\mathcal{I}_k = (\Delta, \cdot^{\mathcal{I}_k})$ such that every $\mathcal{I}_i$, $1 \leq i \leq k$, is a model of $\mathfrak{B}_{X_i}$ where $\mathfrak{B}_{X_i}$ is defined as in Definition 3.11. Let $X_{\mathsf{pos},i}$ and $X_{\mathsf{neg},i}$ respectively denote the sets of positive and negative induced o-axioms

$$X_{\mathsf{pos},i} := \{\alpha \mid \mathsf{b}([\![\alpha]\!]) \in X_i\}, \text{ and}$$
$$X_{\mathsf{neg},i} := \{\alpha \mid \mathsf{b}([\![\alpha]\!]) \in \mathsf{ran}(\mathsf{b}) \setminus X_i\}.$$

By definition of $X_{\mathsf{pos},i}$ and $X_{\mathsf{neg},i}$, $\mathcal{I}_i$ is a model of $\bigwedge_{\alpha \in X_{\mathsf{pos},i}} \alpha$ and $\bigwedge_{\alpha \in X_{\mathsf{neg},i}} \neg\alpha$. Now, let $X_{\mathsf{neg},i} = \{\alpha_{i,1}, \ldots, \alpha_{i,\ell_i}\}$. We have that $\mathcal{I}_i \models \neg\alpha_{i,j}$ for $1 \leq j \leq \ell_i$. By Lemma 5.2, we get that there exists $\mathcal{I}_i'$ such that $\mathcal{I}_i' \models {\alpha_{i,j}}^*$. By induction, we get that $\mathcal{I}_i' \models \mathfrak{O}_{X_i}$.

If all $\mathfrak{O}_{X_i}$ are consistent, there exist interpretations $\mathcal{I}_i$ such that $\mathcal{I}_i \models \mathfrak{O}_{X_i}$. W.l.o.g., we can assume that the interpretations $\mathcal{I}_i$ share the same domain Δ and that individual names are interpreted in the same way. Then, $\mathcal{I}_i$ is a model of $\bigwedge_{\mathsf{b}([\![\alpha]\!]) \in X_i} \alpha$, $\bigwedge_{\mathsf{b}([\![\alpha]\!]) \in \mathsf{ran}(\mathsf{b}) \setminus X_i} \alpha^*$ and $\mathcal{R}_{\mathsf{O}}$. Due to Lemma 5.2, $\mathcal{I}_i$ also models $\bigwedge_{\mathsf{b}([\![\alpha]\!]) \in \mathsf{ran}(\mathsf{b}) \setminus X_i} \neg\alpha$. Hence, $\mathcal{X}$ is admissible. □

In the former case, i.e. $\mathsf{O}_{\mathsf{Crig}} \cup \mathsf{O}_{\mathsf{Rrig}} \neq \emptyset$, we use the renaming technique of [BGL08; BGL12] as in the proof of Theorem 3.19. Due to the interaction of the rigid names, we must reason over all $\mathfrak{B}_{X_i}$ simultaneously.

Definition 5.4 (Renamed axiom, sets of positive and negative induced renamed o-axioms, induced object ontology). *Let $\mathcal{X} = \{X_1, \ldots, X_k\}$ be a set of restricted types and let α be an axiom over* O*. For $\iota \in \mathbb{N}$, the* renamed axiom $\alpha^{(\iota)}$ *is obtained from α by replacing all flexible concept names A, i.e. $A \in \mathsf{O}_\mathsf{C} \setminus \mathsf{O}_{\mathsf{Crig}}$, with a copy $A^{(\iota)}$ and all flexible role names r with a copy $r^{(\iota)}$ where we assume w.l.o.g. that $A^{(\iota)}$ and $r^{(\iota)}$ do not occur in $\mathfrak{B}$.*

Then, the set of positively induced renamed o-axioms X_{pos}*, the* set of negatively induced renamed o-axioms X_{neg}*, the* renamed object RBox $\mathcal{R}_\mathsf{O}{}'$ *and the* induced object ontology $\mathfrak{O}_\mathcal{X} = (\mathcal{O}_\mathcal{X}, \mathcal{R}_\mathsf{O}{}')$ *are defined as follows:*

$$X_{\mathsf{pos}} := \bigcup_{i=1}^{k} \{\alpha^{(i)} \mid \mathsf{b}(\llbracket \alpha \rrbracket) \in X_i\},$$

$$X_{\mathsf{neg}} := \bigcup_{i=1}^{k} \{\alpha^{(i)} \mid \mathsf{b}(\llbracket \alpha \rrbracket) \in \mathsf{ran}(\mathsf{b}) \setminus X_i\},$$

$$\mathcal{R}_\mathsf{O}{}' := \bigcup_{i=1}^{k} \{\alpha^{(i)} \mid \alpha \in \mathcal{R}_\mathsf{O}\}, \text{ and}$$

$$\mathcal{O}_\mathcal{X} := \bigwedge_{\beta \in X_{\mathsf{pos}}} \beta \ \wedge \bigwedge_{\beta \in X_{\mathsf{neg}}} \beta^*.$$

◇

Although the next lemma also holds if no rigid names are present, we state it explicitly with rigid names, as in the other case we will use Lemma 5.3. From now on, let α always denote an original axiom and β an renamed axiom.

Lemma 5.5. *If rigid names are present, i.e. $\mathsf{O}_{\mathsf{Crig}} \cup \mathsf{O}_{\mathsf{Rrig}} \neq \emptyset$, the set $\mathcal{X}$ of restricted types is admissible iff the induced object ontology $\mathfrak{O}_\mathcal{X}$ is consistent.*

Proof. We can reuse the claim made in the proof of Theorem 3.19: $\mathcal{X}$ is admissible iff $\mathfrak{B}_\mathcal{X}$ is consistent where $\mathfrak{B}_\mathcal{X}$ is defined as

$$\mathfrak{B}_\mathcal{X} := \left(\bigwedge\nolimits_{1 \leq \iota \leq k} \left(\bigwedge_{\mathsf{b}(\llbracket \alpha \rrbracket) \in X_\iota} \alpha^{(\iota)} \ \wedge \bigwedge_{\mathsf{b}(\llbracket \alpha \rrbracket) \in \mathsf{ran}(\mathsf{b}) \setminus X_\iota} \neg \alpha^{(\iota)} \right), \quad \mathcal{R}_\mathsf{O}{}' \right).$$

Hence, we only have to show that $\mathfrak{B}_\mathcal{X}$ is consistent iff $\mathfrak{O}_\mathcal{X}$ is consistent. Let $X_{\mathsf{neg}} = \{\beta_1, \ldots, \beta_\ell\}$.

If there exists an O-interpretation $\mathcal{G}$ such that $\mathcal{G} \models \mathfrak{B}_\mathcal{X}$, then we also have that $\mathcal{G} \models \neg\beta_i$ for all $1 \leq i \leq \ell$. Due to Lemma 5.2 and by induction, there is some $\mathcal{G}'$ such that $\mathcal{G}' \models \beta_i^*$ for all $1 \leq i \leq \ell$. Hence, $\mathcal{G}'$ is a model of $\mathfrak{O}_\mathcal{X}$.

Conversely, if there exists an O-interpretation $\mathcal{G}$ such that $\mathcal{G} \models \mathfrak{O}_\mathcal{X}$, then we also have that $\mathcal{G} \models \beta_i^*$ for all $1 \leq i \leq \ell$. By Lemma 5.2, we know that $\mathcal{G} \not\models \beta_i$. Hence, $\mathcal{G} \models \neg\beta_i$ and $\mathcal{G} \models \mathfrak{B}_\mathcal{X}$.

□

To sum up, we can decide admissibility of $\mathcal{X}$ by checking, respectively, $\mathfrak{O}_\mathcal{X}$ or $\mathfrak{O}_{X_i}$ for consistency, depending on whether rigid names are present or not.

5.1.2 Outer consistency

In the decision procedures described in Section 3.3, we always construct the set $\mathcal{X}$ first, and then check whether $\mathfrak{B}^{\mathsf{b}}$ is outer consistent w.r.t. $\mathcal{X}$. In the general case with rigid names we enumerate all sets $\mathcal{X} \subseteq \mathcal{P}(\mathsf{ran(b)})$. When only rigid concept names and no rigid role names are present, we non-deterministically guess a set $\mathcal{X}$. For the case without rigid names we construct the largest possible set $\mathcal{X}$ that is admissible, and argue that any $\mathfrak{B}^{\mathsf{b}}$ that is outer consistent w.r.t. some admissible $\mathcal{X}'$ is also outer consistent w.r.t. $\mathcal{X}$. Hence, we only have to test outer consistency w.r.t. to $\mathcal{X}$. But all these techniques involve the possibly unnecessary, exponentially large construction of $\mathcal{X}$. Alternatively, we can also use the following lemma, which is a direct consequence of Lemma 3.14.

Lemma 5.6. *The $\mathcal{L}_{\mathsf{M}}[\![\mathcal{L}_{\mathsf{O}}]\!]$-BKB $\mathfrak{B}$ is consistent iff there is an* M*-interpretation $\mathcal{H}$ such that $\mathcal{H} \models \mathfrak{B}^{\mathsf{b}}$ and $\mathcal{Z}_{\mathcal{H}} = \{\mathsf{type}^{\mathcal{H}}_{\mathsf{ran}(b)}(d) \mid d \in \Delta^{\mathcal{H}}\}$ is admissible.*

Proof. Let us assume that $\mathfrak{B}$ is consistent. By Lemma 3.14, we know that if $\mathfrak{B}$ is consistent, then there exists an admissible set $\mathcal{X}$ such that $\mathfrak{B}^{\mathsf{b}}$ is outer consistent w.r.t $\mathcal{X}$. By the definition of outer consistency, there exists an M-interpretation $\mathcal{H}$ that models $\mathfrak{B}^{\mathsf{b}}$ and weakly respects $(\mathsf{ran(b)}, \mathcal{X})$. By definition, we have that $\mathcal{Z}_{\mathcal{H}} \subseteq \mathcal{X}$. Since every subset of an admissible set is also admissible, $\mathcal{Z}$ is also admissible.

For the 'if' direction we assume that $\mathcal{H} \models \mathfrak{B}^{\mathsf{b}}$ and that $\mathcal{Z}$ is admissible. If $\mathcal{H} \models \mathfrak{B}^{\mathsf{b}}$, then $\mathfrak{B}^{\mathsf{b}}$ is outer consistent w.r.t $\mathcal{Z}$. Since $\mathcal{Z}$ is admissible, we know, due to Lemma 3.14, that $\mathfrak{B}$ is consistent. □

Due to Lemma 5.6, we do not need to construct the set $\mathcal{X}$ first. We can also enumerate all models $\mathcal{H}$ of $\mathfrak{B}^{\mathsf{b}}$, and check for each $\mathcal{H}$ if the occurring types are admissible. If there exist one model of $\mathfrak{B}^{\mathsf{b}}$, then there exists infinitely many. But we only need to check those that are *essentially different*, i.e. those for which the set of occurring restricted types differs.

Definition 5.7 (Essentially equal interpretations). *For an* M*-interpretation $\mathcal{H}$, the* set of occurring, restricted types *$\mathcal{Z}_{\mathcal{H}}$ is defined as*

$$\mathcal{Z}_{\mathcal{H}} := \{\mathsf{type}^{\mathcal{H}}_{\mathsf{ran}(b)}(d) \mid d \in \Delta^{\mathcal{H}}\}.$$

The two M*-interpretations $\mathcal{H}_1$ and $\mathcal{H}_2$ are* essentially equal *if $\mathcal{Z}_{\mathcal{H}_1} = \mathcal{Z}_{\mathcal{H}_2}$.* ◇

Using the results of Lemma 5.3, 5.5 and 5.6, we can construct a simple algorithm, as depicted in Algorithm 1. Here, $\mathcal{O}_{Z_i}$ and $\mathcal{O}_{\mathcal{Z}_{\mathcal{H}}}$ are respectively defined as in Lemma 5.3 and Definition 5.4. We enumerate all essentially equal models of $\mathfrak{B}^{\mathsf{b}}$ and check for each model $\mathcal{H}$ whether $\mathcal{Z}_{\mathcal{H}}$ is admissible. Note that on the object level, only classical consistency checks are used. On the meta level, the bare information about consistency is not sufficient, since we also need information about the restricted types occurring in the model. Hence, we need a DL reasoner that constructs a model and, if consistent, returns the set of occurring restricted types.

Lemma 5.8. *Algorithm 1 is sound, complete, and terminating.*

Proof. Up to essential equivalence, there are only finitely many models. Checking whether $\mathcal{H}$ is a model and checking whether $\mathfrak{O}_{\mathcal{Z}_{\mathcal{H}}}$ or $\mathfrak{O}_{Z_i}$ is consistent, is decidable. Thus, Algorithm 1 terminates.

Algorithm 1: Algorithm for checking consistency of $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$-BKB $\mathfrak{B}$

Input : $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$-BKB $\mathfrak{B}$
Output : true if $\mathfrak{B}$ is consistent, false otherwise

$\mathfrak{H} := \{\mathcal{H} \mid \mathcal{H} \models \mathfrak{B}^\mathsf{b}$, up to essential equality$\}$
for $\mathcal{H} \in \mathfrak{H}$ **do**
  $\mathcal{Z}_\mathcal{H} := \{\mathsf{type}^{\mathcal{H}}_{\mathsf{ran}(b)}(d) \mid d \in \Delta^{\mathcal{H}}\}$
  if $\mathfrak{B}$ *contains rigid names* **then**
    if $(\mathcal{O}_{\mathcal{Z}_\mathcal{H}}, \mathcal{R}_\mathsf{O}{}')$ *is consistent* **then**
      return true
  else
    $\mathcal{Z}_\mathcal{H} = \{Z_1, \ldots, Z_k\}$
    if $(\mathcal{O}_{Z_i}, \mathcal{R}_\mathsf{O})$ *is consistent for all* $1 \leq i \leq k$ **then**
      return true
return *false*

Let us assume that $\mathfrak{B}$ is consistent. By Lemma 5.6, there exists a model $\mathcal{H}$ of $\mathfrak{B}^\mathsf{b}$ such that $\mathcal{Z}_\mathcal{H}$ is admissible. W.l.o.g., we can assume that $\mathcal{H} \in \mathfrak{H}$. Depending on whether rigid names are present, due to Lemma 5.3 and Lemma 5.5 Algorithm 1 will successfully check that $\mathfrak{O}_{Z_i}$, for all $1 \leq i \leq k$, or $\mathfrak{O}_{\mathcal{Z}_\mathcal{H}}$ is consistent. Hence, it returns true.

Let us assume that the algorithm returns true. Then, there is some $\mathcal{H} \in \mathfrak{H}$ such that $\mathfrak{O}_{Z_i}$, for all $1 \leq i \leq k$, or $\mathfrak{O}_{\mathcal{Z}_\mathcal{H}}$ is consistent. By Lemma 5.3 and Lemma 5.5, we know that $\mathcal{Z}_\mathcal{H}$ is admissible. Lemma 5.6 yields that $\mathfrak{B}$ is consistent. □

In Algorithm 1, we need to enumerate all models of $\mathfrak{B}^\mathsf{b}$ up to essential equivalence. Therefore, we need an algorithm not only deciding the consistency of a DL ontology but where applicable also returning information about a model. We discuss this and further issues about the core algorithm and core reasoner which we use in the next section.

5.2 Contextual Hypertableau

We have to take several arguments into account in order to evaluate, which reasoner or decision procedure we use as core reasoner to decide the two subtasks. First of all as mentioned earlier, in addition to checking consistency on meta level, we also need the set of occurring restricted types of a model. Hence, an reasoner that constructs a model is necessary for enumerating the models of $\mathfrak{B}^\mathsf{b}$. As pointed out in [MLH+15], there exist three major approaches used in OWL reasoners: consequence-, model construction- and rewriting-based. *Consequence-based* approaches are classically employed for the entailment problem in lightweight DLs such as $\mathcal{EL}$, while *rewriting* approaches are usually used for specific reasoning tasks, such as for query answering. *Model construction-based* approaches are utilised for expressive DLs, i.e. any extension of $\mathcal{ALC}$, and try to build a model based on the knowledge base to check consistency. These include tableau and hypertableau techniques. Since we also need the information about the restricted types of a model, we are looking for a model construction-based reasoner.

Furthermore, by means of maintenance, we will use the same reasoner for the meta level and the object level. Since a dominant framework supporting OWL, the OWL API [HB11], is implemented in Java, so is the majority of the available reasoners. Therefore, we also use Java for our OWL-conformant reasoner and implement the OWLReasoner interface of the OWL API.

Last but not least, the performance of the reasoner in consistency checking is important for us. Therefore, we take a look at the OWL Reasoner Evaluation Competition Report [PMG+15], especially at the results of the discipline: *OWL DL Consistency*. This discipline, among others, was won by Konclude [SLG14], a hybrid reasoner that combines tableau calculus with a variant of a consequence-based saturation procedure. However, due to this hybrid approach it is unsuitable in our setting. Besides that, it is implemented in C++ and we focus on Java-based reasoners. The next candidate is HermiT [GHM+14], a Java-implemented reasoner based on the hypertableau calculus [MSH09]. Since it meets all our requirements, we use HermiT as core reasoner in our implementation.

Before we discuss the details of a refinement of our algorithm with regard towards an implementation, we present the relevant parts of the hypertableau algorithm and refer the interested reader to [MSH07b; MSH07a; MSH09]. Like other tableau-based methods, the hypertableau calculus tries to construct an abstraction of a model for a given ontology to check whether that ontology is consistent. But in contrast to other tableau calculi, it operates on a set of *DL-clauses* and an ABox, instead of a TBox and an ABox.

Definition 5.9 (DL-Clause [MSH09]). *The concepts* $\top$*,* $\bot$*, and concepts of the form* A *and* $\neg A$ *for* $A \in \mathsf{M_C}$ *are called* literal concepts. *Let* $\mathsf{M_V}$ *be the set of variables disjoint from the set of individuals* $\mathsf{M_I}$*. An* atom *is an expression of the form* $B(x)$*,* $(\geqslant_n s.B)(x)$*,* $r(x,y)$*, or* $x \approx y$*, for* $x, y \in \mathsf{M_V} \cup \mathsf{M_I}$*,* B *a literal concept,* r *an atomic role,* s *a role, and* n *a positive integer. A* DL-clause *is an expression of the form*

$$U_1 \wedge \cdots \wedge U_m \rightarrow V_1 \vee \cdots \vee V_n$$

where U_i *and* V_j *are atoms,* $m, n \geq 0$*. The conjunction* $U_1 \wedge \cdots \wedge U_m$ *is called the* antecedent*, and the disjunction* $V_1 \vee \cdots \vee V_n$ *is called* consequent*. The empty antecedent and the empty consequent are respectively written as* $\top$ *and* $\bot$*.*

Let $\mathcal{H} = (\Delta^{\mathcal{H}}, \cdot^{\mathcal{H}})$ *be an* M*-interpretation and let* $\mu : \mathsf{M_V} \rightarrow \Delta^{\mathcal{H}}$ *be a mapping from variables to domain elements. We define* $\cdot^{\mathcal{H},\mu}$ *as follows:*

$$c^{\mathcal{H},\mu} := \begin{cases} c^{\mathcal{H}} & \text{if } c \in \mathsf{M_I}, \text{ and} \\ \mu(c) & \text{if } c \in \mathsf{M_V}. \end{cases}$$

Satisfaction of an atom, DL-clause, and a set of DL-clauses $\mathcal{C}$ *in* $\mathcal{H}$ *and* μ *is defined in Table 5.1.* ◇

In the hypertableau algorithm, a few preprocessing steps are necessary, namely *elimination of transitivity axioms*, *normalisation* and *clausification*. The clausification translates a normalised DL ontology $\mathfrak{O} = (\mathcal{O}, \mathcal{R})$ without transitivity axioms into a pair $(\mathcal{C}, \mathcal{A})$, where $\mathcal{C}$ is a set of DL-clauses and $\mathcal{A}$ is an ABox, and $\mathfrak{O}$ and $(\mathcal{C}, \mathcal{A})$ are equisatisfiable. Here, we omit further details of the preprocessing steps, but emphasise that the preprocessing not only preserves satisfiability, but also satisfiability w.r.t. essential equality. Let $\mathcal{K}$ be the input ontology and

Table 5.1: Satisfaction of DL-Clauses

$\mathcal{H},\mu \models C(c)$	iff	$c^{\mathcal{H},\mu} \in C^{\mathcal{H}}$
$\mathcal{H},\mu \models r(c,d)$	iff	$(c^{\mathcal{H},\mu}, d^{\mathcal{H},\mu}) \in r^{\mathcal{H}}$
$\mathcal{H},\mu \models c \approx d$	iff	$c^{\mathcal{H},\mu} = d^{\mathcal{H},\mu}$
$\mathcal{H},\mu \models \bigwedge_{i=1}^{m} U_i \rightarrow \bigvee_{j=1}^{n} V_j$	iff	$\mathcal{H},\mu \models U_i$ for each $1 \leq i \leq m$ implies $\mathcal{H},\mu \models V_j$ for some $1 \leq j \leq n$
$\mathcal{H} \models \bigwedge_{i=1}^{m} U_i \rightarrow \bigvee_{j=1}^{n} V_j$	iff	$\mathcal{H},\mu \models \bigwedge_{i=1}^{m} U_i \rightarrow \bigvee_{j=1}^{n} V_j$ for all mappings μ
$\mathcal{H} \models \mathcal{C}$	iff	$\mathcal{H} \models \mathbf{r}$ for all $\mathbf{r} \in \mathcal{C}$

$(\mathcal{C},\mathcal{A})$ the clausification after normalisation. There exists a model $\mathcal{I}$ of $\mathcal{K}$ iff the exists a model $\mathcal{H}$ of $(\mathcal{C},\mathcal{A})$ such that $\mathcal{I}$ and $\mathcal{H}$ are essentially equal. Hence, it is sufficient to consider the clausification of the outer abstraction of an $\mathcal{L}_{\mathsf{M}}[\![\mathcal{L}_{\mathsf{O}}]\!]$ ontology.

As typical for tableau algorithms, the hypertableau calculus uses a blocking strategy to ensure termination. Intuitively, in the construction of a model, a blocked individual is replaced by its blocker. Thus, by the unravelling technique infinite models can be constructed. Here, it is important for us, that the blocked individual and its blocker have the same restricted type. This holds for all blocking strategies used in the hypertableau algorithm. For the sake of conciseness, we omit the specific blocking strategy in our definition of the hypertableau algorithm and refer to [MSH09] for a complete definition.

Definition 5.10 (Hypertableau Algorithm [MSH09]). *Given a set of named individual* $\mathsf{M_I}$*, the set of* root individuals $\mathsf{M_O}$ *is the smallest set such that* $\mathsf{M_I} \subseteq \mathsf{M_O}$*, and if* $x \in \mathsf{M_O}$*, then* $x.\langle r, B, i\rangle \in \mathsf{M_O}$ *for each role r, literal concept B and positive integer i. The set of* all individuals $\mathsf{M_A}$ *is the smallest set such that* $\mathsf{M_O} \subseteq \mathsf{M_A}$*, and if* $x \in \mathsf{M_A}$*, then* $x.i \in \mathsf{M_A}$ *for each positive integer i.* $x.i$ *is a* successor *of x, and* descendant *is the transitive closure of successor.*

In the hypertableau algorithm, an ABox can additionally contain

- *annotatd equalities,*
- $\bot$*, which is equivalent to the concept assertion* $\bot(a)$ *for some* $a \in \mathsf{M_I}$*,*
- *assertions that contain individuals from* $\mathsf{M_A}$ *and not only* $\mathsf{M_I}$*, and*
- *renamings of the form* $a \mapsto b$*.*

The ABox $\mathsf{prune}_{\mathcal{A}}(\mathsf{c})$ *is obtained from* $\mathcal{A}$ *by removing all assertions containing descendants of c. The ABox* $\mathsf{merge}_{\mathcal{A}}(\mathsf{s} \rightarrow \mathsf{t})$ *is obtained from* $\mathsf{prune}_{\mathcal{A}}(\mathsf{s})$ *by replacing all individuals s with individual t in all assertions and annotation, and, if both s and t are root individuals, adding the renaming* $s \mapsto t$*.*

Table 5.2 specifies derivation rules *that, given an ABox* $\mathcal{A}$ *and a set of DL-Clauses* $\mathcal{C}$*, derive one or more ABoxes* $\mathcal{A}_1, \ldots, \mathcal{A}_n$*.*

An ABox $\mathcal{A}$ *contains a* clash *iff* $\bot \in \mathcal{A}$*. Otherwise,* $\mathcal{A}$ *is* clash-free.

For a set of DL-clauses $\mathcal{C}$ *and an input ABox* $\mathcal{A}$*, a* derivation *is a pair* (T,λ) *where T is a finitely branching tree and* λ *is a function that labels the nodes of T with ABoxes such that for each node* $t \in T$*:*

- $\lambda(t) = \mathcal{A}$ *if t is the root of T,*

Table 5.2: Derivation rules of the hypertableau calculus [MSH09]

Hyp-rule	If 1. $\mathbf{r} \in \mathcal{C}$, where $\mathbf{r} = U_1 \wedge \cdots \wedge U_m \to V_1 \vee \cdots \vee V_n$, and 2. a mapping σ from the variables in $\mathbf{r}$ to the individuals of $\mathcal{A}$ exists such that 2.1. there is no $x \in \mathsf{N_V}$ s.t. $\sigma(x)$ is indirectly blocked, 2.2. $\sigma(U_i) \in \mathcal{A}$ for each $1 \le i \le m$, and 2.3. $\sigma(V_j) \notin \mathcal{A}$ for each $1 \le j \le n$ then $\mathcal{A}_1 := \mathcal{A} \cup \{\bot\}$ if $n = 0$; $\mathcal{A}_j := \mathcal{A} \cup \{\sigma(V_j)\}$ for $1 \le j \le n$ otherwise.
$\geq$-rule	If 1. $(\geqslant_n r.B)(c) \in \mathcal{A}$, 2. c is not blocked in $\mathcal{A}$, and 3. $\mathcal{A}$ does not contain individuals $u_1, \ldots, u_n$ such that 3.1. $\{r(c,u_i), B(u_i) \mid 1 \le i \le n\} \cup \{u_i \not\approx u_j \mid 1 \le i < j \le\} \in \mathcal{A}$, and 3.2. for each $1 \le i \le n$, neither u_i is a successor of c or u_i is not blocked in $\mathcal{A}$, then $\mathcal{A}_1 := \mathcal{A} \cup \{r(c,t_i), B(t_i) \mid 1 \le i \le n\} \cup \{t_i \not\approx t_j \mid 1 \le i < j \le\}$ where $t_1, \ldots, t_n$ are fresh, distinct successors of c.
$\approx$-rule	If 1. $c \approx d \in \mathcal{A}$, 2. $c \neq d$, and 3. neither c nor d is indirectly blocked then $\mathcal{A}_1 := \mathsf{merge}_{\mathcal{A}}(\mathsf{c} \to \mathsf{d})$ if d is a named individual, or d is a root individual and c is not a named individual, or c is a descendant of d; $\mathcal{A}_1 := \mathsf{merge}_{\mathcal{A}}(\mathsf{d} \to \mathsf{c})$ otherwise.
$\bot$-rule	If $c \not\approx c \in \mathcal{A}$ or $\{A(c), \neg A(c)\} \subseteq \mathcal{A}$ where c is not directly blocked then $\mathcal{A}_1 := \mathcal{A} \cup \{\bot\}$
NI-rule	If 1. $c \approx d@^u_{\leqslant_n r.B} \in \mathcal{A}$, 2. u is a root individual, 3. c is a blockable individual that is not a successor of u, 4. d is a blockable individual, and 5. neither c nor d is indirectly blocked then $\mathcal{A}_i := \mathsf{merge}_{\mathcal{A}}(\mathsf{s} \to \Vert \mathsf{u}.\langle \mathsf{r}, \mathsf{B}, \mathsf{i} \rangle \Vert_{\mathcal{A}})$ for each $1 \le i \le n$.

- *t is a leaf of T if $\bot \in \lambda(t)$ or no derivation rule is applicable to $\lambda(t)$ and $\mathcal{C}$, and*
- *t has children $t_1, \ldots, t_n$ such that $\lambda(t_1), \ldots, \lambda(t_n)$ are exactly the results of applying one applicable rule to $\lambda(t)$ and $\mathcal{C}$ in all other cases.*

An ABox is complete *iff it labels some leaf of T.* $\diamondsuit$

Note that the annotation $\ldots @^u_{\leqslant_n r.B}$ used in the *NI*-rule does not effect the meaning of the equality, it only records its provenance. They are introduced in the clausification, but since the details are not important here, we omit them and refer to [MSH08].

Through the rest of this section let $(\mathcal{C}, \mathcal{A})$ be the clausification of the outer abstraction $\mathfrak{O}^{\mathsf{b}}$. Hence, $\mathcal{A}$ is an $\mathcal{L}_{\mathsf{M}}$-ABox over M. In Algorithm 1, we need to enumerate all models for $\mathfrak{B}^{\mathsf{b}}$ up to essential equality. The main idea is that it is sufficient to look at the complete, clash-free ABoxes, since every model of the ontology has such a corresponding ABox.

Lemma 5.11. *In any derivation for $(\mathcal{C}, \mathcal{A})$ and for each model $\mathcal{H}$ of $(\mathcal{C}, \mathcal{A})$, there exists some leaf node labelled with a clash-free ABox $\mathcal{A}'$ such that $\mathcal{H}' \models \mathcal{A}'$, where $\mathcal{H}$ and $\mathcal{H}'$ are essentially equal.*

Proof. We prove the lemma by induction on the derivation rule application. Let $\mathcal{C}$ be a set of DL-clauses, let $\mathcal{A}$ be an ABox and let $\mathcal{H} \models (\mathcal{C}, \mathcal{A})$.

Hyp-rule: Assume that the *Hyp*-rule is applicable for $\mathbf{r} = U_1 \wedge \cdots \wedge U_m \to V_1 \vee \cdots \vee V_n \in \mathcal{C}$. Then there is a mapping σ from the variables in $\mathbf{r}$ to the individuals in $\mathcal{A}$ such that $\sigma(U_i) \in \mathcal{A}$ and $\sigma(V_j) \notin \mathcal{A}$, for each $1 \leq i \leq m$, $1 \leq j \leq n$. Since $\mathcal{H} \models \mathbf{r}$, there exists mapping μ from the variables in $\mathbf{r}$ to elements of $\Delta^{\mathcal{H}}$ such that $\mu(x) = \sigma(x)^{\mathcal{H}}$ and $\mathcal{H}, \mu \models V_j$ for some $1 \leq j \leq n$. Thus, $\mathcal{H} \models \sigma(V_j)$ and, hence, $\mathcal{H} \models \mathcal{A}_j$ for some $1 \leq j \leq n$.

$\geq$-rule: If the $\geq$-rule is applicable, we have that $\geqslant_n r.B(s) \in \mathcal{A}$. Since $\mathcal{H} \models \mathcal{A}$, there exist $d_1, \ldots, d_n \in B^{\mathcal{H}}$ with $(s^{\mathcal{H}}, d_i) \in r^{\mathcal{H}}$. We define $\mathcal{H}'$ like $\mathcal{H}$, but additionally set $t_i^{\mathcal{H}'} = d_i$, where t_i is the fresh successor of c introduced by the $\geq$-rule. Hence, $\mathcal{H}'$ models $\mathcal{A}_1$ and is essentially equal to $\mathcal{H}$.

$\approx$-rule If the $\approx$-rule is applicable, we have that $\mathcal{H} \models s \approx t \in \mathcal{A}$ and, hence, $s^{\mathcal{H}} = t^{\mathcal{H}}$. Thus, $\mathcal{H}$ also models $\mathcal{A}_1 := \mathsf{merge}_{\mathcal{A}}(s \to t)$ or $\mathcal{A}_1 := \mathsf{merge}_{\mathcal{A}}(t \to s)$.

$\bot$-rule: Since $\mathcal{A}$ has a model, the $\bot$-rule is not applicable.

NI-rule analogous to the $\approx$-rule.

$\square$

Similar as for a set of restricted types $\mathcal{X}$ for which we defined admissibility (see Definition 3.11) to assure that the corresponding o-axioms 'fit together', we define admissibility of an ABox.

Definition 5.12 (Admissibility of an ABox). *Let $\mathcal{A}$ be an $\mathcal{L}_{\mathsf{M}}$-ABox over* M *and let $\mathcal{R}_{\mathsf{O}}$ be an object RBox. W.l.o.g., we assume that the meta individuals occurring in $\mathcal{A}$ are $c_1, \ldots, c_k$. We call $\mathcal{A}$* admissible *if there exist* O*-interpretations $\mathcal{I}_1 = (\Delta, \cdot^{\mathcal{I}_1}), \ldots, \mathcal{I}_k = (\Delta, \cdot^{\mathcal{I}_k})$ such that*

- *$x^{\mathcal{I}_i} = x^{\mathcal{I}_j}$ for all $x \in \mathsf{O}_{\mathsf{Crig}} \cup \mathsf{O}_{\mathsf{Rrig}} \cup \mathsf{O}_{\mathsf{I}}$ and all $i, j \in \{1, \ldots, k\}$, and*
- *every $\mathcal{I}_i$, $1 \leq i \leq k$, is a model of the $\mathcal{L}_{\mathsf{O}}$-ontology $\mathfrak{O}_{c_i} = (\mathcal{O}_{c_i}, \mathcal{R}_{\mathsf{O}})$ over* O *where*

$$\mathcal{O}_{c_i} := \bigwedge_{\mathsf{b}(\llbracket\alpha\rrbracket) \in Z^+_{\mathcal{A}}(c_i)} \alpha \wedge \bigwedge_{\mathsf{b}(\llbracket\alpha\rrbracket) \in Z^-_{\mathcal{A}}(c_i)} \alpha^*$$

with

$$Z^+_{\mathcal{A}}(c) := \{A \in \mathsf{ran(b)} \mid A(c) \in \mathcal{A}\} \text{ and}$$
$$Z^-_{\mathcal{A}}(c) := \{A \in \mathsf{ran(b)} \mid \neg A(c) \in \mathcal{A}\}. \qquad \diamondsuit$$

Obviously, if $\mathcal{A}$ is not admissible, no model $\mathcal{H}$ of $\mathcal{A}$ will have an admissible set of occurring restricted types $\mathcal{Z}_{\mathcal{H}}$. If, for example, $\{A_{[\![B(a)]\!]}(s), A_{[\![B\sqsubseteq\bot]\!]}(s)\} \subseteq \mathcal{A}$, then for any model $\mathcal{H}$ the restricted type of $s^{\mathcal{H}}$ will contain $A_{[\![B(a)]\!]}$ and $A_{[\![B\sqsubseteq\bot]\!]}$. Hence, $\mathcal{Z}_{\mathcal{H}}$ is not admissible.

Lemma 5.13. *Let $\mathcal{A}'$ be a $\mathcal{L}_{\mathsf{M}}$-ABox over M such that $\mathcal{A}'$ is complete and clash-free. Then, $\mathcal{A}'$ is admissible if and only if there exists a model $\mathcal{H}$ of $\mathcal{A}'$ such that $\mathcal{Z}_{\mathcal{H}}$ is admissible.*

Proof. In the completeness proof for the Hypertableau Algorithm (see [MSH09], Lemma 6) a model $\mathcal{H}_{(\mathcal{C},\mathcal{A})}$ for $(\mathcal{C},\mathcal{A})$ is constructed by an unravelling based on $\mathcal{A}'$. Here, the domain elements are paths. We can extend this interpretation to $\mathcal{H}_{\mathcal{A}'}$ by additionally interpreting blockable individuals as follows: if s is not blocked, then $s^{\mathcal{H}_{\mathcal{A}'}} := \left[p \mid \frac{s}{s}\right]$ and if s is blocked by t, then $s^{\mathcal{H}_{\mathcal{A}'}} := \left[p \mid \frac{t}{s}\right]$ where p is a path where s does not occur. Hence, $\mathcal{H}_{\mathcal{A}'}$ is also a model of $\mathcal{A}'$.

W.l.o.g., we assume that the meta individuals occurring in $\mathcal{A}'$ are $c_1, \ldots, c_k$. For any interpretation $\mathcal{H}$ such that $\mathcal{H} \models \mathcal{A}'$, let $\mathcal{Z}_{\mathcal{H}} = (Z_1, \ldots, Z_k, Z_{k+1}, \ldots, Z_\ell)$ such that $Z_i = \mathsf{type}^{\mathcal{H}}_{\mathsf{ran}(b)}(c_i^{\mathcal{H}})$ for $1 \le i \le k$. Considering the conjunction of axioms as a set of conjuncts, we have that $\mathcal{O}_{c_i} \subseteq \mathcal{O}_{Z_i}$ where $\mathcal{O}_{c_i}$ (see Def. 5.12) and $\mathcal{O}_{Z_i}$ (see Lem. 5.3) are defined as

$$\mathcal{O}_{c_i} := \bigwedge_{\mathsf{b}([\![\alpha]\!]) \in Z^+_{\mathcal{A}}(c_i)} \alpha \wedge \bigwedge_{\mathsf{b}([\![\alpha]\!]) \in Z^-_{\mathcal{A}}(c_i)} \alpha^*,$$
$$\mathcal{O}_{Z_i} := \bigwedge_{\mathsf{b}([\![\alpha]\!]) \in Z_i} \alpha \wedge \bigwedge_{\mathsf{b}([\![\alpha]\!]) \in \mathsf{ran(b)} \setminus Z_i} \neg\alpha.$$

If $\mathcal{A}'$ is not admissible, then $(Z_1, \ldots, Z_k)$ cannot be admissible either. Hence, $\mathcal{Z}_{\mathcal{H}}$ is also not admissible.

If $\mathcal{A}'$ is admissible, then by definition there exist O-interpretations $\mathcal{I}_1, \ldots, \mathcal{I}_k$ modelling $\mathcal{O}_{c_1}, \ldots, \mathcal{O}_{c_k}$. First, we define $Z^?_{\mathcal{A}}(c_i) := \mathsf{ran(b)} \setminus (Z^+_{\mathcal{A}}(c_i) \cup Z^-_{\mathcal{A}}(c_i))$ as the set of abstracted concept names which do not occur neither as a positive nor negative concept assertion with c_i in $\mathcal{A}'$. Note that $\mathcal{I}_i$ always either models α or $\neg\alpha$ for any o-axiom α, and that by the construction of $\mathcal{H}_{\mathcal{A}'}$, $\mathcal{H}_{\mathcal{A}'} \models A(c)$ iff $A(c) \in \mathcal{A}'$ for any $A \in \mathsf{M_C}$. We define $\mathcal{H}^*$ to be equal with $\mathcal{H}_{\mathcal{A}'}$ except that for all $\mathsf{b}([\![\alpha]\!]) \in \mathsf{ran(b)}$, we define $\mathsf{b}([\![\alpha]\!])^{\mathcal{H}^*}$ as

$$\mathsf{b}([\![\alpha]\!])^{\mathcal{H}^*} := \mathsf{b}([\![\alpha]\!])^{\mathcal{H}_{\mathcal{A}'}} \cup \{c_i^{\mathcal{H}_{\mathcal{A}'}} \mid \mathsf{b}([\![\alpha]\!]) \in Z^?_{\mathcal{A}}(c_i) \text{ and } \mathcal{I}_i \models \alpha\}.$$

Hence, with $Z_i = \mathsf{type}^{\mathcal{H}^*}_{\mathsf{ran}(b)}(c_i)$ we have that $\mathcal{I} \models \mathcal{O}_{Z_i}$. Since $\mathcal{Z}_{\mathcal{H}^*} = (Z_1, \ldots, Z_k)$, $\mathcal{Z}_{\mathcal{H}^*}$ is admissible. $\square$

Note that this lemma yields that for an inadmissible ABox $\mathcal{A}'$, there exists no model of $(\mathcal{C},\mathcal{A})$ and $\mathcal{A}'$ with admissible types. Thus, due to Lemma 5.11, we know that if all ABoxes which label leaf nodes in a derivation are not admissible, then $(\mathcal{C},\mathcal{A})$ is inconsistent. However, the converse does not hold. If $\mathcal{A}'$ is admissible and clash-free, it is still possible that there exists no model $\mathcal{H}$ of $(\mathcal{C},\mathcal{A})$ and $\mathcal{A}'$ such that $\mathcal{Z}_{\mathcal{H}}$ is admissible as the following example shows.

Example 5.14. *Let $\mathfrak{B}_{\mathsf{ex}} = (\mathcal{B}_{\mathsf{ex}}, \emptyset, \emptyset)$ with*

$$\mathcal{B}_{\mathsf{ex}} = \neg C(s) \quad \wedge \quad [\![\neg A(a)]\!] \sqsubseteq C \quad \wedge \quad \neg C \sqsubseteq [\![A \sqsubseteq \bot]\!]$$

be an $\mathcal{ALC}[\![\mathcal{ALC}]\!]$-ontology. Then,

$$\mathfrak{B}^{\mathsf{b}}_{\mathsf{ex}} = \big(\neg C(s) \quad \wedge \quad A_{[\![\neg A(a)]\!]} \sqsubseteq C \quad \wedge \quad \neg C \sqsubseteq A_{[\![A \sqsubseteq \bot]\!]}, \quad \emptyset\big)$$

is the outer abstraction of $\mathfrak{B}_{\mathsf{ex}}$. The normalisation and clausification of $\mathfrak{B}^{\mathsf{b}}_{\mathsf{ex}}$ yields $(\mathcal{C}_{\mathsf{ex}}, \mathcal{A}_{\mathsf{ex}})$ with

$$\mathcal{C}_{\mathsf{ex}} = \{A_{[\![\neg A(a)]\!]}(x) \rightarrow C(x), \quad \top \rightarrow C(x) \vee A_{[\![A \sqsubseteq \bot]\!]}(x)\},$$
$$\mathcal{A}_{\mathsf{ex}} = \{\neg C(s)\}.$$

Any derivation of $(\mathcal{C}_{\mathsf{ex}}, \mathcal{A}_{\mathsf{ex}})$ produces a leaf node labelled with

$$\mathcal{A}' = \{\neg C(s),\ A_{[\![A \sqsubseteq \bot]\!]}(s)\}.$$

Clearly, $\mathcal{A}'$ is admissible and in any model $\mathcal{H} = (\Delta^{\mathcal{H}}, \cdot^{\mathcal{H}})$ of $\mathcal{A}'$ such that $\mathcal{Z}_{\mathcal{H}}$ is admissible there exists $d \in \Delta^{\mathcal{H}}$ with $s^{\mathcal{H}} = d$ and $d \in {A_{[\![A \sqsubseteq \bot]\!]}}^{\mathcal{H}}$. Since $\mathcal{Z}_{\mathcal{H}}$ is admissible, we know that $\mathcal{O}_X := \bigwedge_{\mathsf{b}([\![\alpha]\!]) \in X} \alpha \wedge \bigwedge_{\mathsf{b}([\![\alpha]\!]) \in \mathsf{ran}(\mathsf{b}) \setminus X} \alpha^*$ must be consistent for any $X \in \mathcal{Z}_{\mathcal{H}}$. Hence, in particular $X = \{A_{[\![A \sqsubseteq \bot]\!]}\} \notin \mathcal{Z}_{\mathcal{H}}$ since $\mathcal{O}_X = A \sqsubseteq \bot \ \wedge \ \neg(\neg A(a))$ is inconsistent. Therefore, $A_{[\![\neg A(a)]\!]}$ must also be in the restricted type of d, i.e. $d \in {A_{[\![\neg A(a)]\!]}}^{\mathcal{H}}$. We implicitly obtain an ABox

$$\mathcal{A}'' = \{\neg C(s),\ A_{[\![A \sqsubseteq \bot]\!]}(s),\ A_{[\![\neg A(a)]\!]}(s)\}.$$

Because of $A_{[\![\neg A(a)]\!]}(x) \rightarrow C(x) \in \mathcal{C}_{\mathsf{ex}}$ and $\neg C(s) \in \mathcal{A}_{\mathsf{ex}}$, we know that $\mathcal{H}$ cannot be model of $(\mathcal{C}_{\mathsf{ex}}, \mathcal{A}_{\mathsf{ex}})$. In fact, there exists no model $\mathcal{H}'$ of $(\mathcal{C}_{\mathsf{ex}}, \mathcal{A}_{\mathsf{ex}})$ such that $\mathcal{Z}_{\mathcal{H}'}$ is admissible since $\mathfrak{B}_{\mathsf{ex}}$ is inconsistent. ♢

The above example illustrates that there can exist 'implicitly negated' concept assertions in a complete, clash-free meta ABox $\mathcal{A}'$ like $A_{[\![\neg A(a)]\!]}(s)$ in the above example, i.e. concept assertions that would cause a clash if added to the ABox. In the proof of Lemma 5.13, we had to add certain concept assertions in order to ensure admissibility. To avoid adding concept assertions which cause such a clash, we add special DL-clauses to $\mathcal{C}$. Since for every context $c \in \mathbb{C}$ an object axiom α is either modelled by $\mathcal{I}_c$ or not, we can assume w.l.o.g. that either $c \in \mathsf{b}([\![\alpha]\!])^{\mathcal{J}^{\mathsf{b}}}$ or $c \in \mathsf{b}([\![\alpha^*]\!])^{\mathcal{J}^{\mathsf{b}}}$. Thus, we can add the DL-clause $\top \rightarrow \mathsf{b}([\![\alpha]\!])(x) \vee \mathsf{b}([\![\alpha^*]\!])(x)$ without adding any logical consequences.

Definition 5.15 (Repletion of DL-clauses). *Let $\mathcal{C}$ be a set of $\mathcal{L}_{\mathsf{M}}$-clauses over* M *and $\mathcal{A}$ an $\mathcal{L}_{\mathsf{M}}$-ABox over* M*. The* repletion of $\mathcal{C}$ *is obtained from $\mathcal{C}$ by adding the $\mathcal{L}_{\mathsf{M}}$-clause*

$$\top \rightarrow A_{[\![\alpha]\!]}(x) \ \vee \ A_{[\![\alpha^*]\!]}(x)$$

to $\mathcal{C}$ for each $A_{[\![\alpha]\!]} \in \mathsf{ran}(\mathsf{b})$ that occurs in $\mathcal{C}$ or $\mathcal{A}$, where $A_{[\![\alpha^*]\!]} = \mathsf{b}(\alpha^*)$ and α^* is the weak negation of α.

For an $\mathcal{L}_{\mathsf{M}}$-ontology $\mathcal{O}^{\mathsf{b}}$, the repleted clausification *is obtained from the clausification $(\mathcal{C}, \mathcal{A})$ of $\mathcal{O}^{\mathsf{b}}$ by replacing with its repletion.* ♢

Algorithm 2: Algorithm for checking consistency of $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$-ontology $\mathfrak{O}$ with Hypertableau

Input : $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$-ontology $\mathfrak{O}$
Output : true if $\mathfrak{O}$ is consistent, false otherwise

Preprocessing (results in $(\mathcal{C}, \mathcal{A})$):
1. Elimination of transitivity axioms, normalisation, clausification
2. Repletion of DL-clauses

```
Let (T, λ) be any derivation for (C, A)
𝔄 := {A′ | there exists a leaf node in (T, λ) that is labelled with A′}
for A′ ∈ 𝔄 do
    if A′ is clash-free then
        if 𝔒 contains rigid names then
            if (O_A′, R_O′) is consistent then
                return true
        else
            Let {c_1, ..., c_k} be the individuals occurring in A′
            if (O_c_i, R_O) is consistent for all 1 ≤ i ≤ k then
                return true
return false
```

In the proof of Lemma 5.13, we needed $\mathcal{H}^*$, which models $A_{[\![\alpha]\!]}(c)$ where $A_{[\![\alpha]\!]}(c) \notin \mathcal{A}'$ and $\mathcal{I}_i \models \mathcal{B}_{c_i} \wedge \alpha$. If $\mathcal{C}$ is repleted and $A_{[\![\alpha]\!]}(c) \notin \mathcal{A}'$, then we have $A_{[\![\alpha^*]\!]}(c) \in \mathcal{A}'$. Hence, we know that $\mathcal{I}_i \models \alpha^*$ and, due to Lemma 5.2, $\mathcal{I}_i \not\models \alpha$. Thus,$\mathcal{H}^*$ is not needed anymore.

To sum up, we refine Algorithm 1, so it can be used with the hypertableau calculus. The algorithm is depicted in Algorithm 2, where $\mathcal{O}_{c_i}$ is defined as in Definition 5.12. Instead of enumerating all models of $\mathfrak{O}^\mathsf{b}$ up to essential equality, we traverse the clash-free, complete ABoxes in a derivation of $(\mathcal{C}, \mathcal{A})$ and check whether there is an admissible ABox among them. To check admissibility, we reuse the results of Section 5.1. Thus, $\mathcal{O}_{\mathcal{A}'}$ and $\mathcal{R}_\mathsf{O}{}'$ are defined analogous to Definition 5.4. The next two lemmata show that Algorithm 2 is sound and complete.

Lemma 5.16 (Soundness). *Let $\mathfrak{O}$ be an $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$-ontology. If Algorithm 2 returns* false*, then $\mathfrak{O}$ is inconsistent.*

Proof. If Algorithm 2 returns false, then in any derivation, all complete ABoxes either contain a clash or are not admissible. Assume that $\mathfrak{O}$ is consistent. Then, by Lemma 5.6, we know that there is some M-interpretation $\mathcal{H}$ that models $(\mathcal{C}, \mathcal{A})$, where $\mathcal{Z}_\mathcal{H}$ is admissible. Due to Lemmata 5.11 and 5.13, there then is some complete, clash-free ABox $\mathcal{A}'$ which is admissible. This contradicts the assumption and hence, $\mathfrak{O}$ must be inconsistent. □

Lemma 5.17 (Completeness). *Let $\mathfrak{O}$ be an $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$-ontology. If Algorithm 2 returns* true*, then $\mathfrak{O}$ is consistent.*

Proof. If Algorithm 2 returns true, then there exists some clash-free, complete ABox $\mathcal{A}'$ in any derivation of $(\mathcal{C}, \mathcal{A})$ such that $\mathcal{A}'$ is admissible. Due to Lemma 5.13, there exists an M-interpretation $\mathcal{H}$ such that $\mathcal{H} \models \mathcal{A}'$ and $\mathcal{Z}_{\mathcal{H}}$ is admissible. If $\mathsf{b}([\![\alpha]\!])(c) \notin \mathcal{A}'$, then due to the repletion we know that $\mathsf{b}([\![\alpha^*]\!])(c) \in \mathcal{A}'$. Since $\alpha \wedge \alpha^*$ is inconsistent, we have that $\mathsf{b}([\![\alpha]\!]) \notin \mathsf{type}^{\mathcal{H}}_{\mathsf{ran(b)}}(c^{\mathcal{H}})$. Therefore, in the construction of $\mathcal{H}$ no additional assertions must be assumed and $\mathcal{H}$ also models $(\mathcal{C}, \mathcal{A})$. Hence, by Lemma 5.6, $\mathfrak{O}$ is consistent. □

However, for the repletion, a lot of disjunctive DL-clauses are added to $\mathcal{C}$ which increases the non-determinism tremendously. On closer inspection, we note that changing $\mathcal{H}$ to $\mathcal{H}^*$ in the proof of Lemma 5.13 is not dangerous in general. If $\mathcal{H}^*$ additionally models $A(c)$ with $A \in \mathsf{ran(b)}$ to ensure admissibility of $\mathcal{Z}_{\mathcal{H}}$, where $A(c) \notin \mathcal{A}'$, then it only becomes problematic for a DL-clause containing A in the antecedent. This is shown in Example 5.14. Conversely, we only have to introduce the repletion clauses for A if A appears in the antecedent of some DL-clause. Note here, that in the mapping of Chapter 4 only Axiom (4.11) contains, after clausification, an abstracted meta concept in the antecedent. Thus, if no compartment type plays any roles, then we do not need to add the repletion clauses to the ontology of the role-based model.

5.3 Implementing JConHT and Evaluation

We implemented Algorithm 2 in a reasoner called *JConHT* – a *J*ava-implemented *Con*text description logic reasoner based on *H*ermi*T*. Like HermiT, it is an OWL compliant reasoner implementing the OWLReasoner interface of the OWL API.

Nearly all reasoners use the Web Ontology Language (OWL) in order to represent an ontology. However, in general OWL cannot express contextualised knowledge. In [BGH+03], Bouquet et al. introduce C-OWL, an extension of the OWL syntax and semantics to allow for the representation of contextual ontologies. But their view differs significantly from our approach in what a context is. Therefore, we cannot use C-OWL.

But OWL has other means to enrich an ontology with more information due to *OWL annotations*. Annotations associate information with an ontology, for example name of the creator of the ontology. But also concept, role and individual names, as well as axioms, can be annotated. We use the outer abstraction of an ontology which is a 'normal' DL-ontology and define the connection between an abstracted concept name and the corresponding o-axiom via a special OWL annotation. We decided to use the predefined OWL annotation property `rdfs:isDefinedBy` for that purpose. We simply annotate the o-axiom α with the abstracted meta concept name $A_{[\![\alpha]\!]}$. Let us again consider Example 3.7:

Example 5.18. *Let $\mathcal{O}$ be the $\mathcal{ALC}[\![\mathcal{ALC}]\!]$-ontology*

$$\mathcal{O} := (C \sqsubseteq ([\![A \sqsubseteq \bot]\!]) \ \wedge\ (C \sqcap [\![A(a)]\!])(c),\ \emptyset,\ \emptyset).$$

Then $\mathcal{O}$ is presented in OWL 2 functional syntax as follows:

```
SubclassOf(cls:C cls:A_ASubBot)
ClassAssertion(ObjectIntersectionOf(cls:C cls:A_Aa) ind:c)
SubclassOf(Annotation(rdfs:isDefinedBy cls:A_ASubBot) cls:A owl:Bottom)
ClassAssertion(Annotation(rdfs:isDefinedBy cls:A_Aa) cls:A ind:a)
```
◇

Furthermore, in practical applications we often encounter object axioms that must hold independently of any context. These so-called *global object axioms* are of the form $\top \sqsubseteq [\![\alpha]\!]$. Using the above approach would introduce a new abstracted meta concept for each such axiom and, thus, unnecessarily bloat the OWL ontology. To avoid this, we decided to use another OWL annotation with a special meaning: `rdfs:label "objectGlobal"`. Finally, we need to specify which concept and role names are rigid. Since only the ontology as a whole and single axioms can be annotated, we use annotated declaration axioms. Declaration axioms do not affect the consequences of an OWL 2 ontology, they simply declare the existence of an entity and associate it with an entity type. We again use a special OWL annotation: `rdfs:label "rigid"`. Consider a light variation of example 5.18:

Example 5.19. *Let $\mathcal{O}$ be the $\mathcal{ALC}[\![\mathcal{ALC}]\!]$-ontology*

$$\mathcal{O} := (\top \sqsubseteq ([\![A \sqsubseteq \bot]\!]) \wedge [\![A(a)]\!](c), \emptyset, \emptyset).$$

with $A \in \mathsf{O}_{\mathsf{Crig}}$. Then $\mathcal{O}$ is represented in OWL 2 functional syntax as follows:

```
Declaration(Annotation(rdfs:label "rigid") Class (cls:A))
ClassAssertion(cls:A_Aa ind:c)
ClassAssertion(Annotation(rdfs:isDefinedBy cls:A_Aa) cls:A ind:a)
SubclassOf(Annotation(rdfs:label "objectGlobal") cls:A owl:Bottom)
```
◇

In Chapter 4, we presented a mapping algorithm that translates constrained Σ-compartment role object models into $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$-ontologies. We implemented this algorithm as part of the reference implementation of CROM[1]. The result of the mapping is an OWL ontology in Manchester OWL syntax [HP12] that uses the special annotations introduced above to encode $\mathcal{L}_\mathsf{M}[\![\mathcal{L}_\mathsf{O}]\!]$.

To evaluate the implementation of our reasoner, we performed several benchmarks to test the performance on ontologies that are based on CROM. Therefore, we used a CROM generator, which we explain below in detail, that randomly generates models based on a pseudorandom number generator. These models are then converted to OWL ontologies by the above mentioned mapping. We analyse the performance based on three scenarios:

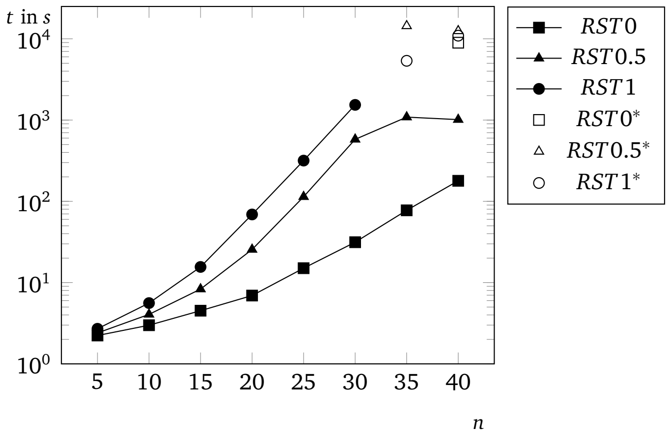
(I) variation of the number of relationship types defined in a compartment type,

(II) variation of the number of role groups defined in a compartment type, and

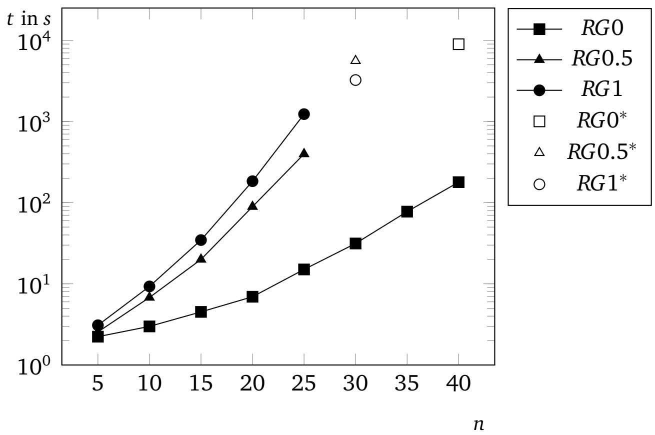
(III) variation of the types that are allowed to fill role types, i.e. whether compartments are allowed to play roles.

The main idea of the CROM generator is to randomly produce role-based models which abstracted and upscaled the banking example introduced in Chapter 4. In detail, the CROM generator works as follows. It has three input parameters. These are an integer n, which defines the size of the model, an integer s, which serves as the seed for the pseudorandom number generator, and a Boolean c which determines whether compartment types can fill role types. Any random choice is realised by a pseudorandom number generator based on seed s. Hence, we can reproduce every generated model. For a given n, the generator creates n compartment types, n natural types, and n^2 role types, i.e. n role types for each

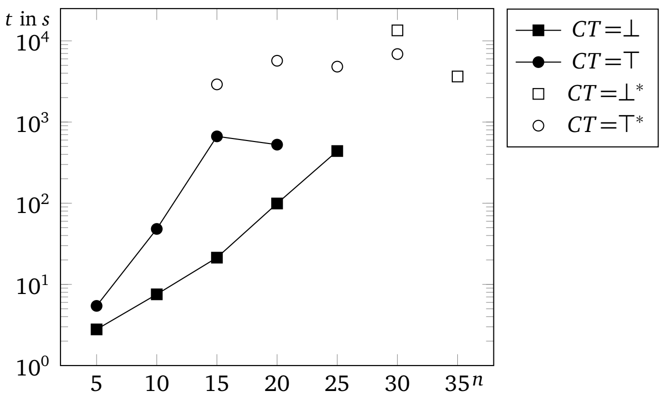
[1] https://github.com/Eden-06/CROM



(a) Scenario (I): Variation of number of relationship types.



(b) Scenario (II): Variation of number of role groups.



(c) Scenario (III): Variation of whether compartments can play roles.

Figure 5.1: Average execution times of JConHT for benchmark ontologies.

compartment type. Depending on the test series, it also generates 0, $n/2$ or n relationship types per compartment type. To generate fills, we randomly select two filler types for each role type. If c is true, these filler types can be any natural type or compartment type. If c is false, i.e. compartment types are not allowed to play roles, then the filler types can only be natural types. The parts-relation is determined by construction, since any role type is already assigned to one compartment type. For rel, we randomly pick two role types of the associated compartment type. Thus, we completely defined a Σ-CROM $\mathcal{M}$. Next, we define the constraint set.

We restricted the pairs of lower and upper bounds to $\{0..0, 0..1, 0..\infty, 1..1, 1..\infty\}$. To construct a role group within a compartment type, we randomly choose two role types of that compartment type and a pair of lower and upper bounds. For the occurrence constraints we assign one pair of lower and upper bound to each role type and each constructed role group. Similarly, for the cardinality constraints we assign two pairs of lower and upper bounds to each relationship type. In Section 4.1.4, we showed that our mapping can only support limited intra-relationship constraints which in turn also restrict the cardinality constraints for that relationship type. Therefore, we omit intra-relationship type constraints in our benchmark.

The basic setting for the different scenarios is that we do not have any relationship types or role groups, and that compartments are not allowed to fill roles. In Series (A) we define 0, $n/2$ or n relationship types per compartment type. Analogously, in Series (B) we construct 0, $n/2$ or n role groups per compartment type. For Scenario (III), we define $n/2$ relationship types and construct $n/2$ role groups. We then distinguish whether or not compartments can play roles. For each series, we start with $n = 5$ and increase n by steps of 5 until the reasoner throws out-of-memory exceptions. At each single configuration, we create 100 models with seeds from 1 to 100. We measure the time that the reasoner needs to decide consistency and calculate the average.

Our tests were conducted on a 64-bit Ubuntu 14-04 machine equipped with an Intel Core i5-2500 quad-core processor with a CPU clock rate of 3.3 GHz and 16 GB main memory. For the execution of the reasoner we used Java8 by OpenJDK and restricted the maximum Java heap size to 12GB. To measure the time needed by the reasoner to decide consistency, we used the shell builtin command `time` of Unix operation systems and measured the accumulated execution time.

Figure 5.1 shows the results of our tests. All diagrams show the average computation time that the reasoner needed to decide consistency. In some cases, the reasoner threw an out-of-memory exception and did not finish. We excluded these data from the average and plotted them separately. For all data sets, their starred version denotes the average time in the case the reasoner exited with an out-of-memory exception. If no such data point exists for a smaller n, then all ontologies could be processed without problems. Note here, that the time axis is logarithmic and that the reasoning time exponentially increases in the size of n, and thus in the size of the input ontology.

In Figure 5.1a, $RST0$ denotes the data set where no relationship types appear. In $RST0.5$, every model has $n/2$ relationship types and in $RST1$ there are n relationship types. Intuitively, the more relationship types and, hence, cardinality constraints a model contains, the harder it is to reason about. If only few constraints appear in the model, it is probable that the first branch of the derivation tree in the hypertableau algorithm already yields a consistent interpretation and no backtracking is needed. With more constraints, backtracking is needed more often, which increases the computation time significantly. An unsatisfiable role-based model is the most expensive to reason upon, since here complete backtracking is necessary before the algorithm can determine that the model is unsatisfiable. Interestingly, there is not much difference between $RST0.5$ and $RST1$. So, once relationship types are introduced, it is not relevant how many of them are defined. Furthermore, once out-of-memory exceptions appear, the reasoning time of the more difficult models is distorted since they would have needed more time if more resources would have been available. This explains the decline of $RST0.5$ for higher n.

The results for Scenario (II), shown in Figure 5.1b, are similar to Scenario (I). Analogously, $RG0$, $RG0.5$ and $RG1$ respectively denote the data sets where no, $n/2$ and n role groups appear in the role-based model. Again, when more constraints are introduced, it gets harder to decide satisfiability. However, it can be realised that introducing role groups is computation-wise costlier than relationship types with cardinality constraints.

In Figure 5.1c, $CT=\bot$ denotes the average reasoning time if no compartments are allowed to play roles. In $CT=\top$, compartments were allowed to play roles. If compartments are allowed to play roles, due to Axiom (4.11), we have to add the repletion clauses. These introduce a large amount of non-determinism. This explains why the reasoning time increases and out-of-memory exceptions appear even for smaller n.

In the end it is hard to tell whether these randomly generated role-based models give realistic test results. The tests clearly show that more constraints directly influence the computation time. Anyway, the application scope for role-based modelling is quite broad and these tests only give an impression for a general setting. However, a further examination of models within a specific topic can result in more target-oriented tests which might even initiate further specialised optimisations in the algorithm, the implementation or both.

Chapter 6

Conclusions

6.1 Major Contributions

In this thesis, we presented an overall workflow to reason on role-based models. Proper formalisation of contexts is crucial for role-based systems, but logical formalisms able to express these, easily tend to become undecidable. We introduced a novel family of description logics that is capable of expressing contextual knowledge, even in the presence of rigid roles, i.e. relational knowledge that is context-independent. For these contextualised description logics we did a thorough analysis on the complexity of the consistency problem, for which we investigated different settings depending on whether rigid role names or rigid concept names are admitted. We showed that for the least expressive setting, in which no rigid names are allowed, the complexity class of the consistency problem does not increase compared to the non-contextual version of that DL, namely the consistency problem is ExpTime-complete up to $\mathcal{SHOQ}[\![\mathcal{SHOQ}]\!]$ and NExpTime-complete for $\mathcal{SHOIQ}[\![\mathcal{SHOIQ}]\!]$. On the other hand, allowing rigid roles, which often causes undecidability in other approaches, only increases the complexity by one exponential. We also looked into a broad variety of description logics ranging from the lightweight DL $\mathcal{EL}$ up to the very expressive DL $\mathcal{SHOIQ}$ and, hence, obtained a nearly complete map of complexity results.

But the purpose of this thesis was not only to theoretically investigate a logical formalism capable of reasoning on role-based models. We also presented a mapping from the formal role-based modelling language CROM into contextualised DL ontologies. An implementation of this mapping is part of the CROM implementation. We proved that the formal semantics of the role-based model is preserved by the mapping and introduced further constraints that exceed the current capabilities of CROM.

Finally, we implemented a reasoner for our contextualised description logics that is based on the highly optimised existing DL reasoner HermiT [GHM+14]. During our analysis of the complexity of the consistency problem, we showed that deciding consistency could be split up into two subtasks. This idea was also used in our implementation. We further refined these subproblems, so they could be processed by HermiT. Due to the special form of the context ontology derived from CROM models, we could introduce a further optimisation step and showed its semantic correctness.

6.2 Future Work

As mentioned earlier, the focus of this thesis was on the overall workflow of modelling and reasoning about context-based domain models, but we also observe several linking points for future research. When investigating $\mathcal{L}_{\mathsf{M}}[\![\mathcal{L}_{\mathsf{O}}]\!]$, we mainly focused on the consistency problem, which was central for our goal. Besides that, query answering with context DLs would be an interesting direction for further investigations. Recently, there has been a lot of work in the area of temporal query answering. As our approach shares a similar setting, we are sure that there is plenty of motivation, applications and methods available to analyse contextualised query answering. Furthermore, there is current work on role-based databases [JKV+16], i.e. database systems that are based on a conceptual, role-based data model to natively represent complex data. Due to the close connection of query answering and database theory, it might be worth investigating query answering involving role-based database systems.

Going back to the consistency problem, we still think one can narrow the gap to undecidability. We added contextualised concepts to $\mathcal{L}_{\mathsf{M}}[\![\mathcal{L}_{\mathsf{O}}]\!]$ which results in undecidability in the presence of rigid roles. Quite certain, there are other, probably more restrictive, means to further extend the expressive power of the logic while preserving decidable.

Another extension to $\mathcal{L}_{\mathsf{M}}[\![\mathcal{L}_{\mathsf{O}}]\!]$ could be towards temporal logics. The combinations of DLs with temporal logics, point-based or interval, are well understood. As both temporal and contextualised DLs adopt a possible worlds semantics, it seems natural to also combine temporal logics with contextualised DLs. On a more abstract level, it might be even possible to analyse common properties of these combinations and deduce an abstract combination of DLs with itself or with other logics. Then temporal DLs or contextualised DLs could be an instance of that abstract combination.

The presented mapping for role-based models is based on the *Compartment Role Object Model (CROM)*. As CROM might be extended in the future, for example by new kinds of constraints, there is always some future work to analyse these upcoming features and to investigate whether they can also be represented by a contextualised DL ontology. Besides that, an investigation where contextualised DLs can be used except for CROM will help to detect any missing expressiveness of $\mathcal{L}_{\mathsf{M}}[\![\mathcal{L}_{\mathsf{O}}]\!]$ if existent.

The last starting point for future work would be the reasoner. While we used a black box approach, combined or integrated (tableaux) algorithms for deciding consistency are conceivable. A different optimisation would be an even more goal-oriented reasoner, which behaves especially well for contextualised ontologies produced from CROM models. Since, for example, the validation of role groups is rather of combinatorial nature which is quite hard for DL reasoners, it might be useful to use SAT solvers internally to improve overall performance.

Bibliography

[AN10] M. Aziz Ahmad and A. Nadeem: 'Consistency checking of UML models using Description Logics: A critical review'. In *Proc. of the 6th Int. Conf. on Emerging Technologies (ICET 2010)*, Islamabad, Pakistan. IEEE, Oct. 2010, pages 310–315 (cited on page 27).

[AF11] Vijay Alturi and David F. Ferraiolo: 'Role-Based Access Control'. In *Encyclopedia of Cryptography and Security, 2nd Ed.* Edited by Henk C. A. van Tilborg and Sushil Jajodia. Springer-Verlag, 2011, pages 1053–1055 (cited on page 1).

[BBL05] Franz Baader, Sebastian Brandt, and Carsten Lutz: 'Pushing the $\mathcal{EL}$ Envelope'. In *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005)*, Edinburgh, Scotland, UK. Edited by Leslie Pack Kaelbling and Alessandro Saffiotti. Morgan Kaufmann, Los Altos, Aug. 2005, pages 364–369 (cited on page 46).

[BCM+07] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors: *The Description Logic Handbook: Theory, Implementation, and Applications*. 2nd edition. Cambridge University Press, 2007 (cited on pages 2, 9).

[BGL08] Franz Baader, Silvio Ghilardi, and Carsten Lutz: 'LTL Over Description Logic Axioms'. In *Proc. of the 11th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2008)*, Sydney, Australia. Edited by Gerhard Brewka and Jérôme Lang. AAAI Press, Sept. 2008, pages 684–694 (cited on pages 4, 33, 36, 39–41, 43, 46–49, 71, 89).

[BGL12] Franz Baader, Silvio Ghilardi, and Carsten Lutz: 'LTL Over Description Logic Axioms'. In *ACM Transactions on Computational Logic* **13**(3): 2012. This is an extended version of [BGL08] (cited on pages 4, 33, 36, 39–41, 43, 46–49, 71).

[BKP12] Franz Baader, Martin Knechtel, and Rafael Peñaloza: 'Context-dependent views to axioms and consequences of Semantic Web ontologies'. In *Journal of Web Semantics* **12**: 2012, pages 22–40 (cited on page 28).

[BD77] Charles W. Bachman and Manilal Daya: 'The Role Concept in Data Models'. In *Proc. of the 3rd Int. Conf. on Very Large Data Bases (VLDB 1977)*, Tokyo, Japan. IEEE Computer Society, Oct. 1977, pages 464–476 (cited on page 1).

[BBT06] Matteo Baldoni, Guido Boella, and Leendert W. N. van der Torre: 'powerJava: Ontologically Founded Roles in Object Oriented Programming Languages'. In *Proc. of the 21st Annual ACM Symposium on Applied Computing (SAC 2006)*, Dijon, France. Edited by Hisham Haddad. ACM, Apr. 2006, pages 1414–1418 (cited on page 1).

[BGE07] Stephanie Balzer, Thomas R. Gross, and Patrick Eugster: 'A Relational Model of Object Collaborations and Its Use in Reasoning About Relationships'. In *Proc. of the 21st European Conf. on Object-Oriented Programming (ECOOP 2007)*, Berlin, Germany. Edited by Erik Ernst. Volume 4609. Lecture Notes in Computer Science. Springer-Verlag, July 2007, pages 323–346 (cited on page 1).

[BVS+09] Jie Bao, George Voutsadakis, Giora Slutzki, and Vasant Honavar: 'Package-Based Description Logics'. In *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*. Edited by Heiner Stuckenschmidt, Christine Parent, and Stefano Spaccapietra. Volume 5445. Lecture Notes in Computer Science. Springer-Verlag, 2009. Chapter 13, pages 349–371 (cited on page 28).

[BAF+06] Djamal Benslimane, Ahmed Arara, Gilles Falquet, Zakaria Maamar, Philippe Thiran, and Faïez Gargouri: 'Contextual Ontologies'. In *Proc. of the 4th Int. Conf. on Advances in Information Systems (ADVIS 2006)*, Izmir, Turkey. Edited by Tatyana M. Yakhno and Erich J. Neuhold. Volume 4243. Lecture Notes in Computer Science. Springer-Verlag, Oct. 2006, pages 168–176 (cited on page 28).

[BCD05] Daniela Berardi, Diego Calvanese, and Giuseppe De Giacomo: 'Reasoning on UML class diagrams'. In *Artificial Intelligence* **168**(1-2): 2005, pages 70–118 (cited on page 27).

[Ber66] Robert Berger: *The Undecidability of the Domino Problem*. Volume 66. Memoirs of the AMS. American Mathematical Society, 1966 (cited on page 51).

[BK17] Stephan Böhme and Thomas Kühn: 'Reasoning on Context-Dependent Domain Models'. In *Proc. of the 7th Joint Int. Conf. on Semantic Technology (JIST 2017)*, Gold Coast, QLD, Australia. Edited by Zhe Wang, Anni-Yasmin Turhan, Kewen Wang, and Xiaowang Zhang. Volume 10675. Lecture Notes in Computer Science. Springer-Verlag, Nov. 2017, pages 69–85 (cited on pages 6, 7).

[BL15a] Stephan Böhme and Marcel Lippmann: 'Decidable Contextualized DLs with Rigid Roles'. In *Proc. of the 28th Int. Workshop on Description Logics (DL 2015)*, Athens, Greece. Edited by Diego Calvanese and Boris Konev. Volume 1350. CEUR Workshop Proceedings. `CEUR-WS.org`, June 2015, pages 92–95 (cited on page 6).

[BL15b] Stephan Böhme and Marcel Lippmann: 'Decidable Description Logics of Context with Rigid Roles'. In *Proc. of the 10th Int. Symp. on Frontiers of Combining Systems (FroCoS 2015)*, Wroclaw, Poland. Edited by Carsten Lutz and Silvio Ranise. Volume 9322. Lecture Notes in Artificial Intelligence. Springer-Verlag, Sept. 2015, pages 17–32 (cited on page 6).

[BL15c] Stephan Böhme and Marcel Lippmann: 'Description Logics of Context with Rigid Roles Revisited'. LTCS-Report 15-04. See `http://lat.inf.tu-dresden.de/research/reports.html`. Chair of Automata Theory, TU Dresden, 2015 (cited on page 6).

[BT15a] Stefan Borgwardt and Veronika Thost: 'LTL over $\mathcal{EL}$ Axioms'. LTCS-Report 15-07. Chair of Automata Theory, Institute of Theoretical Computer Science, Technische Universität Dresden, 2015. URL: `http://lat.inf.tu-dresden.de/research/reports.html` (cited on pages 43, 44).

[BT15b] Stefan Borgwardt and Veronika Thost: 'Temporal Query Answering in the Description Logic $\mathcal{EL}$'. In *Proc. of the 24th Int. Joint Conf. on Artificial Intelligence (IJCAI 2015)*, Buenos Aires, Argentina. Edited by Qiang Yang and Michael Wooldridge. AAAI Press, 2015, pages 2819–2825 (cited on pages 43, 44).

[BGH+03] Paolo Bouquet, Fausto Giunchiglia, Frank van Harmelen, Luciano Serafini, and Heiner Stuckenschmidt: 'C-OWL: Contextualizing Ontologies'. In *The Semantic Web - ISWC 2003, Second International Semantic Web Conference, Sanibel Island, FL, USA, October 20-23, 2003, Proceedings*, Sanibel Island, FL, USA. Edited by Dieter Fensel, Katia P. Sycara, and John Mylopoulos. Volume 2870. Lecture Notes in Computer Science. Springer-Verlag, Oct. 2003, pages 164–179 (cited on pages 28, 81).

[BGH+04] Paolo Bouquet, Fausto Giunchiglia, Frank van Harmelen, Luciano Serafini, and Heiner Stuckenschmidt: 'Contextualizing ontologies'. In *Journal of Web Semantics* **1**(4): 2004, pages 325–343 (cited on page 28).

[CCD+02] Andrea Calì, Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini: 'A Formal Framework for Reasoning on UML Class Diagrams'. In *Proc. of the 13th Int. Symp. on Foundations of Intelligent Systems (ISMIS 2002)*, Lyon, France. Edited by Mohand-Said Hacid, Zbigniew W. Ras, Djamel A. Zighed, and Yves Kodratoff. Volume 2366. Lecture Notes in Computer Science. Springer-Verlag, June 2002, pages 503–513 (cited on page 27).

[CP14] Ismail Ilkan Ceylan and Rafael Peñaloza: 'The Bayesian Description Logic BEL'. In *Proc. of the 7th Int. Joint Conf. on Automated Reasoning (IJCAR 2014)*. Volume 8562. LNCS. Springer-Verlag, 2014, pages 480–494 (cited on page 28).

[CP17] Ismail Ilkan Ceylan and Rafael Peñaloza: 'The Bayesian Ontology Language BEL'. In *Journal of Automated Reasoning* **58**(1): 2017, pages 67–95 (cited on page 28).

[Dij82] Edsger W. Dijkstra: 'On the role of scientific thought'. In *Selected Writings on Computing: A Personal Perspective*. Springer-Verlag, 1982, pages 60–66 (cited on page 1).

[Eva98] Andy Evans: 'Reasoning with UML class diagrams'. In *Proc. of the 2nd IEEE Workshop on Industrial Strength Formal Specification Techniques (WIFT 1998)*, Boca Raton, FL, USA. IEEE Computer Society, Oct. 1998, pages 102–113 (cited on page 27).

[FKC03] David F. Ferraiolo, D. Richard Kuhn, and Ramaswamy Chandramouli: *Role-based Access Control*. Artech House, 2003 (cited on page 1).

[FEL+98] Robert B. France, Andy Evans, Kevin Lano, and Bernhard Rumpe: 'The UML as a formal modeling notation'. In *Computer Standards & Interfaces* **19**(7): 1998, pages 325–334 (cited on page 27).

[GKW+03] Dov M. Gabbay, Ágnes Kurucz, Frank Wolter, and Michael Zakharyaschev: *Many-Dimensional Modal Logics: Theory and Applications*. Elsevier Science, 2003 (cited on page 31).

[GHM+14] Birte Glimm, Ian Horrocks, Boris Motik, Giorgos Stoilos, and Zhe Wang: 'HermiT: An OWL 2 Reasoner'. In *Journal of Automated Reasoning* **53**(3): 2014, pages 245–269 (cited on pages 74, 87).

[GW00a] Nicola Guarino and Christopher A. Welty: 'A Formal Ontology of Properties'. In *Proc. of the 12th Int. Conf. on Knowledge Acquisition, Modeling and Management (EKAW)*, Juan-les-Pins, France. Edited by Rose Dieng and Olivier Corby. Volume 1937. Lecture Notes in Computer Science. Springer-Verlag, Oct. 2000, pages 97–112 (cited on page 15).

[GW00b] Nicola Guarino and Christopher A. Welty: 'Identity, Unity, and Individuality: Towards a Formal Toolkit for Ontological Analysis'. In *Proc. of the 14th European Conf. on Artificial Intelligence (ECAI 2000)*, Berlin, Germany. Edited by Werner Horn. IOS Press, Aug. 2000, pages 219–223 (cited on page 15).

[GW00c] Nicola Guarino and Christopher A. Welty: 'Ontological Analysis of Taxonomic Relationships'. In *Proc. of the 19th Int. Conf. on Conceptual Modeling (ER 2000)*, Salt Lake City, UT, USA. Edited by Alberto H. F. Laender, Stephen W. Liddle, and Veda C. Storey. Volume 1920. Lecture Notes in Computer Science. Springer-Verlag, Oct. 2000, pages 210–224 (cited on page 15).

[GW09] Nicola Guarino and Christopher A. Welty: 'An Overview of OntoClean'. In *Handbook on Ontologies*. Edited by Steffen Staab and Rudi Studer. 2nd edition. International Handbooks on Information Systems. Springer-Verlag, 2009, pages 201–220 (cited on page 15).

[Gui05] Giancarlo Guizzardi: 'Ontological Foundations for Structural Conceptual Models'. PhD thesis. Enschede, Netherlands: University of Twente, 2005 (cited on page 1).

[Hal06] Terry A. Halpin: 'Object-role modeling (ORM/NIAM)'. In *Handbook on Architectures of Information Systems*. Edited by Peter Bernus, Kai Mertins, and Günter Schmidt. 2nd edition. Springer-Verlag, 2006. Chapter 4, pages 81–103 (cited on page 1).

[Her07] Stephan Herrmann: 'A Precise Model for Contextual Roles: The Programming Language ObjectTeams/Java'. In *Applied Ontology* **2**(2): 2007, pages 181–207 (cited on page 1).

[HB11] Matthew Horridge and Sean Bechhofer: 'The OWL API: A Java API for OWL ontologies'. In *Semantic Web* **2**(1): 2011, pages 11–21 (cited on page 74).

[HP12] Matthew Horridge and Peter Patel-Schneider: 'OWL 2 Web Ontology Language Manchester Syntax (Second Edition)'. W3C Note. `http://www.w3.org/TR/2012/NOTE-owl2-manchester-syntax-20121211/`. W3C, Dec. 2012 (cited on page 82).

[HST00] Ian Horrocks, Ulrike Sattler, and Stephan Tobies: 'Practical Reasoning for Very Expressive Description Logics'. In *Journal of the Interest Group in Pure and Applied Logic* **8**(3): 2000, pages 239–263 (cited on page 13).

[JKV+16] Tobias Jäkel, Thomas Kühn, Hannes Voigt, and Wolfgang Lehner: 'Towards a Role-Based Contextual Database'. In *Proc. og the 20th East European Conf. on Advances in Databases and Information Systems (ADBIS 2016)*, Prague, Czech Republic. Edited by Jaroslav Pokorný, Mirjana Ivanovic, Bernhard Thalheim, and Petr Saloun. Volume 9809. Lecture Notes in Computer Science. Springer-Verlag, Aug. 2016, pages 89–103 (cited on page 88).

[KG10] Szymon Klarman and Víctor Gutiérrez-Basulto: '$\mathcal{ALC}_{\mathcal{ALC}}$: A Context Description Logic'. In *Proceedings of the 12th European Conference on Logics in Artificial Intelligence (JELIA 2010)*. Edited by Tomi Janhunen and Ilkka Niemelä. Volume 6341. Lecture Notes in Computer Science. Springer, 2010, pages 208–220 (cited on pages 3, 28, 33, 49, 51).

[KG11a] Szymon Klarman and Víctor Gutiérrez-Basulto: 'Two-Dimensional Description Logics for Context-Based Semantic Interoperability'. In *Proc. of the 25th AAAI Conf. on Artificial Intelligence (AAAI 2011)*, San Francisco, CA, USA. Edited by Wolfram Burgard and Dan Roth. AAAI Press, Aug. 2011 (cited on page 3).

[KG11b] Szymon Klarman and Víctor Gutiérrez-Basulto: 'Two-Dimensional Description Logics of Context'. In *Proceedings of the 24th International Workshop on Description Logics (DL 2011)*. Edited by Riccardo Rosati, Sebastian Rudolph, and Michael Zakharyaschev. Volume 745. CEUR-WS.org, 2011 (cited on page 3).

[KG16] Szymon Klarman and Víctor Gutiérrez-Basulto: 'Description logics of context'. In *Journal of Logic and Computation* **26**(3): 2016, pages 817–854 (cited on pages 3, 28, 33, 51, 54).

[KBG+15] Thomas Kühn, Stephan Böhme, Sebastian Götz, and Uwe Aßmann: 'A combined formal model for relational context-dependent roles'. In *Proc. of the 8th ACM SIGPLAN Int. Conf. on Software Language Engineering (SLE 2015)*, Pittsburgh, PA, USA. Edited by Richard F. Paige, Davide Di Ruscio, and Markus Völter. ACM, Oct. 2015, pages 113–124 (cited on pages 5, 6, 15, 17, 18, 21, 24).

[KLG+14] Thomas Kühn, Max Leuthäuser, Sebastian Götz, Christoph Seidl, and Uwe Aßmann: 'A Metamodel Family for Role-Based Modeling and Programming Languages'. In *Proc. of the 7th Int. Conf. on Software Language Engineering (SLE 2014)*, Västerås, Sweden. Edited by Benoît Combemale, David J. Pearce, Olivier Barais, and Jurgen J. Vinju. Volume 8706. Lecture Notes in Computer Science. Springer-Verlag, Sept. 2014, pages 141–160 (cited on pages 5, 15, 16, 66).

[Lip14] Marcel Lippmann: 'Temporalised Description Logics for Monitoring Partially Observable Events'. PhD thesis. Dresden, Germany: TU Dresden, 2014 (cited on pages 4, 14, 33, 38–42).

[Lut02a] Carsten Lutz: 'Description Logics with Concrete Domains—A Survey'. In *Proc. of the 4th Conf. on Advances in Modal Logic (AiML 2002)*, Toulouse, France. Edited by Philippe Balbiani, Nobu-Yuki Suzuki, Frank Wolter, and Michael Zakharyaschev. King's College Publications, Oct. 2002, pages 265–296 (cited on page 67).

[Lut02b] Carsten Lutz: 'The complexity of description logics with concrete domains'. PhD thesis. Aachen, Germany: RWTH Aachen, 2002 (cited on page 67).

[LWZ08] Carsten Lutz, Frank Wolter, and Michael Zakharyaschev: 'Temporal Description Logics: A Survey'. In *Proc. of the 15th Int. Symp. on Temporal Representation and Reasoning (TIME 2008)*, Montréal, Canada. Edited by Stéphane Demri and Christian S. Jensen. IEEE Press, June 2008, pages 3–14 (cited on pages 28, 31, 49–51).

[MLH+15] Nicolas Matentzoglu, Jared Leo, Valentino Hudhra, Uli Sattler, and Bijan Parsia: 'A Survey of Current, Stand-alone OWL Reasoners'. In *Informal Proc. of the 4th Int. Workshop on OWL Reasoner Evaluation (ORE 2015) co-located with the 28th Int. Workshop on Description Logics (DL 2015)*, Athens, Greece. Edited by Michel Dumontier, Birte Glimm, Rafael S. Gonçalves, Matthew Horridge, Ernesto Jiménez-Ruiz, Nicolas Matentzoglu, Bijan Parsia, Giorgos B. Stamou, and Giorgos Stoilos. Volume 1387. CEUR Workshop Proceedings. `CEUR-WS.org`, June 2015, pages 68–79 (cited on page 73).

[McC87] John McCarthy: 'Generality in Artificial Intelligence'. In *Communications of the ACM* **30**(12): 1987. Turing award lecture, pages 1030–1035 (cited on page 28).

[McC93] John McCarthy: 'Notes on Formalizing Context'. In *Proc. of the 13th Int. Joint Conf. on Artificial Intelligence (IJCAI 1993)*, Chambéry, France. Edited by Ruzena Bajcsy. Morgan Kaufmann, Aug. 1993, pages 555–562 (cited on page 28).

[MSH07a] Boris Motik, Rob Shearer, and Ian Horrocks: 'A Hypertableau Calculus for SHIQ'. In *Proc. of the 20th Int. Workshop on Description Logics (DL 2007)*, Brixen-Bressanone, Italy. Edited by Diego Calvanese, Enrico Franconi, Volker Haarslev, Domenico Lembo, Boris Motik, Anni-Yasmin Turhan, and Sergio Tessaris. Volume 250. CEUR Workshop Proceedings. `CEUR-WS.org`, June 2007 (cited on page 74).

[MSH07b] Boris Motik, Rob Shearer, and Ian Horrocks: 'Optimized Reasoning in Description Logics Using Hypertableaux'. In *Proc. of the 21st Int. Conf. on Automated Deduction (CADE-21)*, Bremen, Germany. Edited by Frank Pfenning. Volume 4603. Lecture Notes in Computer Science. Springer-Verlag, July 2007, pages 67–83 (cited on page 74).

[MSH08] Boris Motik, Rob Shearer, and Ian Horrocks: 'Optimizing the Nominal Introduction Rule in (Hyper)Tableau Calculi'. In *Proc. of the 21st Int. Workshop on Description Logics (DL 2008)*, Dresden, Germany. Edited by Franz Baader, Carsten Lutz, and Boris Motik. Volume 353. CEUR Workshop Proceedings. `CEUR-WS.org`, May 2008 (cited on page 77).

[MSH09] Boris Motik, Rob Shearer, and Ian Horrocks: 'Hypertableau Reasoning for Description Logics'. In *Journal of Artificial Intelligence Research* **36**: 2009, pages 165–228 (cited on pages 74–76, 78).

[PMG+15] Bijan Parsia, Nicolas Matentzoglu, Rafael S. Gonçalves, Birte Glimm, and Andreas Steigmiller: 'The OWL Reasoner Evaluation (ORE) 2015 Competition Report'. In *Proc. of the 11th Int. Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS 2015) co-located with 14th Int. Semantic Web Conference (ISWC 2015)*, Bethlehem, PA, USA. Edited by Thorsten Liebig and Achille Fokoue. Volume 1457. CEUR Workshop Proceedings. `CEUR-WS.org`, Oct. 2015, pages 2–15 (cited on page 74).

[Pra05] Ian Pratt-Hartmann: 'Complexity of the Two-Variable Fragment with Counting Quantifiers'. In *Journal of Logic, Language and Information* **14**(3): 2005, pages 369–395 (cited on pages 14, 38).

[RG98] Dirk Riehle and Thomas Gross: 'Role Model Based Framework Design and Integration'. In *Proc. of the 13th ACM SIGPLAN Conf. on Object-oriented Programming, Systems, Languages, and Applications (OOPSLA98)*, Vancouver, British Columbia, Canada. Edited by Bjørn N. Freeman-Benson and Craig Chambers. ACM, Oct. 1998, pages 117–133 (cited on page 22).

[SCF+96] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman: 'Role-Based Access Control Models'. In *IEEE Computer* **29**(2): Feb. 1996, pages 38–47 (cited on page 1).

[Sch91] Klaus Schild: 'A Correspondence Theory for Terminological Logics: Preliminary Report'. In *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI 1991)*, Sydney, Australia. Edited by John Mylopoulos and Raymond Reiter. Morgan Kaufmann, Aug. 1991, pages 466–471 (cited on pages 14, 43, 46).

[Sch89] Manfred Schmidt-Schauß: 'Subsumption in KL-ONE is Undecidable'. In *Proc. of the 1st Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 1989)*, Toronto, Canada. Edited by Ronald J. Brachman, Hector J. Levesque, and Raymond Reiter. Morgan Kaufmann, May 1989, pages 421–431 (cited on page 56).

[SS91] Manfred Schmidt-Schauß and Gert Smolka: 'Attributive Concept Descriptions with Complements'. In *Artificial Intelligence* **48**(1): 1991, pages 1–26 (cited on page 13).

[SH12] Luciano Serafini and Martin Homola: 'Contextualized knowledge repositories for the Semantic Web'. In *Journal of Web Semantics* **12**: 2012, pages 64–87 (cited on page 28).

[SB05] Quan Z. Sheng and Boualem Benatallah: 'ContextUML: A UML-Based Modeling Language for Model-Driven Development of Context-Aware Web Services'. In *Proc. of the 4th Int. Conf. on Mobile Business (ICMB 2005)*, Sydney, Australia. IEEE Computer Society, July 2005, pages 206–212 (cited on page 27).

[SBH+08] Jocelyn Simmonds, M. Cecilia Bastarrica, Nancy Hitschfeld-Kahler, and Sebastián Rivas: 'A Tool Based on DL for UML Model Consistency Checking'. In *International Journal of Software Engineering and Knowledge Engineering* **18**(6): 2008, pages 713–735 (cited on page 27).

[SSJ+04] Jocelyn Simmonds, Ragnhild Van Der Straeten, Viviane Jonckers, and Tom Mens: 'Maintaining Consistency between UML Models Using Description Logic'. In *L'Objet* **10**(2-3): 2004, pages 231–244 (cited on page 27).

[SLG14] Andreas Steigmiller, Thorsten Liebig, and Birte Glimm: 'Konclude: System description'. In *Journal of Web Semantics* **27**: 2014, pages 78–85 (cited on page 74).

[Ste00] Friedrich Steimann: 'On the representation of roles in object-oriented and conceptual modelling'. In *Data & Knowledge Engineering* **35**(1): Oct. 2000, pages 83–106 (cited on pages 1, 15, 16).

[Ste07] Friedrich Steimann: 'The Role Data Model Revisited'. In *Applied Ontology* **2**(2): 2007, pages 89–103 (cited on page 1).

[SMS+03] Ragnhild Van Der Straeten, Tom Mens, Jocelyn Simmonds, and Viviane Jonckers: 'Using Description Logic to Maintain Consistency between UML Models'. In *Proc. of the 6th Int. Conf. on the Unified Modeling Language, Modeling Languages and Applications (UML 2003)*, San Francisco, CA, USA. Edited by Perdita Stevens, Jon Whittle, and Grady Booch. Volume 2863. Lecture Notes in Computer Science. Springer-Verlag, Oct. 2003, pages 326–340 (cited on page 27).

[Tob00] Stephan Tobies: 'The Complexity of Reasoning with Cardinality Restrictions and Nominals in Expressive Description Logics'. In *Journal of Artificial Intelligence Research* **12**: 2000, pages 199–217 (cited on pages 14, 43, 46).

[Tob01] Stephan Tobies: 'Complexity Results and Practical Algorithms for Logics in Knowledge Representation'. PhD thesis. Aachen, Germany: RWTH Aachen, 2001 (cited on page 14).

[WG01] Christopher A. Welty and Nicola Guarino: 'Supporting ontological analysis of taxonomic relationships'. In *Data & Knowledge Engineering* **39**(1): 2001, pages 51–74 (cited on page 15).

[WS92] William A. Woods and James G. Schmolze: 'The KL-ONE Family'. In *Computers & Mathematics with Applications* **23**(2-5): 1992, pages 133–177 (cited on page 2).