Temporalised Description Logics for Monitoring Partially Observable Events

Dissertation

zur Erlangung des akademischen Grades
Doktor rerum naturalium (Dr. rer. nat.)

vorgelegt an der
Technischen Universität Dresden
Fakultät Informatik

eingereicht von
Dipl.-Inf. Marcel Lippmann
geboren am 11. April 1985 in Dippoldiswalde

verteidigt am 1. Juli 2014

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

Prof. Dr. rer. nat. habil. Frank Wolter
University of Liverpool

Dresden, im Juli 2014

Contents

Chapter 1

Introduction

In this day and age, it is not possible to imagine our world without complex hardware and software systems. Inevitably, it becomes more and more important to verify that the systems that surround us have certain properties. This is indeed unavoidable for safety-critical systems such as power plants and intensive-care units. Throughout this thesis, we refer to the term 'system' in a broad sense: it may be a 'man-made system' (e.g. a computer system) or a 'natural system' (e.g. a patient in an intensive-care unit).

Model Checking [CGP99; BK08] is a prominent field of research that addresses these issues. However, there it is assumed that one has complete knowledge about the functioning of the system, which is not always a reasonable assumption. In the present thesis, we consider an *open-world scenario*. Instead of having a complete model of the system, we assume that we can only observe the behaviour of the actual running system by 'sensors'. Such an abstract sensor could, for instance, sense the blood pressure of a patient or the air traffic observed by radar. Then the observed data are preprocessed appropriately and stored in a *fact base*. Based on the data available in the fact base, situation-awareness tools [BBB+09; End95] are supposed to help the user to detect certain situations, e.g. situations that require intervention by an expert. Such situations could be, for instance, that the heart-rate of a patient is rather high while the blood pressure is low, or that a collision of two aeroplanes is about to happen. Such critical situations can be overcome by reacting accordingly, e.g. giving an appropriate medication to the patient and informing the pilots of the aeroplanes, respectively. Moreover, the information in the fact base can be used by *monitors* to verify that the system has certain properties. Such a property could be, for instance, that nothing 'bad' will happen, i.e. a so-called *safety property*.

It is not realistic, however, to assume that the sensors always yield a complete description of the current state of the observed system. Thus, it makes sense to assume that information that is not present in the fact base is unknown rather than to assume that this information does not hold, which we call the *open-world assumption*. Moreover, very often one has *some* knowledge about the functioning of the system. This *background knowledge* can be used to draw conclusions about the possible future behaviour of the system.

Employing description logics [BCM+07] is one way to deal with these requirements. In this thesis, we tackle the sketched problem in three different contexts: (i) runtime verification using a temporal extension of a description logic, (ii) temporalised query entailment, and (iii) verification in action formalisms based on description logics.

In the remainder of this chapter, we give an abstract overview of the present work. In Section 1.1, we provide the reader with an intuitive understanding of basic notions in description logics. Then in Section 1.2, we go one step further and consider temporal extensions of description logics. After that in Section 1.3, we give details about Context (i), i.e. runtime verification using a temporalised description logic. Section 1.4 deals with Context (ii), i.e. temporalised query entailment, and Section 1.5 respectively deals with Context (iii), i.e. verification in the research area of action formalisms that are based on description logics. Finally, Section 1.6 contains an outline of the present thesis and summarises the main contributions.

1.1 Description Logics

Description logics (DLs) [BCM+07] are a family of logic-based knowledge representation formalisms. Since they are logic-based, DLs have the advantage of being equipped with a formal semantics, which lacks in early knowledge representation formalisms such as Quillian's *Semantic Networks* [Qui67] and Minsky's *Frames* [Min81]. More information about the history of DLs can be found in the *Description Logic Handbook* [BCM+07].

Each DL is defined using a set of *concept names*, a set of *role names*, and a set of *individual names*. These names are used to express knowledge in the respective application domain. In the following, we consider a zoo as application domain in order to be able to give simple examples that are easy to comprehend. In principle, concept names describe simple properties of elements in the domain. For example, the concept name Camel can be used to describe all camels (domain elements) in a zoo (domain). Role names, e.g. $\mathsf{is\text{-}father\text{-}of}$, describe binary relations between domain elements. Finally, individual names give names to specific domain elements, e.g. leah can be used as a name of a specific camel in the zoo. These sets are used by *concept* and *role constructors* to form complex *concepts* and *roles*. Which concept and role constructors are available depends on the specific DL.

The smallest propositionally closed description logic is called $\mathcal{ALC}$ [SS91]. The name $\mathcal{ALC}$ stands for 'attributive language with complements'. There are, however, DLs that are less expressive than $\mathcal{ALC}$ such as $\mathcal{EL}$ [Baa03; Bra04] and extensions of $\mathcal{EL}$ [BBL05; BBL08] for which standard reasoning problems are tractable. With different reasoning problems in mind, other light-weight DLs such as members of the *DL-Lite* family [CDL+05; ACK+09; CDL+09] have been developed. On the other hand, there are also DLs whose expressive power goes far beyond $\mathcal{ALC}$ such as $\mathcal{SROIQ}$ [HKS06].

As an example of a concept that is expressible in $\mathcal{ALC}$, consider

$$\neg\mathsf{Dromedary} \sqcap \exists\mathsf{likes}.\mathsf{Foliage},$$

where $\mathsf{Dromedary}$ and $\mathsf{Foliage}$ are concept names, and likes is a role name. Intuitively, this concept describes all domain elements that are not dromedaries and like foliage.

Besides the logic-based semantics, another advantage of DLs is that most of them can be seen as *decidable* fragments of first-order predicate logic (FOL), which are still more expressive than propositional logic. Hence, important reasoning problems such as concept satisfiability are decidable for DLs. Concepts formulated in $\mathcal{ALC}$, for instance, can be translated into formulas of the two-variable fragment of FOL. The concept above can be formulated as

$$\neg\mathsf{dromedary}(x) \wedge \exists y.\big(\mathsf{likes}(x,y) \wedge \mathsf{foliage}(y)\big),$$

where $\mathsf{dromedary}$ and $\mathsf{foliage}$ are unary predicates, and likes is a binary predicate. In this FOL-formula, the free variable x captures all domain elements that are not dromedaries and like foliage.

There is also a close connection between description logics and modal logics. In fact, $\mathcal{ALC}$ can be seen as a notational variant of the multi-modal logic $\boldsymbol{K}^n$ (see e.g. [GKW+03; BBW07]). For instance, the above concept can be expressed in $\boldsymbol{K}^n$ as

$$\neg\mathsf{dromedary} \wedge \Diamond_{\mathsf{likes}}\mathsf{foliage},$$

where $\mathsf{dromedary}$ and $\mathsf{foliage}$ are propositional variables, and $\Diamond_{\mathsf{likes}}$ is a modal operator.

However, description logics do not only offer a language for describing concepts, but allow to state knowledge in so-called *knowledge bases*, which are split into an assertional part (the *ABox*) and a terminological part (the *TBox*). The ABox consists of a finite sets of *ABox-axioms* (or *assertions*) such as concept assertions and role assertions. For instance, consider the ABox

$$\big\{(\neg\mathsf{Dromedary} \sqcap \exists\mathsf{likes}.\mathsf{Foliage})(\mathsf{leah}),\ \mathsf{is\text{-}father\text{-}of}(\mathsf{hassan}, \mathsf{leah})\big\},$$

which states that Leah is not a dromedary and likes foliage, and that Hassan is Leah's father. The following TBox captures the terminological knowledge:

$$\big\{\neg\mathsf{Dromedary} \sqcap \exists\mathsf{likes}.\mathsf{Foliage} \sqsubseteq \mathsf{NiceCamel}\big\}.$$

It states that each domain element which is not a dromedary and likes foliage is actually a nice camel.

Interesting reasoning problems for such knowledge bases are, for instance, consistency and entailment, i.e. the question whether such a knowledge base has a model and whether it entails certain implicit facts. The formal definition of the notions introduced intuitively above that are relevant for this thesis can be found in Section 2.1.

Description logics are successfully employed in many areas such as natural-language processing, conceptual modelling, and databases. The most notable success, however, has been achieved lately by adopting the *Web Ontology Language (OWL)*, which is based on an expressive DL, as standard language for the semantic web [HPH03]. Recently, the second refinement OWL 2 has been endorsed by the World Wide Web Consortium (W3C) as W3C Recommendation.[1] Additionally, tractable description logics are successfully employed for defining medical ontologies, see e.g. [SBS+07].

However, DLs are not expressive enough to describe the temporal behaviour of systems. Therefore, description logics have been 'temporalised', i.e. equipped with temporal operators. The next section gives a brief overview on temporalised DLs.

1.2 Temporalised Description Logics

In the literature, a plethora of temporalised description logics has been introduced (surveyed e.g. in [AF00; AF05; LWZ08]). To obtain a temporalised DL, one combines a DL with a

[1] See `http://www.w3.org/TR/owl2-overview/`.

temporal logic.[2] For a comprehensive introduction to temporal logics, the reader is referred to e.g. [Eme90; GHR94; BK08]. In *linear-time* temporal logics, the flow of time is assumed to be linear, i.e. each point in time has exactly one successor, whereas in *branching-time* temporal logics, each point in time may have more than one successor, i.e. the flow of time is assumed to be a tree. A well-investigated temporal logic with linear flow of time is *linear-time temporal logic (LTL)* [Pnu77], and logics with branching flow of time include *computation-tree logic (CTL)* [CE82] and its extension *CTL** [EH86]. It is a long-standing debate whether linear-time temporal logics or branching-time temporal logics should be adopted, as both have strengths and weaknesses in terms of expressiveness and computational complexity of reasoning [Var01; NV07].

In this thesis, we consider only combinations of LTL with description logics.[3] The first linear-time temporalised DL, called $\mathrm{LTL}_{\mathcal{ALC}}$, was introduced by Schild [Sch93b]. This temporalised DL is a combination of LTL with the description logic $\mathcal{ALC}$. In $\mathrm{LTL}_{\mathcal{ALC}}$, concepts are built using concept constructors *and* temporal operators, i.e. temporal operators are allowed to occur within concepts. For instance,

$$\neg\mathsf{Dromedary} \sqcap \Diamond\exists\mathsf{likes}.\mathsf{Foliage}$$

is a concept formulated in $\mathrm{LTL}_{\mathcal{ALC}}$. The semantics of $\mathrm{LTL}_{\mathcal{ALC}}$ is two-dimensional, i.e. one dimension describes the flow of time and a second dimension is used for the domains. Thus, the above concept captures all domain elements which are not dromedaries (now), and will like foliage at some point in the future.

An important question is what assumptions are made on the domains. If we make the *constant-domain assumption*, the domain elements are global, i.e. the same domain elements are available at all points in time. If the domains are assumed to be *expanding* (*increasing*), the domain of the next point in time contains always the current domain. Similarly, if the domains are assumed to be *decreasing*, the domain of the next point in time is always contained in the current domain. If no restriction on the domains is imposed, we speak of *varying domains*. However, for the satisfiability problem in $\mathrm{LTL}_{\mathcal{ALC}}$ is is enough to consider constant domains, since the satisfiability problem with expanding, decreasing and varying domains can be polynomially reduced to the satisfiability problem with constant domains [GKW+03]. Note that this reduction works also for other temporalised DLs that allow temporal operators to occur in front of concepts such as the ones in [WZ00].

Moreover, it is often desirable to ensure that the interpretation of certain concept and role names does not change over time. We call such concept and role names *rigid*. For instance, one can argue that the concept name Dromedary should be rigid, i.e. if a domain element is a dromedary, it will always stay a dromedary in the future, and conversely, if a domain element is not a dromedary, it will never become a dromedary. *Rigid concept names* can be simulated in $\mathrm{LTL}_{\mathcal{ALC}}$ and similar temporalised DLs. To ensure that e.g. Dromedary is rigid, the following two TBox-axioms are sufficient:

$$\mathsf{Dromedary} \sqsubseteq \Box\mathsf{Dromedary}, \quad \neg\mathsf{Dromedary} \sqsubseteq \Box\neg\mathsf{Dromedary}.$$

[2] Other approaches to obtain a temporalised DL, which are out of the scope of this thesis, include using fixpoint extensions [FT03; FT11], and encoding time in so-called concrete domains [Lut01].

[3] Combinations of a branching-time temporal logic such as CTL or CTL* with a description logic have been investigated e.g. in [HWZ02; BHW+04; GJL12].

However, *rigid role names* cannot be simulated in $\mathrm{LTL}_{\mathcal{ALC}}$. Interestingly, already one rigid role name causes the satisfiability problem of $\mathrm{LTL}_{\mathcal{ALC}}$-concepts w.r.t. global TBoxes to be undecidable [GKW+03; LWZ08]. Moreover, one can show by a reduction of the recurrent tiling problem that the problem is actually Σ^1_1-hard, i.e. not even recursively enumerable [LWZ08]. The reduction crucially depends on the presence of a global TBox. One way to regain decidability is to restrict the TBox to an *acyclic* one.[4] However, the problem of deciding satisfiability of $\mathrm{LTL}_{\mathcal{ALC}}$-concepts w.r.t. rigid roles and acyclic TBoxes is still hard for non-elementary time [GKW+03].

If no rigid role names are allowed, the satisfiability problem of $\mathrm{LTL}_{\mathcal{ALC}}$-concepts w.r.t. global TBoxes is ExpTime-complete, which was originally stated in [Sch93b].[5] More generally, for any description logic $\mathcal{L}$ between $\mathcal{ALC}$ and $\mathcal{SHIQ}$, the complexity of deciding satisfiability in $\mathrm{LTL}_{\mathcal{L}}$ is the same as the complexity of deciding satisfiability of $\mathcal{L}$-concepts [LWZ08]. Hence, the satisfiability problem of $\mathrm{LTL}_{\mathcal{ALC}}$-concepts (without a global TBox) is PSpace-complete, since the satisfiability problem of $\mathcal{ALC}$-concepts is PSpace-complete [SS91].

Moreover, if one additionally allows temporal operators to occur in front of axioms, i.e. deals with temporal knowledge bases, the complexity of deciding whether such a temporal knowledge base has a model, turns out to be ExpSpace-complete [GKW+03] in $\mathrm{LTL}_{\mathcal{ALC}}$ if no rigid role names are considered.

Due to the high complexity of reasoning and the undecidability results for the case where rigid role names are allowed, in [AKL+07; AKR+10], light-weight DLs have been extended by allowing temporal operators to occur in front of concepts. There, various complexity results for temporal extensions of members of the *DL-Lite* family are shown. However, in [AKL+07], the authors also show that reasoning easily becomes undecidable (if rigid role names are allowed) already in a small temporal extension of $\mathcal{EL}$ that is subsumed by $\mathrm{LTL}_{\mathcal{EL}}$.

Another temporalised DL, which is a combination of $\mathcal{ALC}$ and LTL, is $\mathcal{ALC}$-LTL [BGL12]. In $\mathcal{ALC}$-LTL, temporal operators are not allowed to occur in front of concepts, but rather in front of axioms. More precisely, $\mathcal{ALC}$-LTL is 'LTL over $\mathcal{ALC}$-axioms', i.e. it is LTL with $\mathcal{ALC}$-axioms instead of propositional variables. For instance, the following is an $\mathcal{ALC}$-LTL-formula:

$$\neg\mathsf{Dromedary}(\mathsf{leah}) \wedge \Diamond(\exists\mathsf{likes}.\mathsf{Foliage})(\mathsf{leah}).$$

It states that Leah is not a dromedary (now), and she will like foliage at some point in the future. The complexity of deciding whether an $\mathcal{ALC}$-LTL-formula is satisfiable is investigated in detail in [BGL12]. For $\mathcal{ALC}$-LTL, one needs to distinguish three different settings: (i) no rigid names are available, (ii) only rigid concept names are available, and (iii) rigid role names are available. Since temporal operators are not allowed to occur in front of concepts, rigid concept names are no longer expressible in the logic. However, it is well-known that rigid role names can simulate rigid concept names [BGL12], and thus there are only three cases to consider. Complexity results for the satisfiability problem in all three settings are obtained in [BGL12] for the case of constant domains. If no rigid names are available, the satisfiability problem is ExpTime-complete. If only rigid concept names are available, the complexity increases to NExpTime-complete, and if rigid role names are available, we have 2ExpTime-completeness. Note that it is not clear whether considering different kinds of domains like varying, expanding, or decreasing domains has an impact on the complexity

[4] For a formal definition of the syntax of acyclic TBoxes, see Definition 2.5.

[5] As remarked in [LWZ08], the original proof is incorrect. For a correct proof, see [LWZ08].

of the satisfiability problem, since the reduction from varying, expanding, and decreasing domains to the constant-domain case in [GKW+03] does not work if temporal operators are not allowed to occur in front of concepts. In [BGL12], it is conjectured, however, that in some settings the complexity of the satisfiability problem may decrease.

In this thesis, we use temporalised description logics to formulate knowledge about the temporal behaviour of systems. The next section gives an overview of their application in runtime verification.

1.3 Runtime Verification

Runtime verification [CM04] allows to verify whether an observed system has certain (wanted or unwanted) properties. These properties are usually expressed in a temporal formalism. This property is then 'translated' into a so-called *monitor*. Intuitively, such a monitor solves the following task. Having consumed a finite prefix of the actual behaviour of the system, the monitor indicates whether the property is satisfied or not.

In the literature, there is a plethora of approaches to constructing such monitors, see e.g. [HR04; RH05; Roş12; dR05; BLS10; BLS11; BGH+04; BRH07; BBL09]. Here, we extend the work of [BLS11] where the property is specified in propositional LTL [Pnu77] and a three-valued approach is developed. To illustrate the idea, consider the following example. Suppose that the vital parameters of patient Bob are measured in an emergency ward. If Bob has a high heart rate and a low blood pressure, then an alarm should be raised. This property can be expressed using the propositional LTL-formula

$$\phi_{\text{Bob}} := \Box(\neg(\mathsf{highHeartRateBob} \wedge \mathsf{lowBloodPressureBob})),$$

where highHeartRateBob and lowBloodPressureBob are propositional variables whose validity at each point in time can be checked by evaluating the results of sensing. Intuitively, ϕ_{Bob} expresses that it is always the case that Bob has not both a high hear rate and a low blood pressure. If the formula is violated, then we raise an alarm. The information about Bob's health status at each point in time now yields a finite prefix u. We need to check whether *all* continuations of this prefix satisfy or violate ϕ_{Bob}, i.e. no matter how the system's behaviour evolves over time, we certainly know whether the formula is satisfied or not. Thus, there are three possible answers that a monitor may give having read such a prefix u:

- 'true' if *all* continuations of u satisfy ϕ_{Bob};
- 'false' if *all* continuations of u do *not* satisfy ϕ_{Bob}; and
- 'inconclusive' if none of the above holds, i.e. no definite answer can be given.

In our example, it should be clear that the monitor can never output 'true' since the 'bad' state of the system could still be observed in the future. However, as soon as it outputs 'false', we can raise an alarm and call for intervention by the medical staff.

Note that runtime verification is not about answering such a single question given a prefix u and a propositional LTL-formula ϕ. In fact, since the behaviour of the system is observed over time, this prefix u is continuously extended by adding new observations. The monitor should not answer the questions for the prefixes one after another independently of each other. On the contrary, the monitor should successively read the input, and based on this information it should compute the answer in constant time (if the size of the propositional LTL-formula is assumed to be constant). Thus, the answer of the monitor does *not* depend on the length of the already observed prefix.

This approach presupposes, however, that the relevant propositional variables can be evaluated at each point in time. This is not always realistic due to the following reasons. Firstly, the states of the system may have a complex internal structure, and secondly, the assumption that we have complete information about the system's status at each point in time may be too strict. Temporalised description logics help to overcome these issues. A first step in that direction was done in [BBL09] where the temporalised description logic $\mathcal{ALC}$-LTL [BGL12] was used for constructing monitors. There, $\mathcal{ALC}$-axioms capture the observations and at each point in time, the monitor observes the system's status incompletely by reading an ABox. Unfortunately, in the presence of rigid names, the approach developed in [BBL09] does not work. In Chapter 4, we give a correct monitor construction for the even more expressive temporalised description logic $\mathcal{SHOQ}$-LTL. Moreover, the approach developed here has a better computational complexity, even though it also takes into account background knowledge in the form of another $\mathcal{SHOQ}$-LTL-formula that specifies *temporal* information about the future behaviour of the system.

Considering again the example above, patient Bob's medical status can be captured in ABoxes, whereas additional information about Bob is available from the patient record and added by the medical staff. Such background information is encoded using concepts defined in a medical ontology like SNOMED CT.[6] The observed sequence of ABoxes contain a high-level view of the patient's medical status, which the monitor uses to determine whether a critical situation specified by a $\mathcal{SHOQ}$-LTL-formula has arisen.

In the next section, we consider a similar problem, namely temporalised query entailment.

1.4 Temporalised Query Entailment

In a simple setting, one could realise a situation-awareness tool that helps the user to detect certain situations by using standard database techniques. For that, the information obtained from the sensors is stored in a relational database, and the situations to be recognised are specified by queries in an appropriate query language, e.g. conjunctive queries [AHV95]. However, database systems employ the closed-world assumption (CWA), i.e. knowledge that is not present in the database is assumed to be false rather than unknown. As argued above, this assumption is not always appropriate as the sensors may provide us with incomplete information about the current state of the system. Also, additional global knowledge, i.e. knowledge that holds true at each point in time, may be available, which allows for a limited projection into the future such that maybe more (or less) answers to the query are found.

These requirements are addressed in the research field of ontology-based data access (OBDA) [DEF+99; PCD+08]. There, the fact base is an ABox, which is interpreted with the open-world assumption, and an ontology is used to encode the background knowledge. In OBDA, one usually assumes that the ABox is obtained from external data sources (in the case of situation awareness, the raw sensor data) through appropriate mappings (which in our case realise the preprocessing and aggregation of the sensor data), but here we abstract from

[6] See `http://www.ihtsdo.org/snomed-ct/`.

the mapping step and assume that the result of the preprocessing is explicitly represented in an ABox.

As an example, consider again patient Bob. Assume that the ABox contains the following information:

$$\begin{array}{lll}\mathsf{systolic\text{-}pressure}(\mathsf{bob},\mathsf{p}_1), & \mathsf{High\text{-}pressure}(\mathsf{p}_1), & \\ \mathsf{history}(\mathsf{bob},\mathsf{h}_1), & \mathsf{Hypertension}(\mathsf{h}_1), & \mathsf{Male}(\mathsf{bob}).\end{array}$$

Intuitively, the first line expresses that Bob has high blood pressure, which is information obtained from sensor data. The second line expresses that Bob has a history of hypertension and that he is male, which is information obtained from the patient record. In addition, we have an ontology that states that patients with high blood pressure have hypertension, and that patients that currently have hypertension and also have a history of hypertension are at risk of a heart attack:

$$\begin{aligned}\exists\mathsf{systolic\text{-}pressure}.\mathsf{High\text{-}pressure} &\sqsubseteq \exists\mathsf{finding}.\mathsf{Hypertension},\\ \exists\mathsf{finding}.\mathsf{Hypertension}\sqcap\exists\mathsf{history}.\mathsf{Hypertension} &\sqsubseteq \exists\mathsf{risk}.\mathsf{Myocardial\text{-}infarction}.\end{aligned}$$

Assume that the situation we want to recognise for a given patient x is whether this patient is a male person who is at risk of a heart attack. This situation can be described by the conjunctive query

$$\exists y.\mathsf{Male}(x)\land\mathsf{risk}(x,y)\land\mathsf{Myocardial\text{-}infarction}(y),$$

i.e. 'give me all x such that x is male and at risk of y, which is a heart attack'.

Given the information in the ABox and the axioms in the ontology, we can derive that bob is a *certain answer* to the query in the ABox w.r.t. the ontology, i.e. the following conjunctive query (without free variables)

$$\exists y.\mathsf{Male}(\mathsf{bob})\land\mathsf{risk}(\mathsf{bob},y)\land\mathsf{Myocardial\text{-}infarction}(y)$$

is *entailed* by the ABox and the ontology. Obviously, this answer cannot be derived without the ontology.

The complexity of *query entailment* w.r.t. an ontology, i.e. the complexity of checking whether a given tuple of individual names is a certain answer to a query in an ABox w.r.t. an ontology, has been investigated in detail for cases where the ontology is expressed in an appropriate description and the query is a conjunctive query. One can either consider the *combined complexity*, which is measured in the size of the whole input (consisting of the query, the ontology, and the ABox), or the *data complexity*, which is measured in the size of the ABox only (i.e. the query and the ontology are assumed to be of constant size). The underlying assumption is that the query and the ontology are usually relatively small, whereas the size of the data may be huge.

In the database setting (where there is no ontology and the CWA is used), conjunctive-query entailment is NP-complete w.r.t. combined complexity and in AC^0 w.r.t. data complexity [CM77; AHV95]. For expressive DLs, the complexity of checking certain answers is considerably higher: for $\mathcal{ALC}$, the query entailment problem is ExpTime-complete w.r.t. combined complexity and co-NP-complete w.r.t. data complexity [CDL98; Lut08a; CDL+06]. For this reason, the more light-weight DLs of the *DL-Lite*-family have been developed, for which the entailment problem is still in AC^0 w.r.t. data complexity, and for which computing certain answers can be reduced to answering conjunctive queries in the database setting [CDL+09].

Unfortunately, OBDA as described until now is not sufficient to achieve high-level situation awareness. The reason is that the situations we want to recognise may depend on states of the system at different time points. To illustrate this, consider again the above example. There, it was explicitly stated in the patient records that Bob has a history of hypertension. It makes more sense to derive this piece of information automatically from the data obtained about blood pressure etc.

To achieve this, we propose a temporal query language that extends propositional LTL by allowing conjunctive queries in place of propositional variables, which is very similar to some of the temporalised DLs considered above. In our example, we want to query for male patients with a history of hypertension, which can be expressed by the temporal query

$$\mathsf{Male}(x) \wedge \mathsf{X}^- \Diamond^- \big(\exists y.\mathsf{finding}(x,y) \wedge \mathsf{Hypertension}(y)\big).$$

Intuitively, this temporal query asks for all x such that x is male and at some point in the (strict) past, x had a finding y of hypertension.

As for temporalised description logics, it makes perfect sense to consider rigid concept and role names. For example, we may want to assume that the concept Male is rigid, and thus a patient that is male now also has been male in the past and will stay male in the future.

Thus, our overall setting for recognising situations will be the following. In addition to a global ontology (which describes properties of the system that hold at every point in time), we have a sequence of ABoxes $\mathcal{A}_0, \mathcal{A}_1, \dots, \mathcal{A}_n$, which (incompletely) describe the states of the system at the previous time points $0, 1, \dots, n-1$ and the current time point n. The situation to be recognised is expressed by a temporal conjunctive query, as introduced above, which is evaluated w.r.t. the current time point n. In Chapter 5, we consider the complexity of this approach, i.e. the complexity of temporalised query entailment. There, we adopt $\mathcal{SHQ}$ as description logic, which is an extension of the description logic $\mathcal{ALC}$.

1.5 Verification in DL-Based Action Formalisms

In this section, we consider a different scenario. We assume that we have a system that executes predefined *actions*. Moreover, we assume that we have some knowledge about which actions the system executes. This is captured in a so-called *action program*. We consider non-terminating action programs, and are interested in the *verification problem*, i.e. the problem of deciding whether a certain (temporal) property holds after executing an action program.

High-level action programming languages such as GOLOG [LRL+97] and FLUX [Thi05a] are based on the situation calculus [Rei01] and the fluent calculus [Thi05b], respectively. These calculi encompass full first-order logic, which implies that interesting reasoning problems in them are undecidable. To regain decidability, we restrict our action formalism in two directions: (i) our action formalism is based on a decidable description logic, and (ii) the action program is 'generated' by an ω-automaton (i.e. an automaton working on infinite words).

In [BLM10], it is shown that the verification problem is decidable for the DL-based action formalism introduced in [BLM+05a]. The properties are formulated in a restricted version of the temporalised description logic $\mathcal{ALCO}$-LTL. The authors of [BLM+05a; BLM10] consider, however, only the case where domain constraints are encoded with *acyclic* TBoxes. If one allows general TBoxes, one has to deal with the *ramification problem*, i.e. the question which additional effects the executing of an action has in order to satisfy the domain constraints. In Chapter 6, we introduce a DL-based action formalism that is extended with so-called *causal relationships* that take care of this issue. We prove that important inference problems in the extended action formalism stay decidable and derive complexity results from the obtained decision procedures. Moreover, we continue the work of [BLM10] by considering the verification problem in the extended action formalism.

1.6 Outline and Contributions of the Thesis

In the following, we give a broad outline of the present thesis and summarise the main scientific contributions.

In *Chapter 2*, we formally introduce the basic notions that are needed for the thesis. These are foundations of description logics and of propositional LTL. Moreover, we revisit the relationship between propositional LTL and ω-automata.

In *Chapter 3*, we introduce the temporalised description logic $\mathcal{SHOQ}$-LTL, and prove complexity results for the satisfiability problem in this logic. We consider three different settings: (i) neither concept names nor role names are allowed to be rigid, (ii) only concept names are allowed to be rigid, and (iii) both concept and role names are allowed to be rigid. We can show that the complexity is the same as in the less expressive temporalised description logic $\mathcal{ALC}$-LTL [BGL12], namely ExpTime-complete in Setting (i), NExpTime-complete in Setting (ii), and 2ExpTime-complete in Setting (iii). In order to prove these results, we need to consider also the consistency problem of Boolean $\mathcal{SHOQ}$-knowledge bases (w.r.t. some side condition). We can prove that this problem, which is interesting on its own, can be decided in exponential time. Some of the (ideas of the proofs of the) results of this chapter are already published:

- Franz Baader, Stefan Borgwardt, and Marcel Lippmann: 'Temporalizing Ontology-Based Data Access'. In *Proc. of the 24th Int. Conf. on Automated Deduction (CADE-24)*, Lake Placid, NY, USA. Edited by Maria Paola Bonacina. Volume 7898. Lecture Notes in Artificial Intelligence. Springer-Verlag, June 2013, pages 330–344
- Franz Baader, Stefan Borgwardt, and Marcel Lippmann: 'On the Complexity of Temporal Query Answering'. LTCS-Report 13-01. Chair of Automata Theory, Institute of Theoretical Computer Science, Technische Universität Dresden, Mar. 2013. URL: `http://lat.inf.tu-dresden.de/research/reports.html`

In *Chapter 4*, we consider runtime verification using the temporalised description logic $\mathcal{SHOQ}$-LTL. Before we can construct monitors for $\mathcal{SHOQ}$-LTL-formulas, we need to construct ω-automata for $\mathcal{SHOQ}$-LTL-formulas. For this construction, we reuse certain results from Chapter 3. We are able to show that even in the most complex case where both rigid concept names and rigid role names are allowed, monitors of doubly exponential size can be constructed using doubly exponential time. Moreover, we show that this doubly exponential blow-up in the construction of the monitor cannot be avoided. Indeed, as we show, such a blow-up in unavoidable even for propositional LTL. Finally, we consider the complexity of the deciding *liveness* and *monitorability*, which are two important related decision problems, for the three settings above. Our results are only tight for the case where both rigid concept and role names are allowed. There, both problems are 2EXPTIME-complete. For the other cases, a gap remains: both problems are EXPTIME-hard and in 2EXPTIME if no rigid names are allowed, and CO-NEXPTIME-hard and in 2EXPTIME if only rigid concept names are allowed. However, the exact complexity of these problems are not even known for propositional LTL. Some of the results of this chapter are already published. However, we considered only the less expressive temporalised description logic $\mathcal{ALC}$-LTL in the following publications:

- Franz Baader, Andreas Bauer, and Marcel Lippmann: 'Runtime Verification Using a Temporal Description Logic'. In *Proc. of the 7th Int. Symp. on Frontiers of Combining Systems (FroCoS 2009)*, Trento, Italy. Edited by Silvio Ghilardi and Roberto Sebastiani. Volume 5749. Lecture Notes in Computer Science. Springer-Verlag, Sept. 2009, pages 149–164
- Franz Baader and Marcel Lippmann: 'Runtime Verification Using a Temporal Description Logic Revisited'. LTCS-Report 14-01. Chair of Automata Theory, Institute of Theoretical Computer Science, Technische Universität Dresden, Mar. 2014. URL: `http://lat.inf.tu-dresden.de/research/reports.html`

In *Chapter 5*, we consider temporalised query entailment in any description logic between $\mathcal{ALC}$ and $\mathcal{SHQ}$. After formally introducing the temporal query language that we consider, we provide complexity results of the corresponding entailment problem. For the three settings above, we have shown results both for data complexity and combined complexity. If neither concept nor role names are allowed to be rigid, temporalised query entailment is CO-NP-complete w.r.t. data complexity and EXPTIME-complete w.r.t. combined complexity. If only concept names may be rigid, the problem is CO-NP-complete w.r.t. data complexity and CO-NEXPTIME-complete w.r.t. combined complexity. Finally, if both concept and role names are allowed to be rigid, the problem is CO-NP-hard and in EXPTIME w.r.t. data complexity and 2EXPTIME-complete w.r.t. combined complexity. For showing these results, some results of Chapter 3 are used. Most of the results of this chapter are already published:

- Franz Baader, Stefan Borgwardt, and Marcel Lippmann: 'Temporalizing Ontology-Based Data Access'. In *Proc. of the 24th Int. Conf. on Automated Deduction (CADE-24)*, Lake Placid, NY, USA. Edited by Maria Paola Bonacina. Volume 7898. Lecture Notes in Artificial Intelligence. Springer-Verlag, June 2013, pages 330–344
- Franz Baader, Stefan Borgwardt, and Marcel Lippmann: 'On the Complexity of Temporal Query Answering'. LTCS-Report 13-01. Chair of Automata Theory, Institute of Theoretical Computer Science, Technische Universität Dresden, Mar. 2013. URL: `http://lat.inf.tu-dresden.de/research/reports.html`

Finally, in *Chapter 6*, we consider the verification problem in action formalisms based on description logics between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$. To solve the ramification problem, i.e. the question how to deal with indirect effects caused by domain constraints (which arises if we allow general TBoxes), we extend the DL-based action formalism introduced in [BLM+05a] (which could deal only with acyclic TBoxes) with causal relationships. We show that important inference problems such as the consistency problem and the projection problem are decidable in our new formalism, and continue the work of [BLM10] by generalising the verification problem. We derive a number of complexity results from the obtained decision procedures. Depending on the base DL, the complexity results range from PSPACE-complete to CO-NEXPTIME-hard and in $\mathrm{P}^{\mathrm{NExpTime}}$ for the consistency problem, and from PSPACE-complete to CO-NEXPTIME-complete for the projection problem. For the verification problem, the complexity ranges from in EXPSPACE to in CO-2NEXPTIME, and it is unknown whether these bounds are tight. Some of the results of this chapter are already published:

- Franz Baader, Marcel Lippmann, and Hongkai Liu: 'Using Causal Relationships to Deal with the Ramification Problem in Action Formalisms Based on Description Logics'. In *Proc. of the 17th Int. Conf. on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR-17)*, Yogyakarta, Indonesia. Edited by Christian G. Fermüller and Andrei Voronkov. Volume 6397. Lecture Notes in Computer Science. Springer-Verlag, Oct. 2010, pages 82–96
- Wael Yehia, Hongkai Liu, Marcel Lippmann, Franz Baader, and Mikhail Soutchanski: 'Experimental Results on Solving the Projection Problem in Action Formalisms Based on Description Logics'. In *Proc. of the 25th Int. Workshop on Description Logics (DL 2012)*, Rome, Italy. Edited by Yevgeny Kazakov, Domenico Lembo, and Frank Wolter. Volume 846. CEUR Workshop Proceedings. CEUR-WS.org, June 2012
- Franz Baader, Marcel Lippmann, and Hongkai Liu: 'Adding Causal Relationships to DL-based Action Formalisms'. LTCS-Report 10-01. Chair of Automata Theory, Institute of Theoretical Computer Science, Technische Universität Dresden, Feb. 2010. URL: http://lat.inf.tu-dresden.de/research/reports.html

Chapter 2

Preliminaries

In this chapter, we set a basis for the later chapters by introducing the basic notions that we need. Firstly, we introduce *description logics (DLs)* as the logical formalism that we use throughout the thesis. Secondly, we give the basic definitions of *propositional linear-time temporal logic* and recall the relationship with ω-automata, i.e. automata working on infinite words. These notions are needed to obtain the *linear-time temporalised description logic* $\mathcal{SHOQ}$*-LTL* (see Chapter 3).

More specific notions like the basics of DL-based query answering and action formalisms based on description logics are not covered in this chapter but introduced in the respective later chapters.

2.1 Basic Notions of Description Logics

As already sketched in the Chapter 1, description logics [BCM+07] are a successful family of logic-based knowledge representation formalisms. In this section, we introduce the basic notions of DLs that are relevant for this thesis. For a more thorough introduction to DLs, the interested reader is referred to the *Description Logic Handbook* [BCM+07].

2.1.1 Description Logic Concepts

As discussed in Section 1.1, *concepts* are defined using concept names, role names, individual names, and concept and role constructors. Throughout the thesis, let $\mathsf{N_C}$, $\mathsf{N_R}$, and $\mathsf{N_I}$, respectively, denote pairwise disjoint sets of concept names, role names, and individual names. We introduce now the concept and role constructors that are relevant for this thesis, and show how they are used to define the syntax of concepts (sometimes called *concept descriptions*).

Definition 2.1 (Syntax of concepts). *A* role *r is either a role name, i.e. $r \in \mathsf{N_R}$, or it is of the form s^- for $s \in \mathsf{N_R}$ (inverse role). The set of* concepts *is the smallest set such that*

- *every concept name $A \in \mathsf{N_C}$ is a concept; and*
- *if C, D are concepts, $a \in \mathsf{N_I}$, r is a role, and n is a non-negative integer, then the following are also concepts: $\neg C$ (negation), $C \sqcap D$ (conjunction), $\{a\}$ (nominal), $\exists r.C$ (existential restriction), and $\geq n\, r.C$ (at-least restriction).* ♢

As usual in description logics, we use

- $C \sqcup D$ (disjunction) as an abbreviation for $\neg(\neg C \sqcap \neg D)$;

- $C \rightarrow D$ (implication) as an abbreviation for $\neg C \sqcup D$;
- $\top$ (top) as an abbreviation for $A \sqcup \neg A$ where $A \in \mathsf{N_C}$ is arbitrary but fixed;
- $\bot$ (bottom) as an abbreviation for $\neg\top$;
- $\forall r.C$ (value restriction) as an abbreviation for $\neg(\exists r.\neg C)$; and
- $\leq n\, r.C$ (at-most restriction) as an abbreviation for $\neg(\geq(n+1)\, r.C)$.

Note that there are more concept and role constructors introduced in the literature. These are either beyond the scope of this thesis or introduced where needed.

The semantics of concepts is given in a model-theoretic way using the notion of an interpretation.

Definition 2.2 (Semantics of concepts). *An* interpretation *is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where the* domain $\Delta^{\mathcal{I}}$ *is a non-empty set, and the* interpretation function $\cdot^{\mathcal{I}}$ *assigns to every $A \in \mathsf{N_C}$ a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, to every $r \in \mathsf{N_R}$ a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and to every $a \in \mathsf{N_I}$ an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ such that the* unique-name assumption (UNA) *holds, i.e. for all $a, b \in \mathsf{N_I}$ with $a \neq b$, we have $a^{\mathcal{I}} \neq b^{\mathcal{I}}$. This function is extended to inverse roles and concepts as follows:*

- $(s^-)^{\mathcal{I}} := \{(e,d) \mid (d,e) \in s^{\mathcal{I}}\}$;
- $(\neg C)^{\mathcal{I}} := \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$;
- $(C \sqcap D)^{\mathcal{I}} := C^{\mathcal{I}} \cap D^{\mathcal{I}}$;
- $\{a\}^{\mathcal{I}} := \{a^{\mathcal{I}}\}$;
- $(\exists r.C)^{\mathcal{I}} := \{d \in \Delta^{\mathcal{I}} \mid \text{there exists an } e \in \Delta^{\mathcal{I}} \text{ with } (d,e) \in r^{\mathcal{I}} \text{ and } e \in C^{\mathcal{I}}\}$; *and*
- $(\geq n\, r.C)^{\mathcal{I}} := \big\{d \in \Delta^{\mathcal{I}} \mid |\{e \in \Delta^{\mathcal{I}} \mid (d,e) \in r^{\mathcal{I}} \text{ and } e \in C^{\mathcal{I}}\}| \geq n\big\}$.

We call a concept C satisfiable *if there is a interpretation $\mathcal{I}$ such that $C^{\mathcal{I}} \neq \emptyset$.* ◇

Note that in our definition of the semantics, we make the unique-name assumption, which is an assumption often made in DLs. We continue by giving an example of the notions introduced so far.

Example 2.3. *Let C be the following concept:*

$$\{\mathsf{leah}\} \sqcap \neg\mathsf{Dromedary} \sqcap \exists\mathsf{likes}.\mathsf{Foliage} \sqcap {\leq}1\, \mathsf{has}.\mathsf{Hump} \sqcap \exists\mathsf{is\text{-}father\text{-}of}^-.\{\mathsf{hassan}\}.$$

It is not hard to see that C is satisfiable. Figure 2.4 depicts the graphical representation of an interpretation $\mathcal{I}$ with $C^{\mathcal{I}} = \{\mathsf{leah}^{\mathcal{I}}\} \neq \emptyset$. Note that for this interpretation $\mathcal{I}$, we have $\mathsf{Dromedary}^{\mathcal{I}} = \emptyset$. ◇

It is important to note that the semantics of a concept is entirely given by an interpretation. The concept, role, and individual names itself do not imply anything. For instance, the concept name Dromedary does not necessarily denote dromedaries. It is just a name, and the actual interpretation has to ensure the expected meaning.

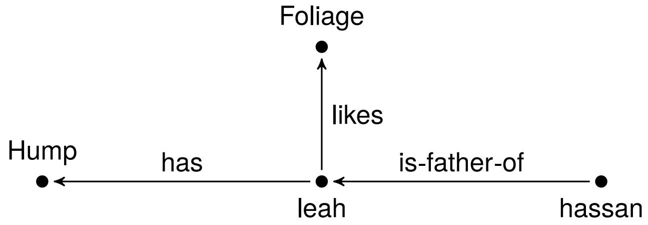

Figure 2.4: An interpretation $\mathcal{I}$ such that $C^{\mathcal{I}} \neq \emptyset$ for the concept C of Example 2.3

2.1.2 Knowledge Bases

To restrict ourselves to certain kinds of interpretations, we capture the domain knowledge in a so-called knowledge base (KB). Each KB consists of three parts: a *TBox* (terminological box), an *RBox* (role box), and an *ABox* (assertional box). Intuitively, the RBox states knowledge about roles, the TBox states knowledge about all domain elements, whilst the ABox states knowledge about specific individuals.[1]

Definition 2.5 (Syntax of TBoxes). *An* concept definition *is of the form $A \equiv C$ where $A \in \mathsf{N_C}$ and C is a concept. A* general concept inclusion (GCI) *is of the form $C \sqsubseteq D$ where C, D are concepts. We call both concept definitions and GCIs* TBox-axioms.

A (general) TBox *is a finite set of TBox-axioms. An* acyclic TBox *$\mathcal{T}$ is a finite set of concept definitions such that the following two conditions are satisfied:*

- *if $A \equiv C$, $A \equiv D \in \mathcal{T}$, then $C = D$ (unambiguity); and*
- *there is no sequence $A_1 \equiv C_1, \ldots, A_n \equiv C_n \in \mathcal{T}$ with $n \geq 1$ such that A_{i+1} occurs in C_i (for $1 \leq i < n$) and A_1 occurs in C_n (no cyclic definitions).*

We call the concept names that occur on the left-hand side of some concept definition in $\mathcal{T}$ defined concept names *whereas we call the others* primitive concept names. ♢

Intuitively, an acyclic TBox consists of 'macros', i.e. definitions of shorthands for (complex) concepts. Therefore, acyclic TBoxes are sometimes called *unfoldable* TBoxes. The semantics of TBoxes can now be defined in a straightforward manner.

Definition 2.6 (Semantics of TBoxes). *The interpretation $\mathcal{I}$ is a* model

- *of the concept definition $A \equiv C$ (written $\mathcal{I} \models A \equiv C$) if $A^{\mathcal{I}} = C^{\mathcal{I}}$; and*
- *of the GCI $C \sqsubseteq D$ (written $\mathcal{I} \models C \sqsubseteq D$) if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.*

$\mathcal{I}$ is a model of the TBox $\mathcal{T}$ (written $\mathcal{I} \models \mathcal{T}$) if it is a model of each TBox-axiom in $\mathcal{T}$. We call a TBox consistent *if it has a model.* ♢

Note that the concept definition $A \equiv C$ can be captured by two GCIs, namely $A \sqsubseteq C$ and $C \sqsubseteq A$. For ease of presentation, we thus often assume in the following that general TBoxes do not contain concept definitions or that a concept definition is an 'abbreviation' for two GCIs.

[1] In the literature, sometimes the information from the RBox is included in the TBox. However, we keep them separately here as this turns out to be useful later.

Definition 2.7 (Syntax of RBoxes). *A* transitivity axiom *is of the form* $\mathsf{trans}(r)$ *where r is a role, and a* role-inclusion axiom *is of the form $r \sqsubseteq s$ where r, s are roles. We call both transitivity axioms and role-inclusion axioms* RBox-axioms. *An* RBox *is a finite set of RBox-axioms.* ♢

Again, the semantics is straightforward.

Definition 2.8 (Semantics of RBoxes). *The interpretation $\mathcal{I}$ is a* model

- *of the transitivity axiom* $\mathsf{trans}(r)$ *(written $\mathcal{I} \models \mathsf{trans}(r)$) if $r^{\mathcal{I}} \circ r^{\mathcal{I}} \subseteq r^{\mathcal{I}}$, i.e. $r^{\mathcal{I}}$ is transitive; and*
- *of the role-inclusion axiom $r \sqsubseteq s$ (written $\mathcal{I} \models r \sqsubseteq s$) if $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$.*

$\mathcal{I}$ is a model of the RBox $\mathcal{R}$ (written $\mathcal{I} \models \mathcal{R}$) if it is a model of each RBox-axiom in $\mathcal{R}$. We call $\mathcal{R}$ consistent *if it has a model.* ♢

Finally, we define the syntax of ABoxes as follows.

Definition 2.9 (Syntax of ABoxes). *A* concept assertion *is of the form $C(a)$ where C is a concept, and $a \in \mathsf{N_I}$. A* role assertion *is of the form $r(a,b)$ where $r \in \mathsf{N_R}$, and $a, b \in \mathsf{N_I}$. We call both concept assertions and role assertions* ABox-axioms. *An ABox-axiom is* atomic *if it is either a role assertion or an atomic concept assertion, i.e. it is of the form $A(a)$ where $A \in \mathsf{N_C}$ and $a \in \mathsf{N_I}$.*

A (complex) ABox *is a finite set of ABox-axioms. A* simple ABox *is a finite set of atomic ABox-axioms.* ♢

We call ABox-axioms sometimes *assertions*. Note that every simple ABox is also a (complex) ABox. The semantics of ABoxes is defined as follows.

Definition 2.10 (Semantics of ABoxes). *The interpretation $\mathcal{I}$ is a* model

- *of the concept assertion $C(a)$ (written $\mathcal{I} \models C(a)$) if $a^{\mathcal{I}} \in C^{\mathcal{I}}$; and*
- *of the role assertion $r(a,b)$ (written $\mathcal{I} \models r(a,b)$) if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$.*

$\mathcal{I}$ is a model of the ABox $\mathcal{A}$ (written $\mathcal{I} \models \mathcal{A}$) if it is a model of each ABox-axiom in $\mathcal{A}$. We call $\mathcal{A}$ consistent *if it has a model.* ♢

Note that in the definition of the syntax of role assertions, we do not allow inverse roles. This is not a real restriction as an ‘assertion’ of the form $r^{-}(a,b)$ can be equivalently expressed by $r(b,a)$.

In the following, we often call TBox-axioms, RBox-axioms, and ABox-axioms simply *axioms*. Now, we are ready to give the formal definition of the syntax and the semantics of knowledge bases.

Definition 2.11 (Knowledge base). *A* knowledge base *is a triple $\mathcal{K} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$ where $\mathcal{A}$ is an ABox, $\mathcal{T}$ is a TBox, and $\mathcal{R}$ is an RBox.*

The interpretation $\mathcal{I}$ is a model *of $\mathcal{K}$ (written $\mathcal{I} \models \mathcal{K}$) if it is a model of $\mathcal{A}$, $\mathcal{T}$, and $\mathcal{R}$. We call $\mathcal{K}$* consistent *if it has a model. We say that $\mathcal{K}$* entails *an axiom α (written $\mathcal{K} \models \alpha$) if every model of $\mathcal{K}$ is also a model of α.* ♢

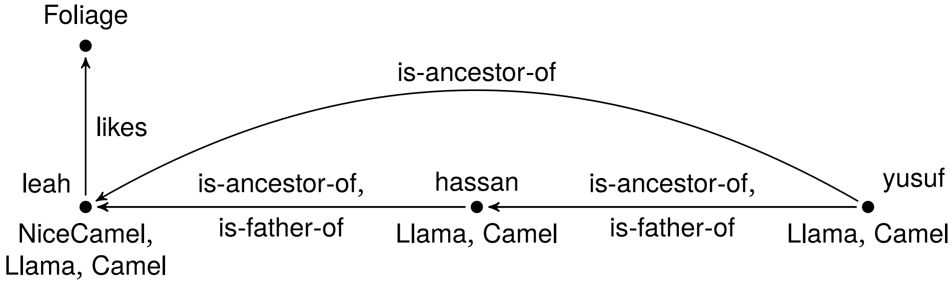

Figure 2.13: A model $\mathcal{I}$ of the KB $\mathcal{K}$ of Example 2.12

If a component of a knowledge base is empty, we may also omit it, i.e. we write e.g. $(\mathcal{A}, \mathcal{T})$ instead of $(\mathcal{A}, \mathcal{T}, \emptyset)$ if the RBox is empty.

We now consider an example of a knowledge base that illustrates how knowledge bases may be used. Note that this example serves only didactic purposes, and its content might not reflect everybody's mindset and is highly debatable.

Example 2.12. *Let $\mathcal{A}$ be the ABox that consists of the following assertions:*

$$\mathsf{NiceCamel}(\mathsf{leah}), \quad \mathsf{is\text{-}father\text{-}of}(\mathsf{hassan}, \mathsf{leah}), \quad \mathsf{is\text{-}father\text{-}of}(\mathsf{yusuf}, \mathsf{hassan}).$$

Intuitively, the first assertion states that Leah is a nice camel. The second one states that Hassan is the father of Leah, and the third one states that Yusuf is the father of Hassan.

Let $\mathcal{R}$ be the RBox that consists of the following two axioms:

$$\mathsf{is\text{-}father\text{-}of} \sqsubseteq \mathsf{is\text{-}ancestor\text{-}of}, \quad \mathsf{trans}(\mathsf{is\text{-}ancestor\text{-}of}).$$

The first axiom states that if d is the father of e, then d is also an ancestor of e. The second axiom states that is-ancestor-of *is transitive.*

Let $\mathcal{T}$ be the TBox that consists of the following GCIs and concept definitions:

- $\mathsf{BactrianCamel} \sqcup \mathsf{Dromedary} \sqsubseteq \mathsf{Camel}$*;*
- $\mathsf{Llama} \sqcup \mathsf{Guanaco} \sqcup \mathsf{Alpaca} \sqcup \mathsf{Vicuña} \sqsubseteq \mathsf{Camel}$*;*
- $\mathsf{TrueCamel} \equiv \mathsf{BactrianCamel} \sqcup \mathsf{Dromedary}$*; and*
- $\mathsf{NiceCamel} \equiv \neg\mathsf{Dromedary} \sqcap \exists \mathsf{likes}.\mathsf{Foliage}.$

Intuitively, the GCIs state that Bactrian camels, dromedaries, llamas, etc. are camels. The first concept definition states that true camels are Bactrian camels or dromedaries, and the second one states that nice camels are no dromedaries and they like foliage.

Figure 2.13 depicts the graphical representation of a model $\mathcal{I}$ of $\mathcal{K} := (\mathcal{A}, \mathcal{T}, \mathcal{R})$. Note that $\mathcal{I}$ is also a model of the axiom Llama(hassan)*, which is, however, not entailed by $\mathcal{K}$.* ◇

2.1.3 Specific Description Logics

As mentioned above, what differs from DL to DL is which concept and role constructors are available. The smallest propositionally closed DL is $\mathcal{ALC}$ [SS91]. In this DL, the allowed concept constructors are negation, conjunction, and existential restrictions, and thus also disjunctions, universal restrictions, and the top and bottom concepts can be expressed.

If additional concept or role constructors are available, this is denoted by concatenating a corresponding letter: $\mathcal{Q}$ means (qualified) number restrictions, $\mathcal{I}$ means inverse roles, $\mathcal{O}$ means nominals, and $\mathcal{H}$ means role-inclusion axioms (role hierarchies). For instance, the DL which is an extension of $\mathcal{ALC}$ and allows inverse roles is called $\mathcal{ALCI}$. The extension of $\mathcal{ALC}$ with transitivity axioms is usually denoted by $\mathcal{S}$ due to its close relationship with the modal logic ***S*4**. Thus, the DL that allows all the concept and role constructors introduced above is called $\mathcal{SHOIQ}$.

Throughout this thesis, we sometimes prefix some notions with the specific DL to make clear which DL is used to construct the concepts or axioms. For instance, we may write '$\mathcal{ALC}$-knowledge base' to make clear that the knowledge base is constructed using concepts expressible in $\mathcal{ALC}$, and does not contain e.g. inverse roles. If the DL under consideration is clear from the context, we omit this prefix for ease of presentation.

Given a knowledge base $\mathcal{K} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$, we say that a role name r is *transitive* (w.r.t. $\mathcal{K}$) if $\mathcal{K} \models \mathsf{trans}(r)$, and r is a *subrole* of a role name s (w.r.t. $\mathcal{K}$) if $\mathcal{K} \models r \sqsubseteq s$. Moreover, we call r *simple* (w.r.t. $\mathcal{K}$) if it has no transitive subrole. Note that entailments of the form $\mathcal{K} \models \mathsf{trans}(r)$ and $\mathcal{K} \models r \sqsubseteq s$ only depend on the RBox $\mathcal{R}$. Such entailments can be decided in time polynomial in the size of $\mathcal{R}$ [HST00]. As shown in [HST00], already for the DL $\mathcal{SHQ}$, the problem of deciding whether a given knowledge base is consistent is undecidable, even if all at-least restrictions are unqualified, i.e. of the form $\geq n\, r.\top$. One reason for that is the occurrence of non-simple role names in such restrictions. To regain decidability of this important inference problem, role names occurring in at-least restrictions are therefore usually required to be simple. In the following, we also make this restriction to the syntax of $\mathcal{SHQ}$ and every of its superlogics.

Under this assumption, the problem of deciding the consistency of knowledge bases is in ExpTime, even if the numbers occurring in at-least restrictions are given in binary encoding [Tob01]. On the other hand, the problem is ExpTime-hard already in $\mathcal{ALC}$ [Sch91]. If we add nominals ($\mathcal{SHOQ}$) [Sch94; HS01] or inverse roles ($\mathcal{SHIQ}$) [Sch94; Tob01], the complexity of this problem stays in ExpTime, but it increases to NExpTime-complete if we include both ($\mathcal{SHOIQ}$) [Sch94; Tob00; Pra05].

2.1.4 Boolean Knowledge Bases

The notion of a knowledge base (see Definition 2.11) can be generalised to Boolean knowledge bases.

Definition 2.14 (Boolean knowledge base). *Let $\mathcal{R}$ be an RBox. The set of* Boolean axiom formulas *w.r.t. $\mathcal{R}$ is the smallest set such that*

- *every ABox-axiom and every TBox-axiom in which at-least restrictions contain only simple roles w.r.t. $\mathcal{R}$ is a Boolean axiom formula; and*
- *if Ψ_1 and Ψ_2 are Boolean axiom formulas, then so are $\neg\Psi_1$ (negation) and $\Psi_1 \wedge \Psi_2$ (conjunction).*

A Boolean knowledge base *is a pair $\mathcal{B} = (\Psi, \mathcal{R})$, where $\mathcal{R}$ is an RBox, and Ψ is a Boolean axiom formula w.r.t. $\mathcal{R}$.*

The interpretation $\mathcal{I}$ is a model *of $\mathcal{B}$ (written $\mathcal{I} \models \mathcal{B}$) iff $\mathcal{I} \models \mathcal{R}$ and $\mathcal{I} \models \Psi$, where the latter is defined inductively as follows:*

- $\mathcal{I} \models \neg\Psi_1$ *iff* $\mathcal{I} \not\models \Psi_1$*; and*
- $\mathcal{I} \models \Psi_1 \wedge \Psi_2$ *iff* $\mathcal{I} \models \Psi_1$ *and* $\mathcal{I} \models \Psi_2$.

We call $\mathcal{B}$ consistent *if it has a model. We say that $\mathcal{B}$* entails *the axiom α (written $\mathcal{B} \models \alpha$) if every model of $\mathcal{B}$ is also a model of α.* ♢

For convenience, we use the Boolean knowledge base $(\Psi_1 \vee \Psi_2, \mathcal{R})$ as an abbreviation for $(\neg(\neg\Psi_1 \wedge \neg\Psi_2), \mathcal{R})$.

The reason why we do not allow RBox-axioms as Boolean axiom formulas is that the notion of simple roles does not make sense w.r.t. a Boolean combination of RBox-axioms.

According to this definition, knowledge bases can be seen as special kinds of Boolean knowledge bases. In fact, the knowledge base $\mathcal{K} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$ *induces* the Boolean knowledge base $\mathcal{B}_\mathcal{K} = (\Psi_\mathcal{K}, \mathcal{R})$ with $\Psi_\mathcal{K} := \bigwedge \mathcal{A} \wedge \bigwedge \mathcal{T}$, where $\bigwedge \mathcal{A}$ denotes $\bigwedge_{\alpha \in \mathcal{A}} \alpha$, and $\bigwedge \mathcal{T}$ denotes $\bigwedge_{\beta \in \mathcal{T}} \beta$. Thus, Boolean knowledge bases generalise classical knowledge bases as introduced in Definition 2.11.

The problem of deciding the consistency of Boolean knowledge bases, however, is not so well-investigated as for 'classical' knowledge bases. It is known that for the description logic $\mathcal{ALC}$, this problem is ExpTime-complete [GKW+03], and we show in Section 3.2.4 that it remains in ExpTime for an extension of the description logic $\mathcal{SHOQ}$.

2.2 Propositional Linear-Time Temporal Logic and ω-Automata

In this section, we recall the definitions for the prominent temporal logic *propositional linear-time temporal logic (LTL)* [Pnu77] that are relevant for this thesis. After introducing the syntax and semantics of propositional LTL in Section 2.2.1, we consider its connection to ω-automata in Section 2.2.2.

2.2.1 Syntax and Semantics of Propositional LTL

Propositional LTL extends propositional logic with modal operators that can be used to talk about the past and the future. The syntax of propositional LTL is defined as follows.

Definition 2.15 (Syntax of propositional LTL). *Let $\mathcal{P} = \{p_1, \ldots, p_m\}$ be a finite set of* propositional variables. *The set of* propositional LTL-formulas over $\mathcal{P}$ *is the smallest set such that*

- *if $p \in \mathcal{P}$, then p is a propositional LTL-formula over $\mathcal{P}$; and*
- *if ϕ_1 and ϕ_2 are propositional LTL-formulas over $\mathcal{P}$, then so are: $\neg\phi_1$ (negation), $\phi_1 \wedge \phi_2$ (conjunction), $\mathsf{X}\,\phi_1$ (next), $\mathsf{X}^-\,\phi_1$ (previous), $\phi_1 \,\mathsf{U}\, \phi_2$ (until), and $\phi_1 \,\mathsf{S}\, \phi_2$ (since).* ♢

If the set of propositional variables is clear from the context or irrelevant, we talk about propositional LTL-formulas rather than propositional LTL-formulas over $\mathcal{P}$.

As usual in temporal logics, we use

- $\phi_1 \vee \phi_2$ (disjunction) as an abbreviation for $\neg(\neg\phi_1 \wedge \neg\phi_2)$;
- $\phi_1 \rightarrow \phi_2$ (implication) as an abbreviation for $\neg\phi_1 \vee \phi_2$;
- true as an abbreviation for an arbitrary but fixed propositional tautology such as $p \vee \neg p$ with $p \in \mathcal{P}$;
- false as an abbreviation for $\neg\mathsf{true}$;
- $\Diamond\phi$ (diamond, which should be read as 'eventually' or 'some time in the future') as an abbreviation for $\mathsf{true}\,\mathsf{U}\,\phi$;
- $\Box\phi$ (box, which should be read as 'always' or 'always in the future') as an abbreviation for $\neg\Diamond\neg\phi$;
- $\Diamond^{-}\,\phi$ (which should be read as 'once' or 'some time in the past') as an abbreviation for $\mathsf{true}\,\mathsf{S}\,\phi$; and
- $\Box^{-}\,\phi$ (which should be read as 'historically' or 'always in the past') as an abbreviation for $\neg\,\Diamond^{-}\,\neg\phi$.

The semantics of propositional LTL is defined using the non-negative integers as discrete linear flow of time. For each point in time, i.e. non-negative integer, the semantic structure determines which of the propositional variables are true at this point. This is captured in the notion of a propositional LTL-structure.

Definition 2.16 (Semantics of propositional LTL). *Let $\mathcal{P} = \{p_1, \ldots, p_m\}$ be a set of propositional variables. A* propositional LTL-structure over *$\mathcal{P}$ is an infinite sequence $\mathfrak{W} = (w_i)_{i\geq 0}$ of sets $w_i \subseteq \mathcal{P}$, which we call* worlds.

Given a propositional LTL-formula ϕ, a propositional LTL-structure $\mathfrak{W} = (w_i)_{i\geq 0}$, and a time point $i \geq 0$, validity of ϕ in $\mathfrak{W}$ at time i *(written $\mathfrak{W}, i \models \phi$) is defined inductively as follows:*

$$\begin{array}{lll}
\mathfrak{W}, i \models p & \textit{iff} & p \in w_i \\
\mathfrak{W}, i \models \neg\phi_1 & \textit{iff} & \mathfrak{W}, i \not\models \phi_1, \textit{ i.e. not } \mathfrak{W}, i \models \phi_1 \\
\mathfrak{W}, i \models \phi_1 \wedge \phi_2 & \textit{iff} & \mathfrak{W}, i \models \phi_1 \textit{ and } \mathfrak{W}, i \models \phi_2 \\
\mathfrak{W}, i \models \mathsf{X}\,\phi_1 & \textit{iff} & \mathfrak{W}, i+1 \models \phi_1 \\
\mathfrak{W}, i \models \mathsf{X}^{-}\,\phi_1 & \textit{iff} & i > 0 \textit{ and } \mathfrak{W}, i-1 \models \phi_1 \\
\mathfrak{W}, i \models \phi_1\,\mathsf{U}\,\phi_2 & \textit{iff} & \textit{there is some } k \geq i \textit{ such that } \mathfrak{W}, k \models \phi_2, \textit{ and} \\
 & & \mathfrak{W}, j \models \phi_1 \textit{ for every } j,\ i \leq j < k \\
\mathfrak{W}, i \models \phi_1\,\mathsf{S}\,\phi_2 & \textit{iff} & \textit{there is some } k,\ 0 \leq k \leq i, \textit{ such that } \mathfrak{W}, k \models \phi_2, \textit{ and} \\
 & & \mathfrak{W}, j \models \phi_1 \textit{ for every } j,\ k < j \leq i
\end{array}$$

If $\mathfrak{W}, 0 \models \phi$, then we call $\mathfrak{W}$ a model *of ϕ. We call the propositional LTL-formula ϕ* satisfiable *if it has a model.*

The satisfiability problem in propositional LTL *is the problem of deciding, given a propositional LTL-formula ϕ, whether ϕ is satisfiable.*

Two propositional LTL-formulas ϕ_1, ϕ_2 are equivalent *(written $\phi_1 \equiv \phi_2$) if they have the same models.* $\Diamond$

Note that we defined here the so-called *non-strict* U and *non-strict* S. For the strict version $\mathsf{U}^{<}$ of U, one needs to replace in the definition of the semantics of U 'there is some $k \geq i$' by

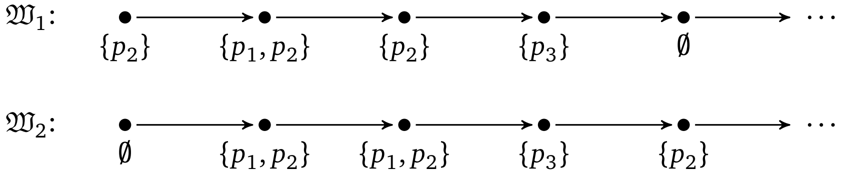

Figure 2.18: The propositional LTL-structures $\mathfrak{W}_1, \mathfrak{W}_2$ of Example 2.17

'there is some $k > i$'. It is not hard to see that, in the presence of X, both U and $\mathsf{U}^{<}$ have the same expressive power. In fact, the formula $\phi_1 \mathbin{\mathsf{U}} \phi_2$ is equivalent to $\phi_2 \vee (\phi_1 \mathbin{\mathsf{U}^{<}} \phi_2)$, and conversely, $\phi_1 \mathbin{\mathsf{U}^{<}} \phi_2$ is equivalent to $\phi_1 \wedge \mathsf{X}(\phi_1 \mathbin{\mathsf{U}} \phi_2)$. Similar arguments apply to the strict version $\mathsf{S}^{<}$ of S.

We continue by giving an example of a propositional LTL-formula.

Example 2.17. *Let $\phi := \mathsf{X}\, p_1 \wedge (p_2 \mathbin{\mathsf{U}} p_3)$ be a propositional LTL-formula. Consider the two propositional LTL-structures $\mathfrak{W}_1, \mathfrak{W}_2$ that are depicted in Figure 2.18 in a graphical representation. We have $\mathfrak{W}_1, 0 \models \phi$, and thus ϕ is satisfiable. Moreover, we have $\mathfrak{W}_2, 0 \not\models \phi$, but $\mathfrak{W}_2, 1 \models \phi$.* ♢

We call the temporal operators X and U *future operators*, whereas we call X^{-} and S *past operators*. Note that propositional LTL is normally defined using only future operators [Pnu77], and the extension with past operators [GPS+80] is usually called *propositional Past-LTL*. It is a well-known result, however, that the past operators do not add expressive power [GPS+80], even though some properties are easier to express using past operators [LPZ85]. Indeed, using Gabbay's separation theorem [Gab89], one can construct for each propositional LTL-formula ϕ with past operators, an equivalent propositional LTL-formula ϕ' that does not contain past operators. However, this construction is in general non-elementary in the size of ϕ, as basically the size of the constructed formula increases by one exponential for each U nested inside an S, and vice versa. This upper bound can be improved, but no constructions of size less than triply exponential in the size of ϕ are known [LMS02]. For the lower bound, it is known that past operators make propositional LTL exponentially more succinct [LMS02], i.e. there is a propositional LTL-formula ϕ with past operators such that the size of an equivalent propositional LTL-formula ϕ' without past operators is bounded by $2^{\Omega(|\phi|)}$, where $|\phi|$ denotes the size of ϕ.

Moreover, the satisfiability problem in propositional LTL is PSPACE-complete irrespective of the use of past operators [SC85; Mar04]. In the next section, we recall the connection between propositional LTL and ω-automata.

2.2.2 ω-Automata and Their Connection to Propositional LTL

For propositional LTL, the satisfiability problem can be decided by first constructing an ω-automaton for the given formula, and then testing this automaton for emptiness. In general, ω-automata accept *ω-words* over an alphabet Σ, i.e. infinite sequences of letters $w = \sigma_0\sigma_1\sigma_2\ldots$ with $\sigma_i \in \Sigma$ for every $i \geq 0$. The set of all ω-words over Σ is denoted by Σ^{ω}, and a subset L of Σ^{ω} is called *ω-language*.

There are various ω-automata models that can be employed for solving the satisfiability problem for propositional LTL such as Büchi-automata [Büc62], Muller-automata [Mul63], Rabin-automata [Rab69], and Streett-automata [Str82]. To keep the explanations simple, in this thesis we focus on (non-deterministic) Büchi-automata.

Definition 2.19 (Generalised Büchi-automaton). *A* generalised Büchi-automaton $\mathcal{G}$ *is a tuple* $\mathcal{G} = (Q, \Sigma, \Delta, Q_0, \mathcal{F})$ *consisting of a finite set of states* Q*, a finite input alphabet* Σ*, a transition relation* $\Delta \subseteq Q \times \Sigma \times Q$*, a set of initial states* $Q_0 \subseteq Q$*, and a set of sets of final states* $\mathcal{F} \subseteq 2^Q$.

Given an ω*-word* $w = \sigma_0\sigma_1\sigma_2 \ldots \in \Sigma^\omega$*, a* run *of* $\mathcal{G}$ *on* w *is an* ω*-word* $q_0q_1q_2 \ldots \in Q^\omega$ *such that* $q_0 \in Q_0$ *and* $(q_i, \sigma_i, q_{i+1}) \in \Delta$ *for every* $i \geq 0$. *This run is* accepting *if for every* $F \in \mathcal{F}$*, there are infinitely many* $i \geq 0$ *such that* $q_i \in F$. *The* language $L_\omega(\mathcal{G})$ accepted by $\mathcal{G}$ *is defined as*

$$L_\omega(\mathcal{G}) := \{w \in \Sigma^\omega \mid \textit{there is an accepting run of } \mathcal{G} \textit{ on } w\}.$$

The emptiness problem *for generalised Büchi-automata is the problem of deciding, given a generalised Büchi-automaton* $\mathcal{G}$*, whether* $L_\omega(\mathcal{G}) = \emptyset$ *or not.*

Moreover, we call $\mathcal{G}$ deterministic *if* $|Q_0| = 1$ *and for every* $q \in Q$ *and* $\sigma \in \Sigma$*, there is at most one* $q' \in Q$ *with* $(q, \sigma, q') \in \Delta$. ◇

Normal Büchi-automata are a special case of a generalised Büchi-automata where $\mathcal{F} = \{F\}$, and are denoted by $\mathcal{N} = (Q, \Sigma, \Delta, Q_0, F)$. It is common knowledge that every generalised Büchi-automaton $\mathcal{G}$, can be transformed into a Büchi automaton $\mathcal{N}$ such that $L_\omega(\mathcal{G}) = L_\omega(\mathcal{N})$ in time polynomial in the size of $\mathcal{G}$ [GPV+96; BK08].

Regarding the complexity of the emptiness problem for Büchi-automata, it is well-known that it can be solved in time polynomial in the size of the Büchi-automaton [VW94]. Together with the arguments of the previous paragraph, this yields that the emptiness problem for generalised Büchi-automata can also be solved in time polynomial in the size of the generalised Büchi-automaton.

Additionally, there is a well-known connection between (generalised) Büchi-automata and propositional LTL. In fact, given a propositional LTL-formula ϕ over $\mathcal{P}$, we can view any propositional LTL-structure $\mathfrak{W} = (w_i)_{i \geq 0}$ as an ω-word $w = w_0w_1w_2 \ldots \in \Sigma_\mathcal{P}^\omega$, where the alphabet $\Sigma_\mathcal{P}$ consists of all subsets of $\mathcal{P}$. It is well-known that one can build a generalised Büchi-automaton that accepts exactly the models of ϕ.

Definition 2.20 (Büchi-automaton for propositional LTL-formula). *Let* ϕ *be a propositional LTL-formula over* $\mathcal{P}$*, and let* $\mathcal{G}$ *be a generalised Büchi-automaton working on the alphabet* $\Sigma_\mathcal{P}$. *We define*

$$L_\omega(\phi) := \{w_0w_1w_2 \ldots \in \Sigma_\mathcal{P}^\omega \mid \mathfrak{W} = (w_i)_{i \geq 0} \textit{ is a model of } \phi\},$$

and say that $\mathcal{G}$ *is a* Büchi-automaton for ϕ *if* $L_\omega(\mathcal{G}) = L_\omega(\phi)$. ◇

If $\mathcal{G}$ is a Büchi-automaton for ϕ, then ϕ is satisfiable iff $L_\omega(\mathcal{G}) \neq \emptyset$. Thus, by constructing a Büchi-automaton for ϕ, we can reduce the satisfiability problem in propositional LTL to the emptiness problem for Büchi-automata. It is well-known that, given a propositional LTL-formula ϕ, one can construct a (generalised) Büchi-automaton for ϕ in time exponential in the size of ϕ [WVS83; VW94; LPZ85]. However, for propositional LTL-formulas involving past-operators, this construction is often not done explicitly. We include it here, and generalise it as follows. Instead of constructing a Büchi-automaton for a propositional LTL-formula ϕ

over $\mathcal{P}$, we define a Büchi-automaton that, given $n \geq 0$, accepts all ω-words $w_0 w_1 w_2 \ldots \in \Sigma_{\mathcal{P}}^{\omega}$ such that ϕ is valid in $\mathfrak{W} = (w_i)_{i \geq 0}$ at time n. This generalisation will prove to be useful in Section 5.2.2.

To define the Büchi-automaton, we need a few more notions. From now on, let ϕ be a propositional LTL-formula over $\mathcal{P}$, and let $n \geq 0$. As usual, the set of *subformulas* of ϕ is the smallest set containing all propositional LTL-formulas occurring in ϕ (including ϕ itself). We define $\mathsf{Cl_p}(\phi)$ to be the closure under negation of the set of subformulas of ϕ. In the following, we identify $\neg\neg\psi$ with ψ for every subformula ψ of ϕ. Thus, the set $\mathsf{Cl_p}(\phi)$ is of size polynomial in the size of ϕ.

Definition 2.21 (Propositional LTL-type). *Let ϕ be a propositional LTL-formula. A* propositional LTL-type *for ϕ is a set $T \subseteq \mathsf{Cl_p}(\phi)$ such that:*

- *for every $\psi_1 \wedge \psi_2 \in \mathsf{Cl_p}(\phi)$, we have $\psi_1 \wedge \psi_2 \in T$ iff $\{\psi_1, \psi_2\} \subseteq T$; and*
- *for every $\neg\psi \in \mathsf{Cl_p}(\phi)$, we have $\neg\psi \in T$ iff $\psi \notin T$.* ◇

Obviously, the set of all propositional LTL-types for a given propositional LTL-formula ϕ is exponential in the size of ϕ.

Now, we are ready to define the generalised Büchi-automaton with the above properties by equipping the set of states with a counter from $\{0, \ldots, n+1\}$. Transitions where the counter is $i = n$ ensure that ϕ is satisfied. We construct the generalised Büchi-automaton $\mathcal{G}_{\phi,n} = (Q, \Sigma_{\mathcal{P}}, \Delta, Q_0, \mathcal{F})$ as follows:

- $Q := \big\{(q,k) \mid q$ is a propositional LTL-type for ϕ, and $0 \leq k \leq n+1\big\}$;
- $((T,k),\ \sigma,\ (T',k')) \in \Delta$ iff
  - $\sigma = T \cap \mathcal{P}$;
  - for every $\mathsf{X}\,\psi \in \mathsf{Cl_p}(\phi)$, we have $\mathsf{X}\,\psi \in T$ iff $\psi \in T'$;
  - for every $\mathsf{X}^-\,\psi \in \mathsf{Cl_p}(\phi)$, we have $\mathsf{X}^-\,\psi \in T'$ iff $\psi \in T$;
  - for every $\psi_1 \,\mathsf{U}\, \psi_2 \in \mathsf{Cl_p}(\phi)$, we have $\psi_1 \,\mathsf{U}\, \psi_2 \in T$ iff (i) $\psi_2 \in T$ or (ii) $\psi_1 \in T$ and $\psi_1 \,\mathsf{U}\, \psi_2 \in T'$;
  - for every $\psi_1 \,\mathsf{S}\, \psi_2 \in \mathsf{Cl_p}(\phi)$, we have $\psi_1 \,\mathsf{S}\, \psi_2 \in T'$ iff (i) $\psi_2 \in T'$ or (ii) $\psi_1 \in T'$ and $\psi_1 \,\mathsf{S}\, \psi_2 \in T$;
  - $k = n$ implies $\phi \in T$; and
  - $k' = \begin{cases} k+1 & \text{if } k \leq n, \text{ and} \\ k & \text{otherwise;} \end{cases}$
- $Q_0 := \big\{(T,0) \mid$ there is no $\mathsf{X}^-\,\psi \in T$, and for every $\psi_1 \,\mathsf{S}\, \psi_2 \in T$, we have $\psi_2 \in T\big\}$; and
- $\mathcal{F} := \big\{F_{\psi_1 \mathsf{U} \psi_2} \times \{n+1\} \mid \psi_1 \,\mathsf{U}\, \psi_2 \in \mathsf{Cl_p}(\phi)\big\}$, where

$$F_{\psi_1 \mathsf{U} \psi_2} := \{T \mid \text{for every } \psi_1 \,\mathsf{U}\, \psi_2 \in T, \text{ we have } \psi_2 \in T\}.$$

We show now that $\mathcal{G}_{\phi,n}$ has indeed the above properties.

Lemma 2.22. *For every ω-word $w = w_0 w_1 w_2 \ldots \in \Sigma_{\mathcal{P}}^{\omega}$, we have $w \in L_\omega(\mathcal{G}_{\phi,n})$ iff ϕ is valid in the propositional LTL-structure $\mathfrak{W} = (w_i)_{i \geq 0}$ at time n.*

Proof. For the ‘only if’ direction, assume that ϕ is valid in the propositional LTL-structure $\mathfrak{W} = (w_i)_{i\geq 0}$ at time n. We define $S_i := \{\psi \in \mathsf{Cl}_\mathsf{p}(\phi) \mid \mathfrak{W}, i \models \psi\}$ for $i \geq 0$. Then,

$$(S_0, 0)(S_1, 1)\dots(S_n, n)(S_{n+1}, n+1)(S_{n+2}, n+1)\dots$$

is a run of $\mathcal{G}_{\phi,n}$ on $w_0 w_1 w_2 \dots$ due to the following reasons:

- We have $(S_i, k) \in Q$ for every $i \geq 0$ and every k, $0 \leq k \leq n+1$, since:
  - For every $\psi_1 \wedge \psi_2 \in \mathsf{Cl}_\mathsf{p}(\phi)$, we have $\mathfrak{W}, i \models \psi_1 \wedge \psi_2$ iff $\mathfrak{W}, i \models \psi_1$ and $\mathfrak{W}, i \models \psi_2$. Thus, we have $\psi_1 \wedge \psi_2 \in S_i$ iff $\{\psi_1, \psi_2\} \subseteq S_i$.
  - For every $\neg\psi \in \mathsf{Cl}_\mathsf{p}(\phi)$, we have either $\mathfrak{W}, i \models \neg\psi$ or $\mathfrak{W}, i \models \psi$. Thus, we have $\neg\psi \in S_i$ iff $\psi \notin S_i$.
- We have for every i, $0 \leq i \leq n$, that

  $$((S_i, i),\ w_i,\ (S_{i+1}, i+1)) \ \in\ \Delta,$$

  and for every $i \geq n+1$ that

  $$((S_i, n+1),\ w_i,\ (S_{i+1}, n+1)) \ \in\ \Delta$$

  since:
  - by the definition of S_i, we have $w_i = S_i \cap \mathcal{P}$;
  - for every $\mathsf{X}\psi \in \mathsf{Cl}_\mathsf{p}(\phi)$, we have $\mathsf{X}\psi \in S_i$ iff $\mathfrak{W}, i \models \mathsf{X}\psi$ iff $\mathfrak{W}, i+1 \models \psi$ iff $\psi \in S_{i+1}$;
  - for every $\mathsf{X}^-\psi \in \mathsf{Cl}_\mathsf{p}(\phi)$, we have $\mathsf{X}^-\psi \in S_{i+1}$ iff $\mathfrak{W}, i+1 \models \mathsf{X}^-\psi$ iff $\mathfrak{W}, i \models \psi$ iff $\psi \in S_i$;
  - for every $\psi_1 \mathbin{\mathsf{U}} \psi_2 \in \mathsf{Cl}_\mathsf{p}(\phi)$, we have $\psi_1 \mathbin{\mathsf{U}} \psi_2 \in S_i$ iff $\mathfrak{W}, i \models \psi_1 \mathbin{\mathsf{U}} \psi_2$ iff (i) $\mathfrak{W}, i \models \psi_2$ or (ii) $\mathfrak{W}, i \models \psi_1$ and $\mathfrak{W}, i+1 \models \psi_1 \mathbin{\mathsf{U}} \psi_2$ iff (i) $\psi_2 \in S_i$ or (ii) $\psi_1 \in S_i$ and $\psi_1 \mathbin{\mathsf{U}} \psi_2 \in S_{i+1}$;
  - for every $\psi_1 \mathbin{\mathsf{S}} \psi_2 \in \mathsf{Cl}_\mathsf{p}(\phi)$, we have $\psi_1 \mathbin{\mathsf{S}} \psi_2 \in S_{i+1}$ iff $\mathfrak{W}, i+1 \models \psi_1 \mathbin{\mathsf{S}} \psi_2$ iff (i) $\mathfrak{W}, i+1 \models \psi_2$ or (ii) $\mathfrak{W}, i+1 \models \psi_1$ and $\mathfrak{W}, i \models \psi_1 \mathbin{\mathsf{S}} \psi_2$ iff (i) $\psi_2 \in S_{i+1}$ or (ii) $\psi_1 \in S_{i+1}$ and $\psi_1 \mathbin{\mathsf{S}} \psi_2 \in S_i$;
  - for $i = n$, we have $\phi \in S_i$ since $\mathfrak{W}, n \models \phi$; and
  - the condition for incrementing the second component of a state (until $n+1$ is reached) is obviously also satisfied.
- We have for every $\mathsf{X}^-\psi \in \mathsf{Cl}_\mathsf{p}(\phi)$ that $\mathfrak{W}, 0 \not\models \mathsf{X}^-\psi$, and thus $\mathsf{X}^-\psi \notin S_0$. Additionally, we have for every $\psi_1 \mathbin{\mathsf{S}} \psi_2 \in S_0$, since $\mathfrak{W}, 0 \models \psi_1 \mathbin{\mathsf{S}} \psi_2$, also $\mathfrak{W}, 0 \models \psi_2$, and thus $\psi_2 \in S_0$. This yields that $(S_0, 0) \in Q_0$.

Moreover, the above run is accepting. We prove this by contradiction. Suppose that for some $\psi_1 \mathbin{\mathsf{U}} \psi_2 \in \mathsf{Cl}_\mathsf{p}(\phi)$, the set $\{i \geq 0 \mid S_i \in F_{\psi_1 \mathsf{U} \psi_2}\}$ is finite. Then there exists some $k \geq 0$ such that $S_\ell \notin F_{\psi_1 \mathsf{U} \psi_2}$ for every $\ell \geq k$. This means $\psi_1 \mathbin{\mathsf{U}} \psi_2 \in S_\ell$ and $\psi_2 \notin S_\ell$ for every $\ell \geq k$. Hence, $\mathfrak{W}, k \models \psi_1 \mathbin{\mathsf{U}} \psi_2$ and $\mathfrak{W}, \ell \not\models \psi_2$ for every $\ell \geq k$. This contradicts the semantics of U.

For the ‘if’ direction, assume that $w_0 w_1 w_2 \dots \in L_\omega(\mathcal{G}_{\phi_n})$, i.e. there is an accepting run

$$(S_0, 0)(S_1, 1)\dots(S_n, n)(S_{n+1}, n+1)(S_{n+2}, n+1)\dots$$

of $\mathcal{G}_{\phi,n}$ on $w_0 w_1 w_2 \dots$. We show that ϕ is valid in $\mathfrak{W} := (w_i)_{i \geq 0}$ at time n. We have $\phi \in S_n$ by the definition of Δ. Thus, it is enough to show that for every $\psi \in \mathsf{Cl}_{\mathsf{p}}(\phi)$ and every $i \geq 0$, we have $\psi \in S_i$ iff $\mathfrak{W}, i \models \psi$. This can be shown by induction on the structure of ψ using the definition of Δ.

- If $\psi \in \mathcal{P}$, we have $\psi \in S_i$ iff $\psi \in w_i$ iff $\mathfrak{W}, i \models \psi$.
- If $\psi = \neg\chi$, we have $\neg\chi \in S_i$ iff $\chi \notin S_i$ iff $\mathfrak{W}, i \not\models \chi$ iff $\mathfrak{W}, i \models \neg\chi$.
- If $\psi = \chi_1 \wedge \chi_2$, we have $\chi_1 \wedge \chi_2 \in S_i$ iff $\{\chi_1, \chi_2\} \subseteq S_i$ iff $\mathfrak{W}, i \models \chi_1$ and $\mathfrak{W}, i \models \chi_2$ iff $\mathfrak{W}, i \models \chi_1 \wedge \chi_2$.
- If $\psi = \mathsf{X}\,\chi$, we have $\mathsf{X}\,\chi \in S_i$ iff $\chi \in S_{i+1}$ iff $\mathfrak{W}, i+1 \models \chi$ iff $\mathfrak{W}, i \models \mathsf{X}\,\chi$.
- If $\psi = \mathsf{X}^-\,\chi$, we have $\mathsf{X}^-\,\chi \in S_i$ iff $i > 0$ and $\chi \in S_{i-1}$ iff $i > 0$ and $\mathfrak{W}, i-1 \models \chi$ iff $\mathfrak{W}, i \models \mathsf{X}^-\,\chi$. The first iff holds because of the definition of Q_0.
- If $\psi = \chi_1 \,\mathsf{U}\, \chi_2$, we prove $\chi_1 \,\mathsf{U}\, \chi_2 \in S_i$ iff $\mathfrak{W}, i \models \chi_1 \,\mathsf{U}\, \chi_2$ as follows. For the 'if' direction, assume that $\mathfrak{W}, i \models \chi_1 \,\mathsf{U}\, \chi_2$. Then there is some $k \geq i$ such that $\mathfrak{W}, k \models \chi_2$ and $\mathfrak{W}, \ell \models \chi_1$ for every ℓ, $i \leq \ell < k$. We show by induction on j that $\chi_1 \,\mathsf{U}\, \chi_2 \in S_{k-j}$ for j, $j \leq k - i$.

  For $j = 0$, we have: $\mathfrak{W}, k \models \chi_2$ implies $\chi_2 \in S_k$ by the outer induction hypothesis, and the definition of Δ yields $\chi_1 \,\mathsf{U}\, \chi_2 \in S_k$.

  For $j > 0$, we have: $\mathfrak{W}, k - j \models \chi_1$ implies $\chi_1 \in S_{k-j}$ by the outer induction hypothesis. By the inner induction hypothesis, we have $\chi_1 \,\mathsf{U}\, \chi_2 \in S_{k-j+1}$. Thus, by the definition of Δ, it follows that $\chi_1 \,\mathsf{U}\, \chi_2 \in S_{k-j}$.

  For the 'only if' direction, assume that $\chi_1 \,\mathsf{U}\, \chi_2 \in S_i$. Since states of $F_{\chi_1 \mathsf{U} \chi_2} \times \{n+1\}$ occur infinitely often among $S_0, S_1, S_2, \dots$, there is some $k \geq i$ such that $S_k \in F_{\chi_1 \mathsf{U} \chi_2}$. Let k be the smallest index with that property. Then it follows that $\chi_1 \,\mathsf{U}\, \chi_2 \in S_\ell$ and $\chi_2 \notin S_\ell$ for every ℓ, $i \leq \ell < k$.

  Since $\chi_1 \,\mathsf{U}\, \chi_2 \in S_\ell$ and $\chi_2 \notin S_\ell$ for every ℓ, $i \leq \ell < k$, we have $\chi_1 \in S_\ell$ because of the definition of Δ. Thus, $\mathfrak{W}, \ell \models \chi_1$ for every ℓ, $i \leq \ell < k$ $(*)$.

  Moreover, $\chi_1 \,\mathsf{U}\, \chi_2 \in S_{k-1}$ and $\chi_2 \notin S_{k-1}$ imply $\chi_1 \,\mathsf{U}\, \chi_2 \in S_k$ because of the definition of Δ. This yields $\chi_2 \in S_k$ since $S_k \in F_{\chi_1 \mathsf{U} \chi_2}$, and thus $\mathfrak{W}, k \models \chi_2$ $(**)$.

  Finally, $(*)$ and $(**)$ yield that $\mathfrak{W}, i \models \chi_1 \,\mathsf{U}\, \chi_2$ by the semantics of U.
- If $\psi = \chi_1 \,\mathsf{S}\, \chi_2$, we prove $\chi_1 \,\mathsf{S}\, \chi_2 \in S_i$ iff $\mathfrak{W}, i \models \chi_1 \,\mathsf{S}\, \chi_2$ similarly. For the 'if' direction, assume that $\mathfrak{W}, i \models \chi_1 \,\mathsf{S}\, \chi_2$. Then there is some k, $0 \leq k \leq i$ such that $\mathfrak{W}, k \models \chi_2$ and $\mathfrak{W}, \ell \models \chi_1$ for every ℓ, $k < \ell \leq i$. We show by induction on j that $\chi_1 \,\mathsf{S}\, \chi_2 \in S_{k+j}$ for j, $j \leq i - k$.

  For $j = 0$, we have: $\mathfrak{W}, k \models \chi_2$ implies $\chi_2 \in S_k$ by the outer induction hypothesis, and the definition of Δ yields $\chi_1 \,\mathsf{S}\, \chi_2 \in S_k$.

  For $j > 0$, we have: $\mathfrak{W}, k + j \models \chi_1$ implies $\chi_1 \in S_{k+j}$ by the outer induction hypothesis. By the inner induction hypothesis, we have $\chi_1 \,\mathsf{S}\, \chi_2 \in S_{k+j-1}$. Thus, by the definition of Δ, it follows that $\chi_1 \,\mathsf{S}\, \chi_2 \in S_{k+j}$.

  For the 'only if' direction, assume that $\chi_1 \,\mathsf{S}\, \chi_2 \in S_i$. There are two cases to consider: either $i = 0$ or $i > 0$.

  For $i = 0$, we have: $\chi_1 \,\mathsf{S}\, \chi_2 \in S_0$ implies $\chi_2 \in S_0$ by the definition of Q_0. This yields $\mathfrak{W}, 0 \models \chi_2$, and thus $\mathfrak{W}, 0 \models \chi_1 \,\mathsf{S}\, \chi_2$.

For $i > 0$, we have again two cases: either $\chi_2 \in S_i$ or $\chi_1 \in S_i$ and $\chi_1 \mathsf{S}\, \chi_2 \in S_{i-1}$. For the case where $\chi_1 \in S_i$, it directly follows that $\mathfrak{W}, i \models \chi_1 \mathsf{S}\, \chi_2$. For the other case where $\chi_1 \in S_i$ and $\chi_1 \mathsf{S}\, \chi_2 \in S_{i-1}$, we have by the inner induction hypothesis that $\mathfrak{W}, i-1 \models \chi_1 \mathsf{S}\, \chi_2$. Thus, there is a k, $0 \leq k \leq i-1$, such that $\mathfrak{W}, k \models \chi_2$ and $\mathfrak{W}, j \models \chi_1$ for every j, $k < j \leq i-1$. Since we have by the outer induction hypothesis also that $\mathfrak{W}, i \models \chi_1$, it follows that there is some k, $0 \leq k \leq i$, such that $\mathfrak{W}, k \models \chi_2$ and $\mathfrak{W}, j \models \chi_1$ for every j, $k < j \leq i$. Hence, $\mathfrak{W}, i \models \chi_1 \mathsf{S}\, \chi_2$. □

We immediately obtain the following corollary.

Corollary 2.23. *The generalised Büchi-automaton $\mathcal{G}_{\phi,0}$ is a Büchi-automaton for ϕ.*

Proof. By Lemma 2.22, we have

$$L_\omega(\mathcal{G}_{\phi,0}) = \{w_0 w_1 w_2 \ldots \in \Sigma_{\mathcal{P}}^\omega \mid \mathfrak{W} = (w_i)_{i \geq 0} \text{ is a model of } \phi\} = L_\omega(\phi). \qquad \square$$

As already mentioned above, one can transform the Büchi-automaton $\mathcal{G}_{\phi,n}$ into a 'normal' Büchi-automaton $\mathcal{N}_{\phi,n}$ such that $L_\omega(\mathcal{G}_{\phi,n}) = L_\omega(\mathcal{N}_{\phi,n})$ in time polynomial in the size of $\mathcal{G}_{\phi,n}$ [GPV+96; BK08]. An analysis of the construction of $\mathcal{G}_{\phi,n}$ yields the following lemma.

Lemma 2.24. *The Büchi-automaton $\mathcal{N}_{\phi,n}$ is of size exponential in the size of ϕ and polynomial in n, and can be constructed in time exponential in the size of ϕ and polynomial in n.*

Proof. Note that $\mathcal{G}_{\phi,n}$, and thus $\mathcal{N}_{\phi,n}$, have exponentially many states in the size of ϕ and linearly many states in the size of n, and each state can be represented using only space polynomial in the size of ϕ and n. Moreover, the alphabet $\Sigma_{\mathcal{P}_\phi}$ is exponential in the size of ϕ. The set of final states of $\mathcal{G}_{\phi,n}$ contains linearly many sets of size at most exponential in ϕ, while the size of the set of initial states and the transition relation is bounded polynomially in the size of the set of states, which is exponential in the size of ϕ and linear in n.

Since all conditions that need to be checked to construct the components of $\mathcal{G}_{\phi,n}$ can be checked in time exponential in the size of ϕ and polynomial in n, and $\mathcal{N}_{\phi,n}$ can be constructed in time polynomial in the size of $\mathcal{G}_{\phi,n}$, we obtain the claim of the lemma. □

We will use the result of this lemma later in Section 5.2.2. Note that instead of constructing the Büchi-automaton $\mathcal{N}_{\phi,n}$, we could also omit the counter and construct a Büchi-automaton for $\mathsf{X}^n\, \phi$, e.g. $\mathcal{N}_{\mathsf{X}^n\, \phi,0}$, that accepts the same ω-language. However, the size of this Büchi-automaton is exponential in the size of $\mathsf{X}^n\, \phi$, and thus also exponential in n. With the construction above, we have shown that we can do better.

Regarding the satisfiability problem in propositional LTL, Lemma 2.24 yields the following. If we first compute an exponentially large Büchi-automaton for ϕ and then apply the emptiness test for Büchi-automata, we obtain an ExpTime decision procedure for the satisfiability problem. In order to reduce the complexity to PSpace, one has to generate the relevant parts of the Büchi-automaton on-the-fly while performing the emptiness check [SC85; LPZ85].

It can be shown that, in the worst case, an exponential blow-up in the construction of the Büchi-automaton for a propositional LTL-formula cannot be avoided (see Theorem 5.42 in [BK08] for a proof). However, there are optimised implementations of the construction that try to keep the number of states as small as possible (see e.g. [GO01; GO03; GPV+96]). Experiments with these implementations show that an exponential blow-up can frequently be

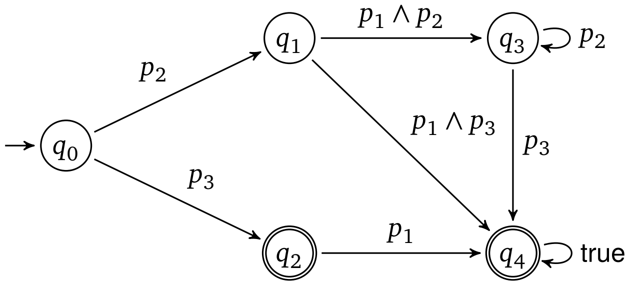

Figure 2.26: A Büchi-automaton for the propositional LTL-formula ϕ of Example 2.17

avoided. For example, the tool *LTL2BA*[2] is widely used in practice to generate Büchi-automata from propositional LTL-formulas.[3]

Example 2.25. *Reconsider the propositional LTL-formula $\phi = \mathsf{X}\, p_1 \wedge (p_2 \mathsf{U}\, p_3)$ of Example 2.17. Figure 2.26 depicts a Büchi-automaton for this formula, which was generated by LTL2BA. Note that edges with propositional formulas ψ as labels are used as abbreviations for sets of edges labelled with those subsets of $\mathcal{P}$ that represent models of ψ. For example, the edge with label $p_1 \wedge p_2$ from q_1 to q_3 stands for two edges between these states, one with label $\{p_1, p_2\}$ and one with label $\{p_1, p_2, p_3\}$.*

This automaton accepts, for example, the ω-word $\{p_2\}\{p_1, p_2\}\{p_2\}\{p_3\}\emptyset\ldots$, which corresponds to the propositional LTL-structure $\mathfrak{W}_1$ of Example 2.17 with $\mathfrak{W}_1, 0 \models \phi$. $\diamondsuit$

In this chapter, we have introduced description logics. Moreover, we have given the basic definitions of propositional LTL and have recalled its relationship with ω-automata. This sets a basis for the later chapters.

[2] See `http://www.lsv.ens-cachan.fr/~gastin/ltl2ba/`.

[3] Unfortunately, like most other such tools, LTL2BA does *not* support propositional LTL *with past-operators*.

Chapter 3

The Temporalised Description Logic $\mathcal{SHOQ}$-LTL

Temporalised description logics extend description logics with temporal modalities. As discussed in Section 1.2, there are many different approaches to temporalising description logics. In this thesis, we follow the approach that was taken by introducing $\mathcal{ALC}$-LTL [BGL12], i.e. a combination of $\mathcal{ALC}$ with propositional LTL where $\mathcal{ALC}$-axioms replace propositional variables, and temporal operators are only allowed to occur in front of $\mathcal{ALC}$-axioms rather than inside of concepts.

The temporalised description logic $\mathcal{SHOQ}$-LTL, which we examine in this chapter, generalises $\mathcal{ALC}$-LTL. In fact, several constructions in the present chapter are adaptations of those for $\mathcal{ALC}$-LTL, in particular the ones used to show Lemmas 4.3 and 6.4 in [BGL12]. Some of the results of this chapter have already been published in [BL14].

This chapter is organised as follows. In Section 3.1, we formally introduce the temporalised description logic $\mathcal{SHOQ}$-LTL. Then, in Section 3.2, we show complexity results for the satisfiability problem in $\mathcal{SHOQ}$-LTL. Finally, we provide a brief summary of the obtained results in Section 3.3.

3.1 Syntax and Semantics of $\mathcal{SHOQ}$-LTL

The temporalised description logic $\mathcal{SHOQ}$-LTL combines the description logic $\mathcal{SHOQ}$ with the temporal logic LTL. Its syntax is very similar to the one of propositional LTL (see Definition 2.15), but in $\mathcal{SHOQ}$-LTL propositional variables are replaced by $\mathcal{SHOQ}$-axioms.

Definition 3.1 (Syntax of $\mathcal{SHOQ}$-LTL). *Let $\mathcal{R}$ be an RBox. The set of $\mathcal{SHOQ}$-*LTL*-formulas* w.r.t. *$\mathcal{R}$ is the smallest set such that*

- *every ABox-axiom and every TBox-axiom in which at-least restrictions contain only simple roles w.r.t. $\mathcal{R}$ is a $\mathcal{SHOQ}$-LTL-formula w.r.t. $\mathcal{R}$; and*
- *if ϕ_1 and ϕ_2 are $\mathcal{SHOQ}$-LTL-formulas w.r.t. $\mathcal{R}$, then so are: $\neg\phi_1$ (negation), $\phi_1 \wedge \phi_2$ (conjunction), $\mathsf{X}\,\phi_1$ (next), $\mathsf{X}^-\,\phi_1$ (previous), $\phi_1\,\mathsf{U}\,\phi_2$ (until), and $\phi_1\,\mathsf{S}\,\phi_2$ (since).* ◇

We denote the set of axioms occurring in a $\mathcal{SHOQ}$-LTL-formula ϕ by $\mathsf{Ax}(\phi)$. Clearly, the cardinality of $\mathsf{Ax}(\phi)$ is bounded by the size of ϕ. Similar to propositional LTL, we use

- $\phi_1 \vee \phi_2$ (disjunction) as an abbreviation for $\neg(\neg\phi_1 \wedge \neg\phi_2)$;
- $\phi_1 \rightarrow \phi_2$ (implication) as an abbreviation for $\neg\phi_1 \vee \phi_2$;

- true as an abbreviation for an arbitrary but fixed tautology such as $A(a) \vee \neg A(a)$ with $A \in \mathsf{N_C}$ and $a \in \mathsf{N_I}$;
- false as an abbreviation for $\neg$true;
- $\Diamond\phi$ (diamond, which should be read as 'eventually' or 'some time in the future') as an abbreviation for true $\mathsf{U}\ \phi$;
- $\Box\phi$ (box, which should be read as 'always' or 'always in the future') as an abbreviation for $\neg\Diamond\neg\phi$;
- $\Diamond^-\ \phi$ (which should be read as 'once' or 'some time in the past') as an abbreviation for true $\mathsf{S}\ \phi$; and
- $\Box^-\ \phi$ (which should be read as 'historically' or 'always in the past') as an abbreviation for $\neg\ \Diamond^-\neg\phi$.

The semantics of $\mathcal{SHOQ}$-LTL is based on DL-LTL-structures, which are infinite sequences of interpretations over the same non-empty domain Δ (*constant-domain assumption*). As discussed in Section 1.2, for some concept and role names it is also not desirable that their interpretation changes over time. Thus, we assume in the following that a subset of the set of concept and role names can be designated as being rigid. Let $\mathsf{N_{RC}}$ denote the set of *rigid concept names* and $\mathsf{N_{RR}}$ the set of *rigid role names* where $\mathsf{N_{RC}} \subseteq \mathsf{N_C}$ and $\mathsf{N_{RR}} \subseteq \mathsf{N_R}$. All concept and role names in $\mathsf{N_C} \setminus \mathsf{N_{RC}}$ and $\mathsf{N_R} \setminus \mathsf{N_{RR}}$ are then called *flexible*. Moreover, we make the rigid-individual assumption, i.e. we assume that every individual name stands for a unique element of the domain Δ.

Definition 3.2 (DL-LTL-structure). *We call an infinite sequence $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$ of interpretations $\mathcal{I}_i = (\Delta, \cdot^{\mathcal{I}_i})$ a DL-LTL-structure if*

- $a^{\mathcal{I}_i} = a^{\mathcal{I}_j}$ *holds for every $a \in \mathsf{N_I}$ and all $i, j \geq 0$ (*rigid-individual assumption*);*
- $A^{\mathcal{I}_i} = A^{\mathcal{I}_j}$ *holds for every $A \in \mathsf{N_{RC}}$ and all $i, j \geq 0$; and*
- $r^{\mathcal{I}_i} = r^{\mathcal{I}_j}$ *holds for every $r \in \mathsf{N_{RR}}$ and all $i, j \geq 0$.* $\Diamond$

This notion is now used to define the semantics of $\mathcal{SHOQ}$-LTL-formulas.

Definition 3.3 (Semantics of $\mathcal{SHOQ}$-LTL). *Given a $\mathcal{SHOQ}$-LTL-formula ϕ w.r.t. an RBox $\mathcal{R}$, a DL-LTL-structure $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$, and a time point $i \geq 0$,* validity of ϕ in $\mathfrak{I}$ at time *i (written $\mathfrak{I}, i \models \phi$) is defined inductively as follows:*

$$
\begin{array}{lll}
\mathfrak{I}, i \models \alpha & \textit{iff} & \mathcal{I}_i \models \alpha \\
\mathfrak{I}, i \models \neg\phi_1 & \textit{iff} & \mathfrak{I}, i \not\models \phi_1,\ \textit{i.e. not}\ \mathfrak{I}, i \models \phi_1 \\
\mathfrak{I}, i \models \phi_1 \wedge \phi_2 & \textit{iff} & \mathfrak{I}, i \models \phi_1\ \textit{and}\ \mathfrak{I}, i \models \phi_2 \\
\mathfrak{I}, i \models \mathsf{X}\,\phi_1 & \textit{iff} & \mathfrak{I}, i+1 \models \phi_1 \\
\mathfrak{I}, i \models \mathsf{X}^-\,\phi_1 & \textit{iff} & i > 0\ \textit{and}\ \mathfrak{I}, i-1 \models \phi_1 \\
\mathfrak{I}, i \models \phi_1\,\mathsf{U}\,\phi_2 & \textit{iff} & \textit{there is some}\ k \geq i\ \textit{such that}\ \mathfrak{I}, k \models \phi_2,\ \textit{and} \\
 & & \mathfrak{I}, j \models \phi_1\ \textit{for every}\ j,\ i \leq j < k \\
\mathfrak{I}, i \models \phi_1\,\mathsf{S}\,\phi_2 & \textit{iff} & \textit{there is some}\ k,\ 0 \leq k \leq i,\ \textit{such that}\ \mathfrak{I}, k \models \phi_2,\ \textit{and} \\
 & & \mathfrak{I}, j \models \phi_1\ \textit{for every}\ j,\ k < j \leq i
\end{array}
$$

Table 3.4: The complexity of the satisfiability problem in $\mathcal{SHOQ}$-LTL

Setting (i)	ExpTime-complete	(Theorems 3.5 and 3.15)
Setting (ii)	NExpTime-complete	(Theorems 3.5 and 3.20)
Setting (iii)	2ExpTime-complete	(Theorems 3.5 and 3.17)

Settings: (i) neither concept names nor role names are allowed to be rigid; (ii) only concept names may be rigid; and (iii) both concept names and role names may be rigid.

If $\mathcal{I}_i \models \mathcal{R}$ for every $i \geq 0$ (written $\mathfrak{I} \models \mathcal{R}$), and $\mathfrak{I}, 0 \models \phi$, then we call $\mathfrak{I}$ a model *of ϕ w.r.t. $\mathcal{R}$. We call ϕ* satisfiable w.r.t. $\mathcal{R}$ *if it has a model w.r.t. $\mathcal{R}$.*

The satisfiability problem in $\mathcal{SHOQ}$-LTL *is the problem of deciding, given a $\mathcal{SHOQ}$-LTL-formula ϕ w.r.t. an RBox $\mathcal{R}$, whether ϕ is satisfiable w.r.t. $\mathcal{R}$.*

Moreover, two $\mathcal{SHOQ}$-LTL-formulas ϕ_1, ϕ_2 w.r.t. an RBox $\mathcal{R}$ are equivalent *(written $\phi_1 \equiv \phi_2$) if they have the same models w.r.t. $\mathcal{R}$.* ◇

In [BGL12], the temporalised description logic $\mathcal{ALC}$-LTL, which is a fragment of $\mathcal{SHOQ}$-LTL, is considered. There, the authors show that satisfiability in $\mathcal{ALC}$-LTL is ExpTime-complete if no rigid names are present. If rigid concept names are allowed, the problem becomes NExpTime-complete, and if additionally rigid role names are allowed, the problem becomes even 2ExpTime-complete.[1] In the next section, we show that the same complexity bounds also apply to the satisfiability problem in $\mathcal{SHOQ}$-LTL (see Table 3.4).[2]

The results obtained in the next sections are used in later chapters of this thesis. For instance, the ideas ideas underlying the decision procedures to show the complexity upper bounds for the cases without rigid names and with rigid concept and role names can also be used to obtain automata-based decision procedures. These constructions of ω-automata are then used in Chapter 4. Moreover, the proof ideas and in particular the results of Section 3.2.4 are used in Chapter 5.

3.2 The Complexity of Satisfiability in $\mathcal{SHOQ}$-LTL

In this section, we examine the complexity of the satisfiability problem in $\mathcal{SHOQ}$-LTL. We consider three different settings: (i) neither concept names nor role names are allowed to be rigid, i.e. $\mathsf{N_{RC}} = \mathsf{N_{RR}} = \emptyset$, (ii) only concept names may be rigid, i.e. $\mathsf{N_{RC}} \neq \emptyset$ and $\mathsf{N_{RR}} = \emptyset$, and (iii) both concept and role names may be rigid, i.e. $\mathsf{N_{RC}} \neq \emptyset$ and $\mathsf{N_{RR}} \neq \emptyset$. It is well-known that one can simulate rigid concept names by rigid role names [BGL12], which is why there are only three cases to consider. The results of this section are summarised in Table 3.4.

Since $\mathcal{ALC}$-LTL is a fragment of $\mathcal{SHOQ}$-LTL, we immediately obtain the following lower bounds for the satisfiability problem.

[1] Additional intermediate cases (such as the case where GCIs occurring in the formula are global, i.e. required to hold at every point in time; or where only the temporal operator ◇ may be used) are also considered in [BGL12], but are not considered in this thesis.

[2] Thus, the complexity of the satisfiability problem in $\mathcal{L}$-LTL is the same as of $\mathcal{ALC}$-LTL for any description logic $\mathcal{L}$ between $\mathcal{ALC}$ and $\mathcal{SHOQ}$.

Theorem 3.5. *The satisfiability problem in $\mathcal{SHOQ}$-LTL is*

1. ExpTime*-hard if* $\mathsf{N_{RC}} = \mathsf{N_{RR}} = \emptyset$*;*
2. NExpTime*-hard if* $\mathsf{N_{RC}} \neq \emptyset$ *and* $\mathsf{N_{RR}} = \emptyset$*; and*
3. 2ExpTime*-hard if* $\mathsf{N_{RC}} \neq \emptyset$ *and* $\mathsf{N_{RR}} \neq \emptyset$*.*

Proof. We reduce the satisfiability problem in $\mathcal{ALC}$-LTL. This problem is ExpTime-complete if $\mathsf{N_{RC}} = \mathsf{N_{RR}} = \emptyset$, NExpTime-complete if $\mathsf{N_{RC}} \neq \emptyset$ and $\mathsf{N_{RR}} = \emptyset$, and 2ExpTime-complete if $\mathsf{N_{RC}} \neq \emptyset$ and $\mathsf{N_{RR}} \neq \emptyset$ as shown in [BGL12].

Let now ϕ be an $\mathcal{ALC}$-LTL-formula. Obviously, ϕ is a $\mathcal{SHOQ}$-LTL-formula w.r.t. the empty RBox. Thus, we have that the $\mathcal{ALC}$-LTL-formula ϕ is satisfiable iff the $\mathcal{SHOQ}$-LTL-formula ϕ is satisfiable w.r.t. $\emptyset$. □

To obtain the corresponding upper bounds, we reduce the satisfiability problem in $\mathcal{SHOQ}$-LTL to two separate satisfiability problems. For that, we use the idea of Lemma 4.3 in [BGL12], where this was done for $\mathcal{ALC}$-LTL.

In the following, let $\mathcal{R}$ be an RBox, and let ϕ be a $\mathcal{SHOQ}$-LTL-formula w.r.t. $\mathcal{R}$. For the first satisfiability problem is called *t-satisfiability*, because it takes care of the *temporal* structure of ϕ. For this, we consider the propositional abstraction. The propositional abstraction of ϕ is constructed by replacing each axiom occurring in ϕ with a propositional variable such that there is a 1–1 relationship between the axioms $\alpha_1, \ldots, \alpha_m$ occurring in ϕ and the propositional variables $p_1, \ldots, p_m$ occurring in its abstraction.

Definition 3.6 (Propositional abstraction). *Let $\mathcal{R}$ be an RBox, and let ϕ be a $\mathcal{SHOQ}$-LTL-formula w.r.t. $\mathcal{R}$. Furthermore, let $\mathcal{P}_\phi$ be a finite set of propositional variables such that there is a bijection* $\mathsf{p}\colon \mathsf{Ax}(\phi) \to \mathcal{P}_\phi$*.*

1. *The propositional LTL-formula ϕ^{p} is obtained from ϕ by replacing every occurrence of an axiom α in ϕ by its* p*-image* $\mathsf{p}(\alpha)$*. We call ϕ^{p} the* propositional abstraction of ϕ w.r.t. p.
2. *Given a DL-LTL-structure $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$, its* propositional abstraction w.r.t. p *is the propositional LTL-structure $\mathfrak{I}^{\mathsf{p}} = (w_i)_{i\geq 0}$ with*

$$w_i := \big\{\mathsf{p}(\alpha) \mid \alpha \in \mathsf{Ax}(\phi) \text{ and } \mathcal{I}_i \models \alpha\big\}$$

for every $i \geq 0$. ◇

In the following, we assume that $\mathsf{p}\colon \mathsf{Ax}(\phi) \to \mathcal{P}_\phi$ is a bijection.[3] For simplicity, for a subformula ψ of ϕ, we denote by ψ^{p} the propositional abstraction of ψ w.r.t. the restriction of p to $\mathsf{Ax}(\psi)$. We now give an example of a propositional abstraction.

Example 3.7. *Let $\phi_{ex} := \mathsf{X}\big(A(a)\big) \wedge \big((A \sqsubseteq B)\,\mathsf{U}\,((\neg B)(a))\big)$ be a $\mathcal{SHOQ}$-LTL-formula w.r.t. the empty RBox, and let* $\mathsf{p}\colon \mathsf{Ax}(\phi_{ex}) \to \{p_1, p_2, p_3\}$ *be the bijection that maps $A(a)$ to p_1, $A \sqsubseteq B$ to p_2, and $(\neg B)(a)$ to p_3. Then the propositional LTL-formula $\mathsf{X}\,p_1 \wedge (p_2 \mathsf{U} p_3)$ is the propositional abstraction of ϕ_{ex} w.r.t.* p. ◇

[3] Obviously, for every $\mathcal{SHOQ}$-LTL-formula ϕ w.r.t. an RBox $\mathcal{R}$, there is a finite set $\mathcal{P}_\phi$ of propositional variables such that a bijection $\mathsf{p}\colon \mathsf{Ax}(\phi) \to \mathcal{P}_\phi$ exists.

The propositional abstraction ϕ^{p} of ϕ w.r.t. p allows us to analyse the temporal structure of ϕ separately from the DL-component. The following lemma states the relationships between ϕ and its propositional abstraction ϕ^{p}.

Lemma 3.8. *Let $\mathfrak{I}$ be a DL-LTL-structure with $\mathfrak{I} \models \mathcal{R}$. Then, $\mathfrak{I}$ is a model of ϕ w.r.t. $\mathcal{R}$ iff $\mathfrak{I}^{\mathsf{p}}$ is a model of ϕ^{p}.*

Proof. Let $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$ be a DL-LTL-structure with $\mathfrak{I} \models \mathcal{R}$, and $\mathfrak{I}^{\mathsf{p}} = (w_i)_{i\geq 0}$ its propositional abstraction w.r.t. p. We prove this lemma by showing that $\mathfrak{I}, i \models \phi$ iff $\mathfrak{I}^{\mathsf{p}}, i \models \phi^{\mathsf{p}}$ for every $i \geq 0$ by induction of the structure of ϕ.

For the base case, let ϕ be an axiom. Then, we have for every $i \geq 0$ that $\mathfrak{I}, i \models \phi$ *iff* $\mathcal{I}_i \models \phi$ *iff* $\mathsf{p}(\phi) \in w_i$ *iff* $w_i \models \phi^{\mathsf{p}}$ *iff* $\mathfrak{I}^{\mathsf{p}}, i \models \phi^{\mathsf{p}}$.

If ϕ is of the form $\neg\phi_1$, we have for every $i \geq 0$ that $\mathfrak{I}, i \models \neg\phi_1$ *iff* $\mathfrak{I}, i \not\models \phi_1$ *iff* $\mathfrak{I}^{\mathsf{p}}, i \not\models \phi_1^{\mathsf{p}}$ *iff* $\mathfrak{I}^{\mathsf{p}} \models (\neg\phi_1)^{\mathsf{p}}$.

If ϕ is of the form $\phi_1 \wedge \phi_2$, we have for every $i \geq 0$ that $\mathfrak{I}, i \models \phi_1 \wedge \phi_2$ *iff* $\mathfrak{I}, i \models \phi_1$ and $\mathfrak{I}, i \models \phi_2$ *iff* $\mathfrak{I}^{\mathsf{p}}, i \models \phi_1^{\mathsf{p}}$ and $\mathfrak{I}^{\mathsf{p}}, i \models \phi_2^{\mathsf{p}}$ *iff* $\mathfrak{I}^{\mathsf{p}}, i \models (\phi_1 \wedge \phi_2)^{\mathsf{p}}$.

If ϕ is of the form $\mathsf{X}\,\phi_1$, we have for every $i \geq 0$ that $\mathfrak{I}, i \models \mathsf{X}\,\phi_1$ *iff* $\mathfrak{I}, i+1 \models \phi_1$ *iff* $\mathfrak{I}^{\mathsf{p}}, i+1 \models \phi_1^{\mathsf{p}}$ *iff* $\mathfrak{I}^{\mathsf{p}}, i \models (\mathsf{X}\,\phi_1)^{\mathsf{p}}$.

If ϕ is of the form $\mathsf{X}^-\,\phi_1$, we have for every $i \geq 0$ that $\mathfrak{I}, i \models \mathsf{X}^-\,\phi_1$ *iff* $i > 0$ and $\mathfrak{I}, i{-}1 \models \phi_1$ *iff* $i > 0$ and $\mathfrak{I}^{\mathsf{p}}, i-1 \models \phi_1^{\mathsf{p}}$ *iff* $\mathfrak{I}^{\mathsf{p}}, i \models (\mathsf{X}^-\,\phi_1)^{\mathsf{p}}$.

If ϕ is of the form $\phi_1 \,\mathsf{U}\, \phi_2$, we have for every $i \geq 0$ that $\mathfrak{I}, i \models \phi_1 \,\mathsf{U}\, \phi_2$ *iff* there is some $k \geq i$ such that $\mathfrak{I}, k \models \phi_2$, and $\mathfrak{I}, j \models \phi_1$ for every j, $i \leq j < k$ *iff* there is some $k \geq i$ such that $\mathfrak{I}^{\mathsf{p}}, k \models \phi_2^{\mathsf{p}}$, and $\mathfrak{I}^{\mathsf{p}}, j \models \phi_1^{\mathsf{p}}$ for every j, $i \leq j < k$ *iff* $\mathfrak{I}^{\mathsf{p}}, i \models (\phi_1 \,\mathsf{U}\, \phi_2)^{\mathsf{p}}$.

Finally, if ϕ is of the form $\phi_1 \,\mathsf{S}\, \phi_2$, we have for every $i \geq 0$ that $\mathfrak{I}, i \models \phi_1 \,\mathsf{S}\, \phi_2$ *iff* there is some k, $0 \leq k \leq i$, such that $\mathfrak{I}, k \models \phi_2$, and $\mathfrak{I}, j \models \phi_1$ for every j, $k < j \leq i$ *iff* there is some k, $0 \leq k \leq i$, such that $\mathfrak{I}^{\mathsf{p}}, k \models \phi_2^{\mathsf{p}}$, and $\mathfrak{I}^{\mathsf{p}}, j \models \phi_1^{\mathsf{p}}$ for every j, $k < j \leq i$ *iff* $\mathfrak{I}^{\mathsf{p}}, i \models (\phi_1 \,\mathsf{S}\, \phi_2)^{\mathsf{p}}$.□

The 'only if' direction of this lemma yields that satisfiability of ϕ w.r.t. $\mathcal{R}$ implies satisfiability of ϕ^{p}. However, the 'if' direction does not yield the converse of this implication. In fact, the propositional LTL-formula ϕ^{p} may turn out to be satisfiable even though the original $\mathcal{SHOQ}$-LTL-formula ϕ is not. The reason is that there may exist propositional LTL-structures that are not propositional abstractions of DL-LTL-structures.

Example 3.9. *Take the $\mathcal{SHOQ}$-LTL-formula ϕ_{ex} and the bijection p of Example 3.7, and consider the propositional LTL-structure $\mathfrak{W} = (w_i)_{i\geq 0}$ with $w_i = \{p_1, p_2, p_3\}$ for every $i \geq 0$. Obviously, we have $\mathfrak{W}, 0 \models \phi_{ex}^{\mathsf{p}}$, but there is no DL-LTL-structure $\mathfrak{I}$ such that $\mathfrak{I}^{\mathsf{p}} = \mathfrak{W}$. In fact, every world of this DL-LTL-structure would need to satisfy the three axioms in $\mathsf{Ax}(\phi_{ex})$ simultaneously, which is clearly not possible.* ◇

To address this problem, we need some more notions. We consider a set $\mathcal{W} \subseteq 2^{\mathcal{P}_\phi}$, which intuitively specifies the worlds that are allowed to occur in a propositional LTL-structure satisfying ϕ^{p}. To express this restriction, we the define the propositional LTL-formula

$$\phi_{\mathcal{W}}^{\mathsf{p}} := \phi^{\mathsf{p}} \wedge \Box\Big(\bigvee_{X\in\mathcal{W}}\Big(\bigwedge_{p\in X} p \wedge \bigwedge_{p\in\mathcal{P}_\phi\setminus X} \neg p\Big)\Big).$$

The second satisfiability problem is called *r-satisfiability*, because it is used to determine whether it is possible to satisfy the *rigidity* constraints for the names in $\mathsf{N_{RC}}$ and $\mathsf{N_{RR}}$. Thus, it can be used to check whether the set $\mathcal{W}$ can indeed be induced by a DL-LTL-structure that is a model of ϕ w.r.t. $\mathcal{R}$.

Definition 3.10 (R-satisfiability). *Let $\mathcal{W} = \{X_1, \ldots, X_k\} \subseteq 2^{\mathcal{P}_\phi}$. We call $\mathcal{W}$* r-satisfiable w.r.t. $\mathcal{R}$ *if there exist interpretations $\mathcal{I}_1 = (\Delta, \cdot^{\mathcal{I}_1})$, …, $\mathcal{I}_k = (\Delta, \cdot^{\mathcal{I}_k})$ such that*

- *$a^{\mathcal{I}_i} = a^{\mathcal{I}_j}$ holds for every $a \in \mathsf{N_I}$ and all i, j, $1 \leq i < j \leq k$;*
- *$A^{\mathcal{I}_i} = A^{\mathcal{I}_j}$ holds for every $A \in \mathsf{N_{RC}}$ and all i, j, $1 \leq i < j \leq k$;*
- *$r^{\mathcal{I}_i} = r^{\mathcal{I}_j}$ holds for every $r \in \mathsf{N_{RR}}$ and all i, j, $1 \leq i < j \leq k$; and*
- *every $\mathcal{I}_i$, $1 \leq i \leq k$, is a model of the Boolean knowledge base*

$$\mathcal{B}_{X_i} := \Bigg(\bigwedge_{p \in X_i} \mathsf{p}^{-1}(p) \wedge \bigwedge_{p \in \mathcal{P}_\phi \setminus X_i} \neg \mathsf{p}^{-1}(p), \quad \mathcal{R} \Bigg).$$

◇

Note that any subset of a set $\mathcal{W}$ that is r-satisfiable w.r.t. $\mathcal{R}$ is again r-satisfiable w.r.t. $\mathcal{R}$. In particular, the empty set is always r-satisfiable w.r.t. $\mathcal{R}$.

The intuition underlying the definition of r-satisfiability is the following. The existence of the interpretations $\mathcal{I}_i$, $1 \leq i \leq k$, ensures that the Boolean knowledge base induced by X_i and $\mathcal{R}$ is consistent. In fact, a set $\mathcal{W}$ containing a set X_i for which this does not hold cannot be induced by a DL-LTL-structure. Moreover, we ensure that the interpretations share the same domain and respect rigid names. As we will see later, for deciding whether a set $\mathcal{W}$ is r-satisfiable w.r.t. $\mathcal{R}$, the difficulty lies in ensuring that the interpretations share the same domain and respect rigid names.

Satisfaction of the temporal structure of ϕ by a DL-LTL-structure built this way is ensured by testing $\phi^{\mathsf{p}}_{\mathcal{W}}$ for satisfiability. This is captured in the notion of t-satisfiability.

Definition 3.11 (T-satisfiability). *Let $\mathcal{W} \subseteq 2^{\mathcal{P}_\phi}$. We call the propositional LTL-formula ϕ^{p}* t-satisfiable w.r.t. $\mathcal{W}$ *if there exists a propositional LTL-structure $\mathfrak{W} = (w_i)_{i \geq 0}$ such that $\mathfrak{W}, 0 \models \phi^{\mathsf{p}}_{\mathcal{W}}$.* ◇

The next two lemmas show that these two satisfiability problems, namely, t-satisfiability and r-satisfiability, can be combined to decidable the satisfiability problem in $\mathcal{SHOQ}$-LTL. The statements of the lemmas were also proved in [BGL12] for $\mathcal{ALC}$-LTL, but again the same arguments can also be used to prove them for $\mathcal{SHOQ}$-LTL. Yet, we repeat these arguments for the sake of completeness.

Lemma 3.12. *For every propositional LTL-structure $\mathfrak{W} = (w_i)_{i \geq 0}$ with $w_i \subseteq \mathcal{P}_\phi$ for every $i \geq 0$, the following two statements are equivalent:*

1. *There is a model $\mathfrak{I}$ of ϕ w.r.t. $\mathcal{R}$ with $\mathfrak{I}^{\mathsf{p}} = \mathfrak{W}$.*
2. *$\mathfrak{W}$ is a model of ϕ^{p} and the set $\mathcal{W} := \{w_i \mid i \geq 0\}$ is r-satisfiable w.r.t. $\mathcal{R}$.*

Proof. For the direction '1 $\Longrightarrow$ 2', assume that there is a DL-LTL-structure $\mathfrak{I} = (\mathcal{I}_i)_{i \geq 0}$ that is a model of ϕ w.r.t. $\mathcal{R}$ with $\mathfrak{I}^{\mathsf{p}} = \mathfrak{W}$. Since $\mathfrak{I} \models \mathcal{R}$, we have by Lemma 3.8 that $\mathfrak{W}$ is a model of ϕ^{p}. Since $w_i \subseteq \mathcal{P}_\phi$ for every $i \geq 0$, we have that $\mathcal{W} = \{w_i \mid i \geq 0\} \subseteq 2^{\mathcal{P}_\phi}$ is finite. Let

$\mathcal{W} = \{X_1, \dots, X_k\}$. We have that for every $i \geq 0$, there is an index $\nu_i \in \{1, \dots, k\}$ such that $\mathcal{I}_i$ induces the set X_{ν_i}, i.e.

$$X_{\nu_i} = \{\mathsf{p}(\alpha) \mid \alpha \in \mathsf{Ax}(\phi) \text{ and } \mathcal{I}_i \models \alpha\},$$

and, conversely, for every $\nu \in \{1, \dots, k\}$, there is an index $i \geq 0$ such that $\nu = \nu_i$. For every i, $1 \leq i \leq k$, the interpretation $\mathcal{J}_i$ is obtained as follows. Let $\ell_1, \dots, \ell_k$ be such that $\nu_{\ell_1} = 1, \dots, \nu_{\ell_k} = k$. Now, if we set $\mathcal{J}_i := \mathcal{I}_{\ell_i}$, then we clearly have $\mathcal{J}_i \models \mathcal{B}_{X_i}$. It is now easy to see that the interpretations $\mathcal{J}_1, \dots, \mathcal{J}_k$ satisfy the conditions for r-satisfiability of $\mathcal{W}$ w.r.t. $\mathcal{R}$.

For the direction '2 $\implies$ 1', assume that $\mathfrak{W}$ is a model of ϕ^{p} and the set $\mathcal{W} = \{w_i \mid i \geq 0\}$ is r-satisfiable w.r.t. $\mathcal{R}$. Since $w_i \subseteq \mathcal{P}_\phi$ for every $i \geq 0$, we have that $\mathcal{W} = \{w_i \mid i \geq 0\} \subseteq 2^{\mathcal{P}_\phi}$ is finite. Let $\mathcal{W} = \{X_1, \dots, X_k\}$. Since $\mathcal{W}$ is r-satisfiable w.r.t. $\mathcal{R}$, there are interpretations $\mathcal{J}_1, \dots, \mathcal{J}_k$ such that the conditions in Definition 3.10 are satisfied. Moreover, we have that for every world w_i, there is exactly one index $\nu_i \in \{1, \dots, k\}$ such that w_i satisfies

$$\bigwedge_{p \in X_{\nu_i}} p \wedge \bigwedge_{p \in \mathcal{P}_\phi \setminus X_{\nu_i}} \neg p.$$

We can now define a DL-LTL-structure $\mathfrak{I} := (\mathcal{I}_i)_{i \geq 0}$ as follows. We set $\mathcal{I}_i := \mathcal{J}_{\nu_i}$ for $i \geq 0$. By construction, we have $\mathfrak{I}^{\mathsf{p}} = \mathfrak{W}$. By Definition 3.10, each $\mathcal{I}_i$ is a model of $\mathcal{B}_{X_{\nu_i}}$, i.e. it is a model of $\mathcal{R}$ and satisfies exactly the axioms specified by the propositional variables in X_{ν_i}. This yields that $\mathfrak{I} \models \mathcal{R}$, and since $\mathfrak{W}, 0 \models \phi^{\mathsf{p}}$, we have by Lemma 3.8 that $\mathfrak{I}, 0 \models \phi$. Thus, $\mathfrak{I}$ is a model of ϕ w.r.t. $\mathcal{R}$. □

The following lemma is an immediate consequence of the previous lemma.

Lemma 3.13. *The $\mathcal{SHOQ}$-LTL-formula ϕ is satisfiable w.r.t. the RBox $\mathcal{R}$ iff there is a set $\mathcal{W} = \{X_1, \dots, X_k\} \subseteq 2^{\mathcal{P}_\phi}$ such that*

- *$\mathcal{W}$ is r-satisfiable w.r.t. $\mathcal{R}$, and*
- *ϕ^{p} is t-satisfiable w.r.t. $\mathcal{W}$.*

Proof. For the 'only if' direction, assume that there is a DL-LTL-structure $\mathfrak{I} = (\mathcal{I}_i)_{i \geq 0}$ that is a model of ϕ w.r.t. $\mathcal{R}$. Let $\mathfrak{I}^{\mathsf{p}} = (w_i)_{i \geq 0}$, and let $\mathcal{W} = \{w_i \mid i \geq 0\}$. By Lemma 3.12, we have that $\mathfrak{I}^{\mathsf{p}}$ is a model of ϕ^{p} and that $\mathcal{W}$ is r-satisfiable w.r.t. $\mathcal{R}$. By construction of $\mathcal{W}$, we have also that $\mathfrak{I}^{\mathsf{p}}$ is a model of $\phi^{\mathsf{p}}_{\mathcal{W}}$. Hence, ϕ^{p} is t-satisfiable w.r.t. $\mathcal{W}$.

For the 'if' direction, assume that there is a set $\mathcal{W} = \{X_1, \dots, X_k\} \subseteq 2^{\mathcal{P}_\phi}$ such that $\mathcal{W}$ is r-satisfiable w.r.t. $\mathcal{R}$ and ϕ^{p} is t-satisfiable w.r.t. $\mathcal{W}$. Hence, there is a propositional LTL-structure $\mathfrak{W} = (w_i)_{i \geq 0}$ such that $\mathfrak{W}$ is a model of $\phi^{\mathsf{p}}_{\mathcal{W}}$. Hence $\mathfrak{W}$ is a model of ϕ^{p}. We define $\mathcal{W}' := \{w_i \mid i \geq 0\}$. Since $\mathfrak{W}$ is a model of $\phi^{\mathsf{p}}_{\mathcal{W}}$, we have that $\mathcal{W}' \subseteq \mathcal{W}$. Since $\mathcal{W}$ is r-satisfiable w.r.t. $\mathcal{R}$, this yields that $\mathcal{W}'$ is r-satisfiable w.r.t. $\mathcal{R}$. Then, Lemma 3.12 yields that there is a model $\mathfrak{I}$ of ϕ w.r.t. $\mathcal{R}$ with $\mathfrak{I}^{\mathsf{p}} = \mathfrak{W}$, i.e. ϕ is satisfiable w.r.t. $\mathcal{R}$. □

To obtain a decision procedure for the satisfiability problem in $\mathcal{SHOQ}$-LTL, we have to non-deterministically guess or construct the set $\mathcal{W}$, and then check the two conditions of Lemma 3.13. Depending on which symbols are allowed to be rigid, we use different constructions to achieve that.

First, we focus on deciding t-satisfiability w.r.t. a given set $\mathcal{W}$. From now on, let $\mathcal{W} \subseteq 2^{\mathcal{P}_\phi}$. Obviously, the size of $\phi^{\mathsf{p}}_{\mathcal{W}}$ may be exponential in the size of ϕ. Since we can decide satisfiability of a propositional LTL-formula in PSPACE [SC85; LPZ85], this yields an EXPSPACE-decision procedure for deciding the satisfiability of $\phi^{\mathsf{p}}_{\mathcal{W}}$. However, using a trick from [BGL12], we can reduce the complexity to EXPTIME.

Lemma 3.14. *Deciding whether ϕ^{p} is t-satisfiable w.r.t. $\mathcal{W}$ can be done in time exponential in the size of ϕ^{p} and linear in the size of $\mathcal{W}$.*

Proof. We first construct a Büchi-automaton for ϕ^{p}, which can be done in time exponential in the size of ϕ^{p} as discussed in Section 2.2.2. Let $\mathcal{N} = (Q, \Sigma_{\mathcal{P}_\phi}, \Delta, Q_0, F)$ be a Büchi-automaton for ϕ^{p}. We obtain the Büchi-automaton $\mathcal{N}' = (Q, \Sigma_{\mathcal{P}_\phi}, \Delta', Q_0, F)$ by removing all transitions that are labelled with a letter $\sigma \in \Sigma_{\mathcal{P}_\phi} \setminus \mathcal{W}$, i.e. we define

$$\Delta' := \big\{(q, \sigma, q') \in \Delta \mid \sigma \in \mathcal{W}\big\}.$$

It is easy to verify that $\mathcal{N}'$ is a Büchi-automaton for $\phi^{\mathsf{p}}_{\mathcal{W}}$.

Note that the Büchi-automaton $\mathcal{N}'$ can be constructed in time polynomial in the size of $\mathcal{N}$ and linear in the size of $\mathcal{W}$, and thus the size of $\mathcal{N}'$ is exponential in the size of ϕ^{p}. Since the emptiness problem for Büchi-automata can be solved in polynomial time [VW94], this yields that t-satisfiability of ϕ^{p} w.r.t. $\mathcal{W}$ can be decided in time exponential in the size of ϕ^{p} and linear in the size of $\mathcal{W}$. □

Due to Lemma 3.13, the complexity of the satisfiability problem in $\mathcal{SHOQ}$-LTL also depends on the complexity of the deciding whether $\mathcal{W}$ is r-satisfiable w.r.t. $\mathcal{R}$. However, this depends on the fact whether there are concept or role names that are allowed to be rigid.

In Section 3.2.1, we consider the case without rigid names, and in Section 3.2.2, we consider the most general case with rigid concept and role names. Finally, we consider the case with rigid concept names in Section 3.2.3. A result that is needed in Section 3.2.3 is proved in a separate section, namely, in Section 3.2.4.

3.2.1 Satisfiability in $\mathcal{SHOQ}$-LTL for the Case without Rigid Names

In this section, we consider the case where neither concept names nor role names are allowed to be rigid, i.e. $\mathsf{N_{RC}} = \mathsf{N_{RR}} = \emptyset$. We establish the following complexity result.

Theorem 3.15. *The satisfiability problem in $\mathcal{SHOQ}$-LTL is in* EXPTIME *if* $\mathsf{N_{RC}} = \mathsf{N_{RR}} = \emptyset$.

Proof. Let $\mathcal{R}$ be an RBox, and let ϕ be a $\mathcal{SHOQ}$-LTL-formula w.r.t. $\mathcal{R}$. We can decide satisfiability of ϕ w.r.t. $\mathcal{R}$ using Lemma 3.13. For that, let $\mathsf{p}\colon \mathsf{Ax}(\phi) \to \mathcal{P}_\phi$ be a bijection, and define

$$\mathcal{W} := \{X \in 2^{\mathcal{P}_\phi} \mid \mathcal{B}_X \text{ is consistent}\},$$

where $\mathcal{B}_X$ is defined as in Definition 3.10. We first show that $\mathcal{W} = \{X_1, \ldots, X_k\}$ is r-satisfiable w.r.t. $\mathcal{R}$. Since every $\mathcal{B}_{X_i}$, $1 \leq i \leq k$ is consistent, there are models $\mathcal{I}_1, \ldots, \mathcal{I}_k$ such that every $\mathcal{I}_i$, $1 \leq i \leq k$, is a model of $\mathcal{B}_{X_i}$. We can assume w.l.o.g. that all of these models have the same domain since we can assume w.l.o.g. that their domains are countably infinite due to to the Löwenheim-Skolem theorem [Löw15; Sko20]. Furthermore, we can assume w.l.o.g.

that all individual names are interpreted by the same domain elements in all models. Since $\mathsf{N_{RC}} = \mathsf{N_{RR}} = \emptyset$, this yields that $\mathcal{W}$ is r-satisfiable w.r.t. $\mathcal{R}$.

Thus, we have by Lemma 3.13 that if ϕ^{p} is t-satisfiable w.r.t. $\mathcal{W}$, then ϕ is satisfiable w.r.t. $\mathcal{R}$. Conversely, again by Lemma 3.13, we have that if ϕ is satisfiable w.r.t. $\mathcal{R}$, then there is a set $\mathcal{W}' \subseteq 2^{\mathcal{P}_\phi}$ such that $\mathcal{W}'$ is r-satisfiable w.r.t. $\mathcal{R}$ and ϕ^{p} is t-satisfiable w.r.t. $\mathcal{W}'$. The definition of $\mathcal{W}$ yields that $\mathcal{W}$ is the maximal subset of $2^{\mathcal{P}_\phi}$ that is r-satisfiable w.r.t. $\mathcal{R}$. Thus, we have that $\mathcal{W}' \subseteq \mathcal{W}$. It is easy to see that the t-satisfiability of ϕ^{p} w.r.t. $\mathcal{W}'$ implies that ϕ^{p} is t-satisfiable w.r.t. $\mathcal{W}$. Hence, we have that ϕ is satisfiable w.r.t. $\mathcal{R}$ iff ϕ^{p} is t-satisfiable w.r.t. $\mathcal{W}$.

Note that $\mathcal{W}$ can be constructed in time exponential in the size of ϕ and $\mathcal{R}$. Indeed, there are exponentially many $X \in 2^{\mathcal{P}_\phi}$, but each $\mathcal{B}_X$ can be constructed in time polynomial in the size of ϕ and $\mathcal{R}$, and is thus of size polynomial in the size of ϕ and $\mathcal{R}$. We show in Corollary 3.34 (see Section 3.2.4) that consistency of a Boolean $\mathcal{SHOQ}$-knowledge base $\mathcal{B}$ can be decided in time exponential in the size of $\mathcal{B}$. Thus, overall, deciding for every $\mathcal{B}_X$ whether it is consistent can be done in time exponential in the size of ϕ and $\mathcal{R}$. Due to Lemma 3.14, deciding whether ϕ^{p} is t-satisfiable w.r.t. $\mathcal{W}$ can be done in time exponential in the size of ϕ^{p} (and thus in time exponential in the size of ϕ) and linear in the size of $\mathcal{W}$. Thus, we can decide whether ϕ is satisfiable w.r.t. $\mathcal{R}$ in time exponential in the size of ϕ and $\mathcal{R}$. □

Together with Theorem 3.5, we obtain that the satisfiability problem in $\mathcal{SHOQ}$-LTL is ExpTime-complete if neither concept nor role names are allowed to be rigid.

3.2.2 Satisfiability in $\mathcal{SHOQ}$-LTL for the Case of Rigid Concept Names and Role Names

In this section, we consider the case where both concept and role names may be rigid, i.e. $\mathsf{N_{RC}} \neq \emptyset$ and $\mathsf{N_{RR}} \neq \emptyset$.

Let us assume that a set $\mathcal{W} = \{X_1, \ldots, X_k\} \subseteq 2^{\mathcal{P}_\phi}$ is given. Note that deciding whether $\mathcal{W}$ is r-satisfiable w.r.t. $\mathcal{R}$ cannot be done by simply checking for each $X \in \mathcal{W}$ whether the Boolean knowledge base $\mathcal{B}_X$ is consistent as we did in Section 3.2.1 for the case without rigid names. In fact, these consistency checks are not independent any longer since one has to ensure that rigid concept and role names are interpreted in the same way. To achieve this, we adopt the renaming technique from [BGL12] that works by introducing copies of the flexible symbols.

For every i, $1 \leq i \leq k$, every *flexible* concept name A occurring in ϕ, and every *flexible* role name r occurring in ϕ or $\mathcal{R}$, we introduce copies $A^{(i)}$ and $r^{(i)}$. We call $A^{(i)}$ the i-th copy of A, and similarly $r^{(i)}$ the i-th copy of r. The axiom $\alpha^{(i)}$ is obtained from the axiom α by replacing every occurrence of a flexible name by its i-th copy. Similarly, the Boolean axiom formula $\Psi^{(i)}$ (RBox $\mathcal{R}^{(i)}$) is obtained from the Boolean axiom formula Ψ (RBox $\mathcal{R}$) by replacing each axiom α occurring in Ψ ($\mathcal{R}$) by $\alpha^{(i)}$.

Moreover, let $\mathcal{B}_{X_i} = (\Psi_{X_i}, \mathcal{R})$, $1 \leq i \leq k$, denote the Boolean knowledge bases defined in Definition 3.10. We define

$$\mathcal{B}_{\mathcal{W}} := \Bigg(\bigwedge_{1 \leq i \leq k} \Psi^{(i)}_{X_i}, \quad \bigcup_{1 \leq i \leq k} \mathcal{R}^{(i)} \Bigg).$$

The next lemma states that consistency of $\mathcal{B}_\mathcal{W}$ is enough for ensuring r-satisfiability of $\mathcal{W}$ w.r.t. $\mathcal{R}$.

Lemma 3.16. *The set $\mathcal{W}$ is r-satisfiable w.r.t. $\mathcal{R}$ iff the Boolean knowledge base $\mathcal{B}_\mathcal{W}$ is consistent.*

Proof. For the 'only if' direction, let $\mathcal{I}_1 = (\Delta, \cdot^{\mathcal{I}_1})$, …, $\mathcal{I}_k = (\Delta, \cdot^{\mathcal{I}_k})$ be the interpretations required by Definition 3.10 for the r-satisfiability of $\mathcal{W}$ w.r.t. $\mathcal{R}$. We construct the interpretation $\mathcal{J} = (\Delta, \cdot^{\mathcal{J}})$ as follows:

- every individual name and every rigid name is interpreted as in $\mathcal{I}_1$; and
- the i-th copy, $1 \le i \le k$, of each flexible name is interpreted like the original name in $\mathcal{I}_i$.

It is easy to verify that $\mathcal{J}$ is a model of $\mathcal{B}_\mathcal{W}$.

For the 'if' direction, let $\mathcal{J}$ be a model of $\mathcal{B}_\mathcal{W}$. We obtain the interpretations $\mathcal{I}_1,\dots,\mathcal{I}_k$ by the inverse construction to the one above:

- the domain of all these interpretations is the domain of $\mathcal{J}$;
- every individual name and every rigid name is interpreted by these interpretations as in $\mathcal{J}$; and
- every flexible name is interpreted in $\mathcal{I}_i$, $1 \le i \le k$, as its i-th copy is interpreted in $\mathcal{J}$.

It is again easy to verify that the interpretations $\mathcal{I}_1,\dots,\mathcal{I}_k$ satisfy the conditions for r-satisfiability of $\mathcal{W}$ w.r.t. $\mathcal{R}$. □

Using this lemma, we can prove our complexity result.

Theorem 3.17. *The satisfiability problem in $\mathcal{SHOQ}$-LTL is in* 2ExpTime *if* $\mathsf{N_{RC}} \neq \emptyset$ *and* $\mathsf{N_{RR}} \neq \emptyset$.

Proof. Let $\mathcal{R}$ be an RBox, let ϕ be a $\mathcal{SHOQ}$-LTL-formula w.r.t. $\mathcal{R}$, and let $\mathsf{p}\colon \mathsf{Ax}(\phi) \to \mathcal{P}_\phi$ be a bijection. We use again Lemma 3.13 for deciding satisfiability of ϕ w.r.t. $\mathcal{R}$. We first enumerate all sets $\mathcal{W} \subseteq 2^{\mathcal{P}_\phi}$, which can be done in time doubly exponential in ϕ and $\mathcal{R}$. For each of these sets $\mathcal{W}$, we check t-satisfiability of ϕ^p w.r.t. $\mathcal{W}$ and r-satisfiability of $\mathcal{W}$ w.r.t. $\mathcal{R}$. By Lemma 3.14, the t-satisfiability check can be done in time exponential in the size of ϕ^p (and thus in time exponential in the size of ϕ) and linear in the size of $\mathcal{W}$.

For the r-satisfiability check, we use Lemma 3.16. We construct the Boolean knowledge base $\mathcal{B}_\mathcal{W}$, which can be done in time exponential in the size of ϕ and $\mathcal{R}$. Also, $\mathcal{B}_\mathcal{W}$ is of size at most exponential in the size of ϕ and $\mathcal{R}$. Consistency of $\mathcal{B}_\mathcal{W}$ can be checked in time exponential in the size of $\mathcal{B}_\mathcal{W}$ by Corollary 3.34, which we prove in Section 3.2.4. Thus, checking whether $\mathcal{W}$ is r-satisfiable w.r.t. $\mathcal{R}$ can be done in time doubly exponential in the size of ϕ and $\mathcal{R}$.

Thus, overall, we can decide whether ϕ is satisfiable w.r.t. $\mathcal{R}$ in time doubly exponential in the size of ϕ and $\mathcal{R}$. □

Together with Theorem 3.5, we obtain that the satisfiability problem in $\mathcal{SHOQ}$-LTL is 2ExpTime-complete if both concept and role names are allowed to be rigid.

3.2.3 Satisfiability in $\mathcal{SHOQ}$-LTL for the Case of Rigid Concept Names

In this section, we consider the case where only concept names may be rigid, i.e. $\mathsf{N_{RC}} \neq \emptyset$ and $\mathsf{N_{RR}} = \emptyset$.

Let us again assume that a set $\mathcal{W} = \{X_1, \ldots, X_k\} \subseteq 2^{\mathcal{P}_\phi}$ is given. By Lemma 3.16, for checking whether $\mathcal{W}$ is r-satisfiable w.r.t. $\mathcal{R}$, it is enough to construct the Boolean knowledge base $\mathcal{B}_\mathcal{W}$ and to check it for consistency. As we have seen in the proof of Theorem 3.17, this yields a 2ExpTime decision procedure. However, we can reduce the complexity to NExpTime by using the ideas of the proof of Lemma 6.3 in [BGL12], where the same complexity result is shown for $\mathcal{ALC}$-LTL. The principal idea is to fix the combinations of rigid concept names that are allowed to occur in the models of the Boolean knowledge bases $\mathcal{B}_{X_i}$, $1 \leq i \leq k$. For that, we need some more notation.

Definition 3.18 (Interpretation respects $\mathcal{D}$). *Let $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ be an interpretation, and let $\mathcal{D} = (\mathcal{U}, \mathcal{Y})$ be a pair such that $\mathcal{U}$ is a set of concept names and $\mathcal{Y} \subseteq 2^\mathcal{U}$.*

We say that $\mathcal{I}$ respects *$\mathcal{D}$ if*

$$\mathcal{Y} = \left\{ Y \subseteq \mathcal{U} \mid \text{there is some } d \in \Delta^\mathcal{I} \text{ with } d \in (C_{\mathcal{U},Y})^\mathcal{I} \right\},$$

where

$$C_{\mathcal{U},Y} := \bigsqcap_{A \in Y} A \quad \sqcap \quad \bigsqcap_{A \in \mathcal{U} \setminus Y} \neg A.$$

◇

Intuitively, this definitions states that every combination of concept names in $\mathcal{Y}$ is realised by a domain element of $\mathcal{I}$, and conversely, every such combination that is realised by a domain element of $\mathcal{I}$ must occur in $\mathcal{Y}$.

Let $\mathsf{RCon}(\phi)$ denote the set of rigid concept names occurring in ϕ, and let $\mathsf{Ind}(\phi)$ denote the set of individual names occurring in ϕ. Furthermore, let $\mathcal{D} = (\mathsf{RCon}(\phi), \mathcal{Y})$ with $\mathcal{Y} \subseteq 2^{\mathsf{RCon}(\phi)}$ be arbitrary, and let τ be a mapping from $\mathsf{Ind}(\phi)$ to $\mathcal{Y}$. The idea is that $\mathcal{D}$ fixes the combinations of rigid concept names that are allowed to occur in the models of $\mathcal{B}_{X_i}$, $1 \leq i \leq k$. The mapping τ assigns to each individual name occurring in ϕ one such combination. We define Ψ_τ to be the following Boolean axiom formula:

$$\Psi_\tau := \bigwedge_{a \in \mathsf{Ind}(\phi)} C_{\mathsf{RCon}(\phi), \tau(a)}(a).$$

The next lemma states how these notions can be used to characterise r-satisfiability of $\mathcal{W}$ w.r.t. $\mathcal{R}$.

Lemma 3.19. *If $\mathsf{N_{RC}} \neq \emptyset$ and $\mathsf{N_{RR}} = \emptyset$, then $\mathcal{W}$ is r-satisfiable w.r.t. $\mathcal{R}$ iff there exist a pair $\mathcal{D} = (\mathsf{RCon}(\phi), \mathcal{Y})$ with $\mathcal{Y} \subseteq 2^{\mathsf{RCon}(\phi)}$ and a mapping $\tau \colon \mathsf{Ind}(\phi) \to \mathcal{Y}$ such that for every i, $1 \leq i \leq k$, the Boolean knowledge base $(\Psi_{X_i} \wedge \Psi_\tau, \mathcal{R})$ has a model that respects $\mathcal{D}$.*

Proof. For the 'if' direction, assume that $\mathcal{I}_i$, $1 \leq i \leq k$, are the models of $(\Psi_{X_i} \wedge \Psi_\tau, \mathcal{R})$, respectively, that respect $\mathcal{D}$. Similar to the proof of Lemma 6.3 in [BGL12], we can assume w.l.o.g. that their domains Δ_i are countably infinite and for each $Y \in \mathcal{Y}$ there are countably infinitely many elements $d \in (C_{\mathsf{RCon}(\phi),Y})^{\mathcal{I}_i}$. This is a consequence of the Löwenheim-Skolem theorem [Löw15; Sko20] and the fact that the countably infinite disjoint union of $\mathcal{I}_i$ with itself is again a model of $(\Psi_{X_i} \wedge \Psi_\tau, \mathcal{R})$.

Consequently, we can partition the domains Δ_i into the countably infinite sets

$$\Delta_i(Y) := \left\{d \in \Delta_i \mid d \in (C_{\mathsf{RCon}(\phi),Y})^{\mathcal{I}_i}\right\}$$

for $Y \in \mathcal{Y}$. By the assumptions above and the fact that every $\mathcal{I}_i$ satisfies Ψ_τ, there are bijections $\pi_i \colon \Delta_1 \to \Delta_i$, $2 \leq i \leq k$, such that

- $\pi_i(\Delta_1(Y)) = \Delta_i(Y)$ for every $Y \in \mathcal{Y}$, and
- $\pi_i(a^{\mathcal{I}_1}) = a^{\mathcal{I}_i}$ for every $a \in \mathsf{Ind}(\phi)$.

Thus, we can assume in the following that the models $\mathcal{I}_i$, $1 \leq i \leq k$, actually share the same domain and interpret the concept names in $\mathsf{RCon}(\phi)$ and the individual names occurring in ϕ in the same way. Note that the interpretation of the names in $\mathsf{N_{RC}} \setminus \mathsf{RCon}(\phi)$ and $\mathsf{N_I} \setminus \mathsf{Ind}(\phi)$ is irrelevant and can be fixed arbitrarily, as long as the UNA is satisfied.

Since for every i, $1 \leq i \leq k$, we have that $\mathcal{I}_i$ is a model of $(\Psi_{X_i} \wedge \Psi_\tau, \mathcal{R})$, we have also that $\mathcal{I}_i$ is a model of $\mathcal{B}_{X_i} = (\Psi_{X_i}, \mathcal{R})$. Thus, the conditions required for the r-satisfiability of $\mathcal{W}$ w.r.t. $\mathcal{R}$ by Definition 3.10 are satisfied.

For the 'only if' direction, assume that $\mathcal{I}_i = (\Delta, \cdot^{\mathcal{I}_i})$, $1 \leq i \leq k$, are the interpretations required for r-satisfiability of $\mathcal{W}$ w.r.t. $\mathcal{R}$ by Definition 3.10. Since these interpretations share the same domain, interpret the rigid concept names in the same way, it follows that for every $Y \subseteq \mathsf{RCon}(\phi)$, we have that $(C_{\mathsf{RCon}(\phi),Y})^{\mathcal{I}_1} = (C_{\mathsf{RCon}(\phi),Y})^{\mathcal{I}_i}$ for every i, $2 \leq i \leq k$. We define $\mathcal{D} := (\mathsf{RCon}(\phi), \mathcal{Y})$ with

$$\mathcal{Y} := \left\{Y \subseteq \mathsf{RCon}(\phi) \mid \text{there is some } d \in \Delta \text{ with } d \in (C_{\mathsf{RCon}(\phi),Y})^{\mathcal{I}_1}\right\}.$$

Moreover, for every $a \in \mathsf{Ind}(\phi)$, we define $\tau(a) := Y \subseteq \mathsf{RCon}(\phi)$ iff $a \in (C_{\mathsf{RCon}(\phi),Y})^{\mathcal{I}_1}$. Since the interpretations $\mathcal{I}_1, \ldots, \mathcal{I}_k$ interpret the individual names in the same way, and for every i, $1 \leq i \leq k$, the interpretation $\mathcal{I}_i$ is a model of $\mathcal{B}_{X_i} = (\Psi_{X_i}, \mathcal{R})$, we have thus that $\mathcal{I}_i$ is also a model of $(\Psi_{X_i} \wedge \Psi_\tau, \mathcal{R})$. Moreover, every $\mathcal{I}_i$, $1 \leq i \leq k$, respects $\mathcal{D}$ by construction of $\mathcal{D}$. □

Using this lemma, we can prove our complexity result.

Theorem 3.20. *The satisfiability problem in $\mathcal{SHOQ}$-LTL is in* NExpTime *if* $\mathsf{N_{RC}} \neq \emptyset$ *and* $\mathsf{N_{RR}} = \emptyset$.

Proof. Let $\mathcal{R}$ be an RBox, let ϕ be a $\mathcal{SHOQ}$-LTL-formula w.r.t. $\mathcal{R}$, and let $\mathsf{p} \colon \mathsf{Ax}(\phi) \to \mathcal{P}_\phi$ be a bijection. Again, we use Lemma 3.13 for deciding whether ϕ is satisfiable w.r.t. $\mathcal{R}$. We first non-deterministically guess a set $\mathcal{W} = \{X_1, \ldots, X_k\} \subseteq 2^{\mathcal{P}_\phi}$, which is of size at most exponential in the size of ϕ and $\mathcal{R}$. Next, we check whether ϕ^{p} is t-satisfiable w.r.t. $\mathfrak{W}$, which can be done in time exponential in the size of ϕ^{p} (and thus in time exponential in the size of ϕ) and linear in the size of $\mathfrak{W}$ by Lemma 3.14.

For the r-consistency check, we use Lemma 3.19. For that, we non-deterministically guess a set $\mathcal{Y} \subseteq 2^{\mathsf{RCon}(\phi)}$ and a mapping $\tau \colon \mathsf{Ind}(\phi) \to \mathcal{Y}$, which also can be done in time exponential in the size of ϕ and $\mathcal{R}$. We define $\mathcal{D} := (\mathsf{RCon}(\phi), \mathcal{Y})$. For every i, $1 \leq i \leq k$, we construct the Boolean knowledge base $(\Psi_{X_i} \wedge \Psi_\tau, \mathcal{R})$, which is of size polynomial in the size of ϕ and $\mathcal{R}$, and can be constructed in time exponential in the size of ϕ and $\mathcal{R}$ (due to the mapping τ). Thus, it is only left to show that we can check whether the Boolean knowledge base $(\Psi_{X_i} \wedge \Psi_\tau, \mathcal{R})$ has a model that respects $\mathcal{D}$ in time exponential in the size of the Boolean knowledge base, and thus exponential in the size of ϕ and $\mathcal{R}$. This follows immediately from Theorem 3.33, which we show in Section 3.2.4.

Thus, overall, we can decide whether ϕ is satisfiable w.r.t. $\mathcal{R}$ in NExpTime. □

Together with Theorem 3.5, we obtain now that the satisfiability problem in $\mathcal{SHOQ}$-LTL is NExpTime-complete if only concept names are allowed to be rigid.

3.2.4 Consistency of Boolean $\mathcal{SHOQ}^{\sqcap}$-knowledge bases

In this section, we prove the result that is needed to finish the proof of Theorem 3.20, namely, that the consistency of a Boolean $\mathcal{SHOQ}$-knowledge base w.r.t. a pair $\mathcal{D}$ can be checked in time exponential in the size of this Boolean knowledge base, where we call a Boolean KB *consistent w.r.t. a pair* $\mathcal{D}$ if it has a model that respects $\mathcal{D}$. Moreover, we derive the corollary that checking the consistency of a Boolean $\mathcal{SHOQ}$-knowledge base (without $\mathcal{D}$) can also be done in time exponential in the size of this Boolean knowledge base. This result then finishes the proofs of Theorems 3.15 and 3.17.

In Chapter 5, we deal with Boolean $\mathcal{SHQ}^{\sqcap}$-knowledge bases. The description logic $\mathcal{SHQ}^{\sqcap}$ extends $\mathcal{SHQ}$ with *role conjunctions* of the form $r_1 \sqcap \cdots \sqcap r_\ell$, $\ell \geq 1$, where $r_1, \ldots, r_\ell$ are *simple* role names. Such role conjunctions are allowed to occur in existential restrictions instead of a single role, but not in at-least restrictions or role assertions. An interpretation $\mathcal{I}$ is extended to role conjunction as follows: $(r_1 \sqcap \cdots \sqcap r_\ell)^{\mathcal{I}} := r_1^{\mathcal{I}} \cap \cdots \cap r_\ell^{\mathcal{I}}$. Therefore, in this section, we consider Boolean $\mathcal{SHOQ}^{\sqcap}$-knowledge bases rather than Boolean $\mathcal{SHOQ}$-knowledge bases.

In the following, let $\mathcal{B} = (\Psi, \mathcal{R})$ be a Boolean $\mathcal{SHOQ}^{\sqcap}$-knowledge base, and let $\mathcal{D} = (\mathcal{U}, \mathcal{Y})$ be a pair such that $\mathcal{U}$ is a set of concept names occurring in $\mathcal{B}$ and $\mathcal{Y} \subseteq 2^{\mathcal{U}}$. We assume here that all GCIs occurring in Ψ are of the form $\top \sqsubseteq C$; this is without loss of generality since any GCI $C \sqsubseteq D$ is equivalent to $\top \sqsubseteq \neg(C \sqcap \neg D)$.

We show that consistency of $\mathcal{B}$ w.r.t. $\mathcal{D}$ can be decided in time exponential in the size of $\mathcal{B}$. This complexity result is tight since already for 'classical' $\mathcal{SHQ}^{\sqcap}$-knowledge bases (without nominals), the consistency problem (without $\mathcal{D}$) is ExpTime-complete [Tob01; Lut08a]. The complexity of this problem even remains in ExpTime when simple role conjunctions are allowed to occur in at-least restrictions and non-simple roles are allowed in role conjunctions in existential restrictions [GK08]. However, if we move to $\mathcal{SHOQ}^{\sqcap}$, i.e. we consider 'classical' $\mathcal{SHOQ}^{\sqcap}$-knowledge bases where simple role conjunctions are allowed to occur in at-least restrictions and non-simple roles are allowed in role conjunctions in existential restrictions, the best known upper bound of the consistency problem (without $\mathcal{D}$) is 2ExpTime [GHS08; Gli07].

The proof of our result is an adaptation of the proof of Lemma 6.4 in [BGL12], which is again an adaptation of the proof of Theorem 2.27 in [GKW+03], which shows that consistency of Boolean $\mathcal{ALC}$-knowledge bases can be decided in exponential time. An earlier version of this proof for $\mathcal{ALC}^{\sqcap}$ can be found in [BBL13a; BBL13b]. There, for role conjunctions, additional concept names are introduced that function as so-called *pebbles* that mark elements that have specific role predecessors, an idea borrowed from [Dan84; DM00; Mas01]. Here, we employ systems of equations over non-negative integers to deal with role conjunctions, transitivity axioms, role-inclusion axioms, and at-least restrictions simultaneously.

For the subsequent construction, we extend the notion of a *quasimodel* from [BGL12], which is an abstract description of a model. Quasimodels characterise domain elements by the concepts they satisfy. We start by introducing several auxiliary notions that we need in the construction.

We define $\mathsf{Con}(\Psi)$ to be the set of all concepts occurring in Ψ, $\mathsf{Ind}(\Psi)$ to be the set of all individual names occurring in Ψ, and $\mathsf{Rol}(\mathcal{B})$ to the set of all role names occurring in $\mathcal{B}$. Then, $\mathsf{Cl}_\mathsf{c}(\mathcal{B})$ is defined to be the closure under negation of the set

$$\begin{aligned}\mathsf{Con}(\Psi) &\cup \left\{\exists r.C \mid \exists s.C \in \mathsf{Con}(\Psi), \mathcal{R} \models r \sqsubseteq s,\ \text{and } \mathcal{R} \models \mathsf{trans}(r)\right\}\\ &\cup \left\{\{a\} \mid a \in \mathsf{Ind}(\Psi)\right\}\\ &\cup \left\{\exists r.\{a\} \mid r \in \mathsf{Rol}(\mathcal{B}) \text{ and } a \in \mathsf{Ind}(\Psi)\right\}.\end{aligned}$$

The reason why we consider these additional sets is that they are needed to properly deal with transitive roles and nominals (see Definition 3.21). Similarly, we define $\mathsf{Cl}_\mathsf{f}(\Psi)$ to be the closure under negation of the set of all subformulas of Ψ.

In the following, we identify $\neg\neg\psi$ with ψ for every concept ψ. Similarly, we identify $\neg\neg\Psi$ with Ψ for every Boolean axiom formula Ψ. Thus, all sets introduced above are of size polynomial in the size of $\mathcal{B}$, and can also be constructed in time polynomial in the size of $\mathcal{B}$.

Definition 3.21 (Concept type). *A* concept type *for $\mathcal{B}$ is a set $\mathbb{c} \subseteq \mathsf{Cl}_\mathsf{c}(\mathcal{B})$ such that:*

- *for every $C \sqcap D \in \mathsf{Cl}_\mathsf{c}(\mathcal{B})$, we have $C \sqcap D \in \mathbb{c}$ iff $\{C, D\} \subseteq \mathbb{c}$;*
- *for every $\neg C \in \mathsf{Cl}_\mathsf{c}(\mathcal{B})$, we have $\neg C \in \mathbb{c}$ iff $C \notin \mathbb{c}$;*
- *for every $\{a\} \in \mathsf{Cl}_\mathsf{c}(\mathcal{B})$, we have $\{a\} \in \mathbb{c}$ implies $\{b\} \notin \mathbb{c}$ for every $\{b\} \in \mathsf{Cl}_\mathsf{c}(\mathcal{B})$ with $\{b\} \neq \{a\}$; and*
- *for every $\exists r.\{a\} \in \mathsf{Cl}_\mathsf{c}(\mathcal{B})$, we have that if $\exists r.\{a\} \in \mathbb{c}$ and $\mathcal{R} \models r \sqsubseteq s$, then $\exists s.\{a\} \in \mathbb{c}$.*

Given two concept types $\mathbb{c}, \mathbb{d}$ for $\mathcal{B}$ and a role name $r \in \mathsf{Rol}(\mathcal{B})$, we say that $\mathbb{c}$ and $\mathbb{d}$ are r-compatible w.r.t. *$\mathcal{R}$ (written $\mathbb{c} \overset{r}{\rightsquigarrow}_{\mathcal{R}} \mathbb{d}$) if the following conditions are satisfied:*

- *for every $\neg(\exists r.D) \in \mathbb{c}$, we have $\neg D \in \mathbb{d}$; and*
- *for every $s \in \mathsf{Rol}(\mathcal{B})$ with $\mathcal{R} \models r \sqsubseteq s$, $\mathcal{R} \models \mathsf{trans}(r)$, and $\neg(\exists s.D) \in \mathbb{c}$, we have $\neg(\exists r.D) \in \mathbb{d}$.* ◇

Obviously, the number of concept types for $\mathcal{B}$ is exponential in the size of $\mathcal{B}$. Intuitively, the r-compatibility of two concept types $\mathbb{c}, \mathbb{d}$ w.r.t. $\mathcal{R}$ indicates that it is possible to connect them with an r-edge without violating the value restrictions in $\mathbb{c}$. These conditions are very similar to the tableau rules $(\forall)$ and $(\forall_+)$ that deal with value restrictions in the presence of role-inclusion axioms and transitivity axioms (see e.g. [HST00]).

Definition 3.22 (Role type). *A* role type *for $\mathcal{B}$ is a set $\mathbb{r} \subseteq \mathsf{Rol}(\mathcal{B})$ such that:*

- *if $\mathcal{R} \models s \sqsubseteq r$, then $s \in \mathbb{r}$ implies $r \in \mathbb{r}$.*

We denote the set of all role types for $\mathcal{B}$ by $\mathfrak{R}(\mathcal{B})$.

Given two concept types $\mathbb{c}, \mathbb{d}$ for $\mathcal{B}$ and a role type $\mathbb{r} \in \mathfrak{R}(\mathcal{B})$, we say that $\mathbb{c}$ and $\mathbb{d}$ are $\mathbb{r}$-compatible w.r.t. *$\mathcal{R}$ (written $\mathbb{c} \overset{\mathbb{r}}{\rightsquigarrow}_{\mathcal{R}} \mathbb{d}$) iff $\mathbb{c} \overset{r}{\rightsquigarrow}_{\mathcal{R}} \mathbb{d}$ for every $r \in \mathbb{r}$.* ◇

Again, the number of role types for $\mathcal{B}$ is exponential in the size of $\mathcal{B}$. Finally, a quasimodel also has to determine which of the axioms in Ψ it satisfies.

Definition 3.23 (Formula type). *A* formula type *for $\mathcal{B}$ is a set $\mathbb{f} \subseteq \mathsf{Cl_f}(\Psi)$ such that:*

- $\Psi \in \mathbb{f}$*;*
- *for every $\neg\psi \in \mathsf{Cl_f}(\Psi)$, we have $\neg\psi \in \mathbb{f}$ iff $\psi \notin \mathbb{f}$; and*
- *for every $\Psi_1 \wedge \Psi_2 \in \mathsf{Cl_f}(\Psi)$, we have $\Psi_1 \wedge \Psi_2 \in \mathbb{f}$ iff $\{\Psi_1, \Psi_2\} \subseteq \mathbb{f}$.* ◇

The number of formula types for $\mathcal{B}$ is exponential in the size of $\mathcal{B}$. Using these definitions, we can now define the notion of a model candidate, and later refine this notion to characterise quasimodels.

Definition 3.24 (Model candidate). *A* model candidate *for $\mathcal{B}$ is a triple $\mathcal{M} = (\mathcal{W}, \iota, \mathbb{f})$ such that:*

- *$\mathcal{W}$ is a set of concept types for $\mathcal{B}$ such that for any $\mathbb{c}, \mathbb{d} \in \mathcal{W}$ with $\mathbb{c} \neq \mathbb{d}$, we have $\mathbb{c} \cap \mathbb{d} \cap \big\{\{a\} \mid a \in \mathsf{Ind}(\Psi)\big\} = \emptyset$;*
- *$\iota \colon \mathsf{Ind}(\Psi) \to \mathcal{W}$ is a function such that $\{a\} \in \iota(a)$ for every $a \in \mathsf{Ind}(\Psi)$; and*
- *$\mathbb{f}$ is a formula type for $\mathcal{B}$.* ◇

Intuitively, the set $\mathcal{W}$ determines the behaviour of the domain elements, while the function ι fixes the interpretation of the named domain elements, and the formula type $\mathbb{f}$ ensures that $\mathcal{B}$ is satisfied. In the following, we denote by $\mathcal{W}_\mathsf{u}$ the set $\mathcal{W} \setminus \iota(\mathsf{Ind}(\Psi))$, i.e. the set of all those concept types for $\mathcal{B}$ that do not contain a nominal concept $\{a\}$ with $a \in \mathsf{Ind}(\Psi)$. Those types represent the unnamed domain elements of the model candidate. To define quasimodels, we add to the definition of a model candidate several conditions that ensure that it can indeed be transformed into a model of $\mathcal{B}$.

To satisfy the (negated) at-least restrictions in the concept types of a model candidate $\mathcal{M} = (\mathcal{W}, \iota, \mathbb{f})$, we introduce for each $\mathbb{c} \in \mathcal{W}$, a system of equations $E_{\mathcal{M},\mathbb{c}}$ with variables ranging over the non-negative integers. In $E_{\mathcal{M},\mathbb{c}}$, we use variables of the form $x_{\mathbb{c},\mathbb{r},\mathbb{d}}$, which determine for a domain element of type $\mathbb{c}$, the number of *unnamed* role successors of type $\mathbb{r}$ (called $\mathbb{r}$-successors) of concept type $\mathbb{d}$, where we require that $\mathbb{c} \overset{\mathbb{r}}{\rightsquigarrow}_{\mathcal{R}} \mathbb{d}$ and $\mathbb{d} \in \mathcal{W}_\mathsf{u}$, i.e. $\mathbb{c}$ and $\mathbb{d}$ are $\mathbb{r}$-compatible w.r.t. $\mathcal{R}$ and $\mathbb{d}$ does not represent a named individual.

Given $\mathbb{c} \in \mathcal{W}$, $C \in \mathsf{Cl_c}(\mathcal{B})$, and $\mathbb{r} \in \mathfrak{R}(\mathcal{B})$, we can now count the number of *unnamed* $\mathbb{r}$-successors of $\mathbb{c}$ that satisfy C using the following expression:

$$\Xi_{\mathcal{M},\mathbb{c},\mathbb{r},C} := \sum_{C \in \mathbb{d} \in \mathcal{W}_\mathsf{u},\ \mathbb{c} \overset{\mathbb{r}}{\rightsquigarrow}_{\mathcal{R}} \mathbb{d}} x_{\mathbb{c},\mathbb{r},\mathbb{d}}.$$

To count the *named* $\mathbb{r}$-successors of $\mathbb{c}$ that satisfy C, we define the following constant:

$$\Gamma_{\mathcal{M},\mathbb{c},\mathbb{r},C} := \Big|\big\{b \in \mathsf{Ind}(\Psi) \mid C \in \iota(b), \text{ and } \exists r.\{b\} \in \mathbb{c} \text{ iff } r \in \mathbb{r}\big\}\Big|.$$

To ensure that the at-least restrictions in $\mathbb{c}$ are satisfied, we add the following equation to $E_{\mathcal{M},\mathbb{c}}$ for each ${\geq}n\, r.C \in \mathbb{c}$:

$$-y_{\mathbb{c},\geq n\, r.C} + \sum_{r \in \mathbb{r} \in \mathfrak{R}(\mathcal{B})} (\Xi_{\mathcal{M},\mathbb{c},\mathbb{r},C} + \Gamma_{\mathcal{M},\mathbb{c},\mathbb{r},C}) = n, \tag{E1}$$

where $y_{\mathbb{c},\geq n\, r.C}$ is a slack variable that is used to obtain an equation instead of an inequation. Similarly, for each $\neg(\geq n\, r.C) \in \mathbb{c}$, we add

$$y_{\mathbb{c},\neg(\geq n\, r.C)} + \sum_{r \in \mathbb{r} \in \mathfrak{R}(\mathcal{B})} (\Xi_{\mathcal{M},\mathbb{c},\mathbb{r},C} + \Gamma_{\mathcal{M},\mathbb{c},\mathbb{r},C}) = n - 1. \tag{E2}$$

For each existential restriction $D = \exists(r_1 \sqcap \cdots \sqcap r_\ell).C \in \mathbb{c}$, we add the following equation to $E_{\mathcal{M},\mathbb{c}}$:

$$-y_{\mathbb{c},D} + \sum_{\{r_1,\ldots,r_\ell\} \subseteq \mathbb{r} \in \mathfrak{R}(\mathcal{B})} (\Xi_{\mathcal{M},\mathbb{c},\mathbb{r},C} + \Gamma_{\mathcal{M},\mathbb{c},\mathbb{r},C}) = 1. \tag{E3}$$

Finally, for each $\neg(\exists(r_1 \sqcap \cdots \sqcap r_\ell).C) \in \mathbb{c}$, we add the equation

$$\sum_{\{r_1,\ldots,r_\ell\} \subseteq \mathbb{r} \in \mathfrak{R}(\mathcal{B})} (\Xi_{\mathcal{M},\mathbb{c},\mathbb{r},C} + \Gamma_{\mathcal{M},\mathbb{c},\mathbb{r},C}) = 0, \tag{E4}$$

where no slack variable is needed since the sum cannot be smaller than 0.

This finishes the description of the system of equations $E_{\mathcal{M},\mathbb{c}}$. Note that this system contains *exponentially* many variables in the size of $\mathcal{B}$, but only *polynomially* many equations, and thus it can be solved in exponential time, even if the numbers are given in binary encoding [Pap81] (for details, see the proof of Theorem 3.33).

Now we are ready to introduce the notion of a quasimodel.

Definition 3.25 (Quasimodel). *The model candidate $\mathcal{M} = (\mathcal{W}, \iota, \mathbb{f})$ for $\mathcal{B}$ is a* quasimodel *for $\mathcal{B}$ if it satisfies the following properties:*

(a) $\mathcal{W}$ is not empty;

(b) for every $A(a) \in \mathsf{Cl_f}(\Psi)$, we have $A(a) \in \mathbb{f}$ iff $A \in \iota(a)$;

(c) for every $r(a,b) \in \mathsf{Cl_f}(\Psi)$, we have $r(a,b) \in \mathbb{f}$ iff $\exists r.\{b\} \in \iota(a)$;

(d) for every $\top \sqsubseteq C \in \mathbb{f}$ and every $\mathbb{c} \in \mathcal{W}$, we have $C \in \mathbb{c}$;

(e) for every $\neg(\top \sqsubseteq C) \in \mathbb{f}$, there is a $\mathbb{c} \in \mathcal{W}$ such that $C \notin \mathbb{c}$;

(f) for every $\mathbb{c} \in \mathcal{W}$, if $\exists r.\{a\} \in \mathbb{c}$, then $\mathbb{c} \overset{r}{\rightsquigarrow}_{\mathcal{R}} \iota(a)$; and

(g) for every $\mathbb{c} \in \mathcal{W}$, the system of equations $E_{\mathcal{M},\mathbb{c}}$ has a solution over the non-negative integers.

The quasimodel $\mathcal{M} = (\mathcal{W}, \iota, \mathbb{f})$ for $\mathcal{B}$ respects *$\mathcal{D} = (\mathcal{U}, \mathcal{Y})$ if it additionally satisfies:*

(h) for every $\mathbb{c} \in \mathcal{W}$, there is a set $Y \in \mathcal{Y}$ such that $Y = \mathbb{c} \cap \mathcal{U}$; and

(i) for every $Y \in \mathcal{Y}$, there is a concept type $\mathbb{c} \in \mathcal{W}$ such that $Y = \mathbb{c} \cap \mathcal{U}$. ◇

We can show to in order to check consistency of $\mathcal{B}$ w.r.t. $\mathcal{D}$ it suffices to search for a quasimodel for $\mathcal{B}$ that respects $\mathcal{D}$.

Lemma 3.26. *Let $\mathcal{B}$ be a Boolean $\mathcal{SHOQ}^\sqcap$-knowledge base, and let $\mathcal{D} = (\mathcal{U}, \mathcal{Y})$ be a pair such that $\mathcal{U}$ is a set of concept names occurring in $\mathcal{B}$ and $\mathcal{Y} \subseteq 2^{\mathcal{U}}$. Then, $\mathcal{B}$ is consistent w.r.t. $\mathcal{D}$ iff there is a quasimodel for $\mathcal{B}$ that respects $\mathcal{D}$.*

Proof. For the 'if' direction, suppose that $\mathcal{M} = (\mathcal{W}, \iota, \mathbb{f})$ is a quasimodel for $\mathcal{B} = (\Psi, \mathcal{R})$ that respects $\mathcal{D}$. Then by Condition (g), for each $\mathbb{c} \in \mathcal{W}$, the system of equations $E_{\mathcal{M},\mathbb{c}}$ has a solution $\nu_{\mathbb{c}}$ that maps the variables in $E_{\mathcal{M},\mathbb{c}}$ to non-negative integers. Let $\mathfrak{z}_{\mathcal{M}}$ be the greatest non-negative integer that occurs in any of these solutions, and let $\mathfrak{Z}$ denote the set $\{1, \dots, \mathfrak{z}_{\mathcal{M}}\}$.

We define the interpretation $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ as follows:

- $\Delta^{\mathcal{J}} := \mathsf{Anon} \cup \mathsf{Ind}(\Psi)$, where $\mathsf{Anon} := \mathcal{W}_{\mathsf{u}} \times \mathfrak{Z} \times \mathfrak{R}(\mathcal{B})$;
- $a^{\mathcal{J}} := a$ for every $a \in \mathsf{Ind}(\Psi)$;[4]
- $A^{\mathcal{J}} := \{(\mathbb{c}, i, \mathbb{r}) \in \mathsf{Anon} \mid A \in \mathbb{c}\} \cup \{a \in \mathsf{Ind}(\Psi) \mid A \in \iota(a)\}$ for every $A \in \mathsf{N_C}$; and
- for every $r \in \mathsf{N_R}$, $(\mathbb{c}, i, \mathbb{r}), (\mathbb{d}, j, \mathbb{s}) \in \mathsf{Anon}$, and $a, b \in \mathsf{Ind}(\Psi)$, we define:

$$\begin{array}{lll}
(a, b) \in r^{\mathcal{J}} & \text{iff} & \exists r.\{b\} \in \iota(a); \\
\big((\mathbb{c}, i, \mathbb{r}), b\big) \in r^{\mathcal{J}} & \text{iff} & \exists r.\{b\} \in \mathbb{c}; \\
(a, (\mathbb{d}, j, \mathbb{s})) \in r^{\mathcal{J}} & \text{iff} & r \in \mathbb{s},\ \iota(a) \overset{\mathbb{s}}{\rightsquigarrow}_{\mathcal{R}} \mathbb{d}, \text{ and } \nu_{\iota(a)}(x_{\iota(a),\mathbb{s},\mathbb{d}}) \geq j; \\
\big((\mathbb{c}, i, \mathbb{r}), (\mathbb{d}, j, \mathbb{s})\big) \in r^{\mathcal{J}} & \text{iff} & r \in \mathbb{s},\ \mathbb{c} \overset{\mathbb{s}}{\rightsquigarrow}_{\mathcal{R}} \mathbb{d}, \text{ and } \nu_{\mathbb{c}}(x_{\mathbb{c},\mathbb{s},\mathbb{d}}) \geq j.
\end{array}$$

Note that $\Delta^{\mathcal{J}} \neq \emptyset$ since even if $\mathsf{Ind}(\Psi) = \emptyset$, we have that $\mathcal{W}_{\mathsf{u}} \neq \emptyset$ by Condition (a).

Now we construct a model $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of $\mathcal{B}$ by defining $\Delta^{\mathcal{I}} := \Delta^{\mathcal{J}}$, for each $A \in \mathsf{N_C}$, $A^{\mathcal{I}} := A^{\mathcal{J}}$, for each $a \in \mathsf{Ind}(\Psi)$, $a^{\mathcal{I}} := a^{\mathcal{J}}$, and for each $r \in \mathsf{N_R}$,

$$r^{\mathcal{I}} := r^{\mathcal{J}} \cup \bigcup_{\mathcal{R} \models s \sqsubseteq r,\ \mathcal{R} \models \mathsf{trans}(s)} (s^{\mathcal{J}})^+,$$

where $\cdot^+$ denotes the transitive closure.

We denote by $\kappa \colon \Delta^{\mathcal{I}} \to \mathcal{W}$ the following function:

$$\kappa(d) := \begin{cases} \mathbb{c} & \text{if } d = (\mathbb{c}, i, \mathbb{r}) \in \mathsf{Anon}, \text{ and} \\ \iota(b) & \text{if } d = b \in \mathsf{Ind}(\Psi). \end{cases}$$

The following claim can be proved by a careful case distinction.

Claim 3.27. *Let $d \in \Delta^{\mathcal{I}}$. If $\neg(\exists r.D) \in \kappa(d)$, and there is an $s \in \mathsf{Rol}(\mathcal{B})$ and an $e \in \Delta^{\mathcal{I}}$ with $\mathcal{R} \models s \sqsubseteq r$ and $(d, e) \in s^{\mathcal{J}}$, then we have:*

- *$\neg D \in \kappa(e)$, and*
- *if $\mathcal{R} \models \mathsf{trans}(s)$, then $\neg(\exists s.D) \in \kappa(e)$.*

Assume first that $e = b \in \mathsf{Ind}(\Psi)$. Then the definition of $\mathcal{J}$ yields that $\exists s.\{b\} \in \kappa(d)$. Since $\kappa(d)$ is a concept type, we by Definition 3.21 that $\exists r.\{b\} \in \kappa(d)$. By Condition (f), this implies $\kappa(d) \overset{r}{\rightsquigarrow}_{\mathcal{R}} \iota(b)$, and thus $\kappa(d) \overset{r}{\rightsquigarrow}_{\mathcal{R}} \kappa(e)$. By Definition 3.21, we obtain $\neg D \in \kappa(e)$. Moreover, by Condition (f), we have $\kappa(d) \overset{s}{\rightsquigarrow}_{\mathcal{R}} \iota(b)$, and thus $\kappa(d) \overset{s}{\rightsquigarrow}_{\mathcal{R}} \kappa(e)$. Hence, if $\mathcal{R} \models \mathsf{trans}(s)$, we have by Definition 3.21 also that $\neg(\exists s.D) \in \kappa(e)$.

[4] For now, we ignore the individual names in $\mathsf{N_I} \setminus \mathsf{Ind}(\Psi)$ since they are irrelevant for the consistency of $\mathcal{B}$. After constructing the model $\mathcal{I}$ below, one can ensure that it respects the UNA by constructing the countably infinite disjoint union of $\mathcal{I}$ with itself to allow for different interpretations of each of these individual names.

Assume now that $e = (\mathbb{d}, j, \mathbb{s}) \in \mathsf{Anon}$. The definition of $\mathcal{J}$ yields that $s \in \mathbb{s}$, $\kappa(d) \overset{\mathbb{s}}{\rightsquigarrow}_{\mathcal{R}} \mathbb{d}$, and $\nu_{\kappa(d)}(x_{\kappa(d),\mathbb{s},\mathbb{d}}) \geq j$. Since $\mathbb{s}$ is a role type, and $\mathcal{R} \models s \sqsubseteq r$, we have $r \in \mathbb{s}$. Thus, we have by Definition 3.22 that $\kappa(d) \overset{r}{\rightsquigarrow}_{\mathcal{R}} \mathbb{d}$. By Definition 3.21, we obtain $\neg D \in \mathbb{d}$, and thus $\neg D \in \kappa(e)$. Moreover, since $s \in \mathbb{s}$, we have by Definition 3.22 that $\kappa(d) \overset{s}{\rightsquigarrow}_{\mathcal{R}} \mathbb{d}$. Hence, if $\mathcal{R} \models \mathsf{trans}(s)$, we have by Definition 3.21 also that $\neg(\exists s.D) \in \mathbb{d}$, and thus $\neg(\exists s.D) \in \kappa(e)$. This finishes the proof of Claim 3.27.

Using Claim 3.27, we now prove the following claim by structural induction.

Claim 3.28. *For every concept $C \in \mathsf{Cl_c}(\mathcal{B})$, we have $C^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} \mid C \in \kappa(d)\}$.*

For the base case, C being a concept name, the definition of $\mathcal{I}$ nd the definition of κ immediately imply the claim.

For the case that C is of the form $\neg D$, we have by the semantics of $\mathcal{SHOQ}^{\sqcap}$, the induction hypothesis, the definition of $\mathcal{I}$, the definition of κ, and the definition of concept types the following for every $d \in \Delta^{\mathcal{I}}$:

$$d \in (\neg D)^{\mathcal{I}} \text{ iff } d \notin D^{\mathcal{I}} \text{ iff } D \notin \kappa(d) \text{ iff } \neg D \in \kappa(d).$$

For the case that C is of the form $D \sqcap E$, we have by similar arguments the following for every $d \in \Delta^{\mathcal{I}}$:

$$d \in (D \sqcap E)^{\mathcal{I}} \text{ iff } d \in D^{\mathcal{I}} \text{ and } d \in E^{\mathcal{I}} \text{ iff } D \in \kappa(d) \text{ and } E \in \kappa(d) \text{ iff } D \sqcap E \in \kappa(d).$$

For the case that C is of the form $\exists(r_1 \sqcap \cdots \sqcap r_\ell).D$, we have by similar arguments the following:

$$\begin{aligned}
&(\exists(r_1 \sqcap \cdots \sqcap r_\ell).D)^{\mathcal{I}} \\
&\qquad = \{d \in \Delta^{\mathcal{I}} \mid \text{there is an } e \in \Delta^{\mathcal{I}} \text{ with } (d,e) \in r_1^{\mathcal{I}} \cap \cdots \cap r_\ell^{\mathcal{I}} \text{ and } e \in D^{\mathcal{I}}\} \\
&\qquad = \{d \in \Delta^{\mathcal{I}} \mid \text{there is an } e \in \Delta^{\mathcal{I}} \text{ with } (d,e) \in r_1^{\mathcal{I}} \cap \cdots \cap r_\ell^{\mathcal{I}} \text{ and } D \in \kappa(e)\} \\
&\qquad \overset{*}{=} \{d \in \Delta^{\mathcal{I}} \mid \exists(r_1 \sqcap \cdots \sqcap r_\ell).D \in \kappa(d)\}
\end{aligned}$$

The starred equality $\overset{*}{=}$ holds due to the following arguments. Assume, for the direction ($\supseteq$), that $d \in \Delta^{\mathcal{I}}$ and $\exists(r_1 \sqcap \cdots \sqcap r_\ell).D \in \kappa(d)$. Since $\nu_{\kappa(d)}$ solves (E3), there is an $\mathbb{r} \in \mathcal{R}(\mathcal{B})$ such that $\{r_1, \ldots, r_\ell\} \subseteq \mathbb{r}$ and

- either there is a $\mathbb{d} \in \mathcal{W}_{\mathsf{u}}$ with $D \in \mathbb{d}$, $\kappa(d) \overset{\mathbb{r}}{\rightsquigarrow}_{\mathcal{R}} \mathbb{d}$, and $\nu_{\kappa(d)}(x_{\kappa(d),\mathbb{r},\mathbb{d}}) \geq 1$; or
- there is a $b \in \mathsf{Ind}(\Psi)$ such that $D \in \iota(b)$ and $\{\exists r_1.\{b\}, \ldots, \exists r_\ell.\{b\}\} \subseteq \kappa(d)$.

The definition of $r_1^{\mathcal{J}}, \ldots, r_\ell^{\mathcal{J}}$ yields in the first case that

$$(d, (\mathbb{d}, 1, \mathbb{r})) \in r_1^{\mathcal{J}} \cap \cdots \cap r_\ell^{\mathcal{J}} \subseteq r_1^{\mathcal{I}} \cap \cdots \cap r_\ell^{\mathcal{I}},$$

and in the second case that

$$(d, b) \in r_1^{\mathcal{J}} \cap \cdots \cap r_\ell^{\mathcal{J}} \subseteq r_1^{\mathcal{I}} \cap \cdots \cap r_\ell^{\mathcal{I}}.$$

Since we have also $D \in \kappa\big((\mathbb{d}, 1, \mathbb{r})\big)$ and $D \in \kappa(b)$, this finishes this direction.

For the other direction ($\subseteq$), take any $d \in \Delta^{\mathcal{I}}$ such that there is an $e \in \Delta^{\mathcal{I}}$ with the property that $(d,e) \in r_1^{\mathcal{I}} \cap \cdots \cap r_\ell^{\mathcal{I}}$ and $D \in \kappa(e)$. We show that $C = \exists(r_1 \sqcap \cdots \sqcap r_\ell).D \in \kappa(d)$. Assume to the contrary that $C \notin \kappa(d)$, and thus $\neg C \in \kappa(d)$.

- For the case $\ell > 1$, we have that $r_1, \ldots, r_\ell$ are *simple* role names, and thus that $(d,e) \in r_1^{\mathcal{J}} \cap \cdots \cap r_\ell^{\mathcal{J}}$. For the case that $e = b \in \mathsf{Ind}(\Psi)$, we have by the definition of $\mathcal{J}$ that $\big\{\exists r_1.\{b\}, \ldots, \exists r_\ell.\{b\}\big\} \in \kappa(d)$. Take the set $\mathbb{r} := \big\{r \in \mathsf{Rol}(\mathcal{B}) \mid \exists r.\{b\} \in \kappa(d)\big\}$. Since $\kappa(d)$ is a concept type, we have by Definition 3.21 that $\mathbb{r}$ is a role type that contains $r_1, \ldots, r_\ell$. Since $D \in \kappa(e) = \iota(b)$, we have that $\Gamma_{\mathcal{M},\kappa(d),\mathbb{r},D} \geq 1$, which contradicts the assumption that (E4) has a solution. For the case that $e = (\mathbb{d}, j, \mathbb{s}) \in \mathsf{Anon}$, we have by the definition of $\mathcal{J}$ that $\{r_1, \ldots, r_\ell\} \subseteq \mathbb{s}$, $\kappa(d) \overset{\mathbb{s}}{\rightsquigarrow}_{\mathcal{R}} \mathbb{d}$, and $\nu_{\kappa(d)}(x_{\kappa(d),\mathbb{s},\mathbb{d}}) \geq j \geq 1$. Since $\nu_{\kappa(d)}$ is a solution of (E4), we must have $\nu_{\kappa(d)}(x_{\kappa(d),\mathbb{s},\mathbb{d}}) = 0$, which is again a contradiction.

- For the case $\ell = 1$, we have by the definition of $r_1^{\mathcal{I}}$ that $(d,e) \in r_1^{\mathcal{J}}$ or $(d,e) \in (s^{\mathcal{J}})^+$ for some $s \in \mathsf{N_R}$ with $\mathcal{R} \models s \sqsubseteq r_1$ and $\mathcal{R} \models \mathsf{trans}(s)$. The first case can be handled as in the case of $\ell > 1$. In the second case, there is a sequence $d_0, \ldots, d_n$ in $\Delta^{\mathcal{I}}$ such that $n \geq 1$, $d_0 = d$, $d_n = e$, and for every k, $0 \leq k \leq n-1$, we have that $(d_k, d_{k+1}) \in s^{\mathcal{J}}$.
  - If $n = 1$, then we have $(d,e) \in s^{\mathcal{J}}$. Thus, we have by Claim 3.27 that $\neg D \in \kappa(e)$, which is a contradiction.
  - If $n > 1$, we have by Claim 3.27 that $\neg(\exists s.D) \in \kappa(d_1)$. Since $\mathcal{R} \models s \sqsubseteq s$, using Claim 3.27 again, we obtain that $\neg(\exists s.D) \in \kappa(d_{n-1})$. By Claim 3.27, we have $\neg D \in \kappa(d_n) = \kappa(e)$, which is again a contradiction.

Finally, consider the case that C is of the form $\geq n\, r.D$. Recall that r must be *simple*, and thus $r^{\mathcal{I}} = r^{\mathcal{J}}$. We first count, for any element $d \in \Delta^{\mathcal{I}}$, the number n_1 of *unnamed* $r^{\mathcal{J}}$-successors that satisfy D. For a fixed role type $\mathbb{s} \in \mathfrak{R}(\mathcal{B})$ and concept type $\mathbb{d} \in \mathcal{W}_{\mathsf{u}}$ with $r \in \mathbb{s}$, $D \in \mathbb{d}$, and $\kappa(d) \overset{\mathbb{s}}{\rightsquigarrow}_{\mathcal{R}} \mathbb{d}$, we have by definition of $\mathcal{J}$ that $(d, (\mathbb{d}, j, \mathbb{s})) \in r^{\mathcal{J}}$ iff $\nu_{\kappa(d)}(x_{\kappa(d),\mathbb{s},\mathbb{d}}) \geq j$. Thus, the number of $r^{\mathcal{J}}$-successors of d that are of the form $(\mathbb{d}, j, \mathbb{s})$ is exactly $\nu_{\kappa(d)}(x_{\kappa(d),\mathbb{s},\mathbb{d}})$. By induction, we obtain the following:

$$\begin{aligned}
n_1 &= |\{(\mathbb{d}, j, \mathbb{s}) \in \mathsf{Anon} \mid (d, (\mathbb{d}, j, \mathbb{s})) \in r^{\mathcal{J}},\ (\mathbb{d}, j, \mathbb{s}) \in D^{\mathcal{I}}\}| \\
&= |\{(\mathbb{d}, j, \mathbb{s}) \in \mathsf{Anon} \mid (d, (\mathbb{d}, j, \mathbb{s})) \in r^{\mathcal{J}},\ D \in \mathbb{d}\}| \\
&= \sum_{\substack{r \in \mathbb{s} \in \mathfrak{R}(\mathcal{B}) \\ D \in \mathbb{d} \in \mathcal{W}_{\mathsf{u}},\ \kappa(d) \overset{\mathbb{s}}{\rightsquigarrow}_{\mathcal{R}} \mathbb{d}}} |\{j \in \mathfrak{Z} \mid (d, (\mathbb{d}, j, \mathbb{s})) \in r^{\mathcal{J}}\}| \\
&= \sum_{\substack{r \in \mathbb{s} \in \mathfrak{R}(\mathcal{B}) \\ D \in \mathbb{d} \in \mathcal{W}_{\mathsf{u}},\ \kappa(d) \overset{\mathbb{s}}{\rightsquigarrow}_{\mathcal{R}} \mathbb{d}}} \nu_{\kappa(d)}(x_{\kappa(d),\mathbb{s},\mathbb{d}}) \ = \sum_{r \in \mathbb{s} \in \mathfrak{R}(\mathcal{B})} \nu_{\kappa(d)}(\Xi_{\mathcal{M},\kappa(d),\mathbb{s},D}).
\end{aligned}$$

Similarly, we count the number n_2 of *named* $r^{\mathcal{J}}$-successors of $d \in \Delta^{\mathcal{I}}$ that satisfy D. Take again the set $\mathbb{r} := \big\{r \in \mathsf{Rol}(\mathcal{B}) \mid \exists r.\{b\} \in \kappa(d)\big\}$. Since $\kappa(d)$ is a concept type, we have by

Definition 3.21 that $\mathbb{r}$ is a role type. By the definition of $\mathcal{J}$, this yields for every $b \in \mathsf{Ind}(\Psi)$ that $(d, b) \in r^{\mathcal{J}}$ iff $\exists r.\{b\} \in \kappa(d)$ iff $r \in \mathbb{r}$. Thus, by induction, we obtain the following:

$$\begin{aligned}
n_2 &= |\{b \in \mathsf{Ind}(\Psi) \mid (d, b) \in r^{\mathcal{J}},\ b \in D^{\mathcal{I}}\}| \\
&= |\{b \in \mathsf{Ind}(\Psi) \mid (d, b) \in r^{\mathcal{J}},\ D \in \iota(b)\}| \\
&= \sum_{r \in \mathbb{s} \in \mathfrak{R}(\mathcal{B})} |\{b \in \mathsf{Ind}(\Psi) \mid D \in \iota(b), \text{ and } (d, b) \in s^{\mathcal{J}} \text{ iff } s \in \mathbb{s}\}| \\
&= \sum_{r \in \mathbb{s} \in \mathfrak{R}(\mathcal{B})} \Big|\Big\{b \in \mathsf{Ind}(\Psi) \mid D \in \iota(b), \text{ and } \exists s.\{b\} \in \kappa(d) \text{ iff } s \in \mathbb{s}\Big\}\Big| \\
&= \sum_{r \in \mathbb{s} \in \mathfrak{R}(\mathcal{B})} \Gamma_{\mathcal{M},\kappa(d),\mathbb{s},D}.
\end{aligned}$$

For every $d \in \Delta^{\mathcal{I}}$, we know that $\nu_{\kappa(d)}$ solves the equations in (E1) and (E2). Thus, we have ${\geq} n\, r.D \in \kappa(d)$ iff the sum of n_1 and n_2 is greater or equal to n iff $d \in ({\geq} n\, r.D)^{\mathcal{I}}$. This finishes the proof of Claim 3.28.

To show that $\mathcal{I}$ is indeed a model of $\mathcal{B}$, we first show the following claim by structural induction.

Claim 3.29. *For all $\psi \in \mathsf{Cl}_{\mathsf{f}}(\Psi)$, we have $\psi \in \mathbb{f}$ iff $\mathcal{I} \models \psi$.*

For the first base case, assume that ψ is of the form $A(a)$ for $A \in \mathsf{N_C}$ and $a \in \mathsf{N_I}$. We have $A(a) \in \mathbb{f}$ iff $A \in \iota(a)$ by Condition (b). Thus, $A(a) \in \mathbb{f}$ iff $a^{\mathcal{I}} = a^{\mathcal{J}} = a \in A^{\mathcal{J}} = A^{\mathcal{I}}$ iff $\mathcal{I} \models A(a)$.

For the second base case, assume that ψ is of the form $r(a, b)$ for $a, b \in \mathsf{N_I}$ and $r \in \mathsf{N_R}$. If $r(a, b) \in \mathbb{f}$, we have by Condition (c) that $\exists r.\{b\} \in \iota(a)$, and thus $(a, b) \in r^{\mathcal{J}}$ by the definition of $r^{\mathcal{J}}$. Since $r^{\mathcal{J}} \subseteq r^{\mathcal{I}}$, $a = a^{\mathcal{I}}$, and $b = b^{\mathcal{I}}$, we obtain $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$, and thus $\mathcal{I} \models r(a, b)$.

Conversely, if $\mathcal{I} \models r(a, b)$, we have by the definition of $r^{\mathcal{I}}$ that $(a, b) \in r^{\mathcal{J}}$ or $(a, b) \in (s^{\mathcal{J}})^+$ for some $s \in \mathsf{N_R}$ with $\mathcal{R} \models s \sqsubseteq r$ and $\mathcal{R} \models \mathsf{trans}(s)$. If $(a, b) \in r^{\mathcal{J}}$, the definition of $r^{\mathcal{J}}$ implies that $\exists r.\{b\} \in \iota(a)$. This yields by Condition (c) that $r(a, b) \in \mathbb{f}$. Otherwise, there is a sequence $d_0, \dots, d_n$ in $\Delta^{\mathcal{I}}$ such that $n \geq 1$, $d_0 = a$, $d_n = b$, and for every k, $0 \leq k \leq n-1$, we have that $(d_k, d_{k+1}) \in s^{\mathcal{J}}$. Assume to the contrary that $\exists s.\{b\} \notin \iota(a)$. Thus, $\neg(\exists s.\{b\}) \in \iota(a)$.

- If $n = 1$, then we have $(a, b) \in s^{\mathcal{J}}$. Thus, we have by Claim 3.27 that $\neg\{b\} \in \iota(b)$, which is a contradiction.
- If $n > 1$, we have by Claim 3.27 that $\neg(\exists s.\{b\}) \in \kappa(d_1)$. Using Claim 3.27 again, we obtain that $\neg(\exists s.\{b\}) \in \kappa(d_{n-1})$. By Claim 3.27, we have $\neg\{b\} \in \kappa(d_n) = \iota(b)$, which is again a contradiction.

Hence, $\exists s.\{b\} \in \iota(a)$. Since $\iota(a)$ is a concept type, Definition 3.21 yields that $\exists r.\{b\} \in \iota(a)$. Thus, by Condition (c), we obtain $r(a, b) \in \mathbb{f}$.

For the third base case, assume that ψ is of the form $\top \sqsubseteq C$. If $\top \sqsubseteq C \in \mathbb{f}$, then for every $\mathbb{c} \in \mathcal{W}$, we have $C \in \mathbb{c}$ by Condition (d). Since ι maps into $\mathcal{W}$, we have that $C \in \kappa(d)$ for every $d \in \Delta^{\mathcal{I}}$. Hence, Claim 3.28 yields $C^{\mathcal{I}} = \Delta^{\mathcal{I}}$. For the converse direction, if $\top \sqsubseteq C \notin \mathbb{f}$, then by the definition of a formula type, $\neg(\top \sqsubseteq C) \in \mathbb{f}$. Then, by Condition (e), there is a

$\mathbb{c} \in \mathcal{W}$ such that $C \notin \mathbb{c}$, which implies $\neg C \in \mathbb{c}$, because $\mathbb{c}$ is a concept type. Hence, there is a $d \in \Delta^{\mathcal{I}}$ such that $\neg C \in \kappa(d)$. Claim 3.28 yields that $d \in (\neg C)^{\mathcal{I}}$. Thus, we have that $C^{\mathcal{I}} \neq \Delta^{\mathcal{I}}$.

For the induction step, assume first that ψ is of the form $\neg\psi_1$. By induction, we have $\psi \in \mathbb{f}$ iff $\psi_1 \notin \mathbb{f}$ iff $\mathcal{I} \not\models \psi_1$ iff $\mathcal{I} \models \neg\psi_1$. Similarly, if ψ is of the form $\psi_1 \wedge \psi_2$, then $\psi \in \mathbb{f}$ iff $\{\psi_1, \psi_2\} \subseteq \mathbb{f}$ iff $\mathcal{I} \models \psi_1$ and $\mathcal{I} \models \psi_2$ iff $\mathcal{I} \models \psi_1 \wedge \psi_2$. This finishes the proof of Claim 3.29.

Since $\mathbb{f}$ is a formula type for Ψ, we have $\Psi \in \mathbb{f}$, and thus together with Claim 3.29 that $\mathcal{I} \models \Psi$. We now show that $\mathcal{I}$ is also a model of $\mathcal{R}$.

Claim 3.30. *For every $\alpha \in \mathcal{R}$, we have $\mathcal{I} \models \alpha$.*

Assume first that α is of the form $r \sqsubseteq s$. Since $r \sqsubseteq s \in \mathcal{R}$, we have also $\mathcal{R} \models r \sqsubseteq s$. We first show that $r^{\mathcal{J}} \subseteq s^{\mathcal{J}}$. For this, take $(d,e) \in r^{\mathcal{J}}$. There are two cases to consider:

- If $e = b \in \mathsf{Ind}(\Psi)$, then the definition of $r^{\mathcal{J}}$ yields that $\exists r.\{b\} \in \kappa(d)$. Since $\mathcal{R} \models r \sqsubseteq s$, we have $\exists s.\{b\} \in \kappa(d)$ since $\kappa(d)$ is a concept type (see Definition 3.21). The definition of $s^{\mathcal{J}}$ yields that $(d,b) \in s^{\mathcal{J}}$.
- If $e = (\mathbb{d}, j, \mathbb{s}) \in \mathsf{Anon}$, then the definition of $r^{\mathcal{J}}$ yields that $r \in \mathbb{s}$, $\kappa(d) \overset{\mathbb{s}}{\rightsquigarrow}_{\mathcal{R}} \mathbb{d}$, and $v_{\kappa(d)}(x_{\kappa(d),\mathbb{s},\mathbb{d}}) \geq j$. Then, $\mathcal{R} \models r \sqsubseteq s$ yields $s \in \mathbb{s}$ since $\mathbb{s}$ is a role type (see Definition 3.22). Hence, the definition of $s^{\mathcal{J}}$ yields that $(d, (\mathbb{d}, j, \mathbb{s})) \in s^{\mathcal{J}}$.

To show that $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$, take $(d,e) \in r^{\mathcal{I}}$. If $(d,e) \in r^{\mathcal{J}}$, we have $(d,e) \in s^{\mathcal{J}}$, and thus $(d,e) \in s^{\mathcal{I}}$. Otherwise, we have $(d,e) \in (t^{\mathcal{J}})^+$ with $\mathcal{R} \models t \sqsubseteq r$ and $\mathcal{R} \models \mathsf{trans}(t)$. Since $\mathcal{R} \models r \sqsubseteq s$, we also have $\mathcal{R} \models t \sqsubseteq s$. The definition of $s^{\mathcal{I}}$ yields that $(t^{\mathcal{J}})^+ \subseteq s^{\mathcal{I}}$, and hence $(d,e) \in s^{\mathcal{I}}$.

Assume now that α is of the form $\mathsf{trans}(r)$. Since $\mathsf{trans}(r) \in \mathcal{R}$, we have also that $\mathcal{R} \models \mathsf{trans}(r)$. Obviously, also $\mathcal{R} \models r \sqsubseteq r$ holds. By the same arguments as above, we have that for each t with $\mathcal{R} \models t \sqsubseteq r$ that $t^{\mathcal{J}} \subseteq r^{\mathcal{J}}$, and thus $(t^{\mathcal{J}})^+ \subseteq (r^{\mathcal{J}})^+$ since the transitive closure is monotonic. This yields that $r^{\mathcal{I}} = (r^{\mathcal{J}})^+$, and thus $\mathcal{I} \models \mathsf{trans}(r)$. This finishes the proof of Claim 3.30.

Thus, we have shown that $\mathcal{I}$ is indeed a model of $\mathcal{B}$. It only remains to be shown that $\mathcal{I}$ respects $\mathcal{D}$.

By Condition (h), we have for every $d \in \Delta^{\mathcal{I}}$ that there is a set $Y \in \mathcal{Y}$ such that $Y = \kappa(d) \cap \mathcal{U}$. Claim 3.28 yields that $d \in (C_{\mathcal{U},Y})^{\mathcal{I}}$. By Condition (i), we have for every $Y \in \mathcal{Y}$ that there is a $d \in \Delta^{\mathcal{I}}$ such that $Y = \kappa(d) \cap \mathcal{U}$. Claim 3.28 yields again that $d \in (C_{\mathcal{U},Y})^{\mathcal{I}}$. This shows that $\mathcal{I}$ respects $\mathcal{D}$, and finishes the proof of the 'if' direction of the lemma.

For the 'only if' direction, assume that there is a model $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of $\mathcal{B} = (\Psi, \mathcal{R})$ that respects $\mathcal{D}$. We now construct a quasimodel for $\mathcal{B}$. Let $\tau(d) := \{C \in \mathsf{Cl_c}(\mathcal{B}) \mid d \in C^{\mathcal{I}}\}$ for $d \in \Delta^{\mathcal{I}}$. We define $\mathcal{M} = (\mathcal{W}, \iota, \mathbb{f})$ as follows:

- $\mathcal{W} := \{\tau(d) \mid d \in \Delta^{\mathcal{I}}\}$;
- $\iota(a) := \tau(a^{\mathcal{I}})$ for every $a \in \mathsf{Ind}(\Psi)$; and
- $\mathbb{f} := \{\psi \in \mathsf{Cl_f}(\Psi) \mid \mathcal{I} \models \psi\}$.

We first show that $\mathcal{W}$ is a set of concept types for $\mathcal{B}$. For this, we take any $d \in \Delta^{\mathcal{I}}$, and show that $\tau(d)$ is a concept type for $\mathcal{B}$. The semantics of $\mathcal{SHOQ}^{\sqcap}$ and the definition of τ yield immediately that for every $C \sqcap D \in \mathsf{Cl_c}(\mathcal{B})$, we have $C \sqcap D \in \tau(d)$ iff $\{C, D\} \subseteq \tau(d)$. Similarly, for every $\neg C \in \mathsf{Cl_c}(\mathcal{B})$, we have $\neg C \in \tau(d)$ iff $C \notin \tau(d)$. Because of the UNA, we have also that for every $\{a\} \in \mathsf{Cl}_\mathsf{c}(\mathcal{B})$, $\{a\} \in \tau(d)$ implies that $\{b\} \notin \tau(d)$ for every $\{b\} \in \mathsf{Cl}_\mathsf{c}(\mathcal{B})$ with $\{b\} \neq \{a\}$. The semantics of $\mathcal{SHOQ}^\sqcap$ and the definition of τ yield also that for every $\exists r.\{a\} \in \mathsf{Cl}_\mathsf{c}(\mathcal{B})$, we have that if $\exists r.\{a\} \in \tau(d)$ and $\mathcal{R} \models r \sqsubseteq s$, then $\exists s.\{a\} \in \tau(d)$.

Obviously, $\mathbb{f}$ is a formula type for $\mathcal{B}$, and we have also, by the UNA, that for any $\mathbb{c}, \mathbb{d} \in \mathcal{W}$ with $\mathbb{c} \neq \mathbb{d}$ that $\mathbb{c} \cap \mathbb{d} \cap \big\{\{a\} \mid a \in \mathsf{Ind}(\Psi)\big\} = \emptyset$. By definition, $\{a\} \in \tau(a^\mathcal{I}) = \iota(a)$ for every $a \in \mathsf{Ind}(\Psi)$. Hence, $\mathcal{M}$ is a model candidate for $\mathcal{B}$. We continue showing the following claim.

Claim 3.31. *For every $d, e \in \Delta^\mathcal{I}$ and every $r \in \mathsf{Rol}(\mathcal{B})$, we have that $(d,e) \in r^\mathcal{I}$ implies $\tau(d) \overset{r}{\rightsquigarrow}_\mathcal{R} \tau(e)$.*

To prove the claim, take any $(d,e) \in r^\mathcal{I}$. For the first condition of r-compatibility (see Definition 3.21), take any $\neg(\exists r.D) \in \tau(d)$, which implies that $d \in (\neg\exists r.D)^\mathcal{I}$. By the semantics of $\mathcal{SHOQ}^\sqcap$, we have $e \in (\neg D)^\mathcal{I}$, and thus $\neg D \in \tau(e)$.

For the second condition of r-compatibility, take any $s \in \mathsf{Rol}(\mathcal{B})$ with $\mathcal{R} \models r \sqsubseteq s$, $\mathcal{R} \models \mathsf{trans}(r)$, and $\neg(\exists s.D) \in \tau(d)$. Since $\mathcal{I}$ is a model of $\mathcal{R}$, we have that $r^\mathcal{I} \subseteq s^\mathcal{I}$ and that $r^\mathcal{I}$ is transitive. Suppose that $\neg(\exists r.D) \notin \tau(e)$, and thus $\exists r.D \in \tau(e)$. Then there is an $e' \in \Delta^\mathcal{I}$ with $e' \in D^\mathcal{I}$ and $(e,e') \in r^\mathcal{I}$. Since $r^\mathcal{I}$ is transitive, we have also $(d,e') \in r^\mathcal{I}$, and thus $(d,e') \in s^\mathcal{I}$, which yields a contradiction to $\neg(\exists s.D) \in \tau(d)$. This finishes the proof of Claim 3.31.

We now use this claim to show that $\mathcal{M}$ is a quasimodel for $\mathcal{B}$ that respects $\mathcal{D}$. Condition (a) is obviously satisfied since $\Delta^\mathcal{I} \neq \emptyset$ by definition.

For Condition (b), we have for every $A(a) \in \mathsf{Cl}_\mathsf{f}(\Psi)$ that $A(a) \in \mathbb{f}$ iff $\mathcal{I} \models A(a)$ iff $a^\mathcal{I} \in A^\mathcal{I}$ iff $A \in \tau(a^\mathcal{I}) = \iota(a)$.

For Condition (c), we have for every $r(a,b) \in \mathsf{Cl}_\mathsf{f}(\Psi)$ that $r(a,b) \in \mathbb{f}$ iff $\mathcal{I} \models r(a,b)$ iff $(a^\mathcal{I}, b^\mathcal{I}) \in r^\mathcal{I}$ iff $a^\mathcal{I} \in (\exists r.\{b\})^\mathcal{I}$ iff $\exists r.\{b\} \in \tau(a^\mathcal{I}) = \iota(a)$.

For Condition (d), take any $\top \sqsubseteq C \in \mathbb{f}$ and any $\mathbb{c} \in \mathcal{W}$. The definition of $\mathbb{f}$ yields $\mathcal{I} \models \top \sqsubseteq C$, and thus $C^\mathcal{I} = \Delta^\mathcal{I}$. Hence, $C \in \tau(d)$ for any $d \in \Delta^\mathcal{I}$, which yields by the definition of $\mathcal{W}$ that $C \in \mathbb{c}$.

For Condition (e), take any $\neg(\top \sqsubseteq C) \in \boldsymbol{f}$. By the definition of $\mathbb{f}$, this implies $\mathcal{I} \not\models \top \sqsubseteq C$. Thus, there is a $d \in \Delta^\mathcal{I}$ with $d \notin C^\mathcal{I}$. Thus, we have $C \notin \tau(d) \in \mathcal{W}$.

For Condition (f), take any $d \in \Delta^\mathcal{I}$ with $\exists r.\{a\} \in \tau(d)$. Then, $d \in (\exists r.\{a\})^\mathcal{I}$, and thus $(d, a^\mathcal{I}) \in r^\mathcal{I}$. Claim 3.31 yields that $\tau(d) \overset{r}{\rightsquigarrow}_\mathcal{R} \tau(a^\mathcal{I})$, i.e. $\tau(d) \overset{r}{\rightsquigarrow}_\mathcal{R} \iota(a)$.

For Condition (g), take any $d \in \Delta^\mathcal{I}$. We construct a solution $\nu_{\tau(d)}$ of the system of equations $E_{\mathcal{M},\tau(d)}$. Let z denote the maximal integer that occurs in any number restriction in $\mathcal{B}$. We denote by $\Delta^\mathcal{I}_\mathsf{u}$ the set $\{d \in \Delta^\mathcal{I} \mid d \neq a^\mathcal{I}$ for every $a \in \mathsf{N_I}\}$ of *unnamed* domain elements, and by $\Delta^\mathcal{I}_\mathsf{n}$ the set $\Delta^\mathcal{I} \setminus \Delta^\mathcal{I}_\mathsf{u}$ of *named* domain elements. We first consider the variables $x_{\tau(d),\mathbb{r},\mathbb{d}}$. Take any $\mathbb{r} \in \mathfrak{R}(\mathcal{B})$ and any $\mathbb{d} \in \mathcal{W}_\mathsf{u}$ such that $\tau(d) \overset{\mathbb{r}}{\rightsquigarrow}_\mathcal{R} \mathbb{d}$. Then we define

$$\nu_{\tau(d)}(x_{\tau(d),\mathbb{r},\mathbb{d}}) := \min\big\{z,\ |\{e \in \Delta^\mathcal{I}_\mathsf{u} \mid \tau(e) = \mathbb{d}, \text{ and } (d,e) \in s^\mathcal{I} \text{ iff } s \in \mathbb{r}\}|\big\}.$$

We set $\nu_{\tau(d)}(x_{\tau(d),\mathbb{r},\mathbb{d}})$ to at most z to ensure that this value is finite. Note that this value counts the *unnamed* role successors of type $\mathbb{r}$ of concept type $\mathbb{d}$.

The following claim implies that the equations of the form (E1) and (E2) are satisfiable by appropriately defining $\nu_{\tau(d)}(y_{\tau(d),\geq n\, r.C})$ and $\nu_{\tau(d)}(y_{\tau(d),\neg(\geq n\, r.C)})$.

Claim 3.32. *For every $\geq n\, r.C \in \mathsf{Cl}_{\mathsf{c}}(\mathcal{B})$, we have*

$$\geq n\, r.C \in \tau(d) \textit{ iff } \sum_{r \in \mathbb{r} \in \mathfrak{R}(\mathcal{B})} (\nu_{\tau(d)}(\Xi_{\mathcal{M},\tau(d),\mathbb{r},C}) + \Gamma_{\mathcal{M},\tau(d),\mathbb{r},C}) \geq n.$$

Assume first that there are $\mathbb{d} \in \mathcal{W}_{\mathsf{u}}$ and $\mathbb{r} \in \mathfrak{R}(\mathcal{B})$ such that $C \in \mathbb{d}$, $r \in \mathbb{r}$, $\tau(d) \overset{\mathbb{r}}{\rightsquigarrow}_{\mathcal{R}} \mathbb{d}$, and $\nu_{\tau(d)}(x_{\tau(d),\mathbb{r},\mathbb{d}}) = z \geq n$. Then by definition of $\nu_{\tau(d)}$, there are at least n unnamed domain elements $e \in \Delta_{\mathsf{u}}^{\mathcal{I}}$ with $C \in \mathbb{d} = \tau(e)$ and $(d,e) \in r^{\mathcal{I}}$, which implies that $d \in (\geq n\, r.C)^{\mathcal{I}}$, and thus $\geq n\, r.C \in \tau(d)$. Additionally, $\nu_{\tau(d)}(\Xi_{\mathcal{M},\tau(d),\mathbb{r},C}) \geq z \geq n$, which shows that Claim 3.32 holds.

We assume in the following that for every $\mathbb{d} \in \mathcal{W}_{\mathsf{u}}$ and $\mathbb{r} \in \mathfrak{R}(\mathcal{B})$ with $C \in \mathbb{d}$, $r \in \mathbb{r}$, and $\tau(d) \overset{\mathbb{r}}{\rightsquigarrow}_{\mathcal{R}} \mathbb{d}$, we have $\nu_{\tau(d)}(x_{\tau(d),\mathbb{r},\mathbb{d}}) = |\{e \in \Delta_{\mathsf{u}}^{\mathcal{I}} \mid \tau(e) = \mathbb{d}, \text{ and } (d,e) \in s^{\mathcal{I}} \text{ iff } s \in \mathbb{r}\}| \leq z$. It now follows that, for each $\mathbb{r} \in \mathfrak{R}(\mathcal{B})$, we have

$$\begin{aligned}
\nu_{\tau(d)}(\Xi_{\mathcal{M},\tau(d),\mathbb{r},C}) &= \sum_{C \in \mathbb{d} \in \mathcal{W}_{\mathsf{u}},\ \tau(d) \overset{\mathbb{r}}{\rightsquigarrow}_{\mathcal{R}} \mathbb{d}} \nu_{\tau(d)}(x_{\tau(d),\mathbb{r},\mathbb{d}}) \\
&= \sum_{C \in \mathbb{d} \in \mathcal{W}_{\mathsf{u}},\ \tau(d) \overset{\mathbb{r}}{\rightsquigarrow}_{\mathcal{R}} \mathbb{d}} |\{e \in \Delta_{\mathsf{u}}^{\mathcal{I}} \mid \tau(e) = \mathbb{d}, \text{ and } (d,e) \in s^{\mathcal{I}} \text{ iff } s \in \mathbb{r}\}| \\
&= |\{e \in C^{\mathcal{I}} \cap \Delta_{\mathsf{u}}^{\mathcal{I}} \mid (d,e) \in s^{\mathcal{I}} \text{ iff } s \in \mathbb{r}\}|,
\end{aligned}$$

where the third equality follows by Claim 3.31. Thus,

$$\sum_{r \in \mathbb{r} \in \mathfrak{R}(\mathcal{B})} \nu_{\tau(d)}(\Xi_{\mathcal{M},\tau(d),\mathbb{r},C}) = |\{e \in C^{\mathcal{I}} \cap \Delta_{\mathsf{u}}^{\mathcal{I}} \mid (d,e) \in r^{\mathcal{I}}\}|.$$

Moreover, we have

$$\begin{aligned}
\sum_{r \in \mathbb{r} \in \mathfrak{R}(\mathcal{B})} \Gamma_{\mathcal{M},\tau(d),\mathbb{r},C} &= \sum_{r \in \mathbb{r} \in \mathfrak{R}(\mathcal{B})} \Big|\big\{b \in \mathsf{Ind}(\Psi) \mid C \in \iota(b), \text{ and } \exists s.\{b\} \in \tau(d) \text{ iff } s \in \mathbb{r}\big\}\Big| \\
&= \Big|\big\{b \in \mathsf{Ind}(\Psi) \mid C \in \iota(b) \text{ and } \exists r.\{b\} \in \tau(d)\big\}\Big| \\
&= |\{b \in \mathsf{Ind}(\Psi) \mid b^{\mathcal{I}} \in C^{\mathcal{I}} \text{ and } (d, b^{\mathcal{I}}) \in r^{\mathcal{I}}\}| \\
&= |\{e \in C^{\mathcal{I}} \cap \Delta_{\mathsf{n}}^{\mathcal{I}} \mid (d,e) \in r^{\mathcal{I}}\}|.
\end{aligned}$$

Since $\{\Delta_{\mathsf{u}}^{\mathcal{I}}, \Delta_{\mathsf{n}}^{\mathcal{I}}\}$ is a partition of $\Delta^{\mathcal{I}}$, we have that $\geq n\, r.C \in \tau(d)$ *iff* $d \in (\geq n\, r.C)^{\mathcal{I}}$ *iff* $|\{e \in C^{\mathcal{I}} \mid (d,e) \in r^{\mathcal{I}}\}| \geq n$ *iff*

$$\sum_{r \in \mathbb{r} \in \mathfrak{R}(\mathcal{B})} (\nu_{\mathbb{c}}(\Xi_{\mathcal{M},\mathbb{c},\mathbb{r},C}) + \Gamma_{\mathcal{M},\mathbb{c},\mathbb{r},C}) \geq n$$

by the above equations, which finishes the proof of Claim 3.32.

Consider now any $E = \exists(r_1 \sqcap \dots \sqcap r_\ell).C \in \mathsf{Cl}_{\mathsf{c}}(\mathcal{B})$. As above, if there are $\mathbb{d} \in \mathcal{W}_{\mathsf{u}}$ and $\mathbb{r} \in \mathfrak{R}(\mathcal{B})$ such that $C \in \mathbb{d}$, $\{r_1, \dots, r_\ell\} \subseteq \mathbb{r}$, $\tau(d) \overset{\mathbb{r}}{\rightsquigarrow}_{\mathcal{R}} \mathbb{d}$, and $\nu_{\tau(d)}(x_{\tau(d),\mathbb{r},\mathbb{d}}) = z \geq 1$, then there is at least one unnamed domain element $e \in \Delta_{\mathsf{u}}^{\mathcal{I}}$ with $C \in \mathbb{d} = \tau(e)$ and $(d,e) \in r_1^{\mathcal{I}} \cap r_\ell^{\mathcal{I}}$. This implies that $d \in E^{\mathcal{I}}$, and thus $E \in \tau(d)$. Also, $\nu_{\tau(d)}(\Xi_{\mathcal{M},\tau(d),\mathbb{r},C}) \geq z \geq 1$, which shows that the corresponding equation of the form (E3) is satisfied if $\nu_{\tau(d)}(y_{\tau(d),E})$ is set appropriately.

We consider now the case where for every $\mathbb{d} \in \mathcal{W}_{\mathsf{u}}$ and every $\mathbb{r} \in \mathfrak{R}(\mathcal{B})$ with $C \in \mathbb{d}$, $\{r_1, \ldots, r_\ell\} \subseteq \mathbb{r}$, and $\tau(d) \overset{\mathbb{r}}{\rightsquigarrow}_{\mathcal{R}} \mathbb{d}$, we have $\nu_{\tau(d)}(x_{\tau(d),\mathbb{r},\mathbb{d}}) \leq \mathbb{z}$. Then, by similar arguments as above, we have:

$$\sum_{\{r_1,\ldots,r_\ell\}\subseteq\mathbb{r}\in\mathfrak{R}(\mathcal{B})} \nu_{\tau(d)}(\Xi_{\mathcal{M},\tau(d),\mathbb{r},C}) = |\{e \in C^{\mathcal{I}} \cap \Delta^{\mathcal{I}}_{\mathsf{u}} \mid (d,e) \in r_1^{\mathcal{I}} \cap \cdots \cap r_\ell^{\mathcal{I}}\}|,$$

and also

$$\sum_{\{r_1,\ldots,r_\ell\}\subseteq\mathbb{r}\in\mathfrak{R}(\mathcal{B})} \Gamma_{\mathcal{M},\tau(d),\mathbb{r},C} = |\{e \in C^{\mathcal{I}} \cap \Delta^{\mathcal{I}}_{\mathsf{n}} \mid (d,e) \in r_1^{\mathcal{I}} \cap \cdots \cap r_\ell^{\mathcal{I}}\}|.$$

Again, this yields that we have $E \in \tau(d)$ *iff* $d \in E^{\mathcal{I}}$ *iff* there is at least one $e \in C^{\mathcal{I}}$ with $(d,e) \in r_1^{\mathcal{I}} \cap \cdots \cap r_\ell^{\mathcal{I}}$ *iff*

$$\sum_{\{r_1,\ldots,r_\ell\}\subseteq\mathbb{r}\in\mathfrak{R}(\mathcal{B})} (\nu_{\tau(d)}(\Xi_{\mathcal{M},\tau(d),\mathbb{r},C}) + \Gamma_{\mathcal{M},\tau(d),\mathbb{r},C}) \geq 1,$$

which shows that the equations of the form (E3) and (E4) can be satisfied by appropriately setting $\nu_{\tau(d)}(y_{\tau(d),E})$. This finishes the proof that $\mathcal{M}$ satisfies Condition (g).

For Condition (h), take any $d \in \Delta^{\mathcal{I}}$. Since $\mathcal{I}$ respects $\mathcal{D}$, there must be a set $Y \in \mathcal{Y}$ such that $d \in (C_{\mathcal{U},Y})^{\mathcal{I}}$. Hence, by definition of $\tau(d)$, we have $Y = \tau(d) \cap \mathcal{U}$.

For Condition (i), let $Y \in \mathcal{Y}$. Since $\mathcal{I}$ respects $\mathcal{D}$, there must be a $d \in (C_{\mathcal{U},Y})^{\mathcal{I}}$. Thus, by definition of $\tau(d)$, we have $Y = \tau(d) \cap \mathcal{U}$ with $\tau(d) \in \mathcal{W}$. □

It remains to be shown that one can check the existence of a quasimodel for $\mathcal{B}$ that respects $\mathcal{D}$ in time exponential in the size of $\mathcal{B}$. For this, consider the following algorithm. Given $\mathcal{B} = (\Psi, \mathcal{R})$ and $\mathcal{D}$, it enumerates all model candidates $(\mathcal{W}, \iota, \mathbb{f})$ for $\mathcal{B}$, where $\mathcal{W}$ is the set of *all* concept types for $\mathcal{B}$. We denote these candidates by $\mathcal{M}_1, \ldots, \mathcal{M}_N$. Note that each of them is of size exponential in the size of $\mathcal{B}$. It should be clear that

$$N \leq 2^{|\mathsf{Cl}_{\mathsf{c}}(\mathcal{B})|\cdot|\mathsf{Ind}(\Psi)|} \cdot 2^{|\mathsf{Cl}_{\mathsf{f}}(\Psi)|},$$

and thus the enumeration of $\mathcal{M}_1, \ldots, \mathcal{M}_N$ can be done in exponential time since $\mathsf{Cl}_{\mathsf{c}}(\mathcal{B})$ and $\mathsf{Cl}_{\mathsf{f}}(\Psi)$ are of size polynomial in the size of $\mathcal{B}$.

The algorithm works as follows. First, initialise $i := 1$ and consider $\mathcal{M}_i = (\mathcal{W}, \iota, \mathbb{f})$.

Step 1. Check whether $\mathcal{M}_i$ satisfies Conditions (b) and (c).
If it does, continue with Step 2. Otherwise, stop considering $\mathcal{M}_i$ and go to Step 5.

Step 2. Check each concept type $\mathbb{c} \in \mathcal{W}$. We call a concept type $\mathbb{c} \in \mathcal{W}$ *defective* if it violates Condition (d) for some $\top \sqsubseteq C \in \mathbb{f}$, it violates Condition (f), or it violates Condition (h).
If we find a defective concept type $\mathbb{c}$, and have $\mathbb{c} \in \mathcal{W}_{\mathsf{u}}$, then we set $\mathcal{W} := \mathcal{W} \setminus \{\mathbb{c}\}$ and continue with Step 2. If we find a defective $\mathbb{c} \notin \mathcal{W}_{\mathsf{u}}$, i.e. $\mathbb{c} \in \iota(\mathsf{Ind}(\Psi))$, then stop considering $\mathcal{M}_i$ and go to Step 5. If we have found no defective concept types in $\mathcal{W}$, continue with Step 3.

Step 3. Consider the model candidate $\mathcal{M}' = (\mathcal{W}', \iota, \mathbb{f})$ obtained from the previous step. For every $\mathbb{c} \in \mathcal{W}'$, check whether $E_{\mathcal{M}',\mathbb{c}}$ has a solution.

If we find a $\mathbb{c} \in \mathcal{W}'_\mathsf{u}$ such that $E_{\mathcal{M}',\mathbb{c}}$ has no solution, then set $\mathcal{W}' := \mathcal{W}' \setminus \{\mathbb{c}\}$ and redo Step 3. If we find a $\mathbb{c} \in \iota(\mathsf{Ind}(\Psi))$ such that $E_{\mathcal{M}',\mathbb{c}}$ has no solution, then go to Step 5. If we have found no such concept type in $\mathcal{W}'$, continue with Step 4.

Step 4. Check whether the model candidate $\mathcal{M}'' = (\mathcal{W}'', \iota, \mathbb{f})$ obtained from Step 3 satisfies Conditions (a), (e), and (i).

If it does, stop with output 'quasimodel that respects $\mathcal{D}$ found'. Otherwise, continue with Step 5.

Step 5. Set $i := i + 1$. If $i \leq N$, continue with Step 1. Otherwise, stop with output 'no quasimodel that respects $\mathcal{D}$ exists'.

Using this algorithm, we are now ready to prove one of the main results of this section.

Theorem 3.33. *Let $\mathcal{B}$ be a Boolean $\mathcal{SHOQ}^{\sqcap}$-knowledge base, and let $\mathcal{D} = (\mathcal{U}, \mathcal{Y})$ be a pair such that $\mathcal{U}$ is a set of concept names occurring in $\mathcal{B}$ and $\mathcal{Y} \subseteq 2^{\mathcal{U}}$. Then, consistency of $\mathcal{B}$ w.r.t. $\mathcal{D}$ can be decided in time exponential in the size of $\mathcal{B}$.*

Proof. By Lemma 3.26, it suffices to show that the algorithm described above to find quasimodels for $\mathcal{B}$ that respect $\mathcal{D}$ is sound, complete, and terminates in time exponential in the size of $\mathcal{B}$.

If the algorithm has constructed a model candidate $\mathcal{M} = (\mathcal{W}, \iota, \mathbb{f})$ that passed all tests, then $\mathcal{M}$ obviously satisfies Conditions (a)–(i) of Definition 3.25.

Conversely, if $\mathcal{M} = (\mathcal{W}, \iota, \mathbb{f})$ is a quasimodel for $\mathcal{B}$ that respects $\mathcal{D}$, then ι and $\mathbb{f}$ must be enumerated by the algorithm at some point. Since ι and $\mathbb{f}$ satisfy Conditions (b) and (c), they pass the tests in Step 1, and we continue with Step 2. There, a model candidate $\mathcal{M}' = (\mathcal{W}', \iota, \mathbb{f})$ is constructed. Note that the concept types in $\mathcal{W}$ cannot be defective because of Conditions (d), (f), and (h). Since ι maps to concept types in $\mathcal{W}$, we indeed obtain $\mathcal{M}'$ with $\mathcal{W} \subseteq \mathcal{W}'$, and continue with Step 3. There, a model candidate $\mathcal{M}'' = (\mathcal{W}'', \iota, \mathbb{f})$ is constructed. Since for every $\mathbb{c} \in \mathcal{W}$, $E_{\mathcal{M},\mathbb{c}}$ has a solution, also $E_{\mathcal{M}'',\mathbb{c}}$ has a solution. Indeed, the additional variables that occur in $E_{\mathcal{M}'',\mathbb{c}}$ but not in $E_{\mathcal{M},\mathbb{c}}$ can be set to 0. Since ι maps to the concept types in $\mathcal{W}$, we indeed obtain a model candidate $\mathcal{M}''$ with $\mathcal{W} \subseteq \mathcal{W}''$ that satisfies Condition (g), and continue with Step 4. Finally, $\mathcal{W}''$ satisfies the Conditions (a), (e), and (i), because $\mathcal{W} \subseteq \mathcal{W}''$. This shows that the algorithm detects the existence of a quasimodel for $\mathcal{B}$ that respects $\mathcal{D}$.

To analyse the running time of the algorithm, observe first that r-compatibility w.r.t. $\mathcal{R}$ can be checked in polynomial time since this only involves inclusion tests for two concept types, which are sets of polynomial size, and entailment tests of role axioms w.r.t. $\mathcal{R}$, which can be done in time polynomial in the size of $\mathcal{R}$ [HST00].

As mentioned above, the number N of model candidates is at most exponential in the size of $\mathcal{B}$, while each model candidate $\mathcal{M}_i$ is of size exponential in the size of $\mathcal{B}$. Also the sequence of model candidates $\mathcal{M}_1, \ldots, \mathcal{M}_N$ can be enumerated in time exponential in the size of $\mathcal{B}$.

For each of these exponentially many model candidates, the checks in Step 1 can be done in time polynomial in the size of $\mathcal{B}$, and the checks in Step 2 are done at most exponentially often since each time one of the exponentially many concept types in $\mathcal{W}$ is removed. Each of these checks can be done in time exponential in the size of $\mathcal{B}$ since the following conditions are checked for at most exponentially many concept types $\mathbb{c}$:

- for Condition (d), we check for inclusion of polynomially many concepts in $\mathbb{c}$;
- for Condition (f), we have polynomially many r-compatibility tests; and
- for Condition (h), we enumerate all (at most exponentially many) elements of $\mathcal{Y}$ and do a simple test.

By similar arguments as above, the checks in Step 3 are done at most exponentially often since each time one of the exponentially many concept types in $\mathcal{W}'$ is removed. Each time Step 3 is performed, for exponentially many concept types $\mathbb{c} \in \mathcal{W}'$, it must be checked whether $E_{\mathcal{M}',\mathbb{c}}$ has a solution. We denote the number of variables in $E_{\mathcal{M}',\mathbb{c}}$ by n, and the number of equations in $E_{\mathcal{M}',\mathbb{c}}$ by m. Note that n may be exponential in the size of $\mathcal{B}$ since there are exponentially many concept types and role types. However, m is polynomial in the size of $\mathcal{B}$ since we have one equation per at-least and existential restriction occurring in Ψ. In [Pap81], it was shown that $E_{\mathcal{M}',\mathbb{c}}$ can be solved in time $O(n^{2m+2}(ma)^{(m+1)(2m+1)})$, where a denotes the value of the largest number appearing in the equations. Thus, even if the numbers in the at-least restrictions occurring in Ψ are given in binary encoding, checking whether the system of equations $E_{\mathcal{M}',\mathbb{c}}$ has a solution can be done in time exponential in the size of $\mathcal{B}$. Overall, Step 3 takes only time exponential in the size of $\mathcal{B}$.

Finally, Step 4 can also be done in time exponential in the size of $\mathcal{B}$. Checking Condition (a) is trivial. Checking Condition (e) involves enumerating polynomially many $\neg(\top \sqsubseteq C) \in \mathbb{f}$ and at most exponentially many concept types in $\mathcal{W}''$ to do an inclusion test. For Condition (i), we enumerate at most exponentially many elements of $\mathcal{Y}$ and at most exponentially many concept types in $\mathcal{W}''$ to do a simple test.

Overall, the algorithm runs in time exponential in the size of $\mathcal{B}$. □

The following corollary captures the special case of Boolean $\mathcal{SHOQ}$-knowledge bases without a set $\mathcal{D}$.

Corollary 3.34. *Let $\mathcal{B}$ be a Boolean $\mathcal{SHOQ}$-knowledge base. Then, consistency of $\mathcal{B}$ can be decided in time exponential in the size of $\mathcal{B}$.*

Proof. Obviously, $\mathcal{B}$ is also a Boolean $\mathcal{SHOQ}^\sqcap$-knowledge base as $\mathcal{SHOQ}$ is a fragment of $\mathcal{SHOQ}^\sqcap$. We define $\mathcal{D} := (\mathcal{U}, \mathcal{Y})$ with $\mathcal{U} := \emptyset$ and $\mathcal{Y} := \{\emptyset\}$. It is easy to see that $\mathcal{B}$ is consistent iff $\mathcal{B}$ is consistent w.r.t. $\mathcal{D}$. Indeed, the ‘if’ direction is trivial. For the ‘only if’ direction, assume that $\mathcal{B}$ is consistent. Then, there is a model $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ of $\mathcal{B}$. Note that $C_{\emptyset,\emptyset}$ is equivalent to $\top$, and thus $(C_{\emptyset,\emptyset})^\mathcal{I} = \Delta^\mathcal{I} \neq \emptyset$. Hence, we have

$$\mathcal{Y} = \{\emptyset\} = \left\{Y \subseteq \mathcal{U} = \emptyset \mid \text{there is a } d \in \Delta^\mathcal{I} \text{ with } d \in (C_{\mathcal{U},Y})^\mathcal{I}\right\},$$

which shows that $\mathcal{I}$ respects $\mathcal{D}$. Thus, Theorem 3.33 yields that consistency of $\mathcal{B}$ can be decided in time exponential in the size of $\mathcal{B}$. □

Theorem 3.33 and Corollary 3.34 yield the results that are needed in the proofs of Theorems 3.15, 3.17, and 3.20.

3.3 Summary

In this chapter, we obtained complexity results for the satisfiability problem in the temporalised description logic $\mathcal{SHOQ}$-LTL as shown in Table 3.4. More precisely, we considered the satisfiability problem in the settings where (i) neither concept names nor role names are allowed to be rigid, (ii) only concept names may be rigid, and (iii) both concept names and role names may be rigid. It turned out that in all three settings, the satisfiability problem in $\mathcal{SHOQ}$-LTL has the same complexity as the satisfiability problem in the less expressive temporalised description logic $\mathcal{ALC}$-LTL. Hence, for every description logic $\mathcal{L}$ between $\mathcal{ALC}$ and $\mathcal{SHOQ}$, we have that the satisfiability problem in $\mathcal{L}$-LTL is ExpTime-complete in Setting (i), which is the same complexity as the satisfiability problem in $\mathcal{L}$. Moreover, the satisfiability problem in $\mathcal{L}$-LTL is NExpTime-complete if we allow rigid concept names (but no rigid role names), i.e. in Setting (ii), and 2ExpTime-complete in Setting (iii), where we further allow rigid role names.

Chapter 4

Runtime Verification Using the Temporalised Description Logic $\mathcal{SHOQ}$-LTL

Runtime verification deals with the problem of verifying properties about the behaviour of observed systems. In this chapter, we investigate runtime verification using the temporalised description logic $\mathcal{SHOQ}$-LTL. We show how monitors for $\mathcal{SHOQ}$-LTL-formulas can be constructed and establish complexity results for related decision problems. Some of the results of this chapter have already been published in [BL14].

This chapter is organised as follows. In Section 4.1, we first consider propositional runtime verification, and revisit known results to be able to compare them to one we establish. Then, in Section 4.2 we show how to construct Büchi-automata for $\mathcal{SHOQ}$-LTL-formulas. This section is very related to Chapter 3 where we established complexity results for the satisfiability problem in $\mathcal{SHOQ}$-LTL. After that, we consider the actual monitor construction for $\mathcal{SHOQ}$-LTL in Section 4.3. Then, in Section 4.4, we show some complexity results about the important related decision problems 'liveness' and 'monitorability'. Finally, Section 4.5 gives a brief summary of the main results of this chapter.

4.1 Runtime Verification Using Propositional LTL

In propositional runtime verification [BLS10; BLS11], one observes the actual behaviour of the given system since it started, which at any point in time can be described by a finite word u over $\Sigma_{\mathcal{P}}$. Here $\mathcal{P}$ is a finite set of propositional variables whose truth values at any point in time can be determined by observing the system. Given such a word $u = u_0 u_1 \ldots u_t \in \Sigma_{\mathcal{P}}^*$, we say that the propositional LTL-structure $\mathfrak{W} = (w_i)_{i \geq 0}$ *extends* u if $u_i = w_i$ for every i, $0 \leq i \leq t$. In this case, we also call $\mathfrak{W}$ an *extension of* u. In principle, a monitor for a propositional LTL-formula ϕ needs to realise the following *monitoring function* $\mathsf{m}_\phi : \Sigma_{\mathcal{P}}^* \to \{\top, \bot, ?\}$:

$$\mathsf{m}_\phi(u) := \begin{cases} \top & \text{if } \mathfrak{W}, 0 \models \phi \text{ for every propositional LTL-structure } \mathfrak{W} \text{ that extends } u; \\ \bot & \text{if } \mathfrak{W}, 0 \models \neg\phi \text{ for every propositional LTL-structure } \mathfrak{W} \text{ that extends } u; \text{ and} \\ ? & \text{otherwise.} \end{cases}$$

As mentioned above, this function should not be computed from scratch whenever a new observation $\sigma \in \Sigma_{\mathcal{P}}$ is added. In particular, the time needed for computing the next function value $\mathsf{m}_\phi(u\sigma)$ should not depend on the length of the already observed word u. This can be achieved by constructing a deterministic Moore-automaton as monitor.

Definition 4.1 (Deterministic Moore-automaton). *A* deterministic Moore-automaton *is a tuple $\mathcal{M} = (S, \Sigma, \delta, s_0, \Gamma, \lambda)$ consisting of a finite set of states S, a finite input alphabet Σ, a transition function $\delta\colon S \times \Sigma \to S$, an initial state $s_0 \in S$, a finite output alphabet Γ, and an output function $\lambda\colon S \to \Gamma$.*

The transition function and the output function can be extended to functions $\delta^*\colon S \times \Sigma^* \to S$ and $\lambda^*\colon \Sigma^* \to \Gamma$ as follows:

- *$\delta^*(s, \varepsilon) := s$ where ε denotes the empty word;*
- *$\delta^*(s, u\sigma) := \delta(\delta^*(s, u), \sigma)$ where $u \in \Sigma^*$ and $\sigma \in \Sigma$; and*

for every $u \in \Sigma^*$, $\lambda^*(u) := \lambda(\delta^*(s_0, u))$. ◇

Such a deterministic Moore-automaton is a monitor for ϕ if its extended output function is the monitoring function for ϕ.

Definition 4.2 (Monitor for propositional LTL-formula). *Let $\mathcal{P}$ be a finite set of propositional variables, and let ϕ be a propositional LTL-formula over $\mathcal{P}$. The deterministic Moore-automaton $\mathcal{M} = (S, \Sigma_{\mathcal{P}}, \delta, s_0, \{\top, \bot, ?\}, \lambda)$ is a* monitor for ϕ *if $\lambda^*(u) = \mathsf{m}_\phi(u)$ holds for every $u \in \Sigma_{\mathcal{P}}^*$.* ◇

Given a propositional LTL-formula ϕ over $\mathcal{P}$, a monitor for ϕ can effectively be computed [BLS11]. Basically, on proceeds as follows. First, one computes Büchi-automata for ϕ and $\neg\phi$. These automata are then determinised (viewed as finite automata rather than Büchi-automata). Finally, one builds the product of the two deterministic automata. The output function is computed using reachability tests in the Büchi-automata (see [BLS11] and the monitor construction for $\mathcal{SHOQ}$-LTL in Section 4.3 below for details). The monitor obtained this way is in the worst case of doubly exponential size and be be computed in doubly exponential time.

One can actually show that this doubly exponential blow-up in the construction of the monitor cannot be avoided. Such a doubly exponential lower bound was already claimed in [BLS11], referring to a result of Kupferman and Vardi [KV01]. However, a closer look at Theorem 3.3 in [KV01] shows that it only yields a lower bound of $2^{2^{\sqrt{n}}}$. Fortunately, a more recent result by Kupferman and Rosenberg [KR10] can be used to show a lower bound of 2^{2^n}. We include a proof of this tight lower bound for the monitor construction here for the sake of completeness. This lower bound can also be used to show optimality of our monitor constructions for $\mathcal{SHOQ}$-LTL.

Kupferman and Rosenberg show (see Theorem 3 in [KR10]) that there exists a sequence $(L_n)_{n\geq 1}$ of ω-languages and a sequence $(\phi_n)_{n\geq 1}$ of propositional LTL-formulas such that the following holds for every $n \geq 1$:

1. the ω-language L_n can be accepted by a deterministic Büchi-automaton, but the number of states of any deterministic Büchi-automaton accepting L_n is at least 2^{2^n}; and
2. $L_n = L_\omega(\phi_n)$ and the size of ϕ_n is linear in n.

Using an argument similar to the one employed in [KR10], we can show that the number of states of any monitor for ϕ_n is at least 2^{2^n}. For this purpose, we first recall the definition of the languages L_n from [KR10].

For every $n \geq 0$, we consider the alphabet $\Sigma_n := \{a_1, \ldots, a_n\} \cup \{b_1, \ldots, b_n\} \cup \{\#, \$\}$, and define

$$
\begin{aligned}
T_n &:= \{a_1, b_1\} \cdot \ldots \cdot \{a_n, b_n\},\\
S_n &:= \{\#\} \cdot (T_n \cdot \{\#\})^* \cdot \{\$\} \cdot T_n \cdot \{\#\}^\omega,\\
R_n &:= \bigcup_{w \in T_n} \Sigma_n^* \cdot \{\#\} \cdot \{w\} \cdot \{\#\} \cdot \Sigma_n^* \cdot \{\$\} \cdot \Sigma_n^* \cdot \{w\} \cdot \{\#\}^\omega,\\
L_n &:= S_n \cap R_n.
\end{aligned}
$$

Thus, the language T_n consist of the words of length n such that the letter at position i is a_i or b_i. Obviously, there are 2^n such words. The ω-language S_n consists of ω-words that start with a finite sequence of elements of T_n, which are separated by the #-symbol. This sequence is terminated by the \$-symbol, which is followed by exactly one element of T_n. Then comes an infinite sequence of #-symbols. Intersecting the ω-language S_n with the ω-language R_n has the following effect: it ensures that the element w of T_n that follows the \$-symbol has already occurred in the sequence of elements of T_n before the \$-symbol.

The propositional LTL-formulas ϕ_n representing the ω-languages L_n are built over sets of propositional variables with $2n+2$ elements, i.e. over $\mathcal{P}_n := \{p_1, \ldots, p_{2n+2}\}$. Recall that such a propositional LTL-formula defines a ω-language over the alphabet $\Sigma_{\mathcal{P}_n}$, whose letters are the subsets of $\mathcal{P}_n$. Of these exponentially many letters, the language L_n uses only the (linearly many) singleton sets, where

- $\{p_i\}$ represents the letter a_i for i, $1 \leq i \leq n$;
- $\{p_{n+i}\}$ represents the letter b_i for i, $1 \leq i \leq n$;
- $\{p_{2n+1}\}$ represents the letter #; and
- $\{p_{2n+2}\}$ represents the letter \$.

In order to increase readability, we continue to use the letters from Σ_n rather than these singleton sets in our argument below.

Theorem 4.3. *There is a sequence $(\phi_n)_{n\geq 1}$ of propositional LTL-formulas of size linear in n such that the number of states of any monitor for ϕ_n is at least 2^{2^n}.*

Proof. Let $(\phi_n)_{n\geq 1}$ be the sequence of propositional LTL-formulas constructed in the proof of Theorem 3 of [KR10]. It is shown in that proof that $L_n = L_\omega(\phi_n)$ and that the size of ϕ_n is linear in n.

Now assume that $\mathcal{M}_n = (S_n, \Sigma_n, \delta_n, s_{0,n}, \{\top, \bot, ?\}, \lambda_n)$ is a monitor for ϕ_n with less than 2^{2^n} states. Given a set $T \subseteq T_n$, we enumerate its elements in lexicographic order (where a_i comes before b_i). Assume that $w_1, \ldots, w_m$ is the enumeration of the elements of T in this order. Then we define

$$\mathsf{w}(T) := \#w_1\# \ldots \#w_m\#.$$

Moreover, let $\mathsf{s}(T)$ be the state reached in $\mathcal{M}_n$ with input $\mathsf{w}(T)$ when starting at the initial state $s_{0,n}$. Since there are 2^{2^n} different subsets of T_n, but less than 2^{2^n} states, there must be two different such subsets T, T' such that $\mathsf{s}(T) = \mathsf{s}(T')$. We assume without loss of generality that there is a word $w \in T \setminus T'$.[1] Now consider the state s reached from $s_{0,n}$ on input $\mathsf{w}(T)w\#$. Since $\mathsf{s}(T) = \mathsf{s}(T')$, this is the same state as the one reached from $s_{0,n}$ on input $\mathsf{w}(T')w\#$. Since $\mathcal{M}_n$ is a monitor for ϕ_n, this implies that

$$\mathsf{m}_{\phi_n}(\mathsf{w}(T)w\#) = \lambda_n(s) = \mathsf{m}_{\phi_n}(\mathsf{w}(T')w\#).$$

This, however, yields a contradiction since actually we have

$$\mathsf{m}_{\phi_n}(\mathsf{w}(T)w\#) = ? \neq \bot = \mathsf{m}_{\phi_n}(\mathsf{w}(T')w\#).$$

In fact, we can extend $\mathsf{w}(T)w\#$ to an ω-word belonging to L_n (and thus satisfying ϕ_n) by adding an infinite sequence of #-symbols. Any other extension of $\mathsf{w}(T)w\#$ does not belong to L_n. This shows that $\mathsf{m}_{\phi_n}(\mathsf{w}(T)w\#) = ?$. The word $\mathsf{w}(T')w\#$, however, cannot be extended to an element of L_n since w does not occur in the sequence before the \$-symbol. This shows that $\mathsf{m}_{\phi_n}(\mathsf{w}(T')w\#) = \bot$.

Summing up, we have seen that our assumption that there is a monitor for ϕ_n with less than 2^{2^n} states leads to a contradiction, which shows that any monitor for ϕ_n must have at least 2^{2^n} states. □

Before building a monitor for a propositional LTL-formula ϕ, it makes sense to check whether the monitor will actually be able to give reasonable answers. For example, a monitor that always, i.e. for every finite word, returns the answer ? is clearly useless. Similarly, when running the monitor, it makes sense to check whether, according to what has been seen of the system's behaviour until now, i.e. the finite word read by the monitor until now, it makes sense to continue running the monitor. This leads to the following definition of monitorability [PZ06; FFM09; Bau10].

Definition 4.4 (Monitorability). *Let ϕ be a propositional LTL-formula over $\mathcal{P}$, and let u be a finite word over $\Sigma_{\mathcal{P}}$. We say that ϕ is* u-monitorable *if there is a finite word $v \in \Sigma^*$ such that $\mathsf{m}_\phi(uv) \neq ?$. Moreover, we call ϕ* monitorable *if it is u-monitorable for every finite word $u \in \Sigma_{\mathcal{P}}^*$.* ◊

Given a monitor $\mathcal{M}$ for ϕ, one can easily decide monitorability through reachability tests in $\mathcal{M}$. We call a state in $\mathcal{M}$ *good* if one can reach from it a state whose output is different to ?. Then

- ϕ is u-monitorable iff the state reached from the initial state with input u is good; and
- ϕ is monitorable iff every state reachable from the initial state is good.

This shows that monitorability can be decided in time doubly exponential in the size of the input formula. More precisely, one can obtain an upper bound of ExpSpace by constructing the relevant parts of the monitor on-the-fly while performing the reachability test.[2] To the best of our knowledge, it is an open problem whether this upper bound of ExpSpace is tight. The only known lower bound is one of PSpace, which can be obtained using a reduction from

[1] The case where $T' \setminus T$ is non-empty can be treated symmetrically.

[2] This is the same idea underlying the automata-based PSpace satisfiability check for propositional LTL [SC85; LPZ85] mentioned in Section 2.2.2.

the satisfiability problem.[3] Interestingly, the same is true for the related but simpler-looking problem of liveness [AS85].

Definition 4.5 (Liveness). *Let ϕ be a propositional LTL-formula over $\mathcal{P}$. We say that ϕ* expresses a liveness property *if every finite word $u \in \Sigma_{\mathcal{P}}^*$ has an extension to an ω-word that satisfies ϕ.* ♢

Using the monitoring function, liveness of ϕ can thus be expressed as follows: ϕ expresses a liveness property iff $\mathsf{m}_\phi(u) \neq \bot$ for every $u \in \Sigma_{\mathcal{P}}^*$. Consequently, given a monitor for ϕ, liveness of ϕ can again be tested by checking reachability in the monitor, which yields an upper bound of ExpSpace. Again, it is open whether this upper bound is tight. The only known lower bound is again one of PSpace, which can again be obtained using a reduction from satisfiability.[4]

In the next sections, we show how to extend the notions of this section to obtain monitors for $\mathcal{SHOQ}$-LTL-formulas.

4.2 Büchi-Automata for $\mathcal{SHOQ}$-LTL-Formulas

The decision procedures in Section 3.2 for the satisfiability problem in $\mathcal{SHOQ}$-LTL are not based on Büchi-automata. We show in this section, however, that the ideas underlying these decision procedures can be used to obtain automata-based decision procedures. The Büchi-automata constructed in this section will be the building blocks of our monitors.

In the following, let $\mathcal{R}$ be an RBox, and let ϕ be a $\mathcal{SHOQ}$-LTL-formula w.r.t. $\mathcal{R}$. In principle, we want to construct a Büchi-automaton $\mathcal{N}_{\phi,\mathcal{R}}$ that accepts exactly the models of ϕ w.r.t. $\mathcal{R}$. This is very similar to what is done for propositional LTL; see Definition 2.20. Since there are infinitely many interpretations, we would end up with an infinite alphabet for this Büchi-automaton. For this reason, we abstract from specific interpretations and consider only the axioms in ϕ that they satisfy.

For a given interpretation $\mathcal{I}$, we denote by $\tau_\phi(\mathcal{I})$ the set of all axioms in $\mathsf{Ax}(\phi)$ that $\mathcal{I}$ is a model of, i.e.

$$\tau_\phi(\mathcal{I}) := \{\alpha \in \mathsf{Ax}(\phi) \mid \mathcal{I} \models \alpha\}.$$

Note that if $\mathcal{I} \models \mathcal{R}$, we have that $\mathcal{I}$ is a model of the Boolean knowledge base

$$\Big(\bigwedge_{\alpha \in \tau_\phi(\mathcal{I})} \alpha \wedge \bigwedge_{\alpha \in \mathsf{Ax}(\phi) \setminus \tau_\phi(\mathcal{I})} \neg\alpha, \quad \mathcal{R} \Big).$$

This motivates the following definition.

Definition 4.6 (Axiom type). *The set of axioms T is an* axiom type for ϕ w.r.t. $\mathcal{R}$ *if the following two properties are satisfied:*

- *$T \subseteq \mathsf{Ax}(\phi)$; and*

[3] Note that the proof of an upper bound of PSpace given in [Bau10] actually does not go through.

[4] Note that the proof of an upper bound of PSpace sketched in [UW01] actually does not go through.

- *the Boolean knowledge base*

$$\mathcal{B}_T := \left(\bigwedge_{\alpha \in T} \alpha \wedge \bigwedge_{\alpha \in \mathsf{Ax}(\phi) \setminus T} \neg\alpha, \quad \mathcal{R} \right)$$

is consistent. ◇

We denote the set of all axiom types for ϕ w.r.t. $\mathcal{R}$ with $\mathfrak{T}_{\phi,\mathcal{R}}$. The following lemma is an easy consequence of this definition.

Lemma 4.7. *Let $\mathcal{I}$ be an interpretation, and let T be a set of axioms.*

1. *If $\mathcal{I}$ is a model of $\mathcal{R}$, then $\tau_\phi(\mathcal{I})$ is an axiom type for ϕ w.r.t. $\mathcal{R}$.*
2. *If T is any axiom type for ϕ w.r.t. $\mathcal{R}$, then there is a model $\mathcal{J}$ of $\mathcal{R}$ such that $T = \tau_\phi(\mathcal{J})$.*

Proof. For Part 1 of the lemma, assume that $\mathcal{I}$ is a model of $\mathcal{R}$. We have $\tau_\phi(\mathcal{I}) \subseteq \mathsf{Ax}(\phi)$ by definition, and as argued above that $\mathcal{I} \models \mathcal{B}_{\tau_\phi(\mathcal{I})}$.

For Part 2 of the lemma, assume that T is an axiom type for ϕ w.r.t. $\mathcal{R}$. Hence, $\mathcal{B}_T$ is consistent. Let $\mathcal{J}$ be a model of $\mathcal{B}_T$. Note that we have $\mathcal{J} \models \mathcal{R}$. Furthermore, it is easy to see that, by construction of $\mathcal{B}_T$, we have that $T = \tau_\phi(\mathcal{J})$. □

In the following, for $\mathcal{I} \models \mathcal{R}$, we call $\tau_\phi(\mathcal{I})$ the *axiom type of* $\mathcal{I}$. This notion can be further extended to DL-LTL-structures. For a given DL-LTL-structure $\mathfrak{I} = (\mathcal{I}_i)_{i \geq 0}$, we define

$$\tau_\phi(\mathfrak{I}) := \tau_\phi(\mathcal{I}_0)\tau_\phi(\mathcal{I}_1)\tau_\phi(\mathcal{I}_2)\ldots,$$

and, for $\mathfrak{I} \models \mathcal{R}$, we call $\tau_\phi(\mathfrak{I})$ the *axiom type of* $\mathfrak{I}$. Note that the axiom type of $\mathfrak{I}$ is an ω-word over the alphabet $\mathfrak{T}_{\phi,\mathcal{R}}$.

Whether a given DL-LTL-structure $\mathfrak{I}$ with $\mathfrak{I} \models \mathcal{R}$ is a model of ϕ w.r.t. $\mathcal{R}$ only depends on its axiom type. This is stated formally in the following lemma.

Lemma 4.8. *Let $\mathfrak{I}$ and $\mathfrak{J}$ be DL-LTL-structures such that $\mathfrak{I} \models \mathcal{R}$, $\mathfrak{J} \models \mathcal{R}$, and $\tau_\phi(\mathfrak{I}) = \tau_\phi(\mathfrak{J})$. Then, $\mathfrak{I}$ is a model of ϕ w.r.t. $\mathcal{R}$ iff $\mathfrak{J}$ is a model of ϕ w.r.t. $\mathcal{R}$.*

Proof. Let $\mathfrak{I} = (\mathcal{I}_i)_{i \geq 0}$ and $\mathfrak{J} = (\mathcal{J}_i)_{i \geq 0}$ be DL-LTL-structures such that $\mathfrak{I} \models \mathcal{R}$, $\mathfrak{J} \models \mathcal{R}$, and $\tau_\phi(\mathfrak{I}) = \tau_\phi(\mathfrak{J})$. It is enough to prove that $\mathfrak{I}, i \models \phi$ iff $\mathfrak{J}, i \models \phi$ for every $i \geq 0$, which we show by induction on the structure of ϕ.

For the case where ϕ is an axiom, we have for every $i \geq 0$ that $\mathfrak{I}, i \models \phi$ *iff* $\mathcal{I}_i \models \phi$ *iff* $\phi \in \tau_\phi(\mathcal{I}_i)$ *iff* $\phi \in \tau_\phi(\mathcal{J}_i)$ *iff* $\mathcal{J}_i \models \phi$ *iff* $\mathfrak{J}, i \models \phi$.

If ϕ is of the form $\neg\phi_1$, we have for every $i \geq 0$ that $\mathfrak{I}, i \models \neg\phi_1$ *iff* $\mathfrak{I}, i \not\models \phi_1$ *iff* $\mathfrak{J}, i \not\models \phi_1$ *iff* $\mathfrak{J}, i \models \neg\phi_1$.

If ϕ is of the form $\phi_1 \wedge \phi_2$, we have for every $i \geq 0$ that $\mathfrak{I}, i \models \phi_1 \wedge \phi_2$ *iff* $\mathfrak{I}, i \models \phi_1$ and $\mathfrak{I}, i \models \phi_2$ *iff* $\mathfrak{J}, i \models \phi_1$ and $\mathfrak{J}, i \models \phi_2$ *iff* $\mathfrak{J}, i \models \phi_1 \wedge \phi_2$.

If ϕ is of the form $\mathsf{X}\,\phi_1$, we have for every $i \geq 0$ that $\mathfrak{I}, i \models \mathsf{X}\,\phi_1$ *iff* $\mathfrak{I}, i+1 \models \phi_1$ *iff* $\mathfrak{J}, i+1 \models \phi_1$ *iff* $\mathfrak{J}, i \models \mathsf{X}\,\phi_1$.

If ϕ is of the form $\mathsf{X}^-\,\phi_1$, we have for every $i \geq 0$ that $\mathfrak{I}, i \models \mathsf{X}^-\,\phi_1$ *iff* $i > 0$ and $\mathfrak{I}, i-1 \models \phi_1$ *iff* $i > 0$ and $\mathfrak{J}, i-1 \models \phi_1$ *iff* $\mathfrak{J}, i \models \mathsf{X}^-\,\phi_1$.

If ϕ is of the form $\phi_1 \mathsf{U} \phi_2$, we have for every $i \geq 0$ that $\mathfrak{I}, i \models \phi_1 \mathsf{U} \phi_2$ *iff* there is some $k \geq i$ such that $\mathfrak{I}, k \models \phi_2$, and $\mathfrak{I}, j \models \phi_1$ for every j, $i \leq j < k$ *iff* there is some $k \geq i$ such that $\mathfrak{J}, k \models \phi_2$, and $\mathfrak{J}, j \models \phi_1$ for every j, $i \leq j < k$ *iff* $\mathfrak{J}, i \models \phi_1 \mathsf{U} \phi_2$.

Finally, if ϕ is of the form $\phi_1 \mathsf{S} \phi_2$, we have for every $i \geq 0$ that $\mathfrak{I}, i \models \phi_1 \mathsf{S} \phi_2$ *iff* there is some k, $0 \leq k \leq i$, such that $\mathfrak{I}, k \models \phi_2$, and $\mathfrak{I}, j \models \phi_1$ for every j, $k < j \leq i$ *iff* there is some k, $0 \leq k \leq i$, such that $\mathfrak{J}, k \models \phi_2$, and $\mathfrak{J}, j \models \phi_1$ for every j, $k < j \leq i$ *iff* $\mathfrak{J}, i \models \phi_1 \mathsf{S} \phi_2$. □

This lemma justifies considering Büchi-automata that receive axiom types of DL-LTL-structures as input rather than the DL-LTL-structures themselves. The next definition is very similar to the one for propositional LTL; see Definition 2.20.

Definition 4.9 (Büchi-automaton for $\mathcal{SHOQ}$-LTL-formula). *Let $\mathcal{R}$ be an RBox, let ϕ be a $\mathcal{SHOQ}$-LTL-formula w.r.t. $\mathcal{R}$, and let $\mathcal{N}$ be a Büchi-automaton working on the alphabet $\mathfrak{T}_{\phi,\mathcal{R}}$. We define*

$$L_\omega(\phi, \mathcal{R}) := \left\{\tau_\phi(\mathfrak{I}) \in \mathfrak{T}^\omega_{\phi,\mathcal{R}} \mid \mathfrak{I} = (\mathcal{I}_i)_{i \geq 0} \textit{ is a model of } \phi \textit{ w.r.t. } \mathcal{R}\right\},$$

and say that $\mathcal{N}$ is a Büchi-automaton for ϕ w.r.t. $\mathcal{R}$ *if $L_\omega(\mathcal{N}) = L_\omega(\phi, \mathcal{R})$.* ◊

Instead of constructing such Büchi-automata directly for $\mathcal{SHOQ}$-LTL-formulas, we build their propositional abstractions and then reuse the known construction for the propositional case.[5] For that, we reuse also the results shown in Chapter 3. In the following, let $\mathsf{p}\colon \mathsf{Ax}(\phi) \to \mathcal{P}_\phi$ be a bijection. It turns out that it is convenient to define the notion of r-satisfiability (see Definition 3.10) on the level of axiom types.

Definition 4.10 (R-consistency). *Let $\mathfrak{T} = \{T_1, \ldots, T_k\} \subseteq \mathfrak{T}_{\phi,\mathcal{R}}$. We call $\mathfrak{T}$* r-consistent *if there exist interpretations $\mathcal{I}_1 = (\Delta, \cdot^{\mathcal{I}_1})$, …, $\mathcal{I}_k = (\Delta, \cdot^{\mathcal{I}_k})$ such that*

- *$a^{\mathcal{I}_i} = a^{\mathcal{I}_j}$ holds for every $a \in \mathsf{N_I}$ and all i, j, $1 \leq i < j \leq k$;*
- *$A^{\mathcal{I}_i} = A^{\mathcal{I}_j}$ holds for every $A \in \mathsf{N_{RC}}$ and all i, j, $1 \leq i < j \leq k$;*
- *$r^{\mathcal{I}_i} = r^{\mathcal{I}_j}$ holds for every $r \in \mathsf{N_{RR}}$ and all i, j, $1 \leq i < j \leq k$; and*
- *$\mathcal{I}_i \models \mathcal{R}$ and $\tau_\phi(\mathcal{I}_i) = T_i$ holds for every i, $1 \leq i \leq k$.* ◊

Note that any subset of an r-consistent set of axiom types for ϕ w.r.t. $\mathcal{R}$ is again r-consistent. In particular, the empty set is always r-consistent.

We denote the set of all r-consistent sets of axiom types for ϕ w.r.t. $\mathcal{R}$ with $\mathfrak{C}_{\phi,\mathcal{R}}$. Moreover, we denote by $\mathsf{p}(T)$, for $T \subseteq \mathsf{Ax}(\phi)$, the following subset of $\mathcal{P}_\phi$:

$$\mathsf{p}(T) := \left\{\mathsf{p}(\alpha) \mid \alpha \in T\right\}.$$

Conversely, we denote by $\mathsf{p}^{-1}(w)$, for $w \subseteq \mathcal{P}_\phi$, the following subset of $\mathsf{Ax}(\phi)$:

$$\mathsf{p}^{-1}(w) := \left\{\mathsf{p}^{-1}(p) \mid p \in w\right\}.$$

[5] One could also define the Büchi-automata directly as done in [BBL09; Lip09], but the approach developed below is more modular and also easier to implement. In fact, the approach does not depend on a specific algorithm for generating Büchi-automata for propositional LTL-formulas. Thus, any existing tool for transforming a propositional LTL-formula into a Büchi-automaton can be used.

It is easy to see that for any DL-LTL-structure $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$, we have that $\mathfrak{I}^{\mathsf{p}} = (\mathsf{p}(\tau_\phi(\mathcal{I}_i)))_{i\geq 0}$. Moreover, we obtain the following relationship between r-satisfiability and r-consistency.

Lemma 4.11. *Let $\mathfrak{T} = \{T_1, \dots, T_k\} \subseteq \mathfrak{T}_{\phi,\mathcal{R}}$, and let $\mathcal{W} = \{X_1, \dots, X_k\} \subseteq 2^{\mathcal{P}_\phi}$.*

1. *If $\mathfrak{T}$ is r-consistent, then $\{\mathsf{p}(T_1), \dots, \mathsf{p}(T_k)\}$ is r-satisfiable w.r.t. $\mathcal{R}$.*
2. *If $\mathcal{W}$ is r-satisfiable w.r.t. $\mathcal{R}$, then $\{\mathsf{p}^{-1}(X_1), \dots, \mathsf{p}^{-1}(X_k)\}$ is r-consistent.*

Proof. For Part 1 of the lemma, assume that $\mathfrak{T} = \{T_1, \dots, T_k\} \subseteq \mathfrak{T}_{\phi,\mathcal{R}}$ is r-consistent. Then there are interpretations $\mathcal{I}_1 = (\Delta, \cdot^{\mathcal{I}_1})$, ..., $\mathcal{I}_k = (\Delta, \cdot^{\mathcal{I}_k})$ such that the conditions of Definition 4.10 are satisfied. Note that the first three conditions coincide with the ones of Definition 3.10. Thus, to show that $\{\mathsf{p}(T_1), \dots, \mathsf{p}(T_k)\}$ is r-satisfiable w.r.t. $\mathcal{R}$, it only remains to show that every $\mathcal{I}_i$, $1 \leq i \leq k$, is a model of the Boolean knowledge base $\mathcal{B}_{\mathsf{p}(T_i)}$, which is defined as in Definition 3.10. This is satisfied, since for every i, $1 \leq i \leq k$, we have by Definition 4.10 and the arguments above that $\mathcal{I}_i$ is a model of

$$\mathcal{B}_{\mathsf{p}(T_i)} := \left(\bigwedge_{p\in \mathsf{p}(T_i)} \mathsf{p}^{-1}(p) \wedge \bigwedge_{p\in \mathcal{P}_\phi \setminus \mathsf{p}(T_i)} \neg \mathsf{p}^{-1}(p), \quad \mathcal{R} \right) = \left(\bigwedge_{\alpha\in\tau_\phi(\mathcal{I}_i)} \alpha \wedge \bigwedge_{\alpha\in \mathsf{Ax}(\phi)\setminus\tau_\phi(\mathcal{I}_i)} \neg\alpha, \quad \mathcal{R} \right).$$

For Part 2 of the lemma, assume that $\mathcal{W} = \{X_1, \dots, X_k\} \subseteq 2^{\mathcal{P}_\phi}$ is r-satisfiable w.r.t. $\mathcal{R}$. Then there are interpretations $\mathcal{I}_1 = (\Delta, \cdot^{\mathcal{I}_1})$, ..., $\mathcal{I}_k = (\Delta, \cdot^{\mathcal{I}_k})$ such that the conditions of Definition 3.10 are satisfied. Note again that the first three conditions coincide with the ones of Definition 4.10. Thus, to show that $\{\mathsf{p}^{-1}(X_1), \dots, \mathsf{p}^{-1}(X_k)\}$ is r-consistent, it only remains to show that for every i, $1 \leq i \leq k$, we have that $\mathcal{I}_i \models \mathcal{R}$ and $\tau_\phi(\mathcal{I}_i) = \mathsf{p}^{-1}(X_i)$. This is satisfied, since for every i, $1 \leq i \leq k$, we have by Definition 3.10 that $\mathcal{I}_i$ is a model of

$$\mathcal{B}_{X_i} := \left(\bigwedge_{p\in X_i} \mathsf{p}^{-1}(p) \wedge \bigwedge_{p\in \mathcal{P}_\phi \setminus X_i} \neg \mathsf{p}^{-1}(p), \quad \mathcal{R} \right) = \left(\bigwedge_{\alpha\in \mathsf{p}^{-1}(X_i)} \alpha \wedge \bigwedge_{\alpha\in \mathsf{Ax}(\phi)\setminus \mathsf{p}^{-1}(X_i)} \neg\alpha, \quad \mathcal{R} \right),$$

which yields that $\mathcal{I}_i \models \mathcal{R}$ and $\tau_\phi(\mathcal{I}_i) = \{\alpha \in \mathsf{Ax}(\phi) \mid \mathcal{I}_i \models \alpha\} = \mathsf{p}^{-1}(X_i)$. □

We are now ready to define Büchi-automata for $\mathcal{SHOQ}$-LTL-formulas. We consider here only the case without rigid names, i.e. $\mathsf{N_{RC}} = \mathsf{N_{RR}} = \emptyset$, and the case with rigid concept and role names, i.e. $\mathsf{N_{RC}} \neq \emptyset$ and $\mathsf{N_{RR}} \neq \emptyset$. These two cases are treated in Sections 4.2.1 and 4.2.2. The intermediate case where only concept names are allowed to be rigid, i.e. $\mathsf{N_{RC}} \neq \emptyset$ and $\mathsf{N_{RR}} = \emptyset$, is not considered. As we will see below, the corresponding monitors are of size doubly exponential in the size of the input $\mathcal{SHOQ}$-LTL-formula (and the RBox) irrespective of the fact whether rigid names are allowed or not.

4.2.1 The Case without Rigid Names

In this section, we consider the case where neither concept names nor role names are allowed to be rigid, i.e. $\mathsf{N_{RC}} = \mathsf{N_{RR}} = \emptyset$. With the lemmas above, we are able to establish the following result.

Theorem 4.12. *Let $\mathcal{R}$ be an RBox, let ϕ a $\mathcal{SHOQ}$-LTL-formula w.r.t. $\mathcal{R}$, and let* $\mathsf{p}\colon \mathsf{Ax}(\phi) \to \mathcal{P}_\phi$ *be a bijection. If* $\mathsf{N_{RC}} = \mathsf{N_{RR}} = \emptyset$ *and* $\mathcal{N}_{\phi^\mathsf{p}} = (Q, \Sigma_{\mathcal{P}_\phi}, \Delta, Q_0, F)$ *is a Büchi-automaton for the propositional abstraction* ϕ^p *of* ϕ*, then* $\mathcal{N}_{\phi,\mathcal{R}} := (Q, \mathfrak{T}_{\phi,\mathcal{R}}, \Delta', Q_0, F)$ *with*

$$\Delta' := \big\{(q, T, q') \mid (q, \mathsf{p}(T), q') \in \Delta \textit{ and } T \in \mathfrak{T}_{\phi,\mathcal{R}}\big\}$$

is a Büchi-automaton for ϕ *w.r.t.* $\mathcal{R}$.

Proof. We have to show that $L_\omega(\mathcal{N}_{\phi,\mathcal{R}}) = L_\omega(\phi, \mathcal{R})$.

For the direction '$\subseteq$', assume that $T_0T_1T_2 \ldots \in L_\omega(\mathcal{N}_{\phi,\mathcal{R}})$. The definition of $\mathcal{N}_{\phi,\mathcal{R}}$ yields that $\mathsf{p}(T_0)\mathsf{p}(T_1)\mathsf{p}(T_2)\ldots \in L_\omega(\mathcal{N}_{\phi^\mathsf{p}})$. Since $\mathcal{N}_{\phi^\mathsf{p}}$ is a Büchi-automaton for ϕ^p, we obtain that $\mathfrak{W} = (\mathsf{p}(T_i))_{i\geq 0}$ is a model of ϕ^p. We define $\mathfrak{T} := \{T_i \mid i \geq 0\}$. Since for every $i \geq 0$, we have $T_i \in \mathfrak{T}_{\phi,\mathcal{R}}$, it follows that $\mathfrak{T} = \{T'_1, \ldots, T'_k\} \subseteq \mathfrak{T}_{\phi,\mathcal{R}}$. By Lemma 4.7, we have that for every i, $1 \leq i \leq k$, there is a model $\mathcal{J}_i$ of $\mathcal{R}$ such that $T'_i = \tau_\phi(\mathcal{J})$. We can assume w.l.o.g. that all of these models have the same domain since we can assume w.l.o.g. that their domains are countably infinite due to the Löwenheim-Skolem theorem [Löw15; Sko20]. Moreover, we can assume w.l.o.g. that all individual names are interpreted by the same domain elements in all models. Since $\mathsf{N_{RC}} = \mathsf{N_{RR}} = \emptyset$, this yields that $\mathfrak{T}$ is r-consistent. By Lemma 4.11, we have that $\mathcal{W} := \{\mathsf{p}(T'_1), \ldots, \mathsf{p}(T'_k)\} = \{\mathsf{p}(T_i) \mid i \geq 0\}$ is r-satisfiable w.r.t. $\mathcal{R}$. Then, Lemma 3.12 yields that there is a model $\mathfrak{I}$ of ϕ w.r.t. $\mathcal{R}$ with $\mathfrak{I}^\mathsf{p} = \mathfrak{W}$. Consequently, $\tau_\phi(\mathfrak{I}) = T_0T_1T_2 \ldots \in L_\omega(\phi, \mathcal{R})$.

For the direction '$\supseteq$', assume that $T = T_0T_1T_2 \ldots \in L_\omega(\phi, \mathcal{R})$. Then there is a model $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$ of ϕ w.r.t. $\mathcal{R}$ with $\tau_\phi(\mathfrak{I}) = T$. By Lemma 4.7, for every $i \geq 0$, the letter T_i is an axiom type for ϕ w.r.t. $\mathcal{R}$, i.e. $T_i \in \mathfrak{T}_{\phi,\mathcal{R}}$. By Lemma 3.12, we have that $\mathfrak{I}^\mathsf{p} = (\mathsf{p}(T_i))_{i\geq 0}$ is a model of ϕ^p, and thus the ω-word $\mathsf{p}(T_0)\mathsf{p}(T_1)\mathsf{p}(T_2)\ldots$ is accepted by $\mathcal{N}_{\phi^\mathsf{p}}$. Consequently, we have $T_0T_1T_2 \ldots \in L_\omega(\mathcal{N}_{\phi,\mathcal{R}})$. □

As an immediate consequence of this theorem, the satisfiability problem in $\mathcal{SHOQ}$-LTL (for the case $\mathsf{N_{RC}} = \mathsf{N_{RR}} = \emptyset$) can be reduced to the emptiness problem for Büchi-automata.

Corollary 4.13. *If* $\mathsf{N_{RC}} = \mathsf{N_{RR}} = \emptyset$*, the* $\mathcal{SHOQ}$*-LTL-formula* ϕ *is satisfiable w.r.t. the RBox* $\mathcal{R}$ *iff* $L_\omega(\mathcal{N}_{\phi,\mathcal{R}}) \neq \emptyset$.

It remains to analyse the complexity of the decision procedure for the satisfiability problem obtained by this reduction.

The size of the Büchi-automaton $\mathcal{N}_{\phi,\mathcal{R}}$ is obviously exponential in the size of ϕ (and $\mathcal{R}$) since the Büchi-automaton $\mathcal{N}_{\phi^\mathsf{p}}$ for ϕ^p is of size exponential in the size of ϕ^p (and thus exponential in the size of ϕ since the size of ϕ^p is linearly bounded by the size of ϕ) as discussed in Section 2.2.2. In addition, the Büchi-automaton $\mathcal{N}_{\phi,\mathcal{R}}$ can be constructed in exponential time. As shown in Section 2.2.2, a Büchi-automaton for a propositional LTL-formula can be constructed in time exponential in the size of the input formula. Thus, we can construct $\mathcal{N}_{\phi^\mathsf{p}}$ in time exponential in the size of ϕ^p (and thus in time exponential in the size of ϕ). To obtain $\mathcal{N}_{\phi,\mathcal{R}}$ from $\mathcal{N}_{\phi^\mathsf{p}}$, we basically have to remove all transitions labelled with a letter σ such that $\mathsf{p}^{-1}(\sigma) \notin \mathfrak{T}_{\phi,\mathcal{R}}$. For the remaining transitions, we then simply replace the letter σ with $\mathsf{p}^{-1}(\sigma)$. In order to check whether $\mathsf{p}^{-1}(\sigma)$ belongs to $\mathfrak{T}_{\phi,\mathcal{R}}$, we need to check the Boolean knowledge base $\mathcal{B}_{\mathsf{p}^{-1}(\sigma)}$ for consistency. By Corollary 3.34, this can be done in time exponential in the size of this Boolean knowledge base. Since there are exponentially many letters σ, but the size of each Boolean knowledge base $\mathcal{B}_{\mathsf{p}^{-1}(\sigma)}$ is linearly bounded by the size of ϕ and $\mathcal{R}$, we need to perform exponentially many checks (where each needs exponential time), which yields an overall complexity of exponential time for the construction of $\mathcal{N}_{\phi,\mathcal{R}}$.

Since the emptiness problem for Büchi-automata can be solved in time polynomial in the size of the Büchi-automaton [VW94], this yields an alternative proof of the fact that the satisfiability problem in $\mathcal{SHOQ}$-LTL is in ExpTime if $\mathsf{N_{RC}} = \mathsf{N_{RR}} = \emptyset$ (see Theorem 3.15).

4.2.2 The Case of Rigid Concept and Role Names

In this section, we consider the case where both concept and role names may be rigid, i.e. $\mathsf{N_{RC}} \neq \emptyset$ and $\mathsf{N_{RR}} \neq \emptyset$. If rigid names are allowed, the Büchi-automaton needs to check whether the set of axiom types seen within a run are r-consistent. This is achieved by using tuples (q_1, q_2) as states, where q_1 is a state of the Büchi-automaton $\mathcal{N}_{\phi,\mathcal{R}}$ introduced in Theorem 4.12, and q_2 is an r-consistent set of axiom types for ϕ w.r.t. $\mathcal{R}$.

Theorem 4.14. *Let $\mathcal{R}$ be an RBox, let ϕ a $\mathcal{SHOQ}$-LTL-formula w.r.t. $\mathcal{R}$, and let $\mathsf{p}\colon \mathsf{Ax}(\phi) \to \mathcal{P}_\phi$ be a bijection. If $\mathcal{N}_{\phi^{\mathsf{p}}} = (Q, \Sigma_{\mathcal{P}_\phi}, \Delta, Q_0, F)$ is a Büchi-automaton for the propositional abstraction ϕ^{p} of ϕ, then $\mathcal{N}^{\mathsf{r}}_{\phi,\mathcal{R}} := (Q \times \mathfrak{C}_{\phi,\mathcal{R}}, \mathfrak{T}_{\phi,\mathcal{R}}, \Delta', Q_0 \times \{\emptyset\}, F \times \mathfrak{C}_{\phi,\mathcal{R}})$ with*

$$\Delta' := \left\{((q_1,q_2), T, (q_1',q_2')) \mid (q_1, \mathsf{p}(T), q_1') \in \Delta \text{ and } q_2' = q_2 \cup \{T\} \in \mathfrak{C}_{\phi,\mathcal{R}}\right\}$$

is a Büchi-automaton for ϕ w.r.t. $\mathcal{R}$.

Proof. We have to show that $L_\omega(\mathcal{N}^{\mathsf{r}}_{\phi,\mathcal{R}}) = L_\omega(\phi, \mathcal{R})$.

For the direction '⊆', assume that $T = T_0T_1T_2\ldots \in L_\omega(\mathcal{N}^{\mathsf{r}}_{\phi,\mathcal{R}})$, and let

$$(q_1^{(0)}, q_2^{(0)})(q_1^{(1)}, q_2^{(1)})(q_1^{(2)}, q_2^{(2)})\ldots$$

be an accepting run of $\mathcal{N}^{\mathsf{r}}_{\phi,\mathcal{R}}$ on T. It is easy to see that the projection $q_1^{(0)}q_1^{(1)}q_1^{(2)}\ldots$ of this run to the first component is an accepting run of $\mathcal{N}_{\phi,\mathcal{R}}$ on T. As argued in the proof of Theorem 4.12, we have that $\mathsf{p}(T_0)\mathsf{p}(T_1)\mathsf{p}(T_2)\ldots \in L_\omega(\mathcal{N}_{\phi^{\mathsf{p}}})$, and that $\mathfrak{W} = (\mathsf{p}(T_i))_{i\geq 0}$ is a model of ϕ^{p}. We define $\mathfrak{T} := \{T_i \mid i \geq 0\}$. Since for every $i \geq 0$, we have $T_i \in \mathfrak{T}_{\phi,\mathcal{R}}$, it follows that $\mathfrak{T} = \{T_1', \ldots, T_k'\} \subseteq \mathfrak{T}_{\phi,\mathcal{R}}$. It remains to show that $\mathfrak{T}$ is r-consistent. In fact, once this is shown, Lemma 4.11 yields that $\mathfrak{W} := \{\mathsf{p}(T_1'), \ldots, \mathsf{p}(T_k')\} = \{\mathsf{p}(T_i) \mid i \geq 0\}$ is r-satisfiable w.r.t. $\mathcal{R}$. Then, Lemma 3.12 yields that there is a model $\mathfrak{I}$ of ϕ w.r.t. $\mathcal{R}$ with $\mathfrak{I}^{\mathsf{p}} = \mathfrak{W}$. Consequently, $\tau_\phi(\mathfrak{I}) = T_0T_1T_2\ldots \in L_\omega(\phi, \mathcal{R})$. To see that $\mathfrak{T}$ is r-consistent, we note that the second components $q_2^{(j)}$, $j \geq 0$, of the states in the run are r-consistent sets of axiom types for ϕ w.r.t. $\mathcal{R}$ satisfying $q_2^{(j)} = \{T_0, \ldots, T_{j-1}\}$. Since there are only finitely many axiom types for ϕ w.r.t. $\mathcal{R}$, there is an index $\ell \geq 1$ such that $q_2^{(k)} = \{T_i \mid i \geq 0\}$, and thus this set is r-consistent.

For the direction '⊇', assume that $T_0T_1T_2\ldots \in L_\omega(\phi, \mathcal{R})$. Then there is a model $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$ of ϕ w.r.t. $\mathcal{R}$ with $\tau_\phi(\mathfrak{I}) = T$. Using the arguments in the proof of Lemma 4.12 (direction '⊇'), we obtain that $T_0T_1T_2\ldots \in L_\omega(\mathcal{N}_{\phi,\mathcal{R}})$. Let $q_1^{(0)}q_1^{(1)}q_1^{(2)}\ldots$ be an accepting run of $\mathcal{N}_{\phi,\mathcal{R}}$ on T. Moreover, by Lemma 3.12, we obtain that $\mathfrak{I}^{\mathsf{p}} = (\mathsf{p}(T_i))_{i\geq 0}$ is a model of ϕ^{p} and $\mathcal{W} = \{X_1, \ldots, X_k\} := \{\mathsf{p}(T_i) \mid i \geq 0\}$ is r-satisfiable w.r.t. $\mathcal{R}$. By Lemma 4.11, we obtain that the set $\{\mathsf{p}^{-1}(X_1), \ldots, \mathsf{p}^{-1}(X_k)\} = \{T_i \mid i \geq 0\}$ is r-consistent. If we define $q_2^{(j)} := \{T_0, \ldots, T_{j-1}\}$ for every $j \geq 0$, then these sets are r-consistent since they are subsets of the r-consistent set $\{T_i \mid i \geq 0\}$. Consequently,

$$(q_1^{(0)}, q_2^{(0)})(q_1^{(1)}, q_2^{(1)})(q_1^{(2)}, q_2^{(2)})\ldots$$

is an accepting run of $\mathcal{N}^{\mathfrak{T}}_{\phi,\mathcal{R}}$ on T. □

As an immediate consequence of this theorem, we obtain that the satisfiability problem in $\mathcal{SHOQ}$-LTL (even for the case where $\mathsf{N}_{\mathsf{RC}} \neq \emptyset$ and $\mathsf{N}_{\mathsf{RR}} \neq \emptyset$) can be reduced to the emptiness problem for Büchi-automata.

Corollary 4.15. *The $\mathcal{SHOQ}$-LTL-formula ϕ is satisfiable w.r.t. the RBox $\mathcal{R}$ iff $L_\omega(\mathcal{N}^{\mathfrak{T}}_{\phi,\mathcal{R}}) \neq \emptyset$.*

The complexity of the decision procedure for the satisfiability problem obtained by this reduction is, however, higher than the complexity of the decision procedure for the case without rigid names.

The size of the Büchi-automaton $\mathcal{N}^{\mathfrak{T}}_{\phi,\mathcal{R}}$ is doubly exponential in the size of ϕ and $\mathcal{R}$. This is due to the fact that the set $\mathfrak{C}_{\phi,\mathcal{R}}$ of all r-consistent sets of axiom types for ϕ w.r.t. $\mathcal{R}$ may contain doubly exponentially many elements since these sets are subsets of the exponentially large set $\mathfrak{T}_{\phi,\mathcal{R}}$ of all axiom types for ϕ w.r.t. $\mathcal{R}$. Each element of $\mathfrak{C}_{\phi,\mathcal{R}}$ may be of exponential size.

Next, we show that the Büchi-automaton $\mathcal{N}^{\mathfrak{T}}_{\phi,\mathcal{R}}$ can be constructed in doubly exponential time. In addition to constructing the Büchi-automaton $\mathcal{N}_{\phi,\mathcal{R}}$, i.e. the Büchi-automaton constituting the first component of $\mathcal{N}^{\mathfrak{T}}_{\phi,\mathcal{R}}$, we must also compute the set $\mathfrak{C}_{\phi,\mathcal{R}}$. For this, we consider all sets of axiom types for ϕ w.r.t. $\mathcal{R}$. There are doubly exponentially many such sets, each of size at most exponential in the size of ϕ and $\mathcal{R}$. By Lemma 4.11, checking such a set $\mathfrak{T} = \{T_1, \ldots, T_k\}$ for r-consistency amounts to check the set $\mathcal{W} := \{\mathsf{p}(T_1), \ldots, \mathsf{p}(T_k)\}$ for r-satisfiability w.r.t. $\mathcal{R}$. By Lemma 3.16, this amounts to checking the Boolean knowledge base $\mathcal{B}_{\mathcal{W}}$ (as defined in Section 3.2.2) for consistency. Since the size of $\mathcal{B}_{\mathcal{W}}$ is exponential in the size of ϕ and $\mathcal{R}$, we obtain by Corollary 3.34 that this can be done in time doubly exponential in the size of ϕ and $\mathcal{R}$. Overall, the computation of $\mathfrak{C}_{\phi,\mathcal{R}}$ requires doubly exponentially many such tests, each requiring doubly exponential time. This shows that $\mathfrak{C}_{\phi,\mathcal{R}}$, and thus also the Büchi-automaton $\mathcal{N}^{\mathfrak{T}}_{\phi,\mathcal{R}}$ can be constructed in doubly exponential time.

Since the emptiness problem for Büchi-automata can be solved in time polynomial in the size of the Büchi-automaton [VW94], this yields an alternative proof of the fact that the satisfiability problem in $\mathcal{SHOQ}$-LTL is in 2ExpTime (see Theorem 3.17).

4.3 Monitoring $\mathcal{SHOQ}$-LTL-Formulas

In this section, we extend existing definitions and results for runtime verification from propositional LTL to $\mathcal{SHOQ}$-LTL. We restrict the attention to the case with rigid names since the complexity of the monitor construction for this more general case is actually the same (doubly exponential) as for the case without rigid names. Thus, it does not make sense to treat the restricted case separately. In addition to considering a more expressive logic, our notion of monitoring extends the one for propositional logic in two directions.

On the one hand, we do not assume that the monitor has complete knowledge about the states of the system. In the propositional case, as introduced in Section 4.1, at each point in time the monitor 'knows' which of the propositional variables are true at this point and which are not. In our setting, $\mathcal{SHOQ}$-axioms take the place of propositional variables, but we do not assume that we have complete knowledge about their truth value. For some of the relevant axioms, we may know that they are true, for others that they are false, but it also may be the case that we have no information regarding the truth status of a certain axiom.

On the other hand, we take background knowledge about the working of the system into account. This background knowledge could, for example, be a global TBox, i.e. a finite set of GCIs that are known to hold for every state of the system. In this case, the formula describing the background knowledge is of the form $\psi = \Box \bigwedge \mathcal{T}$, where $\mathcal{T}$ is a TBox. The presence of background knowledge enables the monitor to give more often definite answers (i.e. $\top$ or $\bot$) rather than the answer ?.

4.3.1 Basic Definitions

In the following, we extend the notion of a monitoring function and a monitor, as introduced in Section 4.1 for propositional LTL-formulas, to the case of $\mathcal{SHOQ}$-LTL-formulas ϕ w.r.t. an RBox $\mathcal{R}$. We assume that the background knowledge is described by an additional $\mathcal{SHOQ}$-LTL-formula ψ w.r.t. $\mathcal{R}$, and that the monitor receives the information about the current state of the system in the form of a Boolean knowledge base $\mathcal{O}$ (observations) that provides (partial) information about the truth values of certain axioms from a fixed finite set of axioms. Without loss of generality, we can also assume that the axioms occurring in ψ and $\mathcal{O}$ also occur in ϕ. This assumption is indeed without loss of generality since for every such axiom α, which does not occur in ϕ, we can define $\phi' := \phi \wedge (\alpha \vee \neg\alpha)$. Obviously, every model of ϕ is also a model of ϕ', and vice versa. We make this assumption throughout this section without explicitly mentioning it.

To simplify the subsequent definitions, we introduce the following notation. A *literal of ϕ* is either an axiom $\alpha \in \mathsf{Ax}(\phi)$ (*positive* literal) or the negation $\neg\alpha$ of an axiom $\alpha \in \mathsf{Ax}(\phi)$ (*negative* literal).

Definition 4.16 (Partial axiom type). *A finite conjunction $\mathcal{O} = L_1 \wedge \cdots \wedge L_m$ of literals of ϕ is a* partial axiom type for ϕ w.r.t. $\mathcal{R}$ *if the Boolean knowledge base $(\mathcal{O}, \mathcal{R})$ is consistent.* ◇

We denote the set of all partial axiom types for ϕ w.r.t. $\mathcal{R}$ by $\mathfrak{P}_{\phi,\mathcal{R}}$. Note that, up to equivalence, there are at most exponentially many (in the size of ϕ and $\mathcal{R}$) partial axiom types for ϕ w.r.t. $\mathcal{R}$.

Given an axiom type T for ϕ w.r.t. $\mathcal{R}$, the first component of the corresponding Boolean knowledge base $\mathcal{B}_T = (\mathcal{O}_T, \mathcal{R})$, i.e.

$$\mathcal{O}_T := \bigwedge_{\alpha \in T} \alpha \wedge \bigwedge_{\alpha \in \mathsf{Ax}(\phi) \setminus T} \neg\alpha,$$

is a partial axiom type for ϕ w.r.t. $\mathcal{R}$. In this case, every axiom of ϕ occurs either positively or negatively in $\mathcal{O}_T$. For an arbitrary partial axiom type $\mathcal{O}$ for ϕ w.r.t. $\mathcal{R}$, this need not be the case. Some axioms of ϕ may not occur at all in $\mathcal{O}$.

Definition 4.17 (Extensions). *Let $\mathcal{R}$ be an RBox, ϕ and ψ be $\mathcal{SHOQ}$-LTL-formulas w.r.t. $\mathcal{R}$, and $\mathfrak{O} = \mathcal{O}_0\mathcal{O}_1 \ldots \mathcal{O}_t$ be a* finite *sequence of partial axioms types for ϕ and $\mathcal{R}$.*

1. *We say that the DL-LTL-structure $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$* extends $\mathfrak{O}$ w.r.t. ψ and $\mathcal{R}$ *if $\mathfrak{I}$ is a model of ψ w.r.t. $\mathcal{R}$, and $\mathcal{I}_i \models \mathcal{O}_i$ for every i, $0 \leq i \leq t$.*
2. *We write $\mathfrak{O}, \psi, \mathcal{R} \mathrel{|\!\!\approx_\exists} \phi$ if there is a DL-LTL-structure $\mathfrak{I}$ that extends $\mathfrak{O}$ w.r.t. ψ and $\mathcal{R}$, and is a model of ϕ w.r.t. $\mathcal{R}$. If this is not the case, we write $\mathfrak{O}, \psi, \mathcal{R} \mathrel{|\!\!\not\approx_\exists} \phi$.*
3. *We write $\mathfrak{O}, \psi, \mathcal{R} \mathrel{|\!\!\approx_\forall} \phi$ if every DL-LTL-structure $\mathfrak{I}$ extending $\mathfrak{O}$ w.r.t. ψ and $\mathcal{R}$ is a model of ϕ w.r.t. $\mathcal{R}$. If this is not the case, we write $\mathfrak{O}, \psi, \mathcal{R} \mathrel{|\!\!\not\approx_\forall} \phi$.* ◇

The notions introduced in Part 2 and Part 3 of this definition are dual to each other in the following sense:

$$\mathfrak{O}, \psi, \mathcal{R} \mathrel{|\!\!\approx_\exists} \phi \text{ iff } \mathfrak{O}, \psi, \mathcal{R} \mathrel{|\!\!\not\approx_\forall} \neg\phi \qquad \text{and} \qquad \mathfrak{O}, \psi, \mathcal{R} \mathrel{|\!\!\approx_\forall} \phi \text{ iff } \mathfrak{O}, \psi, \mathcal{R} \mathrel{|\!\!\not\approx_\exists} \neg\phi.$$

We assume that our system actually respects rigid names and satisfies the background knowledge ψ and the RBox $\mathcal{R}$ in the sense that any run of the system corresponds to a DL-LTL-structure that is a model of ψ w.r.t. $\mathcal{R}$. Thus, if the monitor receives information about the partial axiom types of a finite prefix of such a run, this finite sequence of partial axiom types can actually be extended to a DL-LTL-structure satisfying ψ w.r.t. $\mathcal{R}$. In this case, the following lemma holds.

Lemma 4.18. *Let $\mathcal{R}$ be an RBox, ϕ and ψ be $\mathcal{SHOQ}$-LTL-formulas w.r.t. $\mathcal{R}$, and $\mathfrak{O}$ be a finite sequence of partial axiom types for ϕ w.r.t. $\mathcal{R}$ such that there is a DL-LTL-structure extending $\mathfrak{O}$ w.r.t. ψ and $\mathcal{R}$. Then $\mathfrak{O}, \psi, \mathcal{R} \mathrel{|\!\!\approx_\forall} \phi$ and $\mathfrak{O}, \psi, \mathcal{R} \mathrel{|\!\!\approx_\forall} \neg\phi$ cannot both be true.*

Proof. Let $\mathfrak{I}$ be a DL-LTL-structure extending $\mathfrak{O}$ w.r.t. ψ and $\mathcal{R}$. Then we have $\mathfrak{I}, 0 \models \phi$ or $\mathfrak{I}, 0 \models \neg\phi$. In the first case, we have $\mathfrak{O}, \psi, \mathcal{R} \mathrel{|\!\!\not\approx_\forall} \neg\phi$, and in the second one, we have $\mathfrak{O}, \psi, \mathcal{R} \mathrel{|\!\!\not\approx_\forall} \phi$. □

The monitoring function receives as input a finite sequence of partial axiom types for ϕ w.r.t. $\mathcal{R}$, i.e. a finite word over the alphabet $\mathfrak{P}_{\phi,\mathcal{R}}$.

Definition 4.19 (Monitoring function). *Let $\mathcal{R}$ be an RBox, and let ϕ and ψ be $\mathcal{SHOQ}$-LTL-formulas w.r.t. $\mathcal{R}$. The* monitoring function for ϕ w.r.t. ψ and $\mathcal{R}$ *is defined to be the function $\mathsf{m}_{\phi,\psi,\mathcal{R}} \colon \mathfrak{P}^*_{\phi,\mathcal{R}} \to \{\top, \bot, ?, \lightning\}$ with*

$$\mathsf{m}_{\phi,\psi,\mathcal{R}}(\mathfrak{O}) := \begin{cases} \top & \text{if } \mathfrak{O}, \psi, \mathcal{R} \mathrel{|\!\!\approx_\forall} \phi \text{ and } \mathfrak{O}, \psi, \mathcal{R} \mathrel{|\!\!\not\approx_\forall} \neg\phi; \\ \bot & \text{if } \mathfrak{O}, \psi, \mathcal{R} \mathrel{|\!\!\not\approx_\forall} \phi \text{ and } \mathfrak{O}, \psi, \mathcal{R} \mathrel{|\!\!\approx_\forall} \neg\phi; \\ ? & \text{if } \mathfrak{O}, \psi, \mathcal{R} \mathrel{|\!\!\not\approx_\forall} \phi \text{ and } \mathfrak{O}, \psi, \mathcal{R} \mathrel{|\!\!\not\approx_\forall} \neg\phi; \text{ and} \\ \lightning & \text{if } \mathfrak{O}, \psi, \mathcal{R} \mathrel{|\!\!\approx_\forall} \phi \text{ and } \mathfrak{O}, \psi, \mathcal{R} \mathrel{|\!\!\approx_\forall} \neg\phi. \end{cases}$$ ◇

Compared to the definition of the monitoring function in the propositional setting, we have added the fourth possible output $\lightning$ in order to have a well-defined value also for sequences $\mathfrak{O} \in \mathfrak{P}^*_{\phi,\mathcal{R}}$ that have no extension w.r.t. ψ and $\mathcal{R}$. In fact, if there is no DL-LTL-structure extending $\mathfrak{O}$ w.r.t. ψ and $\mathcal{R}$, the monitoring function yields the value $\lightning$. In practice, this value should not be encountered since we assume that the observed system actually respects rigid names and satisfies the background knowledge ψ and $\mathcal{R}$. Thus, no finite sequence of partial axiom types obtained by observing the system can yields this case.[6] The monitoring function returns the value $\top$ if there is at least one extension of $\mathfrak{D}$ w.r.t. ψ and $\mathcal{R}$ (expressed by $\mathfrak{D}, \psi, \mathcal{R} \mathrel{|\!\!\not\approx_{\forall}} \neg\phi$), and all such extensions satisfy ϕ (expressed by $\mathfrak{D}, \psi, \mathcal{R} \mathrel{|\!\!\approx_{\forall}} \phi$). Similarly, it returns the value $\bot$ if there is at least one extension of $\mathfrak{D}$ w.r.t. ψ and $\mathcal{R}$, and all such extensions satisfy $\neg\phi$. Finally, it returns he value ? if there is an extension of $\mathfrak{D}$ w.r.t. ψ and $\mathcal{R}$ that satisfies ϕ, and there is another extension of $\mathfrak{D}$ w.r.t. ψ and $\mathcal{R}$ that satisfies $\neg\phi$.

We are interested in constructing a monitor that realises the monitoring function defined above. As in the propositional case, this monitor is a deterministic Moore-automaton whose output function is equal to the monitoring function.

Definition 4.20 (Monitor for $\mathcal{SHOQ}$-LTL-formula). *Let $\mathcal{R}$ be an RBox, and ϕ, ψ be $\mathcal{SHOQ}$-LTL-formulas w.r.t. $\mathcal{R}$. The deterministic Moore-automaton $\mathcal{M} = (S, \mathfrak{P}_{\phi,\mathcal{R}}, \delta, s_0, \{\top, \bot, ?, \lightning\}, \lambda)$ is a* monitor for ϕ w.r.t. ψ and $\mathcal{R}$ *if $\lambda^*(\mathfrak{D}) = \mathsf{m}_{\phi,\psi,\mathcal{R}}(\mathfrak{D})$ holds for every $\mathfrak{D} \in \mathfrak{P}^*_{\phi,\mathcal{R}}$.* ◇

Before we construct the monitor, we need an auxiliary automaton that we define next.

4.3.2 An Auxiliary Deterministic Finite Automaton

In this section, we define a *deterministic* finite automaton that accepts exactly those sequences of partial axiom types $\mathfrak{D} \in \mathfrak{P}^*_{\phi,\mathcal{R}}$ such that $\mathfrak{D}, \psi, \mathcal{R} \mathrel{|\!\!\approx_{\forall}} \phi$. We know that requiring $\mathfrak{D}, \psi, \mathcal{R} \mathrel{|\!\!\approx_{\forall}} \phi$ is the same as requiring $\mathfrak{D}, \psi, \mathcal{R} \mathrel{|\!\!\not\approx_{\exists}} \neg\phi$. Thus, the automaton needs to accept all words $\mathfrak{D} \in \mathfrak{P}^*_{\phi,\mathcal{R}}$ that have no extension w.r.t. ψ and $\mathcal{R}$ that satisfies $\neg\phi$. To construct this automaton, we take the Büchi-automaton $\mathcal{N}^{\mathfrak{T}}_{\neg\phi\wedge\psi,\mathcal{R}}$ for $\neg\phi \wedge \psi$ w.r.t. $\mathcal{R}$ as defined in Theorem 4.14, and make it deterministic by applying an appropriate modification of the power-set construction to the first components of the states of $\mathcal{N}^{\mathfrak{T}}_{\neg\phi\wedge\psi,\mathcal{R}}$. The second component of a state of $\mathcal{N}^{\mathfrak{T}}_{\neg\phi\wedge\psi,\mathcal{R}}$ collects the axiom types encountered on the path leading to this state, which enables the automaton to check whether this collection of axiom types is r-consistent. Instead, our deterministic automaton collects the partial axiom types encountered on a path, and checks whether this set is related in an appropriate way to an r-consistent set of axiom types.

Before we can define this relation, we need to introduce some notation. Given a partial axiom type $\mathcal{O} = L_1 \wedge \cdots \wedge L_m$, we define $\mathsf{Pos}(\mathcal{O}) := \{L_i \mid 1 \leq i \leq m,\ L_i \text{ is positive}\}$ and $\mathsf{Neg}(\mathcal{O}) := \{\alpha_i \mid 1 \leq i \leq m,\ L_i = \neg\alpha_i \text{ is negative}\}$. Given an axiom type T for ϕ w.r.t. $\mathcal{R}$ and a partial axiom type $\mathcal{O}$ for ϕ w.r.t. $\mathcal{R}$, we define

$$\mathcal{O} <^{\phi,\mathcal{R}} T \quad \text{iff} \quad \mathsf{Pos}(\mathcal{O}) \subseteq T \text{ and } \mathsf{Neg}(\mathcal{O}) \cap T = \emptyset.$$

If $T = \tau_{\phi}(\mathcal{I})$ for a model $\mathcal{I}$ of $\mathcal{R}$, then we obviously have $\mathcal{I} \models (\mathcal{O}, \mathcal{R})$ iff $\mathcal{O} <^{\phi,\mathcal{R}} T$. We now lift the relation $<^{\phi,\mathcal{R}}$ from (partial) axiom types to sets of (partial) axiom types.

Definition 4.21 (Realisation). *Let $\mathfrak{T}$ be a set of axiom types for ϕ w.r.t. $\mathcal{R}$, and let $\mathfrak{P}$ be a set of partial axiom types for ϕ w.r.t. $\mathcal{R}$. We say that $\mathfrak{T}$* realises *$\mathfrak{P}$ and write $\mathfrak{P} \prec^{\phi,\mathcal{R}} \mathfrak{T}$ if the following property is satisfied: for every $\mathcal{O} \in \mathfrak{P}$, there is a $T \in \mathfrak{T}$ such that $\mathcal{O} <^{\phi,\mathcal{R}} T$.* ◇

[6] If it does, then the modelling of the properties of the system using ψ, $\mathcal{R}$, and the rigididy of symbols was incorrect, or the sensors that generated the sequence $\mathfrak{D}$ were faulty.

This relation can be used to characterise r-consistency of a set of partial axiom types for ϕ w.r.t. $\mathcal{R}$.

Definition 4.22 (R-consistency of partial axiom types). *Let $\mathfrak{P} = \{\mathcal{O}_1, \dots, \mathcal{O}_k\}$ be a set of partial axiom types for ϕ w.r.t. $\mathcal{R}$. We call $\mathfrak{P}$* r-consistent *if there exist interpretations $\mathcal{I}_1 = (\Delta, \cdot^{\mathcal{I}_1})$, ..., $\mathcal{I}_k = (\Delta, \cdot^{\mathcal{I}_k})$ such that*

- *$a^{\mathcal{I}_i} = a^{\mathcal{I}_j}$ holds for every $a \in \mathsf{N_I}$ and all i, j, $1 \leq i < j \leq k$;*
- *$A^{\mathcal{I}_i} = A^{\mathcal{I}_j}$ holds for every $A \in \mathsf{N_{RC}}$ and all i, j, $1 \leq i < j \leq k$;*
- *$r^{\mathcal{I}_i} = r^{\mathcal{I}_j}$ holds for every $r \in \mathsf{N_{RR}}$ and all i, j, $1 \leq i < j \leq k$; and*
- *$\mathcal{I}_i \models (\mathcal{O}_i, \mathcal{R})$ holds for every i, $1 \leq i \leq k$.* ◇

We denote the set of all r-consistent sets of partial axiom types for ϕ w.r.t. $\mathcal{R}$ with $\mathfrak{C}^{\mathrm{p}}_{\phi,\mathcal{R}}$.

Lemma 4.23. *The set $\mathfrak{P}$ of partial axiom types for ϕ w.r.t. $\mathcal{R}$ is r-consistent iff there is an r-consistent set $\mathfrak{T}$ of axiom types for ϕ w.r.t. $\mathcal{R}$ such that $\mathfrak{P} \prec^{\phi,\mathcal{R}} \mathfrak{T}$.*

Proof. For the 'only if' direction, assume that $\mathfrak{P} = \{\mathcal{O}_1, \dots, \mathcal{O}_k\}$ is r-consistent. Then there are interpretations $\mathcal{I}_1, \dots, \mathcal{I}_k$ that share the same domain, coincide on the individual names and the rigid concept and role names, and satisfy $\mathcal{I}_i \models (\mathcal{O}_i, \mathcal{R})$ for every i, $1 \leq i \leq k$. If we define $\mathfrak{T} := \{\tau_\phi(\mathcal{I}_1), \dots, \tau_\phi(\mathcal{I}_k)\}$, then this set of axiom types is obviously r-consistent, and we have $\mathcal{O}_i <^{\phi,\mathcal{R}} \tau_\phi(\mathcal{I}_i)$ for every i, $1 \leq i \leq k$. This shows $\mathfrak{P} \prec^{\phi,\mathcal{R}} \mathfrak{T}$.

Conversely, for the 'if' direction, let $\mathfrak{P} = \{\mathcal{O}_1, \dots, \mathcal{O}_k\}$ and assume that $\mathfrak{T} = \{T_1, \dots, T_m\}$ is an r-consistent set of axiom types for ϕ w.r.t. $\mathcal{R}$ such that $\mathfrak{P} \prec^{\phi,\mathcal{R}} \mathfrak{T}$. Then there are interpretations $\mathcal{I}_1, \dots, \mathcal{I}_m$ that share the same domain, coincide on the individual names and the rigid concept and role names, and satisfy $\mathcal{I}_i \models \mathcal{R}$ and $T_i = \tau_\phi(\mathcal{I}_i)$ for every i, $1 \leq i \leq m$. In addition, for every j, $1 \leq j \leq k$, there is an index ν_j, $1 \leq \nu_j \leq m$, such that $\mathcal{O}_j <^{\phi,\mathcal{R}} T_{\nu_j}$. The interpretations $\mathcal{I}_{\nu_1}, \dots, \mathcal{I}_{\nu_k}$ share the same domain, coincide on the individual names and the rigid concept and role names, and satisfy $\mathcal{I}_{\nu_j} \models (\mathcal{O}_j, \mathcal{R})$ for every j, $1 \leq j \leq k$. This shows that $\mathfrak{P}$ is r-consistent. □

We are now ready to define a deterministic finite automaton that accepts exactly those sequences of partial axiom types $\mathfrak{O} \in \mathfrak{P}^*_{\phi,\mathcal{R}}$ such that $\mathfrak{O}, \psi, \mathcal{R} \models\!\!\approx_\forall \phi$. But first, for the sake of completeness, let us recall the definition of a deterministic finite automaton.

Definition 4.24 (Deterministic finite automaton). *A* deterministic finite automaton *is a tuple $\mathcal{D} = (S, \Sigma, \delta, s_0, E)$ consisting of a finite set of states S, a finite input alphabet Σ, a transition function $\delta : S \times \Sigma \to S$, an initial state $s_0 \in S$, and a set of final states $E \subseteq S$.*

The transition function can be extended to a function $\delta^* : S \times \Sigma^* \to S$ as follows:

- *$\delta^*(s, \varepsilon) := s$ where ε denotes the empty word; and*
- *$\delta^*(s, u\sigma) := \delta(\delta^*(s, u), \sigma)$ where $u \in \Sigma^*$ and $\sigma \in \Sigma$.*

The language $L(\mathcal{D})$ accepted by *$\mathcal{D}$ is defined as as*

$$L(\mathcal{D}) := \{u \in \Sigma^* \mid \delta^*(s_0, u) \in E\}.$$ ◇

As mentioned above, the deterministic finite automaton to be defined is based on the Büchi-automaton $\mathcal{N}^{\mathrm{r}}_{\neg\phi\wedge\psi,\mathcal{R}}$ for the $\mathcal{SHOQ}$-LTL-formula $\neg\phi \wedge \psi$ w.r.t. $\mathcal{R}$ as introduced in Theorem 4.14. Recall that, according to our assumption, all the axioms occurring in ψ already occur in ϕ. Thus, the alphabet of this Büchi-automaton is actually $\mathfrak{T}_{\phi,\mathcal{R}}$ and the second components of the states are r-consistent sets of axiom types for ϕ w.r.t. $\mathcal{R}$, i.e. we have

$$\mathcal{N}^{\mathrm{r}}_{\neg\phi\wedge\psi,\mathcal{R}} = (Q \times \mathfrak{C}_{\phi,\mathcal{R}}, \mathfrak{T}_{\phi,\mathcal{R}}, \Delta, Q_0 \times \{\emptyset\}, F \times \mathfrak{C}_{\phi,\mathcal{R}}).$$

Given a state $(q, \mathfrak{T})$ of $\mathcal{N}^{\mathrm{r}}_{\neg\phi\wedge\psi,\mathcal{R}}$, we denote the Büchi-automaton obtained from this automaton by replacing the set of initial states with $\{(q, \mathfrak{T})\}$ by $\mathcal{N}^{\mathrm{r}}_{\neg\phi\wedge\psi,\mathcal{R}}(q, \mathfrak{T})$. The deterministic finite automaton $\mathcal{D}_{\phi,\psi,\mathcal{R}} = (S, \mathfrak{P}_{\phi,\mathcal{R}}, \delta, s_0, E)$ is defined as follows:

- $S := 2^Q \times \mathfrak{C}^{\mathrm{p}}_{\phi,\mathcal{R}}$;
- $s_0 := (Q_0, \emptyset)$;
- $\delta : S \times \mathfrak{P}_{\phi,\mathcal{R}} \to S$ is defined as follows:
  - if $\mathfrak{P} \cup \{\mathcal{O}\} \notin \mathfrak{C}^{\mathrm{p}}_{\phi,\mathcal{R}}$, then $\delta((P, \mathfrak{P}), \mathcal{O}) := (\emptyset, \emptyset)$;
  - if $\mathfrak{P} \cup \{\mathcal{O}\} \in \mathfrak{C}^{\mathrm{p}}_{\phi,\mathcal{R}}$, then $\delta((P, \mathfrak{P}), \mathcal{O}) := (P', \mathfrak{P} \cup \{\mathcal{O}\})$ where

$$P' := \bigcup_{q\in P} \{q' \in Q \mid \text{there is } ((q, \mathfrak{T}), T, (q', \mathfrak{T} \cup \{T\})) \in \Delta \text{ such that } \mathcal{O} <^{\phi,\mathcal{R}} T,\ \mathfrak{P} \prec^{\phi,\mathcal{R}} \mathfrak{T}, \text{ and } L_\omega(\mathcal{N}^{\mathrm{r}}_{\neg\phi\wedge\psi,\mathcal{R}}(q', \mathfrak{T} \cup \{T\})) \neq \emptyset\};$$

- $E := \{\emptyset\} \times \mathfrak{C}^{\mathrm{p}}_{\phi,\mathcal{R}}$.

Final states are those whose first component is the empty set. Note that these states reproduce themselves: states whose first component is the empty set have only successor states for which this is again the case. There are two possible reasons for reaching such a state with letter $\mathcal{O}$ from a state $(P, \mathfrak{P})$ whose first component P is non-empty. Either the set $\mathfrak{P} \cup \{\mathcal{O}\}$ is not r-consistent, or there are no states $q' \in Q$ satisfying the conditions in the definition of P'.

The following lemma states that this deterministic finite automaton behaves as intended.

Lemma 4.25. *For every finite sequence of partial axiom types $\mathfrak{O} \in \mathfrak{P}^*_{\phi,\mathcal{R}}$, we have $\mathfrak{O}, \psi, \mathcal{R} \mid\!\approx_\forall \phi$ iff $\mathfrak{O} \in L(\mathcal{D}_{\phi,\psi,\mathcal{R}})$.*

Proof. For the 'if' direction, assume to the contrary that $\mathfrak{O} = \mathcal{O}_0\mathcal{O}_1 \ldots \mathcal{O}_t \in L(\mathcal{D}_{\phi,\psi,\mathcal{R}})$ and $\mathfrak{O}, \psi, \mathcal{R} \mid\!\not\approx_\forall \phi$. Then we have $\mathfrak{O}, \psi, \mathcal{R} \mid\!\approx_\exists \neg\phi$, i.e. there is a DL-LTL-structure $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$ that extends $\mathfrak{O}$ w.r.t. ψ and $\mathcal{R}$, and is a model of $\neg\phi$ w.r.t. $\mathcal{R}$. This means that $\mathfrak{I}$ is a model of $\neg\phi \wedge \psi$ w.r.t. $\mathcal{R}$, and $\mathcal{I}_i \models \mathcal{O}_i$ for every i, $0 \leq i \leq t$. Thus, $\tau_\phi(\mathfrak{I}) \in L_\omega(\neg\phi \wedge \psi, \mathcal{R})$, and since $\mathcal{N}^{\mathrm{r}}_{\neg\phi\wedge\psi,\mathcal{R}}$ is a Büchi-automaton for $\neg\phi\wedge\psi$ w.r.t. $\mathcal{R}$, we have $\tau_\phi(\mathfrak{I}) \in L_\omega(\mathcal{N}^{\mathrm{r}}_{\neg\phi\wedge\psi,\mathcal{R}})$. This means that there is an accepting run $(q_0, \mathfrak{T}_0)(q_1, \mathfrak{T}_1)\ldots$ of $\mathcal{N}^{\mathrm{r}}_{\neg\phi\wedge\psi,\mathcal{R}}$ on $\tau_\phi(\mathfrak{I})$. In particular, this yields $L_\omega(\mathcal{N}^{\mathrm{r}}_{\neg\phi\wedge\psi,\mathcal{R}}(q_i, \mathfrak{T}_i)) \neq \emptyset$ for every $i \geq 0$.

Moreover, we have by the construction of $\mathcal{N}^{\mathrm{r}}_{\neg\phi\wedge\psi,\mathcal{R}}$ that $\mathfrak{T}_i = \{\tau_\phi(\mathcal{I}_j) \mid 0 \leq j < i\}$ for every $i \geq 0$. We define $\mathfrak{P}_i := \{\mathcal{O}_j \mid 0 \leq j < i\}$ for every i, $0 \leq i \leq t + 1$. Note that we have $\mathcal{O}_i <^{\phi,\mathcal{R}} \tau_\phi(\mathcal{I}_i)$ for every i, $0 \leq i \leq t + 1$. Hence, $\mathfrak{P}_i \prec^{\phi,\mathcal{R}} \mathfrak{T}_i$ holds for every i, $0 \leq i \leq t + 1$. By the definition of $\mathcal{N}^{\mathrm{r}}_{\neg\phi\wedge\psi,\mathcal{R}}$, the sets $\mathfrak{T}_i$ are r-consistent, and thus Lemma 4.23 yields that $\mathfrak{P}_i$ is r-consistent for every i, $0 \leq i \leq t + 1$. Thus, we have $\delta^*(s_0, \mathcal{O}_0 \ldots \mathcal{O}_i) = (P_{i+1}, \mathfrak{P}_{i+1})$ with

$q_{i+1} \in P_{i+1}$ for every i, $0 \leq i \leq t$. In particular, $\delta^*(s_0, \mathfrak{O}) = (P_{t+1}, \mathfrak{P}_{t+1})$ with $q_{t+1} \in P_{t+1}$, which shows that $P_{t+1} \neq \emptyset$. Consequently, $(P_{t+1}, \mathfrak{P}_{t+1}) \notin E$, which is a contradiction to the assumption that $\mathfrak{O} \in L(\mathcal{D}_{\phi,\psi,\mathcal{R}})$.

For the 'only if' direction, assume to the contrary that $\mathfrak{O} = \mathcal{O}_0\mathcal{O}_1 \ldots \mathcal{O}_t \notin L(\mathcal{D}_{\phi,\psi,\mathcal{R}})$ and $\mathfrak{O}, \psi, \mathcal{R} \mathrel{|\!\!\approx_\forall} \phi$, i.e. every DL-LTL-structure $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$ that extends $\mathfrak{O}$ w.r.t. ψ and $\mathcal{R}$ is a model of ϕ w.r.t. $\mathcal{R}$.

The first assumption implies that $\delta^*(s_0, \mathfrak{O}) \notin E$, i.e. $\delta^*(s_0, \mathfrak{O}) = (P_{t+1}, \mathfrak{P}_{t+1}) \in 2^Q \times \mathfrak{C}^{\mathrm{p}}_{\phi,\mathcal{R}}$ with $P_{t+1} \neq \emptyset$. This yields intermediate states $(P_i, \mathfrak{P}_i) \in 2^Q \times \mathfrak{C}^{\mathrm{p}}_{\phi,\mathcal{R}}$, $0 \leq i \leq t$, such that $P_0 = Q_0$, $\mathfrak{P}_0 = \emptyset$, and $\delta^*(s_0, \mathcal{O}_0 \ldots \mathcal{O}_i) = (P_{i+1}, \mathfrak{P}_{i+1}) \in 2^Q \times \mathfrak{C}^{\mathrm{p}}_{\phi,\mathcal{R}}$ with $\mathfrak{P}_{i+1} = \mathfrak{P}_i \cup \{\mathcal{O}_i\}$ and $P_{i+1} \neq \emptyset$ for every i, $0 \leq i \leq t$. Moreover, we have that there are for every i, $0 \leq i \leq t$, a state $q_i \in P_i$, an axiom type $T_i \in \mathfrak{T}_{\phi,\mathcal{R}}$, and an r-consistent set of axiom types $\mathfrak{T}_i \in \mathfrak{C}_{\phi,\mathcal{R}}$ such that $((q_i, \mathfrak{T}_i), T_i, (q_{i+1}, \mathfrak{T}_{i+1})) \in \Delta$, $\mathfrak{T}_{i+1} = \mathfrak{T}_i \cup \{T_i\}$, $\mathcal{O}_i <^{\phi,\mathcal{R}} T_i$, $\mathfrak{P}_i \prec^{\phi,\mathcal{R}} \mathfrak{T}_i$, and $L_\omega(\mathcal{N}^{\mathrm{r}}_{\neg\phi\wedge\psi,\mathcal{R}}(q_{i+1}, \mathfrak{T}_{i+1})) \neq \emptyset$. Note that $q_0 \in Q_0$ since $q_0 \in P_0$ and $P_0 = Q_0$.

We define $\mathfrak{T}'_i := \{T_j \mid 0 \leq j < i\}$ for every i, $0 \leq i \leq t+1$. Obviously, we then have $\mathfrak{T}'_i \subseteq \mathfrak{T}_i$ for every i, $0 \leq i \leq t+1$. Since every subset of an r-consistent set of axiom types is again r-consistent, this shows $\mathfrak{T}'_i \in \mathfrak{C}_{\phi,\mathcal{R}}$ for every i, $0 \leq i \leq t+1$. Moreover, since $\mathfrak{P}_i = \{\mathcal{O}_j \mid 0 \leq j < i\}$ for every i, $0 \leq i \leq t+1$, the fact that $\mathcal{O}_i <^{\phi} T_i$ for every i, $0 \leq i \leq t+1$, implies $\mathfrak{P}_i \prec^{\phi,\mathcal{R}} \mathfrak{T}'_i$ for every i, $0 \leq i \leq t+1$. In addition, we have $((q_i, \mathfrak{T}'_i), T_i, (q_{i+1}, \mathfrak{T}'_{i+1})) \in \Delta$ for every i, $0 \leq i \leq t$.

Since $L_\omega(\mathcal{N}^{\mathrm{r}}_{\neg\phi\wedge\psi,\mathcal{R}}(q_{t+1}, \mathfrak{T}_{t+1})) \neq \emptyset$, there is a ω-word $T \in \mathfrak{T}^{\omega}_{\phi,\mathcal{R}}$ such that there is an accepting run of $\mathcal{N}^{\mathrm{r}}_{\neg\phi\wedge\psi,\mathcal{R}}(q_{t+1}, \mathfrak{T}_{t+1})$ on T. Using similar arguments as above, we can transform this run into an accepting run of $\mathcal{N}^{\mathrm{r}}_{\neg\phi\wedge\psi,\mathcal{R}}(q_{t+1}, \mathfrak{T}'_{t+1})$ on T. Hence, we have that $T \in L_\omega(\mathcal{N}^{\mathrm{r}}_{\neg\phi\wedge\psi,\mathcal{R}}(q_{t+1}, \mathfrak{T}'_{t+1}))$. Overall, we obtain that the ω-word $T_0T_1 \ldots T_t \cdot T$ is in $L_\omega(\mathcal{N}^{\mathrm{r}}_{\neg\phi\wedge\psi,\mathcal{R}})$. Since $\mathcal{N}^{\mathrm{r}}_{\neg\phi\wedge\psi,\mathcal{R}}$ is a Büchi-automaton for $\neg\phi \wedge \psi$ w.r.t. $\mathcal{R}$, this shows that there exists a DL-LTL-structure $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$ such that $\tau_\phi(\mathfrak{I}) = T_0T_1 \ldots T_t \cdot T$ and $\mathfrak{I}$ is a model of $\neg\phi \wedge \psi$ w.r.t. $\mathcal{R}$.

For every i, $0 \leq i \leq t$, we have $\mathcal{O}_i <^{\phi,\mathcal{R}} \tau_\phi(\mathcal{I}_i)$ since $\tau_\phi(\mathcal{I}_i) = T_i$. This yields $\mathcal{I}_i \models \mathcal{O}_i$ for every i, $0 \leq i \leq t$. Since $\mathfrak{I}$ is a model of ψ w.r.t. $\mathcal{R}$, we obtain that $\mathfrak{I}$ extends $\mathfrak{O}$ w.r.t. ψ and $\mathcal{R}$. Hence, there is a DL-LTL-structure, namely $\mathfrak{I}$, extending $\mathfrak{O}$ w.r.t. ψ and $\mathcal{R}$ that is a model of $\neg\phi$ w.r.t. $\mathcal{R}$, which contradicts our assumption that $\mathfrak{O}, \psi, \mathcal{R} \mathrel{|\!\!\approx_\forall} \phi$. □

It remains to analyse the complexity of the construction of the deterministic finite automaton $\mathcal{D}_{\phi,\psi,\mathcal{R}}$. The size of $\mathcal{D}_{\phi,\psi,\mathcal{R}}$ is doubly exponential in the size of ϕ, ψ, and $\mathcal{R}$. This is due to the fact that the size of Q may be exponential and the fact that the set $\mathfrak{C}^{\mathrm{p}}_{\phi,\mathcal{R}}$ of all r-consistent partial axiom types for ϕ w.r.t. $\mathcal{R}$ may contain doubly exponentially many elements since these sets are subsets of the exponentially large set $\mathfrak{P}_{\phi,\mathcal{R}}$ of all partial axiom types for ϕ w.r.t. $\mathcal{R}$. Each element of $\mathfrak{C}^{\mathrm{p}}_{\phi,\mathcal{R}}$ may be of exponential size.

Next, we show that $\mathcal{D}_{\phi,\psi,\mathcal{R}}$ can be constructed in doubly exponential time. In addition to constructing the Büchi-automaton $\mathcal{N}^{\mathrm{r}}_{\neg\phi\wedge\psi,\mathcal{R}}$, we must also compute the set $\mathfrak{C}^{\mathrm{p}}_{\phi,\mathcal{R}}$. As shown in Section 4.2.2, the Büchi-automaton $\mathcal{N}^{\mathrm{r}}_{\neg\phi\wedge\psi,\mathcal{R}}$, and thus also the set $\mathfrak{C}_{\phi,\mathcal{R}}$, can be constructed in time doubly exponential in the size of ϕ, ψ, and $\mathcal{R}$. To compute $\mathfrak{C}^{\mathrm{p}}_{\phi,\mathcal{R}}$, we use Lemma 4.23, which yields

$$\mathfrak{C}^{\mathrm{p}}_{\phi,\mathcal{R}} = \big\{\mathfrak{P} \subseteq \mathfrak{P}_{\phi,\mathcal{R}} \mid \mathfrak{P} \prec^{\phi,\mathcal{R}} \mathfrak{T} \text{ for some } \mathfrak{T} \in \mathfrak{C}_{\phi,\mathcal{R}}\big\}.$$

We consider all sets of partial axiom types for ϕ w.r.t. $\mathcal{R}$. There are doubly exponentially many such sets, each of size at most exponential in the size of ϕ and $\mathcal{R}$. For each such set $\mathfrak{P} = \{\mathcal{O}_1, \ldots, \mathcal{O}_k\}$, we need to check whether there is a set $\mathfrak{T} = \{T_1, \ldots, T_m\} \in \mathfrak{C}_{\phi,\mathcal{R}}$ such that $\mathfrak{P} \prec^{\phi,\mathcal{R}} \mathfrak{T}$. Since $\mathfrak{C}_{\phi,\mathcal{R}}$ is of doubly exponential size, there are at most doubly exponentially many such tests for each $\mathfrak{P}$. The test $\mathfrak{P} \prec^{\phi,\mathcal{R}} \mathfrak{T}$ itself amounts to checking for each $\mathcal{O}_i$, $1 \leq i \leq k$, whether there is a T_j, $1 \leq j \leq m$, such that $\mathsf{Pos}(\mathcal{O}_i) \subseteq T_j$ and $\mathsf{Neg}(\mathcal{O}_i) \cap T_j = \emptyset$, which can be done in exponential time since both k and m are at most exponential in the size of ϕ and $\mathcal{R}$. Overall, we can thus compute $\mathfrak{C}^{\mathrm{p}}_{\phi,\mathcal{R}}$ is doubly exponential time. Using these arguments, the fact that $\mathcal{N}^{\mathrm{r}}_{\neg\phi\wedge\psi,\mathcal{R}}$ can be constructed in doubly exponential time, and the fact that the emptiness problem for Büchi-automata can be solved in time polynomial in the size of the Büchi-automaton [VW94], it is easy to see that the transition function δ and the set of final states E can be computed in doubly exponential time. Overall, we have shown that $\mathcal{D}_{\phi,\psi,\mathcal{R}}$ can be constructed in time doubly exponential in the size of ϕ, ψ, and $\mathcal{R}$.

4.3.3 The Monitor Construction

Given the construction of the deterministic finite automaton of the previous section, it is now a simple exercise to construct the monitor for ϕ w.r.t. ψ and $\mathcal{R}$. Such a monitor is obtained by first constructing the auxiliary deterministic finite automata $\mathcal{D}_{\phi,\psi,\mathcal{R}}$ and $\mathcal{D}_{\neg\phi,\psi,\mathcal{R}}$, and then building the product of these two automata. The output of the monitor is determined by the final states of the auxiliary automata.

Theorem 4.26. *Let $\mathcal{R}$ be an RBox, and let ϕ and ψ be $\mathcal{SHOQ}$-LTL-formulas w.r.t. $\mathcal{R}$. If $\mathcal{D}_{\phi,\psi,\mathcal{R}} = (S, \mathfrak{P}_{\phi,\mathcal{R}}, \delta, s_0, E)$ and $\mathcal{D}_{\neg\phi,\psi,\mathcal{R}} = (S', \mathfrak{P}_{\phi,\mathcal{R}}, \delta', s_0', E')$ are the deterministic finite automata introduced in Section 4.3.2, then $\mathcal{M}_{\phi,\psi,\mathcal{R}} := (S \times S', \mathfrak{P}_{\phi,\mathcal{R}}, \widehat{\delta}, (s_0, s_0'), \{\top, \bot, ?, \lightning\}, \lambda)$ with $\widehat{\delta}((s,s'), \mathcal{O}) := (\delta(s,\mathcal{O}), \delta'(s',\mathcal{O}))$ and*

$$\lambda((s,s')) := \begin{cases} \top & \text{if } s \in E \text{ and } s' \notin E'; \\ \bot & \text{if } s \notin E \text{ and } s' \in E'; \\ ? & \text{if } s \notin E \text{ and } s' \notin E'; \text{ and} \\ \lightning & \text{if } s \in E \text{ and } s' \in E'. \end{cases}$$

is a monitor for ϕ w.r.t. ψ and $\mathcal{R}$.

Proof. We have to prove that for every $\mathfrak{O} \in \mathfrak{P}^*_{\phi,\mathcal{R}}$, we have $\lambda^*(\mathfrak{O}) = \mathsf{m}_{\phi,\psi,\mathcal{R}}(\mathfrak{O})$. This is an immediate consequence of the definition of the monitoring function (Definition 4.19) and the following facts:

- $\widehat{\delta}^*((s_0, s_0'), \mathfrak{O}) = (\delta^*(s_0, \mathfrak{O}), (\delta')^*(s_0', \mathfrak{O}))$ for every $\mathfrak{O} \in \mathfrak{P}^*_{\phi,\mathcal{R}}$;
- $\delta^*(s_0, \mathfrak{O}) \in E$ iff $\mathfrak{O}, \psi, \mathcal{R} \mathrel{|\!\!\approx_\forall} \phi$ for every $\mathfrak{O} \in \mathfrak{P}^*_{\phi,\mathcal{R}}$ (by Lemma 4.25); and
- $(\delta')^*(s_0', \mathfrak{O}) \in E'$ iff $\mathfrak{O}, \psi, \mathcal{R} \mathrel{|\!\!\approx_\forall} \neg\phi$ for every $\mathfrak{O} \in \mathfrak{P}^*_{\phi,\mathcal{R}}$ (by Lemma 4.25).

To show the theorem formally, take any $\mathfrak{O} \in \mathfrak{P}^*_{\phi,\mathcal{R}}$. Using the above facts, we have:

$$\lambda^*(\mathfrak{O}) = \top \quad \text{iff} \quad \lambda(\widehat{\delta}^*((s_0, s_0'), \mathfrak{O})) = \top$$

$$\begin{aligned}
&\text{iff} \quad \widehat{\delta}^*((s_0,s_0'),\mathfrak{D}) = (s,s') \text{ with } s \in E \text{ and } s' \notin E'\\
&\text{iff} \quad \delta^*(s_0,\mathfrak{D}) \in E \text{ and } (\delta')^*(s_0',\mathfrak{D}) \notin E'\\
&\text{iff} \quad \mathfrak{D} \in L(\mathcal{D}_{\phi,\psi,\mathcal{R}}) \text{ and } \mathfrak{D} \notin L(\mathcal{D}_{\neg\phi,\psi,\mathcal{R}})\\
&\text{iff} \quad \mathfrak{D},\psi,\mathcal{R} \mathrel{|\!\approx}_{\forall} \phi \text{ and } \mathfrak{D},\psi,\mathcal{R} \mathrel{|\!\not\approx}_{\forall} \neg\phi\\
&\text{iff} \quad \mathsf{m}_{\phi,\psi,\mathcal{R}}(\mathfrak{D}) = \top.
\end{aligned}$$

Moreover, we have:

$$\begin{aligned}
\lambda^*(\mathfrak{D}) = \bot \quad &\text{iff} \quad \lambda(\widehat{\delta}^*((s_0,s_0'),\mathfrak{D})) = \bot\\
&\text{iff} \quad \widehat{\delta}^*((s_0,s_0'),\mathfrak{D}) = (s,s') \text{ with } s \notin E \text{ and } s' \in E'\\
&\text{iff} \quad \delta^*(s_0,\mathfrak{D}) \notin E \text{ and } (\delta')^*(s_0',\mathfrak{D}) \in E'\\
&\text{iff} \quad \mathfrak{D} \notin L(\mathcal{D}_{\phi,\psi,\mathcal{R}}) \text{ and } \mathfrak{D} \in L(\mathcal{D}_{\neg\phi,\psi,\mathcal{R}})\\
&\text{iff} \quad \mathfrak{D},\psi,\mathcal{R} \mathrel{|\!\not\approx}_{\forall} \phi \text{ and } \mathfrak{D},\psi,\mathcal{R} \mathrel{|\!\approx}_{\forall} \neg\phi\\
&\text{iff} \quad \mathsf{m}_{\phi,\psi,\mathcal{R}}(\mathfrak{D}) = \bot,
\end{aligned}$$

and also:

$$\begin{aligned}
\lambda^*(\mathfrak{D}) = {?} \quad &\text{iff} \quad \lambda(\widehat{\delta}^*((s_0,s_0'),\mathfrak{D})) = {?}\\
&\text{iff} \quad \widehat{\delta}^*((s_0,s_0'),\mathfrak{D}) = (s,s') \text{ with } s \notin E \text{ and } s' \notin E'\\
&\text{iff} \quad \delta^*(s_0,\mathfrak{D}) \notin E \text{ and } (\delta')^*(s_0',\mathfrak{D}) \notin E'\\
&\text{iff} \quad \mathfrak{D} \notin L(\mathcal{D}_{\phi,\psi,\mathcal{R}}) \text{ and } \mathfrak{D} \notin L(\mathcal{D}_{\neg\phi,\psi,\mathcal{R}})\\
&\text{iff} \quad \mathfrak{D},\psi,\mathcal{R} \mathrel{|\!\not\approx}_{\forall} \phi \text{ and } \mathfrak{D},\psi,\mathcal{R} \mathrel{|\!\not\approx}_{\forall} \neg\phi\\
&\text{iff} \quad \mathsf{m}_{\phi,\psi,\mathcal{R}}(\mathfrak{D}) = {?}.
\end{aligned}$$

This shows that $\mathcal{M}_{\phi,\psi,\mathcal{R}}$ is indeed a monitor for ϕ w.r.t. ψ and $\mathcal{R}$. □

It remains to analyse the complexity of the construction. As shown in Section 4.3.2, the size of the auxiliary deterministic finite automata $\mathcal{D}_{\phi,\psi,\mathcal{R}}$ and $\mathcal{D}_{\neg\phi,\psi,\mathcal{R}}$ is doubly exponential in the size of ϕ, ψ, and $\mathcal{R}$. Furthermore, they can be constructed in doubly exponential time. Hence, the size of $\mathcal{M}_{\phi,\psi,\mathcal{R}}$ is also doubly exponential in the size of ϕ, ψ, and $\mathcal{R}$, and it can be constructed in doubly exponential time.

This doubly exponential blow-up in the construction of the monitor cannot be avoided, since Theorem 4.3 yields that such a blow-up is unavoidable even for propositional LTL.

4.4 The Complexity of Deciding Liveness and Monitorability in $\mathcal{SHOQ}$-LTL

In this section, we extend the definitions and results about liveness and monitorability from propositional LTL to $\mathcal{SHOQ}$-LTL. In Section 4.4.1, we consider the simpler-looking problem of liveness and in Section 4.4.2, we consider monitorability.

4.4.1 Deciding Liveness

First, we extend the notion of liveness from propositional LTL (see Definition 4.5) to the temporalised description logic $\mathcal{SHOQ}$-LTL and the presence of background knowledge.

Definition 4.27 (Liveness). *Let $\mathcal{R}$ be an RBox, and ϕ and ψ be $\mathcal{SHOQ}$-LTL-formulas w.r.t. $\mathcal{R}$. We say that ϕ* expresses a liveness property w.r.t. ψ and $\mathcal{R}$ *if for every finite sequence of partial axiom types $\mathfrak{D} \in \mathfrak{P}^*_{\phi,\mathcal{R}}$ that has an extension w.r.t. ψ and $\mathcal{R}$, we have $\mathfrak{D}, \psi, \mathcal{R} \mathrel{|}\!\!\approx_\exists \phi$.* ◇

Note that, in this definition, we restrict ourselves to the finite sequences of partial axiom types that have an extension w.r.t. ψ and $\mathcal{R}$. In fact, these are the sequences that we expect to see in practice since we assume that the system satisfies ψ, $\mathcal{R}$, and respects rigid names.

As in the propositional case, liveness of ϕ w.r.t. ψ and $\mathcal{R}$ can be expressed using the monitoring function.

Lemma 4.28. *Let $\mathcal{R}$ be an RBox, and ϕ and ψ be $\mathcal{SHOQ}$-LTL-formulas w.r.t. $\mathcal{R}$. Then ϕ expresses a liveness property w.r.t. ψ and $\mathcal{R}$ iff $\mathsf{m}_{\phi,\psi,\mathcal{R}}(\mathfrak{D}) \neq \bot$ for every $\mathfrak{D} \in \mathfrak{P}^*_{\phi,\mathcal{R}}$.*

Proof. For the 'only if' direction, assume that ϕ expresses a liveness property w.r.t. ψ and $\mathcal{R}$, and consider $\mathfrak{D} \in \mathfrak{P}^*_{\phi,\mathcal{R}}$. If $\mathfrak{D}$ does not have an extension w.r.t. ψ and $\mathcal{R}$, then $\mathsf{m}_{\phi,\psi,\mathcal{R}}(\mathfrak{D}) = \lightning \neq \bot$. Otherwise, the fact that ϕ expresses a liveness property w.r.t. ψ and $\mathcal{R}$ implies that $\mathfrak{D}, \psi, \mathcal{R} \mathrel{|}\!\!\approx_\exists \phi$. Consequently, we have $\mathfrak{D}, \psi, \mathcal{R} \mathrel{|}\!\!\not\approx_\forall \neg\phi$, which yields $\mathsf{m}_{\phi,\psi,\mathcal{R}}(\mathfrak{D}) \neq \bot$.

For the 'if' direction, assume that $\mathsf{m}_{\phi,\psi,\mathcal{R}}(\mathfrak{D}) \neq \bot$ for every $\mathfrak{D} \in \mathfrak{P}^*_{\phi,\mathcal{R}}$. Consider a finite sequence of partial axiom types $\mathfrak{D} \in \mathfrak{P}^*_{\phi,\mathcal{R}}$ that has an extension w.r.t. ψ and $\mathcal{R}$. The existence of this extension implies that $\mathsf{m}_{\phi,\psi,\mathcal{R}}(\mathfrak{D}) \neq \lightning$. Thus, we know that $\mathsf{m}_{\phi,\psi,\mathcal{R}}(\mathfrak{D}) \in \{\top, ?\}$. In both cases, $\mathfrak{D}, \psi, \mathcal{R} \mathrel{|}\!\!\not\approx_\forall \neg\phi$ holds, which is equivalent to $\mathfrak{D}, \psi, \mathcal{R} \mathrel{|}\!\!\approx_\exists \phi$. □

Consequently, given a monitor for ϕ w.r.t. ψ and $\mathcal{R}$, liveness of ϕ w.r.t. ψ and $\mathcal{R}$ can be tested by checking reachability in the monitor, which yields an upper bound of 2ExpTime. The lower bound can be obtained by a reduction of the *un*satisfiability problem in $\mathcal{ALC}$-LTL [BGL12].

Lemma 4.29. *The problem of deciding whether a $\mathcal{SHOQ}$-LTL-formula ϕ expresses a liveness property w.r.t. a $\mathcal{SHOQ}$-LTL-formula ψ and an RBox $\mathcal{R}$ is as hard as the unsatisfiability problem in $\mathcal{ALC}$-LTL.*

Proof. Let ψ be an $\mathcal{ALC}$-LTL-formula (and thus a $\mathcal{SHOQ}$-LTL-formula w.r.t. the empty RBox $\mathcal{R} := \emptyset$). We prove that ψ is unsatisfiable iff the $\mathcal{SHOQ}$-LTL-formula $\phi := \mathsf{false}$ expresses a liveness property w.r.t. ψ and $\mathcal{R}$.

In fact, if ψ is unsatisfiable, then there is *no* sequence $\mathfrak{D} \in \mathfrak{P}^*_{\phi,\mathcal{R}}$ such that $\mathfrak{D}$ has extension w.r.t. ψ and $\mathcal{R}$. Consequently, the condition in the definition of liveness quantifies over the empty set of sequences, and is thus trivially true.

Conversely, if ψ is satisfiable, then there is some $\mathfrak{D} \in \mathfrak{P}^*_{\phi,\mathcal{R}}$ (e.g. the empty sequence) that has an extension w.r.t. ψ and $\mathcal{R}$. But then $\mathfrak{D}, \psi, \mathcal{R} \mathrel{|}\!\!\not\approx_\exists \phi$ since ϕ is unsatisfiable. Hence, ϕ does not express a liveness property w.r.t. ψ and $\mathcal{R}$. □

Due to the complexity results for the satisfiability problem in $\mathcal{ALC}$-LTL (see [BGL12]), we obtain from these lemmas the following theorem.

Theorem 4.30. *The problem of deciding whether a $\mathcal{SHOQ}$-LTL-formula ϕ expresses a liveness property w.r.t. a $\mathcal{SHOQ}$-LTL-formula ψ and an RBox $\mathcal{R}$ is*

- EXPTIME-*hard and in* 2EXPTIME *if* $\mathsf{N_{RC}} = \mathsf{N_{RR}} = \emptyset$*;*
- CO-NEXPTIME-*hard and in* 2EXPTIME *if* $\mathsf{N_{RC}} \neq \emptyset$ *and* $\mathsf{N_{RR}} = \emptyset$*; and*
- 2EXPTIME-*complete if* $\mathsf{N_{RC}} \neq \emptyset$ *and* $\mathsf{N_{RR}} \neq \emptyset$*.*

Proof. Regarding the upper bounds, Lemma 4.28 implies that ϕ expresses a liveness property w.r.t. ψ and $\mathcal{R}$ iff in the monitor $\mathcal{M}_{\phi,\psi,\mathcal{R}}$ no state with output $\bot$ is reachable from the initial state. Since this monitor is of doubly exponential size and reachability can be decided in linear time in the size of the automaton, this yields the required upper bounds of 2EXPTIME.

The lower bounds follow immediately from Lemma 4.29 and the complexity results for the satisfiability problem in $\mathcal{ALC}$-LTL [BGL12]. Indeed, this problem is EXPTIME-complete if $\mathsf{N_{RC}} = \mathsf{N_{RR}} = \emptyset$, NEXPTIME-complete if $\mathsf{N_{RC}} \neq \emptyset$ and $\mathsf{N_{RR}} = \emptyset$, and 2EXPTIME-complete if $\mathsf{N_{RC}} \neq \emptyset$ and $\mathsf{N_{RR}} \neq \emptyset$. Since both EXPTIME and 2EXPTIME are closed under complement, we obtain the complexity lower bounds of our theorem. □

Note that only in the case $\mathsf{N_{RC}} \neq \emptyset$ and $\mathsf{N_{RR}} \neq \emptyset$, we have a tight complexity result. Recall, however, that the exact complexity for this problem is not even known for propositional LTL.

Also, our hardness proof (Lemma 4.29) strongly depends on the presence of background knowledge. Without background knowledge (i.e. in the case where $\psi = \mathsf{true}$ and $\mathcal{R} = \emptyset$), we can only show an EXPTIME-hardness result by a reduction of the satisfiability problem in $\mathcal{ALC}$-LTL (without rigid names).

Theorem 4.31. *The problem of deciding whether a $\mathcal{SHOQ}$-LTL-formula ϕ expresses a liveness property w.r.t. the $\mathcal{SHOQ}$-LTL-formula* true *and the empty RBox is* EXPTIME-*hard.*

Proof. Consider an $\mathcal{ALC}$-LTL-formula ϕ and assume $\mathsf{N_{RC}} = \mathsf{N_{RR}} = \emptyset$. We prove that ϕ is satisfiable iff $\Diamond\phi$ expresses a liveness property w.r.t. true and the empty RBox $\mathcal{R} := \emptyset$. Since the satisfiability problem in $\mathcal{ALC}$-LTL is EXPTIME-complete if $\mathsf{N_{RC}} = \mathsf{N_{RR}} = \emptyset$ (see [BGL12]), this shows EXPTIME-hardness of liveness w.r.t. true and the empty RBox.

If ϕ is unsatisfiable, then obviously $\Diamond\phi$ is unsatisfiable as well, and thus no sequence of partial axiom types can be extended to a model of $\Diamond\phi$. In addition, there is a sequence of partial axiom types (e.g. the empty sequence) that can be extended to a model of true and $\mathcal{R}$. Consequently, $\Diamond\phi$ does not express a liveness property w.r.t. true and $\mathcal{R}$.

Conversely, assume that ϕ is satisfiable, and let $\mathfrak{O} = \mathcal{O}_0\mathcal{O}_1\ldots\mathcal{O}_{t-1} \in \mathfrak{P}^*_{\phi,\mathcal{R}}$ be a sequence of partial axiom types. Satisfiability of ϕ yields a model $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$ of ϕ. In addition, since partial axiom types are by definition consistent, there are interpretations $\mathcal{I}'_i$, $0 \leq i \leq t-1$, such that $\mathcal{I}'_i \models \mathcal{O}_i$. It is easy to see that the DL-LTL-structure

$$\mathfrak{J} := (\mathcal{J}_i)_{i\geq 0} \text{ with } \mathcal{J}_i = \mathcal{I}'_i,\ 0 \leq i \leq t-1, \text{ and } \mathcal{J}_{i+t} = \mathcal{I}_i,\ i \geq 0,$$

is a model of $\Diamond\phi$ that extends $\mathfrak{O}$ w.r.t. true and $\mathcal{R}$, i.e. $\mathfrak{O}, \mathsf{true}, \mathcal{R} \mathrel{|\!\!\approx_{\exists}} \phi$. This shows that $\Diamond\phi$ expresses a liveness property w.r.t. true and $\mathcal{R}$. □

Unfortunately, the proof of this theorem does not go through in the presence of rigid names. In fact, the DL-LTL-structure $\mathfrak{J}$ constructed there need not respect rigid names.

4.4.2 Deciding Monitorability

We first extend the notion of monitorability from propositional LTL (see Definition 4.4) to the temporalised description logic $\mathcal{SHOQ}$-LTL and the presence of background knowledge.

Definition 4.32 (Monitorability). *Let $\mathcal{R}$ be an RBox, let ϕ and ψ be $\mathcal{SHOQ}$-LTL-formulas w.r.t. $\mathcal{R}$, and let $\mathfrak{O} \in \mathfrak{P}^*_{\phi,\mathcal{R}}$. We say that ϕ is* $\mathfrak{O}$-monitorable w.r.t. ψ and $\mathcal{R}$ *if there is a finite word $\mathfrak{O}' \in \mathfrak{P}^*_{\phi,\mathcal{R}}$ such that $\mathsf{m}_{\phi,\psi,\mathcal{R}}(\mathfrak{O} \cdot \mathfrak{O}') \in \{\top, \bot\}$. Moreover, we call ϕ* monitorable w.r.t. ψ and $\mathcal{R}$ *if it is $\mathfrak{O}$-monitorable for every finite sequence of partial axiom types $\mathfrak{O} \in \mathfrak{P}^*_{\phi,\mathcal{R}}$ that has an extension w.r.t. ψ and $\mathcal{R}$.* ◇

Monitorability can thus be expressed using the monitoring function as follows: ϕ is monitorable w.r.t. ψ and $\mathcal{R}$ iff for every finite sequence of partial axiom types $\mathfrak{O} \in \mathfrak{P}^*_{\phi,\mathcal{R}}$ with $\mathsf{m}_{\phi,\psi,\mathcal{R}}(\mathfrak{O}) \neq \lightning$, there exists a finite sequence of partial axiom types $\mathfrak{O}' \in \mathfrak{P}^*_{\phi,\mathcal{R}}$ satisfying $\mathsf{m}_{\phi,\psi,\mathcal{R}}(\mathfrak{O} \cdot \mathfrak{O}') \in \{\top, \bot\}$. This can again be checked using reachability tests in the monitor.

Lemma 4.33. *The problem of deciding monitorability of a $\mathcal{SHOQ}$-LTL-formula ϕ w.r.t. a $\mathcal{SHOQ}$-LTL-formula ψ and an RBox $\mathcal{R}$ is in* 2ExpTime.

Proof. To decide monitorability of ϕ w.r.t. ψ and $\mathcal{R}$, we construct the monitor $\mathcal{M}_{\phi,\psi,\mathcal{R}}$. In this monitor, we compute all the states with output different from $\lightning$ that are reachable from the initial state. For each of these states, we then check whether a state with output $\top$ or $\bot$ is reachable. If this is the case, then ϕ is monitorable w.r.t. ψ and $\mathcal{R}$. Otherwise, i.e. if there is a state reachable from the initial state such that every state reachable from it has output ? or $\lightning$, then ϕ is not monitorable w.r.t. ψ and $\mathcal{R}$.

Since the monitor can be constructed in doubly exponential time and each of the doubly exponentially many reachability tests requires at most doubly exponential time, this yields a 2ExpTime procedure for deciding monitorability. □

For the lower bound, we again reduce the *un*satisfiability problem in $\mathcal{ALC}$-LTL. For monitorability, such a reduction is possible even for the case without background knowledge.

Lemma 4.34. *The problem of deciding monitorability of a $\mathcal{SHOQ}$-LTL-formula ϕ w.r.t.* true *and the empty RBox is as hard as the unsatisfiability problem in $\mathcal{ALC}$-LTL.*

Proof. Note that the lower bounds of the satisfiability problem in $\mathcal{ALC}$-LTL hold also for $\mathcal{ALC}$-LTL-formulas without past operators [BGL12]. Thus, let ψ be an $\mathcal{ALC}$-LTL-formula without past operators. We define the $\mathcal{SHOQ}$-LTL-formula ϕ as $\phi := \Diamond\psi \wedge \Box\Diamond A(a)$ where the flexible concept name A and the individual name a do not occur in ψ. We prove that ψ is unsatisfiable iff ϕ is monitorable w.r.t. true and the empty RBox $\mathcal{R} := \emptyset$.

If ψ is unsatisfiable, we have that $\phi \equiv \Diamond\mathsf{false} \wedge \Box\Diamond A(a) \equiv \mathsf{false} \wedge \Box\Diamond A(a) \equiv \mathsf{false}$, i.e. ϕ is also unsatisfiable. Take now any $\mathfrak{O} \in \mathfrak{P}^*_{\phi,\psi}$ that has an extension w.r.t. true and $\mathcal{R}$. Since ϕ is unsatisfiable, we have $\mathfrak{O}, \mathsf{true}, \mathcal{R} \mathrel{|\!\approx}_\forall \neg\phi$. Thus, Lemma 4.18 yields $\mathfrak{O}, \mathsf{true}, \mathcal{R} \not\mathrel{|\!\approx}_\forall \phi$, which shows that $\mathsf{m}_{\phi,\mathsf{true},\mathcal{R}}(\mathfrak{O}) = \bot$. Consequently, ϕ is $\mathfrak{O}$-monitorable w.r.t. true and $\mathcal{R}$ (take $\mathfrak{O}'$ to be the empty word). Since $\mathfrak{O}$ was an arbitrary element of $\mathfrak{P}^*_{\phi,\mathcal{R}}$ that has an extension w.r.t. true and $\mathcal{R}$, this shows that ϕ is monitorable w.r.t. true and $\mathcal{R}$.

Conversely, if ψ is satisfiable, then there is a model $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$ of ψ. We define

$$\mathcal{O}_i := \bigwedge_{\alpha \in \tau_\psi(\mathcal{I}_i)} \alpha \wedge \bigwedge_{\alpha \in \mathsf{Ax}(\psi)\setminus\tau_\psi(\mathcal{I}_i)} \neg\alpha$$

for every $i \geq 0$. Obviously, $\mathcal{I}_i$ is a model of $\mathcal{O}_i$ and the empty RBox $\mathcal{R}$, and thus $\mathcal{O}_i$ is a partial axiom type, i.e. $\mathcal{O}_i \in \mathfrak{P}_{\phi,\mathcal{R}}$ for every $i \geq 0$. Since there only finitely many partial axiom types, there are finitely many partial axiom types $\mathcal{O}'_1,\ldots,\mathcal{O}'_k$ such that $\{\mathcal{O}'_1,\ldots,\mathcal{O}'_k\} = \{\mathcal{O}_i \mid i \geq 0\}$. Then there is a surjective function $\nu\colon \mathbb{N} \to \{1,\ldots,k\}$ such that $\mathcal{O}_i = \mathcal{O}'_{\nu(i)}$.

To show that ϕ is not monitorable w.r.t. true and $\mathcal{R}$, we consider the finite sequence of partial axiom types $\mathfrak{O} := \mathcal{O}'_1 \ldots \mathcal{O}'_k$. Since the function ν is surjective, any partial axiom type $\mathcal{O}'_i$ in this sequence has at least one of the interpretation $\mathcal{I}_j$ as model, and since $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$ is a DL-LTL-structure, these models of $\mathcal{O}'_1,\ldots,\mathcal{O}'_k$ share the same domain and coincide on the individual names and the rigid concept and role names. Also, these models satisfy the empty RBox $\mathcal{R}$. Consequently, the set $\{\mathcal{O}'_1,\ldots,\mathcal{O}'_k\}$ is r-consistent. Obviously, this implies that $\mathfrak{O}$ has an extension w.r.t. true and $\mathcal{R}$.

To disprove monitorability of ϕ w.r.t. true and $\mathcal{R}$, it is thus sufficient to show that ϕ is not $\mathfrak{O}$-monitorable w.r.t. true and $\mathcal{R}$. For this purpose, we take any finite sequence $\mathfrak{O}' = \mathcal{O}''_1 \ldots \mathcal{O}''_m \in \mathfrak{P}^*_{\phi,\mathcal{R}}$ and show that $\mathsf{m}_{\phi,\mathsf{true},\mathcal{R}}(\mathfrak{O}\cdot\mathfrak{O}') \notin \{\top,\bot\}$.

If $\mathfrak{O}\cdot\mathfrak{O}'$ does not have an extension w.r.t. true and $\mathcal{R}$ (which can happen due to the presence of rigid names), then $\mathfrak{O}\cdot\mathfrak{O}', \mathsf{true}, \mathcal{R} \not\models\!\!\approx_\forall \phi$ and $\mathfrak{O}\cdot\mathfrak{O}', \mathsf{true}, \mathcal{R} \not\models\!\!\approx_\forall \neg\phi$, and thus $\mathsf{m}_{\phi,\mathsf{true},\mathcal{R}}(\mathfrak{O}\cdot\mathfrak{O}') = \lightning \notin \{\top,\bot\}$ as required.

Otherwise, let $\mathfrak{J} = (\mathcal{J}_i)_{i\geq 0}$ be an extension of $\mathfrak{O}\cdot\mathfrak{O}'$ w.r.t. true and $\mathcal{R}$. We define a new DL-LTL-structure $\mathfrak{J}' := (\mathcal{J}'_i)_{i\geq 0}$ with

$$\mathcal{J}'_i := \begin{cases} \mathcal{J}_i & \text{if } 0 \leq i \leq k+m-1\text{; and} \\ \mathcal{J}_{\nu(i-k-m)} & \text{otherwise.} \end{cases}$$

By definition, $\mathfrak{J}'$ consists of interpretations occurring in $\mathfrak{J}$, and thus is indeed a DL-LTL-structure, i.e. all interpretations occurring in $\mathfrak{J}'$ share the same domain and coincide on the individual names and the rigid concept and role names. Additionally, by definition $\mathfrak{J}'$ coincides with $\mathfrak{J}$ on the first $k+m$ interpretations, which shows that it extends $\mathfrak{O}\cdot\mathfrak{O}'$ w.r.t. true and $\mathcal{R}$. Moreover, since every $\mathcal{O}'_i$, $1 \leq i \leq k$, contains complete information about the axioms in ψ, we have that

$$\tau_\psi(\mathfrak{I}) = \tau_\psi(\mathcal{J}'_{k+m})\tau_\psi(\mathcal{J}'_{k+m+1})\ldots.$$

By Lemma 4.8, this shows that $(\mathcal{J}'_{k+m+i})_{i\geq 0}$ is a model of ψ. Since ψ does not contain past operators, this implies that $\mathfrak{J}'$ is a model of $\Diamond\psi$. Since ψ does not contain the concept name A and the individual name a, this is independent on how A and a are interpreted. In addition, since A is flexible, changing its interpretation does not change the fact that rigid names are respected.

Let now $\mathfrak{J}_A$ and $\mathfrak{J}_{\neg A}$ be DL-LTL-structures such that:

1. $\mathfrak{J}_A$ and $\mathfrak{J}_{\neg A}$ coincide for all points in time with $\mathfrak{J}'$ on the interpretation domain as well as on the interpretation of all individual names, role names, and concept names different from A.

2. $\mathfrak{J}_A$ and $\mathfrak{J}_{\neg A}$ coincide with $\mathfrak{J}'$ for all points in time up to $k+m-1$ also on the interpretation of A.
3. In $\mathfrak{J}_A$, the interpretation of A consists of the individual interpreting a at all points in time strictly after $k+m-1$.
4. In $\mathfrak{J}_{\neg A}$, the interpretation of A is empty at all points in time strictly after $k+m-1$.

Obviously, both $\mathfrak{J}_A$ and $\mathfrak{J}_{\neg A}$ are models of $\Diamond\psi$ and they extend $\mathfrak{D}\cdot\mathfrak{D}'$ w.r.t. true and $\mathcal{R}$. However, only $\mathfrak{J}_A$ is also a model of $\Box\Diamond A(a)$. Thus, $\mathfrak{J}_A$ is an extension of $\mathfrak{D}\cdot\mathfrak{D}'$ w.r.t. true and $\mathcal{R}$ that satisfies ϕ, and $\mathfrak{J}_{\neg A}$ is an extension of $\mathfrak{D}\cdot\mathfrak{D}'$ w.r.t. true and $\mathcal{R}$ that satisfies $\neg\phi$. This shows that we have $\mathfrak{D}\cdot\mathfrak{D}', \mathsf{true}, \mathcal{R} \not\approx_{\forall} \phi$ and $\mathfrak{D}\cdot\mathfrak{D}', \mathsf{true}, \mathcal{R} \not\approx_{\forall} \neg\phi$. Consequently, $\mathsf{m}_{\phi,\mathsf{true},\mathcal{R}}(\mathfrak{D}\cdot\mathfrak{D}') \notin \{\top,\bot\}$, which finishes the proof that ϕ is not monitorable w.r.t. true and $\mathcal{R}$. □

Putting the previous two lemmas together, we obtain the following theorem.

Theorem 4.35. *The problem of deciding monitorability of a $\mathcal{SHOQ}$-LTL-formula ϕ w.r.t. a $\mathcal{SHOQ}$-LTL-formula ψ and an RBox $\mathcal{R}$ is*

- ExpTime-*hard and in* 2ExpTime *if* $\mathsf{N_{RC}} = \mathsf{N_{RR}} = \emptyset$*;*
- co-NExpTime-*hard and in* 2ExpTime *if* $\mathsf{N_{RC}} \neq \emptyset$ *and* $\mathsf{N_{RR}} = \emptyset$*; and*
- 2ExpTime-*complete if* $\mathsf{N_{RC}} \neq \emptyset$ *and* $\mathsf{N_{RR}} \neq \emptyset$*.*

The lower bounds hold already for the special case where ϕ is an $\mathcal{ALC}$-LTL-formula, $\psi = \mathsf{true}$, and $\mathcal{R} = \emptyset$.

Proof. The upper bounds follow immediately from Lemma 4.33. The lower bounds follow immediately from Lemma 4.34, the fact that the $\mathcal{SHOQ}$-LTL-formula ϕ constructed in the proof of this lemma is actually an $\mathcal{ALC}$-LTL-formula, and the complexity results for the satisfiability problem in $\mathcal{ALC}$-LTL [BGL12]. Indeed, this problem is ExpTime-complete if $\mathsf{N_{RC}} = \mathsf{N_{RR}} = \emptyset$, NExpTime-complete if $\mathsf{N_{RC}} \neq \emptyset$ and $\mathsf{N_{RR}} = \emptyset$, and 2ExpTime-complete if $\mathsf{N_{RC}} \neq \emptyset$ and $\mathsf{N_{RR}} \neq \emptyset$. Since both ExpTime and 2ExpTime are closed under complement, we obtain the complexity lower bounds of our theorem. □

Note again that only in the case $\mathsf{N_{RC}} \neq \emptyset$ and $\mathsf{N_{RR}} \neq \emptyset$, we have a tight complexity result. Recall, however, that the exact complexity for this problem is not even known for propositional LTL.

4.5 Summary

In this chapter, we have investigated runtime verification using the temporalised description logic $\mathcal{SHOQ}$-LTL. More precisely, we have shown how to construct monitors for $\mathcal{SHOQ}$-LTL-formulas w.r.t. background knowledge that can deal with incomplete knowledge in the form of partial $\mathcal{SHOQ}$-axiom types. The complexity of the monitor construction is quite high. We have seen that the size of a monitor is doubly exponential in the size of the input. However, this cannot be avoided as this doubly exponential blow-up also occurs for propositional LTL, which we have shown in Section 4.1. It should be noted that the complexity of the monitor construction is a worst-case complexity. Minimisation of the intermediate Büchi-automata and the auxiliary deterministic finite automata may lead to much smaller monitors than the ones defined above.

Moreover, we have considered the decision problems of liveness and monitorability. For these problems, we have shown that they are as hard as *un*satisfiability in $\mathcal{ALC}$-LTL and in 2ExpTime. For liveness, the proof of the lower bounds depends on the fact that background knowledge is available. If this is not the case, we have shown that we obtain a lower bound of ExpTime for the liveness problem. For the monitorability problem, we could prove the lower bounds without using the background knowledge. Overall, we have obtained both liveness and monitorability are ExpTime-hard if no rigid names are available, co-NExpTime-hard if only rigid concept names are allowed, and 2ExpTime-complete if both concept names and role names may be rigid. Unfortunately, this leaves gaps for the cases without rigid role names. However, the precise complexity of those problems is unknown even for propositional LTL.

Future work will include trying to close those gaps. Note that since the satisfiability problem in $\mathcal{ALC}$-LTL is harder than in propositional LTL, it may be easier to come up with new complexity results in the case of $\mathcal{ALC}$-LTL and $\mathcal{SHOQ}$-LTL. Furthermore, an implementation of the monitor construction and a thorough empirical evaluation of it is an important direction of future research.

Chapter 5

Temporalised Query Entailment in the Description Logic $\mathcal{SHQ}$

Ontology-based data access (OBDA) [DEF+99; PCD+08] generalises query answering in databases: firstly, the data are not assumed to be complete, i.e. we do not make the closed-world assumption, and secondly, the interpretation of the predicates occurring in the queries is constrained by background knowledge encoded in a knowledge base.

In this chapter, we investigate a *temporalised* version of OBDA and its corresponding decision problem: temporalised query entailment. We show how temporalised query entailment can be decided in the description logic $\mathcal{SHQ}$, and provide complexity results. Most of the results contained in this chapter have already been published in [BBL13b; BBL13a].

This chapter is organised as follows. In Section 5.1, we formally introduce the temporal query language that we investigate in this chapter and discuss similar approaches to temporalising OBDA. After that, in Section 5.2, we show complexity results for temporalised query entailment in our temporal query language. Finally, in Section 5.3, we give a brief summary of the results that we have obtained in this chapter.

5.1 The Temporal Query Language

Unless stated otherwise, we assume throughout this chapter that all ABoxes are *simple* (see Definition 2.9). Note that, however, every complex ABox can be rewritten to a simple ABox using a set of GCIs. More precisely, one can rewrite that concept assertion $C(a)$ to $A(a)$ where $A \in \mathsf{N_C}$ is a fresh concept name, and add the GCIs $A \sqsubseteq C$ and $C \sqsubseteq A$ to the TBox. However, this restriction is useful to separate the influence of the ABox and the TBox on the complexity of reasoning problems.

Our temporal query language is a combination of conjunctive queries [AHV95] and propositional LTL [Pnu77]. This language is very similar to the temporalised description logics $\mathcal{ALC}$-LTL [BGL12] and $\mathcal{SHOQ}$-LTL introduced in Section 3.1. The main difference is that we allow conjunctive queries to occur in place of description logic axioms. Thus, the results we obtained build on existing results about $\mathcal{ALC}$-LTL and also $\mathcal{SHOQ}$-LTL. Since our temporal query language generalises $\mathcal{ALC}$-LTL, some of our hardness results for complexity follow easily from the results in [BGL12]. Moreover, we will use the results obtained in Section 3.2.4 about the consistency problem for Boolean $\mathcal{SHOQ}^{\sqcap}$-knowledge bases. For more information about temporalising description logics, see Section 1.2.

However, most work on temporalised description logics focuses on the satisfiability problem in such logics rather than query answering. In the following, we describe relevant related work in temporalising OBDA. The approaches from the literature have mainly been developed for light-weight languages of the *DL-Lite* family [CDL+09].

For instance, in [AKL+07], various light-weight DLs are extended by allowing the temporal operators to interfere with the DL-component. Extending the work of [AKL+07], in [AKW+13] a temporal extension of *DL-Lite* is presented, which allows the temporal operators $\Diamond^-$ and $\Diamond$ on the left-hand side of GCIs and role-inclusion axioms. In this logic, *first-order rewritability* of conjunctive queries w.r.t. *DL-Lite*-knowledge bases is preserved from the atemporal case, i.e. answering a query over a knowledge base can be reduced to answering a rewritten query (in a different language) over a database induced by the knowledge base. This means that techniques from temporal relational databases can be used to answer temporal queries that can refer to specific points in time.

An approach to temporal query answering in *DL-Lite* that is more similar to the one considered in this chapter is presented in [BLT13b; BLT13a; BLT13c]. There, conjunctive queries are used as atoms in negation-free temporal formulas. This allows for reuse of results about atemporal first-order rewritability in *DL-Lite*. The paper also presents an algorithm to answer such temporal queries over temporal relational databases, which generalises an algorithm from [Cho95; CT05]. The main advantage of this algorithm is that it achieves a so-called *bounded-history encoding*, i.e. the amount of space needed to answer a temporal query does not depend on the length of the observed history. Thus, it is enough to keep track of the relevant data and storing it in the database instead of storing all information from the past.

A similar approach is pursued in [GK12]. There, the authors propose a generic framework to combine a generic DL-query component with a linear temporal dimension. To simplify the decision procedures, both components are decoupled via an autoepistemic modal operator. This allows to use atemporal query-answering algorithms as a black-box inside a temporal satisfiability algorithm.

In [Mot12], temporal query answering over temporalised RDF-triples [GHV05] using an extension of the query language SPARQL is considered.

Furthermore, in [AFW+02], the very expressive temporalised description logic $\mathcal{DLR}_{\mathcal{US}}$ is introduced, which is an extension of $\mathcal{DLR}$ that allows temporal operators to occur within concepts and roles. Moreover, the query-containment problem of non-recursive Datalog queries under constraints defined in $\mathcal{DLR}_{\mathcal{US}}$ is investigated. It turns out that this problem is in general undecidable, but becomes decidable in the fragment $\mathcal{DLR}^-_{\mathcal{US}}$, where no temporal operators are allowed to occur within roles. The query-containment problem is then in 2ExpTime, whereas other reasoning problems such as the satisfiability problem and the subsumption problem in $\mathcal{DLR}^-_{\mathcal{US}}$ are ExpSpace-complete.

Whilst in principle our temporal query language can be parametrised with any description logic, we focus in this chapter on the description logics between $\mathcal{ALC}$ and $\mathcal{SHQ}$. The relative expressivity of these DLs is depicted in Figure 5.1.

To summarise, the work described in [AKW+13; BLT13b; AFW+02; GK12] is most closely related to our approach. Nonetheless, this related work differs from our approach in several ways:

1. We consider the expressive description logic $\mathcal{SHQ}$ instead of inexpressive description logics such as members of the *DL-Lite* family [AKW+13; BLT13b].

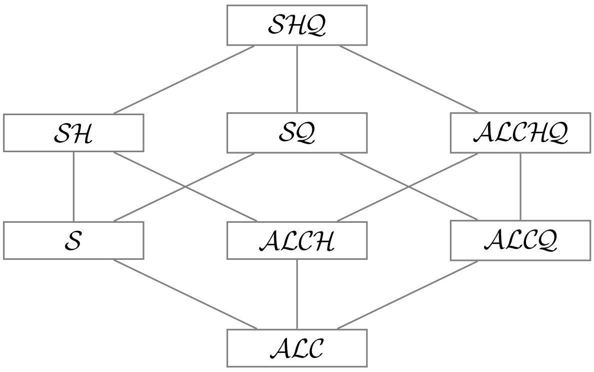

Figure 5.1: The relative expressivity of the DLs between $\mathcal{ALC}$ and $\mathcal{SHQ}$

2. We consider a temporal query language instead of temporalising the ontology language [AKW+13; AFW+02].
3. In contrast to [GK12], we consider also the case of rigid concept and role names. In [BLT13b; BLT13a; BLT13c; AKW+13], rigid names are also used, but in the context of inexpressive DLs.

Since we deal with the expressive description logic $\mathcal{SHQ}$, we take up on the results about the complexity of atemporal conjunctive-query entailment in expressive description logics [OCE06; Lut08a; GHL+08]. For the proofs of our results, it is, however, not sufficient to only apply these results, but we need to adapt the proof methods that were developed in these papers to show these results. To be more precise, we adapt, for instance, the constructions involving forest models and equivalence relations over individual names from [GHL+08], and we use the results about spoilers in $\mathcal{SHQ}^{\sqcap}$ from [Lut08a]. We make this connection explicit in later sections of this chapter.

As the temporal component of our query language is propositional LTL, we also use well-known results from that area of research. As such, we adapt the automata construction for propositional LTL satisfiability from [WVS83; VW94], which is mentioned in Section 2.2.

For our temporal query language, we investigate both the *combined complexity* and the *data complexity* of the temporalised query-entailment problem in three different settings as summarised in Table 5.2. These results hold for all description logics between $\mathcal{ALC}$ and $\mathcal{SHQ}$. In fact, we show that the hardness results already hold for $\mathcal{ALC}$, and we prove the complexity upper bounds for the more expressive description logic $\mathcal{SHQ}$.

Note that in [BBL13a; BBL13b], the complexity results were shown only for $\mathcal{ALC}$. Even though our complexity results are the same for $\mathcal{ALC}$ and $\mathcal{SHQ}$, and in principle the approaches used below to prove the upper bounds for $\mathcal{SHQ}$ are similar to the ones employed in [BBL13a; BBL13b], the proof details are considerably more complex for $\mathcal{SHQ}$. For the combined complexity, the complexity results listed in Table 5.2 are actually identical to the ones for $\mathcal{ALC}$-LTL [BGL12], though the upper bounds are considerably harder to show. The data complexity results in Settings (i) and (ii) coincide with the ones for atemporal query entailment, which is co-NP-complete w.r.t. data complexity. For Setting (iii), we can show that the temporalised query entailment problem is in ExpTime w.r.t. data complexity (in contrast to 2ExpTime-completeness w.r.t. combined complexity), but we do not have a matching lower bound.

Table 5.2: The complexity of temporalised query entailment for all DLs between $\mathcal{ALC}$ and $\mathcal{SHQ}$ in three different settings

	Data complexity	*Combined complexity*
Setting (i)	co-NP-complete (Corollary 5.12 and Theorem 5.21)	ExpTime-complete (Theorems 5.11 and 5.21)
Setting (ii)	co-NP-complete (Corollary 5.12 and Theorem 5.26)	co-NExpTime-complete (Theorems 5.11 and 5.39)
Setting (iii)	co-NP-hard / in ExpTime (Corollary 5.12 and Theorem 5.23)	2ExpTime-complete (Theorems 5.11 and 5.23)

Settings: (i) neither concept names nor role names are allowed to be rigid; (ii) only concept names may be rigid; and (iii) both concept names and role names may be rigid.

From now on, we consider an arbitrary (but fixed) DL between $\mathcal{ALC}$ and $\mathcal{SHQ}$. Before we formally define our temporal query language in Section 5.1.3, we first introduce *conjunctive queries* and related notions in Section 5.1.1, and *temporal knowledge bases* in Section 5.1.2.

5.1.1 Conjunctive Queries

Our query language is based on conjunctive queries [AHV95], which is a subset of first-order queries that are well-investigated in database theory. Basically, conjunctive queries correspond to select-project-join queries in relational algebra, and to select-from-where queries in SQL.

Definition 5.3 (Syntax of conjunctive queries). *Let* $\mathsf{N_V}$ *be a set of variables. A* conjunctive query (CQ) *is of the form* $\phi = \exists y_1, \ldots, y_m.\psi$*, where* $y_1, \ldots, y_m \in \mathsf{N_V}$*, and* ψ *is a (possibly empty) finite conjunction of* atoms *of the form*

- $A(z)$ *with* $A \in \mathsf{N_C}$ *and* $z \in \mathsf{N_V} \cup \mathsf{N_I}$ (concept atom)*; or*
- $r(z_1, z_2)$ *with* $r \in \mathsf{N_R}$ *and* $z_1, z_2 \in \mathsf{N_V} \cup \mathsf{N_I}$ (role atom).

The empty conjunction is denoted by true.

A union of conjunctive queries *(UCQ) is of the form* $\phi_1 \vee \cdots \vee \phi_n$ *with* $n \geq 1$*, where* $\phi_1, \ldots, \phi_n$ *are CQs.* ◇

We denote the set of individual names occurring in a (U)CQ ϕ by $\mathsf{Ind}(\phi)$, the set of variables occurring in ϕ by $\mathsf{Var}(\phi)$, the set of free variables occurring in ϕ by $\mathsf{FVar}(\phi)$, and the set of atoms occurring in ϕ by $\mathsf{At}(\phi)$. We call ϕ *Boolean* if $\mathsf{FVar}(\phi) = \emptyset$. Moreover, we denote the set of individual names occurring in a knowledge base $\mathcal{K}$ by $\mathsf{Ind}(\mathcal{K})$.

Given a (U)CQ ϕ and a knowledge base $\mathcal{K}$, a basic reasoning task is finding so-called *certain answers* to ϕ w.r.t. $\mathcal{K}$, i.e. instantiations of the free variables in ϕ with individual names from $\mathsf{Ind}(\mathcal{K})$ such that the resulting formula is satisfied in every model of $\mathcal{K}$. Thus, answering (U)CQs w.r.t. knowledge bases generalises the entailment of ABox-axioms, i.e. deciding whether $\mathcal{K} \models \alpha$ holds for a given knowledge base $\mathcal{K}$ and a given ABox-axiom α.

We now define the semantics of Boolean (U)CQs, using the notion of homomorphisms [CM77]. This is then extended to answering arbitrary UCQs.

Definition 5.4 (Homomorphism, entailment, certain answer). *Let $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$ be an interpretation and ϕ be a Boolean CQ. A mapping $\pi\colon \mathsf{Var}(\phi) \cup \mathsf{Ind}(\phi) \to \Delta$ is a* homomorphism *of ϕ into $\mathcal{I}$ if*

- $\pi(a) = a^{\mathcal{I}}$ *for every $a \in \mathsf{Ind}(\phi)$;*
- $\pi(z) \in A^{\mathcal{I}}$ *for every concept atom $A(z) \in \mathsf{At}(\phi)$; and*
- $(\pi(z_1), \pi(z_2)) \in r^{\mathcal{I}}$ *for every role atom $r(z_1, z_2) \in \mathsf{At}(\phi)$.*

We say that $\mathcal{I}$ is a model *of ϕ (written $\mathcal{I} \models \phi$) if there is such a homomorphism. Moreover, $\mathcal{I}$ is a* model *of a Boolean UCQ $\phi_1 \vee \cdots \vee \phi_n$ if it is a model of ϕ_i for some i, $1 \leq i \leq n$.*

A Boolean UCQ ϕ is entailed *by a knowledge base $\mathcal{K}$ (written $\mathcal{K} \models \phi$) if every model of $\mathcal{K}$ is also a model of ϕ.*

Given a (not necessarily Boolean) UCQ ϕ, we call a mapping $\mathfrak{a}\colon \mathsf{FVar}(\phi) \to \mathsf{Ind}(\mathcal{K})$ a certain answer *to ϕ w.r.t. $\mathcal{K}$ if $\mathcal{K} \models \mathfrak{a}(\phi)$, where $\mathfrak{a}(\phi)$ is the Boolean UCQ obtained from ϕ by replacing the free variables according to $\mathfrak{a}$.* ◇

For a UCQ ϕ and a knowledge base $\mathcal{K}$, one can compute all certain answers by enumerating all candidate mappings $\mathfrak{a}\colon \mathsf{FVar}(\phi) \to \mathsf{Ind}(\mathcal{K})$ and then solving the entailment problem $\mathcal{K} \models \mathfrak{a}(\phi)$ for each $\mathfrak{a}$. Since there are $|\mathsf{Ind}(\mathcal{K})|^{|\mathsf{FVar}(\phi)|}$ such mappings, we have to solve exponentially many such entailment problems.

To analyse the complexity of deciding the entailment $\mathcal{K} \models \mathfrak{a}(\phi)$, it obviously suffices to consider the case where the UCQ is Boolean. As discussed in Section 1.4, usually one considers two kinds of complexity measures: the combined complexity and data complexity. For the *combined complexity*, all parts of the input, i.e. the UCQ ϕ and the knowledge base $\mathcal{K} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$, are taken into account. For the *data complexity*, however, ϕ, $\mathcal{T}$, and $\mathcal{R}$ are assumed to be of constant size, and the complexity is measured only w.r.t. the data, i.e. the ABox $\mathcal{A}$. For this analysis, we assume in the following that the query does not introduce new names, i.e. it contains only concept and role names that also occur in the TBox or the RBox. This is without loss of generality since we can always introduce trivial axioms like $A \sqsubseteq A$ or $r \sqsubseteq r$ into the TBox and RBox without affecting data complexity or combined complexity.

Regarding data complexity, the entailment problem for *concept assertions* and $\mathcal{ALC}$-knowledge bases is already CO-NP-hard [Sch93a; DLN+94], and a matching upper bound has been established for the entailment problem for UCQs and $\mathcal{SHQ}$-knowledge bases [GHL+08].

The entailment problem for concept assertions and $\mathcal{ALC}$-knowledge bases is EXPTIME-hard w.r.t. combined complexity [BCM+07], and a matching upper bound is known for the entailment problem for UCQs and $\mathcal{ALCHQ}$-knowledge bases [Lut08a]. For the description logic $\mathcal{W}$, the problem is already CO-NEXPTIME-hard, while it becomes 2EXPTIME-hard for the description logic $\mathcal{SH}$ [ELO+09]. In this chapter, we focus on a variant of the UCQ-entailment problem that is EXPTIME-complete even for $\mathcal{SHQ}$-knowledge bases, namely, we restrict to *simple* queries, which are not allowed to use non-simple role names. Note that this is only a restriction in extensions of $\mathcal{W}$.

Before we are ready to consider the temporalised query-entailment problem, we formally introduce temporal knowledge bases.

5.1.2 Temporal Knowledge Bases

We extend the notion of knowledge bases and models into the temporal setting. The setting is that there is a global TBox and a global RBox that define the terminology, and several ABoxes that contain information about the state of the world at the time points we have observed so far.

Definition 5.5 (Syntax of temporal knowledge bases). *A* temporal knowledge base (temporal KB) $\mathcal{K} = ((\mathcal{A}_i)_{0\leq i\leq n}, \mathcal{T}, \mathcal{R})$ *consists of a non-empty finite sequence of ABoxes* $\mathcal{A}_i$*,* $0 \leq i \leq n$*, of* length $n+1 > 0$*, a TBox* $\mathcal{T}$*, and an RBox* $\mathcal{R}$*.* ◇

As for atemporal knowledge bases, we denote by $\mathsf{Ind}(\mathcal{K})$ the set of all individual names occurring in a temporal KB $\mathcal{K}$. The semantics of temporal KBs is based on DL-LTL-structures, which were introduced in Definition 3.3.

Definition 5.6 (Semantics of temporal knowledge bases). *We call the DL-LTL-structure* $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$ *a* model *of the temporal KB* $\mathcal{K} = ((\mathcal{A}_i)_{0\leq i\leq n}, \mathcal{T}, \mathcal{R})$ *(written* $\mathfrak{I} \models \mathcal{K}$*) if*

- $\mathcal{I}_i \models \mathcal{A}_i$ *for every* i*,* $0 \leq i \leq n$*; and*
- $\mathcal{I}_i \models \mathcal{T}$ *and* $\mathcal{I}_i \models \mathcal{R}$ *for every* $i \geq 0$*.* ◇

Recall that according to Definition 3.2, we make the constant-domain assumption and the rigid-individual assumption.

Now we are ready to formally introduce our temporal query language.

5.1.3 Temporal Conjunctive Queries

We combine the notions of conjunctive queries and propositional LTL-formulas into a new formalism, which we call *temporal conjunctive queries*.

Definition 5.7 (Syntax of temporal conjunctive queries). *The set of* temporal conjunctive queries *(TCQs) is the smallest set such that*

- *every conjunctive query is a TCQ; and*
- *if* ϕ_1 *and* ϕ_2 *are TCQs, then so are:* $\neg\phi_1$ *(negation),* $\phi_1 \wedge \phi_2$ *(conjunction),* $\mathsf{X}\,\phi_1$ *(next),* $\mathsf{X}^{-}\,\phi_1$ *(previous),* $\phi_1 \,\mathsf{U}\, \phi_2$ *(until), and* $\phi_1 \,\mathsf{S}\, \phi_2$ *(since).* ◇

As for UCQs, we denote the set of individual names occurring in a TCQ ϕ by $\mathsf{Ind}(\phi)$, and the set of free variables occurring in ϕ by $\mathsf{FVar}(\phi)$. Moreover, a *Boolean TCQ* is a TCQ without free variables.

As usual in temporal logics, we use again

- $\phi_1 \vee \phi_2$ (disjunction) as an abbreviation for $\neg(\neg\phi_1 \wedge \neg\phi_2)$;
- $\phi_1 \rightarrow \phi_2$ (implication) as an abbreviation for $\neg\phi_1 \vee \phi_2$;
- false as an abbreviation for $\neg\mathsf{true}$;[1]

[1] Recall that true denotes the empty conjunction, which is a CQ and thus also a TCQ (see Definition 5.3).

- $\Diamond\phi$ (diamond, which should be read as 'eventually' or 'some time in the future') as an abbreviation for $\mathsf{true}\,\mathsf{U}\,\phi$;
- $\Box\phi$ (box, which should be read as 'always' or 'always in the future') as an abbreviation for $\neg\Diamond\neg\phi$;
- $\Diamond^-\,\phi$ (which should be read as 'once' or 'some time in the past') as an abbreviation for $\mathsf{true}\,\mathsf{S}\,\phi$; and
- $\Box^-\,\phi$ (which should be read as 'historically' or 'always in the past') as an abbreviation for $\neg\,\Diamond^-\,\neg\phi$.

We denote the set of conjunctive queries occurring in a TCQ ϕ by $\mathsf{CQ}(\phi)$. As before, we first define the semantics for Boolean TCQs, which is a straightforward extension of the semantics of CQs and propositional LTL-formulas (see Definition 2.16), similar to the semantics of $\mathcal{SHOQ}$-LTL from Definition 3.3. Also, the notion of certain answers can then be defined exactly as in the atemporal case.

Definition 5.8 (Semantics of TCQs). *For a Boolean TCQ ϕ, a DL-LTL-structure $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$, and a time point $i \geq 0$,* validity of ϕ in $\mathfrak{I}$ at time i *(written $\mathfrak{I}, i \models \phi$) is defined inductively as follows:*

$$\begin{array}{lll}
\mathfrak{I}, i \models \exists y_1,\dots,y_m.\psi & \textit{iff} & \mathcal{I}_i \models \exists y_1,\dots,y_m.\psi \\
\mathfrak{I}, i \models \neg\phi_1 & \textit{iff} & \mathfrak{I}, i \not\models \phi_1, \textit{ i.e. not } \mathfrak{I}, i \models \phi_1 \\
\mathfrak{I}, i \models \phi_1 \land \phi_2 & \textit{iff} & \mathfrak{I}, i \models \phi_1 \textit{ and } \mathfrak{I}, i \models \phi_2 \\
\mathfrak{I}, i \models \mathsf{X}\,\phi_1 & \textit{iff} & \mathfrak{I}, i+1 \models \phi_1 \\
\mathfrak{I}, i \models \mathsf{X}^-\,\phi_1 & \textit{iff} & i > 0 \textit{ and } \mathfrak{I}, i-1 \models \phi_1 \\
\mathfrak{I}, i \models \phi_1\,\mathsf{U}\,\phi_2 & \textit{iff} & \textit{there is some } k \geq i \textit{ such that } \mathfrak{I}, k \models \phi_2\textit{, and} \\
 & & \mathfrak{I}, j \models \phi_1 \textit{ for every } j,\ i \leq j < k \\
\mathfrak{I}, i \models \phi_1\,\mathsf{S}\,\phi_2 & \textit{iff} & \textit{there is some } k,\ 0 \leq k \leq i\textit{, such that } \mathfrak{I}, k \models \phi_2\textit{, and} \\
 & & \mathfrak{I}, j \models \phi_1 \textit{ for every } j,\ k < j \leq i
\end{array}$$

Given a temporal KB $\mathcal{K} = ((\mathcal{A}_i)_{0\leq i\leq n}, \mathcal{T}, \mathcal{R})$, we say that $\mathfrak{I}$ is a model of ϕ w.r.t. $\mathcal{K}$ *if $\mathfrak{I} \models \mathcal{K}$ and $\mathfrak{I}, n \models \phi$. We call ϕ* satisfiable w.r.t. $\mathcal{K}$ *if it has a model w.r.t. $\mathcal{K}$, and it is* entailed by $\mathcal{K}$ *(written $\mathcal{K} \models \phi$) if every model $\mathfrak{I}$ of $\mathcal{K}$ satisfies $\mathfrak{I}, n \models \phi$.*

Let $\mathcal{L}$ be a description logic. The TCQ-satisfiability problem in $\mathcal{L}$ *is to decide, given a TCQ ϕ and a temporal $\mathcal{L}$-KB $\mathcal{K}$, whether ϕ is satisfiable w.r.t. $\mathcal{K}$. Moreover, the* temporalised query-entailment problem in $\mathcal{L}$ *is the problem of deciding, given a TCQ ϕ and a temporal $\mathcal{L}$-KB $\mathcal{K}$, whether ϕ is entailed by $\mathcal{K}$.*

Given a (not necessarily Boolean) TCQ ϕ, we call a mapping $\mathfrak{a}\colon \mathsf{FVar}(\phi) \to \mathsf{Ind}(\mathcal{K})$ a certain answer *to ϕ w.r.t. $\mathcal{K}$ if $\mathcal{K} \models \mathfrak{a}(\phi)$, where $\mathfrak{a}(\phi)$ is the Boolean TCQ obtained from ϕ by replacing the free variables according to $\mathfrak{a}$.* ◊

Note that in this definition of a model, the point of reference is not the first time point 0, as in propositional LTL and $\mathcal{SHOQ}$-LTL, but rather the last time point n of a given temporal knowledge base. Intuitively, this can be seen as the current time point, at which we have information (e.g. sensor data) about the past, but not yet about the future.

As in the atemporal case, one can compute all certain answers to a TCQ ϕ w.r.t. a temporal KB $\mathcal{K}$ by enumerating the (exponentially many) mappings $\mathfrak{a}\colon \mathsf{FVar}(\phi) \to \mathsf{Ind}(\mathcal{K})$ and then solving the entailment problem $\mathcal{K} \models \mathfrak{a}(\phi)$ for each $\mathfrak{a}$. We therefore focus on deciding the entailment problem for the case where ϕ is Boolean. It turns out to be easier to analyse the complexity of deciding the temporalised query *non-entailment* problem $\mathcal{K} \not\models \phi$. This problem has the same complexity as the TCQ-satisfiability problem of ϕ w.r.t. $\mathcal{K}$. In fact, $\mathcal{K} \not\models \phi$ iff $\neg\phi$ has a model w.r.t. $\mathcal{K}$, and conversely ϕ has a model w.r.t. $\mathcal{K}$ iff $\mathcal{K} \not\models \neg\phi$.

Note that, for the data complexity, we have to measure the complexity in the size of the sequence of ABoxes in the temporal knowledge base, instead of just a single ABox. In the following, we assume without loss of generality that the query contains only concept and role names that also occur in the global TBox or the global RBox.

Obviously, the temporalised query-entailment problem includes as a special case the entailment of CQs by atemporal knowledge bases, which can be seen as temporal knowledge bases with a sequence of ABoxes of length 1, i.e. having $n = 0$. Although models of such temporal knowledge bases are formally infinite sequences of interpretations (DL-LTL-structures), all but the first interpretation are irrelevant for the semantics of CQs.

On the temporal side, the TCQ-satisfiability problem generalises the satisfiability problem in $\mathcal{ALC}$-LTL (and $\mathcal{SHQ}$-LTL) since assertions can be seen as simple instances of Boolean CQs. Although $\mathcal{ALC}$-LTL-formulas may additionally contain GCIs, they can equivalently be expressed by negated CQs (see the proof of Theorem 5.11 for details). On the other hand, TCQs are more expressive than $\mathcal{SHQ}$-LTL-formulas since CQs such as $\exists y.r(y,y)$, which says that there is a loop in the model without naming the individual which has the loop, can clearly not even be expressed in $\mathcal{ALC}$.

An assumption on TCQs that was made in [BBL13b] is that all Boolean CQs we encounter are *connected* in the sense that the variables and individual names are related by roles, as defined e.g. in [Tes01; RG10].

Definition 5.9 (Connected Boolean CQs). *We call a Boolean CQ ϕ* connected *if for all $x, y \in \mathsf{Var}(\phi) \cup \mathsf{Ind}(\phi)$, there exists a sequence $x_1, \dots, x_n \in \mathsf{Var}(\phi) \cup \mathsf{Ind}(\phi)$ such that $x_1 = x$, $x_n = y$, and for every i, $1 \leq i < n$, there is an $r \in \mathsf{N_R}$ such that either $r(x_i, x_{i+1}) \in \mathsf{At}(\phi)$ or $r(x_{i+1}, x_i) \in \mathsf{At}(\phi)$.*

A collection of Boolean CQs $\phi_1, \dots, \phi_n$ is a partition *of ϕ if*

- $\mathsf{At}(\phi) = \mathsf{At}(\phi_1) \cup \dots \cup \mathsf{At}(\phi_n)$;
- *the sets $\mathsf{Var}(\phi_i) \cup \mathsf{Ind}(\phi_i)$, $1 \leq i \leq n$, are pairwise disjoint; and*
- *each ϕ_i, $1 \leq i \leq n$, is connected.* ◇

Similar to [Tes01; RG10], in [BBL13b], it is assumed without loss of generality that Boolean TCQs contain only connected CQs. Indeed, if a Boolean TCQ ϕ contains a CQ ψ that is not connected, we can replace ψ by the conjunction $\psi_1 \land \dots \land \psi_n$, where $\psi_1, \dots, \psi_n$ is a partition of ψ. This conjunction is of linear size in the size of ψ, and the resulting TCQ has exactly the same models as ϕ since every homomorphism of ψ into an interpretation $\mathcal{I}$ can be uniquely represented as a collection of homomorphisms of $\psi_1, \dots, \psi_n$ into $\mathcal{I}$. Thus, in [BBL13b] it was always assumed without loss of generality that Boolean TCQs contain only connected CQs. Even though this assumption is without loss of generality, it turns out that we can show the results in this chapter also without making this assumption.

Before we investigate the complexity of the temporalised query-entailment problem, we recall all the assumptions that we have made so far:

- Every at-least restriction contains only simple roles, since otherwise even the problem of deciding whether a knowledge base is consistent would be undecidable [HST00].
- Every role atom in a query contains only simple roles. We make this restriction since then the combined complexity of atemporal query entailment is ExpTime-complete in all description logics between $\mathcal{ALC}$ and $\mathcal{SHQ}$. This enables us to state our complexity results for all these logics at the same time. Without this restriction, the combined complexity would increase whenever transitivity axioms are allowed [ELO+09].
- The queries contain only concept and role names that also occur in the TBox or the RBox. This restriction is without loss of generality.

In the next section, we will investigate the complexity of temporalised query entailment in our temporal query language. We show how to obtain the results of Table 5.2.

5.2 The Complexity of Temporalised Query Entailment

In this section, we analyse the complexity of the temporalised query-entailment problem in DLs between $\mathcal{ALC}$ and $\mathcal{SHQ}$. As mentioned before, all our complexity results hold for any DL between $\mathcal{ALC}$ and $\mathcal{SHQ}$, i.e. we show the lower bounds for $\mathcal{ALC}$ and the upper bounds for $\mathcal{SHQ}$.

We first take a look at an atemporal special case of the TCQ-satisfiability problem, which will prove to be useful for analysing the temporalised query-entailment problem for arbitrary TCQs. A *CQ-literal* is either a Boolean CQ or a negated Boolean CQ. Note that a *conjunction of CQ-literals* ϕ is a special case of a Boolean TCQ. Since ϕ does not contain any temporal operators, for the deciding satisfiability, it suffices to consider a single interpretation instead of a DL-LTL-structure $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$. Extending the notation for UCQs, we often write $\mathcal{I}_i \models \phi$ instead of $\mathfrak{I}, i \models \phi$ in this case. Moreover, it is sufficient to consider temporal KBs with only one ABox, which can be viewed as 'normal' knowledge bases. The following theorem states the complexity of deciding satisfiability in this special case.

Theorem 5.10. *Let $\mathcal{L}$ be a DL between $\mathcal{ALC}$ and $\mathcal{SHQ}$. Deciding whether a conjunction of CQ-literals is satisfiable w.r.t. an $\mathcal{L}$-knowledge base is*

- ExpTime-*complete w.r.t. combined complexity, and*
- NP-*complete w.r.t. data complexity.*

Proof. The problem of deciding the entailment of concept assertions w.r.t. $\mathcal{ALC}$-knowledge bases is ExpTime-hard w.r.t. combined complexity [BCM+07] and co-NP-hard w.r.t. data complexity [CDL+06; Sch93a; DLN+94]. Note that this entailment problem is a special case of the complement of our problem.

Let now $\mathcal{K} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$ be an $\mathcal{SHQ}$-knowledge base, and let ζ be a conjunction of CQ-literals. To check whether there is an interpretation $\mathcal{I}$ with $\mathcal{I} \models \mathcal{K}$ and $\mathcal{I} \models \zeta$, we reduce this problem to a query non-entailment problem of known complexity. Let

$$\zeta = \chi_1 \land \dots \land \chi_\ell \land \neg\rho_1 \land \dots \land \neg\rho_m$$

for Boolean CQs $\chi_1, \dots, \chi_\ell, \rho_1, \dots, \rho_m$. First, we instantiate the non-negated CQs $\chi_1, \dots, \chi_\ell$ by omitting the existential quantifiers and replacing the variables with fresh individual names. The set $\mathcal{A}'$ of all resulting atoms can thus be viewed as an additional ABox that restricts the interpretation $\mathcal{I}$.

However, we also have to ensure that the UNA is respected for the newly introduced individual names. To achieve this, we employ a trick from [GHL+08], which consists in guessing an equivalence relation $\approx$ on $\mathsf{Ind}(\mathcal{A} \cup \mathcal{A}')$ (i.e. the set of individual names occurring in $\mathcal{A}$ or $\mathcal{A}'$) that specifies which individual names are allowed to be mapped to the same domain element, with the additional restriction that each equivalence class can contain at most one element from $\mathsf{Ind}(\mathcal{A})$. For such a relation $\approx$, we fix a representative for each equivalence class such that every class that contains an $a \in \mathsf{Ind}(\mathcal{A})$ has a as its representative. We denote by $\mathcal{A}_\approx$ the ABox resulting from $\mathcal{A}'$ by replacing each new individual name by the representative of its equivalence class. Note that there are exponentially many such equivalence relations, each of which is of size polynomial in the size of ζ.

We now show that the existence of an interpretation $\mathcal{I}$ with $\mathcal{I} \models \mathcal{K}$ and $\mathcal{I} \models \zeta$ is equivalent to the existence of an equivalence relation $\approx$ as above and an interpretation $\mathcal{I}'$ with $\mathcal{I}' \models (\mathcal{A} \cup \mathcal{A}_\approx, \mathcal{T}, \mathcal{R})$ and $\mathcal{I}' \models \neg\rho_1 \land \dots \land \neg\rho_m$.

For the 'if' direction, assume that $\approx$ is an equivalence relation on the individual names and $\mathcal{I}'$ is a model of $\mathcal{A}$, $\mathcal{T}$, $\mathcal{R}$, $\mathcal{A}_\approx$, and $\neg\rho_1 \land \dots \land \neg\rho_m$. By mapping each variable occurring in $\chi_1 \land \dots \land \chi_\ell$ to the interpretation of the representative of the equivalence class of the corresponding fresh individual name, we obtain homomorphisms from χ_i into $\mathcal{I}'$ for each i, $1 \leq i \leq n$. This shows that $\mathcal{I}'$ is also a model of ζ.

For the 'only if' direction, assume that $\mathcal{I} \models \mathcal{K}$ and $\mathcal{I} \models \zeta$. Thus, there are homomorphisms from χ_i into $\mathcal{I}$ for every i, $1 \leq i \leq n$. We define any pair of individual names occurring in $\mathcal{A} \cup \mathcal{A}'$ equivalent w.r.t. $\approx$ iff they are mapped to the same domain element by their respective homomorphisms or $\mathcal{I}$. The extension of $\mathcal{I}$ that maps each representative of its equivalence class to exactly this domain element is obviously a model of $\mathcal{A}_\approx$. It still satisfies $\mathcal{A}$, $\mathcal{T}$, $\mathcal{R}$, and $\neg\rho_1 \land \dots \land \neg\rho_m$ since they do not contain the new individual names, and thus it is of the required form.

The above problem is thus equivalent to finding an equivalence relation $\approx$ and an interpretation $\mathcal{I}$ with $\mathcal{I} \models (\mathcal{A} \cup \mathcal{A}_\approx, \mathcal{T}, \mathcal{R})$ and $\mathcal{I} \not\models \rho$ where $\rho := \rho_1 \lor \dots \lor \rho_m$ is the Boolean UCQ that results from negating the conjunction of all negated CQs in ζ. This is the same as asking whether $(\mathcal{A} \cup \mathcal{A}_\approx, \mathcal{T}, \mathcal{R})$ does not entail ρ.

For the combined complexity, we can enumerate all equivalence relations $\approx$ in exponential time, and check the above non-entailment for the polynomial-size $\mathcal{SHQ}$-knowledge base and UCQ resulting from each relation $\approx$, which can be done in ExpTime [Lut08a]. For the data complexity, we can guess a relation $\approx$ in non-deterministic polynomial time, and check the non-entailment in NP [OCE06]. Hence, we obtain the desired complexity results for the satisfiability problem of a conjunction of CQ-literals. □

In the remainder of this section, we present several constructions, most of which use the above theorem, to derive the complexity results shown in Table 5.2 for temporalised query entailment. As mentioned several times, the results depend on which symbols are allowed to be rigid. It is well-known that one can simulate rigid concept names by rigid role names [BGL12], which is why there are only three cases to consider.

As argued above, we show the lower bounds for the DL $\mathcal{ALC}$, and the upper bounds for the DL $\mathcal{SHQ}$. Hence, in Section 5.2.1, we consider the lower bounds for temporalised query entailment in $\mathcal{ALC}$, whereas in Section 5.2.2, we consider the upper bounds for temporalised query entailment in $\mathcal{SHQ}$. However, the upper bounds for the most complex case of rigid concept names are treated separately in Sections 5.2.3 and 5.2.4.

5.2.1 Lower Bounds for Temporalised Query Entailment in $\mathcal{ALC}$

In this section, we investigate the lower bounds for temporalised query entailment of Table 5.2.

For the combined complexity, we obtain the lower bounds by a simple reduction of the satisfiability problem of $\mathcal{ALC}$-LTL [BGL12].

Theorem 5.11. *With respect to combined complexity, the temporalised query-entailment problem in $\mathcal{ALC}$ is*

- ExpTime-*hard if* $\mathsf{N_{RC}} = \mathsf{N_{RR}} = \emptyset$*;*
- co-NExpTime-*hard if* $\mathsf{N_{RC}} \neq \emptyset$ *and* $\mathsf{N_{RR}} = \emptyset$*; and*
- 2ExpTime-*hard if* $\mathsf{N_{RC}} \neq \emptyset$ *and* $\mathsf{N_{RR}} \neq \emptyset$*.*

Proof. As shown in [BGL12], the satisfiability problem of $\mathcal{ALC}$-LTL is ExpTime-complete if $\mathsf{N_{RC}} = \mathsf{N_{RR}} = \emptyset$, NExpTime-complete if $\mathsf{N_{RC}} \neq \emptyset$ and $\mathsf{N_{RR}} = \emptyset$, and 2ExpTime-complete if $\mathsf{N_{RC}} \neq \emptyset$ and $\mathsf{N_{RR}} \neq \emptyset$.

Let now ϕ be an $\mathcal{ALC}$-LTL-formula, $C_1 \sqsubseteq D_1, \ldots, C_p \sqsubseteq D_p$ be all GCIs occurring in ϕ, and $E_1(a_1), \ldots, E_m(a_m)$ be all concept assertions occurring in ϕ, where $E_1, \ldots, E_m$ are arbitrary concepts. We define ψ to be the Boolean TCQ obtained from ϕ by replacing each $C_i \sqsubseteq D_i$ by $\neg(\exists x. A_i(x))$ and each E_j by B_j, where A_i, B_j are concept names that do not occur in ϕ, for every i, $1 \leq i \leq p$, and every j, $1 \leq j \leq m$. Moreover, we define

$$\mathcal{T} := \{A_i \equiv C_i \sqcap \neg D_i \mid 1 \leq i \leq p\} \cup \{B_j \equiv E_j \mid 1 \leq j \leq m\}.$$

Then ϕ is satisfiable iff $(\emptyset, \mathcal{T}, \emptyset) \not\models \neg\psi$. We have thus reduced the satisfiability problem in $\mathcal{ALC}$-LTL to the temporalised query *non-entailment* problem in $\mathcal{ALC}$, which yields the claimed lower bounds. □

For the data complexity, we obtain the lower bounds directly from Theorem 5.10.

Corollary 5.12. *With respect to data complexity, the temporalised query-entailment problem in $\mathcal{ALC}$ is* co-NP-*hard.*

Proof. Theorem 5.10 states that deciding whether a conjunction of CQ-literals ζ is satisfiable w.r.t. an atemporal $\mathcal{ALC}$-knowledge base $\mathcal{K}$ is NP-complete w.r.t. data complexity. Since ζ is a special TCQ and rigid names are irrelevant in the atemporal case, we obtain co-NP-hardness w.r.t. data complexity for the temporalised query-entailment problem in all the settings listed in Table 5.2. □

Theorem 5.11 and Corollary 5.12 yield the lower bounds for temporalised query entailment as shown in Table 5.2.

In the following sections, we present the ideas for the upper bounds w.r.t. combined complexity and data complexity. For the former, we can match all lower bounds that we have from Theorem 5.11. For the latter, however, we cannot match the lower bound of co-NP in the case where both concept names and role names may be rigid. While our constructions need to deal with CQs and the additional expressivity of $\mathcal{SHQ}$ in an appropriate way, the basic ideas are similar to those presented for $\mathcal{SHOQ}$-LTL in Chapter 3. However, there are several differences to the constructions of Chapter 3. Firstly, we have to deal with conjunctive queries instead of axioms, and secondly, we do not allow nominals in this chapter. Thirdly, in the semantics of TCQs (see Definition 5.8), the point of reference is the last time point n and a temporal knowledge base has to be taken into account. Hence, although similar, the constructions in the subsequent sections differ from the ones in Chapter 3.

5.2.2 Upper Bounds for Temporalised Query Entailment in $\mathcal{SHQ}$

Similar to what was done for $\mathcal{ALC}$-LTL in Lemma 4.3 in [BGL12] and also for $\mathcal{SHOQ}$-LTL in Lemma 3.13, we reduce the *TCQ-satisfiability problem* in $\mathcal{SHQ}$ to two separate satisfiability problems.

In the following, let $\mathcal{K} = ((\mathcal{A}_i)_{0\leq i\leq n}, \mathcal{T}, \mathcal{R})$ be a temporal $\mathcal{SHQ}$-KB, and let ϕ be a Boolean TCQ, for which we want to decide whether ϕ has a model w.r.t. $\mathcal{K}$.

We again consider the propositional abstraction of ϕ. Its definition is very similar to propositional abstraction of a $\mathcal{SHOQ}$-LTL-formula (see Definition 3.6).

Definition 5.13 (Propositional abstraction). *Let ϕ be a TCQ, and let $\mathcal{P}_\phi$ be a finite set of propositional variables such that there is a bijection $\mathsf{p}\colon \mathsf{CQ}(\phi) \to \mathcal{P}_\phi$.*

1. *The propositional LTL-formula ϕ^{p} is obtained from ϕ by replacing every occurrence of a CQ ψ in ϕ by its p-image $\mathsf{p}(\psi)$. We call ϕ^{p} the* propositional abstraction of ϕ w.r.t. p.
2. *Given a DL-LTL-structure $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$, its* propositional abstraction w.r.t. p *is the propositional LTL-structure $\mathfrak{I}^{\mathsf{p}} = (w_i)_{i\geq 0}$ with*

$$w_i := \left\{\mathsf{p}(\psi) \mid \psi \in \mathsf{CQ}(\phi) \text{ and } \mathcal{I}_i \models \alpha\right\}$$

for every $i \geq 0$. ◇

In the following, we assume that $\mathsf{p}\colon \mathsf{CQ}(\phi) \to \mathcal{P}_\phi$ is a bijection.[2] Again for simplicity, for a sub-TCQ ψ of ϕ, we denote by ψ^{p} the propositional abstraction of ψ w.r.t. the restriction of p to $\mathsf{CQ}(\psi)$. The propositional abstraction ϕ^{p} of ϕ w.r.t. p is a propositional LTL-formula that allows us to analyse the temporal structure of ϕ separately from the CQ-component. The following lemma is very similar to Lemma 3.8, and its proof is analogous.

Lemma 5.14. *Let $\mathfrak{I}$ be a DL-LTL-structure with $\mathfrak{I} \models \mathcal{K}$. Then, $\mathfrak{I}$ is a model of ϕ w.r.t. $\mathcal{K}$ iff $\mathfrak{I}^{\mathsf{p}}$ is a model of ϕ^{p}.*

[2] As for $\mathcal{SHOQ}$-LTL-formulas, it is obvious that such a set $\mathcal{P}_\phi$ and such a bijection p exists for every TCQ.

Proof. Let $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$ be a DL-LTL-structure with $\mathfrak{I} \models \mathcal{K}$, and $\mathfrak{I}^{\mathsf{p}} = (w_i)_{i\geq 0}$ its propositional abstraction w.r.t. p. Moreover, let $i \geq 0$. We prove this lemma by showing that $\mathfrak{I}, i \models \phi$ iff $\mathfrak{I}^{\mathsf{p}}, i \models \phi^{\mathsf{p}}$ by induction of the structure of ϕ.

For the base case, let ϕ be an CQ. Then, we have $\mathfrak{I}, i \models \phi$ *iff* $\mathcal{I}_i \models \phi$ *iff* $\mathsf{p}(\phi) \in w_i$ *iff* $w_i \models \phi^{\mathsf{p}}$ *iff* $\mathfrak{I}^{\mathsf{p}}, i \models \phi^{\mathsf{p}}$.

For the induction steps, we obtain the claim by using the definition of the semantics and the induction hypothesis. This can be done exactly as we did in the proof of Lemma 3.8. □

Again, the 'only if' direction of this lemma yields that satisfiability of ϕ w.r.t. $\mathcal{K}$ implies satisfiability of ϕ^{p}. Note that, however, the 'if' direction does not yield the converse of this implication as already argued for Lemma 3.8.

We again consider a set $\mathcal{W} \subseteq 2^{\mathcal{P}_\phi}$, which intuitively specifies the worlds that are allowed to occur in an LTL-structure satisfying ϕ^{p}. To express this restriction, we define the propositional LTL-formula

$$\phi^{\mathsf{p}}_{\mathcal{W}} := \phi^{\mathsf{p}} \wedge \Box^{-}\Box\Big(\bigvee_{X\in\mathcal{W}}\Big(\bigwedge_{p\in X} p \wedge \bigwedge_{p\in\mathcal{P}_\phi\setminus X} \neg p\Big)\Big).$$

Note that a propositional LTL-formula $\Box^{-}\Box\psi$ is satisfied iff ψ holds at every point in time.[3] The next lemma formalises the immediate connection between ϕ and $\phi^{\mathsf{p}}_{\mathcal{W}}$.

Lemma 5.15. *If ϕ is satisfiable w.r.t. $\mathcal{K}$, then there is a set $\mathcal{W} \subseteq 2^{\mathcal{P}_\phi}$ and a propositional LTL-structure $\mathfrak{W}$ such that $\phi^{\mathsf{p}}_{\mathcal{W}}$ is valid in $\mathfrak{W}$ at time n.*

Proof. Let $\mathfrak{I}$ be a DL-LTL-structure that is a model of ϕ w.r.t. $\mathcal{K}$, and let $\mathfrak{I}^{\mathsf{p}} = (w_i)_{i\geq 0}$ be its propositional abstraction w.r.t. p. We consider the finite set $\mathcal{W} := \{w_i \mid i \geq 0\}$ induced by $\mathfrak{I}$. Using Lemma 5.14, it is easy to verify that the fact that $\mathfrak{I} \models \mathcal{K}$ and ϕ is valid in $\mathfrak{I}$ at time n implies that $\phi^{\mathsf{p}}_{\mathcal{W}}$ is valid in $\mathfrak{I}^{\mathsf{p}}$ at time n. □

As argued above, guessing a set $\mathcal{W}$ and then checking whether there is a propositional LTL-structure $\mathfrak{W}$ such that the induced propositional LTL-formula $\phi^{\mathsf{p}}_{\mathcal{W}}$ is valid in $\mathfrak{W}$ at time n is not sufficient for checking whether ϕ has a model w.r.t. $\mathcal{K}$. We must also check whether $\mathcal{W}$ can indeed be induced by some DL-LTL-structure that is a model of $\mathcal{K}$. For that, we extend the notion of r-satisfiability from Definition 3.10.

Definition 5.16 (R-satisfiability). *Let $\mathcal{W} = \{X_1, \ldots, X_k\} \subseteq 2^{\mathcal{P}_\phi}$, and let ι be a mapping from $\{0, \ldots, n\}$ into $\{1, \ldots, k\}$. We call $\mathcal{W}$* r-satisfiable w.r.t. ι and $\mathcal{K}$ *if there exist interpretations $\mathcal{J}_1 = (\Delta, \cdot^{\mathcal{J}_1})$, …, $\mathcal{J}_k = (\Delta, \cdot^{\mathcal{J}_k})$, and $\mathcal{I}_0 = (\Delta, \cdot^{\mathcal{I}_0})$, …, $\mathcal{I}_n = (\Delta, \cdot^{\mathcal{I}_n})$ such that*

- *$a^{\mathcal{J}_i} = a^{\mathcal{J}_j} = a^{\mathcal{I}_\ell}$ holds for every $a \in \mathsf{N_I}$ and all i, j, ℓ, $1 \leq i < j \leq k$ and $0 \leq \ell \leq n$;*
- *$A^{\mathcal{J}_i} = A^{\mathcal{J}_j} = A^{\mathcal{I}_\ell}$ holds for every $A \in \mathsf{N_{RC}}$ and all i, j, ℓ, $1 \leq i < j \leq k$ and $0 \leq \ell \leq n$;*
- *$r^{\mathcal{J}_i} = r^{\mathcal{J}_j} = r^{\mathcal{I}_\ell}$ holds for every $r \in \mathsf{N_{RR}}$ and all i, j, ℓ, $1 \leq i < j \leq k$ and $0 \leq \ell \leq n$; and*
- *every $\mathcal{J}_i$, $1 \leq i \leq k$, and every $\mathcal{I}_j$, $0 \leq j \leq n$, is a model of $\mathcal{T}$ and $\mathcal{R}$;*

[3] Note also that the propositional LTL-formula $\phi^{\mathsf{p}}_{\mathcal{W}}$ for a $\mathcal{SHOQ}$-LTL-formula ϕ as defined in Section 3.2 does not use the $\Box^{-}$-operator. We need this here due to the definition of the semantics where the point of reference is time point n rather than 0.

- *every $\mathcal{J}_i$, $1 \leq i \leq k$, is a model of the conjunction of CQ-literals*

$$\zeta_{X_i} := \bigwedge_{p \in X_i} \mathsf{p}^{-1}(p) \wedge \bigwedge_{p \in \mathcal{P}_\phi \setminus X_i} \neg \mathsf{p}^{-1}(p); \text{ and}$$

- *every $\mathcal{I}_i$, $0 \leq i \leq n$, is a model of $\mathcal{A}_i$ and $\zeta_{X_{\iota(i)}}$.* ◇

The intuition underlying this definition is the following. The existence of the interpretations $\mathcal{J}_i$, $1 \leq i \leq k$, ensures that the conjunction ζ_{X_i} of the CQ-literals induced by X_i is consistent. In fact, a set $\mathcal{W}$ containing a set X_i for which this does not hold cannot be induced by a DL-LTL-structure. The interpretations $\mathcal{I}_i$, $0 \leq i \leq n$, constitute the first $n+1$ interpretations in such a DL-LTL-structure. In addition to inducing a set $X_{\iota(i)} \in \mathcal{W}$ and thus satisfying the corresponding conjunction $\zeta_{X_{\iota(i)}}$, the interpretation $\mathcal{I}_i$ must also satisfy the ABox $\mathcal{A}_i$. Moreover, we ensure that the interpretations share the same domain, respect rigid names, and satisfy the TBox $\mathcal{T}$ and the RBox $\mathcal{R}$. Note that we can use Theorem 5.10 to check whether interpretations satisfying the last three conditions of Definition 5.16 exist. As we will see below, the difficulty lies in ensuring that the interpretations share the same domain and respect rigid names.

Satisfaction of the temporal structure of ϕ by a DL-LTL-structure built this way is ensured by testing $\phi^{\mathsf{p}}_{\mathcal{W}}$ for satisfiability w.r.t. a side condition that ensures that the first n worlds are those chosen by ι. For that, we extend the notion of t-satisfiability from Definition 3.11.

Definition 5.17 (T-satisfiability). *Let $\mathcal{W} = \{X_1, \ldots, X_k\} \subseteq 2^{\mathcal{P}_\phi}$, and let ι be a mapping from $\{0, \ldots, n\}$ into $\{1, \ldots, k\}$. We call the propositional LTL-formula ϕ^{p}* t-satisfiable w.r.t. $\mathcal{W}$ and ι *if there exists a propositional LTL-structure $\mathfrak{W} = (w_i)_{i \geq 0}$ such that*

- *$\mathfrak{W}, n \models \phi^{\mathsf{p}}_{\mathcal{W}}$, and*
- *$w_i = X_{\iota(i)}$ for every i, $0 \leq i \leq n$.* ◇

The next lemma shows that these two satisfiability problems, namely, t-satisfiability and r-satisfiability, can be combined to decidable the TCQ-satisfiability problem in $\mathcal{SHQ}$. The proof of the lemma is very similar to the proofs of Lemmas 3.12 and 3.13.

Lemma 5.18. *The TCQ ϕ is satisfiable w.r.t. the temporal knowledge base $\mathcal{K}$ iff there is a set $\mathcal{W} = \{X_1, \ldots, X_k\} \subseteq 2^{\mathcal{P}_\phi}$ and a mapping $\iota \colon \{0, \ldots, n\} \to \{1, \ldots, k\}$ such that*

- *$\mathcal{W}$ is r-satisfiable w.r.t. ι and $\mathcal{K}$, and*
- *ϕ^{p} is t-satisfiable w.r.t. $\mathcal{W}$ and ι.*

Proof. For the ‘only if’ direction, assume that there is a DL-LTL-structure $\mathfrak{I} = (\mathcal{I}_i)_{i \geq 0}$ that is a model of ϕ w.r.t. $\mathcal{K}$, i.e. $\mathfrak{I} \models \mathcal{K}$ and $\mathfrak{I}, n \models \phi$. Let $\mathfrak{I}^{\mathsf{p}} = (w_i)_{i \geq 0}$ be the propositional abstraction of $\mathfrak{I}$ w.r.t. p. Recall that we have already seen in the proof of Lemma 5.15 that $\mathfrak{I}$ induces a finite set $\mathcal{W} := \{w_i \mid i \geq 0\} = \{X_1, \ldots, X_k\} \subseteq 2^{\mathcal{P}_\phi}$ such that $\phi^{\mathsf{p}}_{\mathcal{W}}$ is valid in $\mathfrak{I}^{\mathsf{p}}$ at time n. Moreover, we have that for every $i \geq 0$, there is an index $\nu_i \in \{1, \ldots, k\}$ such that $\mathcal{I}_i$ induces the set X_{ν_i}, i.e.

$$X_{\nu_i} = \{\mathsf{p}(\psi) \mid \psi \in \mathsf{CQ}(\phi) \text{ and } \mathcal{I}_i \models \psi\},$$

and, conversely, for every $\nu \in \{1,\dots,k\}$, there is an index $i \geq 0$ such that $\nu = \nu_i$. We define the mapping ι as follows: $\iota(i) = \nu_i$ for every i, $0 \leq i \leq n$. By definition of ι, X_{ν_i} and $\mathfrak{I}^{\mathsf{p}}$, we also have $w_i = X_{\iota(i)}$ for every i, $0 \leq i \leq n$. Thus, ϕ^{p} is t-satisfiable w.r.t. $\mathcal{W}$ and ι. For every i, $1 \leq i \leq k$, the interpretation $\mathcal{J}_i$ is obtained as follows. Let $\ell_1,\dots,\ell_k$ be such that $\nu_{\ell_1} = 1, \dots, \nu_{\ell_k} = k$. Now, if we set $\mathcal{J}_i := \mathcal{I}_{\ell_i}$, then we clearly have that $\mathcal{J}_i$ is a model of ζ_{X_i}. It is now easy to see that the interpretations $\mathcal{J}_1,\dots,\mathcal{J}_k$, and $\mathcal{I}_0,\dots,\mathcal{I}_n$ satisfy the conditions for r-satisfiability of $\mathcal{W}$ w.r.t. ι and $\mathcal{K}$.

For the 'if' direction, assume that there is a set $\mathcal{W} = \{X_1,\dots,X_k\} \subseteq 2^{\mathcal{P}_\phi}$ and a mapping $\iota\colon \{0,\dots,n\} \to \{1,\dots,k\}$ such that $\mathcal{W}$ is r-satisfiable w.r.t. ι and $\mathcal{K}$ and ϕ^{p} is t-satisfiable w.r.t. $\mathcal{W}$ and ι. Hence, there is a propositional LTL-structure $\mathfrak{W} = (w_i)_{i\geq 0}$ such that $\phi^{\mathsf{p}}_{\mathcal{W}}$ is valid in $\mathfrak{W}$ at time n and $w_i = X_{\iota(i)}$ for every i, $0 \leq i \leq n$, and there are interpretations $\mathcal{J}_1,\dots,\mathcal{J}_k$, and $\mathcal{I}_0,\dots,\mathcal{I}_n$ such that the conditions in Definition 5.16 are satisfied.

By the definition of $\phi^{\mathsf{p}}_{\mathcal{W}}$, we have that for every world w_i, there is exactly one index $\nu_i \in \{1,\dots,k\}$ such that w_i satisfies

$$\bigwedge_{p\in X_{\nu_i}} p \wedge \bigwedge_{p\in \mathcal{P}_\phi \setminus X_{\nu_i}} \neg p.$$

Since every w_i, $0 \leq i \leq n$, satisfies exactly the propositional variables of $X_{\iota(i)}$, we have $\iota(i) = \nu_i$. We can now define a DL-LTL-structure $\mathfrak{I} := (\mathcal{I}_i)_{i\geq 0}$ as follows. We set $\mathcal{I}_i := \mathcal{J}_{\nu_i}$ for $i > n$. By Definition 5.16, each $\mathcal{I}_i$ is a model of $\zeta_{X_{\nu_i}}$, i.e. it satisfies exactly the CQs specified by the propositional variables in X_{ν_i}. This yields since $\mathfrak{W}, n \models \phi^{\mathsf{p}}_{\mathcal{W}}$, that $\mathfrak{I}, n \models \phi$. It also follows directly from Definition 5.16 that $\mathfrak{I} \models \mathcal{K}$. Hence, we have that ϕ is satisfiable w.r.t. $\mathcal{K}$. □

To obtain a decision procedure for the TCQ-satisfiability problem in $\mathcal{SHQ}$, we have to nondeterministically guess or construct the set $\mathcal{W}$ and the mapping ι, and then check the two conditions of Lemma 5.18. Depending on which symbols are allowed to be rigid, we use different constructions to achieve that. First, we focus on deciding t-satisfiability w.r.t. a given set $\mathcal{W}$ and a given mapping ι.

Deciding T-satisfiability

From now on, let $\mathcal{W} = \{X_1,\dots,X_k\} \subseteq 2^{\mathcal{P}_\phi}$, and let $\iota\colon \{0,\dots,n\} \to \{1,\dots,k\}$ be a mapping that specifies a set $X_{\iota(i)}$ for each of the ABoxes $\mathcal{A}_i$, $0 \leq i \leq n$. We proceed similar to the proof of Lemma 3.14, where we have shown that we can decide t-satisfiability of the propositional abstraction of a $\mathcal{SHOQ}$-LTL-formula w.r.t. a set of worlds $\mathcal{W}$ in time exponential in the size of the propositional abstraction and linear in the size of $\mathcal{W}$. For that, we constructed a Büchi-automaton for the propositional abstraction and removed all transitions that are labelled with a letter that is not contained in $\mathcal{W}$. However, the difference here is that we have to take care of the second condition of t-satisfiability (see Definition 5.17), and that we have to ensure that ϕ^{p} is satisfied as time point n rather than 0.

We check this by using the same idea that we employed in Section 2.2.2 when we constructed a Büchi-automaton that ensures that a propositional LTL-formula is satisfied at a given time point. We attach a counter from $\{0,\dots,n+1\}$ to the states of the Büchi-automaton. Transitions where the counter is $i < n+1$ check if the current world corresponds to $X_{\iota(i)}$ and increase the counter by 1.

In the following, let $\mathcal{N} = (Q, \Sigma_{\mathcal{P}_\phi}, \Delta, Q_0, F)$ be a Büchi-automaton such that for every ω-word $w = w_0 w_1 w_2 \ldots \in \Sigma_{\mathcal{P}_\phi}^\omega$, we have that $w \in L_\omega(\mathcal{N})$ iff ϕ^{p} is valid in the propositional LTL-structure $\mathfrak{W} = (w_i)_{i \geq 0}$ at time n. We have shown in Section 2.2.2 how such a Büchi-automaton can be constructed. Moreover, we have shown in Lemma 2.24 that we can construct such a Büchi-automaton in time exponential in the size of ϕ^{p} and polynomial in n.

We define the Büchi-automaton $\mathcal{N}' = (Q', \Sigma_{\mathcal{P}_\phi}, \Delta', Q'_0, F')$ as follows:

- $Q' := Q \times \{0, \ldots, n+1\}$;
- $((q, \ell),\ \sigma,\ (q', \ell'))\ \in\ \Delta'$ iff
  - $(q, \sigma, q') \in \Delta$,
  - $\sigma \in \mathcal{W}$,
  - $\ell \leq n$ implies $\sigma = X_{\iota(\ell)}$, and
  - $\ell' = \begin{cases} \ell + 1 & \text{if } \ell \leq n \text{, and} \\ \ell & \text{otherwise;} \end{cases}$
- $Q'_0 := Q_0 \times \{0\}$; and
- $F' := F \times \{n+1\}$.

The next lemma shows that this Büchi-automaton is correct.

Lemma 5.19. *For every ω-word $w = w_0 w_1 w_2 \ldots \in \Sigma_{\mathcal{P}_\phi}^\omega$, we have $w \in L_\omega(\mathcal{N}')$ iff $\phi_{\mathcal{W}}^{\mathsf{p}}$ is valid in the propositional LTL-structure $\mathfrak{W} = (w_i)_{i \geq 0}$ at time n, and $w_i = X_{\iota(i)}$ for every i, $0 \leq i \leq n$.*

Proof. For the 'only if' direction, assume that $\phi_{\mathcal{W}}^{\mathsf{p}}$ is valid in the propositional LTL-structure $\mathfrak{W} = (w_i)_{i \geq 0}$ at time n, and that we have $w_i = X_{\iota(i)}$ for every i, $0 \leq i \leq n$. Obviously, we have also $\mathfrak{W}, n \models \phi^{\mathsf{p}}$ since ϕ^{p} is a conjunct of $\phi_{\mathcal{W}}^{\mathsf{p}}$. This yields that $w = w_0 w_1 w_2 \ldots \in L_\omega(\mathcal{N})$. Thus, there is a run $S_0 S_1 S_2 \ldots$ of $\mathcal{N}$ on w. Then,

$$(S_0, 0)(S_1, 1) \ldots (S_n, n)(S_{n+1}, n+1)(S_{n+2}, n+1) \ldots$$

is a accepting run of $\mathcal{N}'$ on w due to the following reasons:

- Obviously, we have $(S_i, \ell) \in Q'$ for every $i \geq 0$ and every ℓ, $0 \leq k \leq n+1$.
- We have for every i, $0 \leq i \leq n$, that

  $$((S_i, i),\ w_i,\ (S_{i+1}, i+1))\ \in\ \Delta',$$

  and for every $i \geq n+1$ that

  $$((S_i, n+1),\ w_i,\ (S_{i+1}, n+1))\ \in\ \Delta'$$

  since:
  - $(S_i, w_i, S_{i+1}) \in \Delta$ by construction;
  - $w_i \in \mathcal{W}$ since $\mathfrak{W}$ is a model of $\phi_{\mathcal{W}}^{\mathsf{p}}$;
  - $i \leq n$ implies $w_i = X_{\iota(i)}$ by construction; and
  - the condition for incrementing the second component of a state (until $n+1$ is reached) is obviously also satisfied.

- Since $S_0 \in Q_0$, we have $(S_0, 0) \in Q'_0$.
- Since $S_0 S_1 S_2 \ldots$ is an accepting run of $\mathcal{N}$ on w, there are infinitely many $j \geq 0$ such that $S_j \in F$. The definition of F' yields now that the above run is accepting.

For the 'if' direction, assume that $w = w_0 w_1 w_2 \ldots \in L_\omega(\mathcal{N}')$, i.e. there is an accepting run

$$(S_0, 0)(S_1, 1) \ldots (S_n, n)(S_{n+1}, n+1)(S_{n+2}, n+1) \ldots$$

of $\mathcal{N}'$ on w.

By the definition of Δ', we have $w_i = X_{\iota(i)}$ for every i, $0 \leq i \leq n$. To show that $\phi^{\mathsf{p}}_{\mathcal{W}}$ is valid in $\mathfrak{W} := (w_i)_{i \geq 0}$ at time n observe that we have $w_i \in \mathcal{W}$ for every $i \geq 0$ again by the definition of Δ'. Thus, the conjunct

$$\Box^- \Box \Bigg(\bigvee_{X \in \mathcal{W}} \Big(\bigwedge_{p \in X} p \wedge \bigwedge_{p \in \mathcal{P}_\phi \setminus X} \neg p \Big) \Bigg)$$

of $\phi^{\mathsf{p}}_{\mathcal{W}}$ is clearly satisfied by $\mathfrak{W}$ (at any time point).

Moreover, we have that $S_0 S_1 S_2 \ldots$ is an accepting run of $\mathcal{N}$ on w by the definition of Q'_0, Δ', and F'. Thus, ϕ^{p} is valid in $\mathfrak{W}$ at time n. Hence, we obtain that $\phi^{\mathsf{p}}_{\mathcal{W}}$ is valid in $\mathfrak{W}$ at time n. □

This lemma implies that $L_\omega(\mathcal{N}') \neq \emptyset$ iff ϕ^{p} is t-satisfiable w.r.t. $\mathcal{W}$ and ι. We can thus decide the latter problem by checking $\mathcal{N}'$ for emptiness, which yields the following complexity result.

Lemma 5.20. *Deciding whether ϕ^{p} is t-satisfiable w.r.t. $\mathcal{W}$ and ι can be done in time exponential in the size of ϕ^{p}, linear in the size of $\mathcal{W}$, and polynomial in n.*

Proof. As mentioned above the Büchi-automaton $\mathcal{N}$ can be constructed in time exponential in the size of ϕ^{p} and polynomial in n (see Lemma 2.24). Note that the Büchi-automaton $\mathcal{N}'$ can be constructed in time linear in the size of $\mathcal{N}$, the size of $\mathcal{W}$, and n, and thus the size of $\mathcal{N}'$ is linear in the size of $\mathcal{N}$ and n, and thus exponential in the size of ϕ^{p} and polynomial in n. Since the emptiness problem for Büchi-automata can be solved in polynomial time [VW94], this yields that t-satisfiability of ϕ^{p} w.r.t. $\mathcal{W}$ and ι can be decided in time exponential in the size of ϕ^{p}, linear in the size of $\mathcal{W}$, and polynomial in n. □

However, due to Lemma 5.18, the complexity of the TCQ-satisfiability problem also depend on the complexity of deciding whether $\mathcal{W}$ is r-satisfiable w.r.t. ι and $\mathcal{K}$. This depends again on the fact whether there are concept or role names that are allowed to be rigid.

In the following sections, we establish some results as to this complexity in the cases without rigid names, and with rigid concept and role names. The case without rigid role names, but with rigid concept names, is considered in Section 5.2.3 for data complexity and in Section 5.2.4 for combined complexity.

The Case without Rigid Names

In this section, we consider the case where neither concept names nor role names are allowed to be rigid, i.e. $\mathsf{N}_{\mathsf{RC}} = \mathsf{N}_{\mathsf{RR}} = \emptyset$. We establish the following complexity results.

Theorem 5.21. *If* $\mathsf{N_{RC}} = \mathsf{N_{RR}} = \emptyset$*, the temporalised query-entailment problem in* $\mathcal{SHQ}$ *is*

- *in* ExpTime *w.r.t. combined complexity and*
- *in* co-NP *w.r.t. data complexity.*

Proof. Let ϕ be a Boolean TCQ, and let $\mathcal{K} = ((\mathcal{A}_i)_{0\le i\le n}, \mathcal{T}, \mathcal{R})$ be a temporal $\mathcal{SHQ}$-knowledge base. As argued above, the temporalised query *non-entailment* problem has the same complexity as the TCQ-satisfiability problem. We can decide whether ϕ is satisfiable w.r.t. $\mathcal{K}$ using Lemma 5.18. For that, let $\mathsf{p}\colon \mathsf{CQ}(\phi) \to \mathcal{P}_\phi$ be a bijection.

For combined complexity, we proceed as follows. We define

$$\mathcal{W} := \{X \in 2^{\mathcal{P}_\phi} \mid \zeta_X \text{ is satisfiable w.r.t. } (\emptyset, \mathcal{T}, \mathcal{R})\},$$

where ζ_X is defined as in Definition 5.16. Note that $\mathcal{W} = \{X_1, \dots, X_k\}$ can be constructed in time exponential in the size of ϕ, $\mathcal{T}$, and $\mathcal{R}$. Indeed, there are exponentially many sets $X \in 2^{\mathcal{P}_\phi}$, but each ζ_X can be constructed in time polynomial in the size of ϕ, and is thus of size polynomial in the size of ϕ. By Theorem 5.10, the problem of checking whether the conjunction of CQ-literals ζ_X is satisfiable w.r.t. $(\emptyset, \mathcal{T}, \mathcal{R})$ is ExpTime-complete. Thus, we obtain the set $\mathcal{W}$ after exponentially many ExpTime-tests, i.e. in time exponential in the size of ϕ, $\mathcal{T}$, and $\mathcal{R}$. Moreover, we enumerate all possible mappings $\iota\colon \{0, \dots, n\} \to \{1, \dots, k\}$ in time exponential in the size of ϕ and $\mathcal{K}$. For each such ι and every i, $0 \le i \le n$, we check whether the conjunction of CQ-literals $\zeta_{X_{\iota(i)}}$ is satisfiable w.r.t. $(\mathcal{A}_i, \mathcal{T}, \mathcal{R})$ in time exponential in the size of ϕ and $\mathcal{K}$ (using again Theorem 5.10). After that, we check for ever mapping ι that passes this test, whether ϕ^{p} is t-satisfiable w.r.t. $\mathcal{W}$ and ι, which, by Lemma 5.20, can be done in time exponential in the size of ϕ^{p} (and thus in time exponential in the size of ϕ), linear in the size of $\mathcal{W}$, and polynomial in n.

We now show that for every mapping ι that passes the above tests, we have that $\mathcal{W}$ is r-satisfiable w.r.t. ι and $\mathcal{K}$. Since every ζ_{X_i}, $1 \le i \le k$ is satisfiable w.r.t. $(\emptyset, \mathcal{T}, \mathcal{R})$, there are models $\mathcal{J}_1, \dots, \mathcal{J}_k$ such that every $\mathcal{J}_i$, $1 \le i \le k$, is a model of ζ_{X_i} w.r.t. $(\emptyset, \mathcal{T}, \mathcal{R})$. Moreover, since every $\zeta_{X_{\iota(j)}}$, $0 \le j \le n$, is satisfiable w.r.t. $(\mathcal{A}_j, \mathcal{T}, \mathcal{R})$, there are models $\mathcal{I}_0, \dots, \mathcal{I}_n$ such that every $\mathcal{I}_i$, $0 \le j \le n$, is a model of $\zeta_{X_{\iota(j)}}$ w.r.t. $(\emptyset, \mathcal{T}, \mathcal{R})$. We can assume w.l.o.g. that all of these models have the same domain since we can assume w.l.o.g. that their domains are countably infinite due to to the Löwenheim-Skolem theorem [Löw15; Sko20]. Furthermore, we can assume w.l.o.g. that all individual names are interpreted by the same domain elements in all models. Since $\mathsf{N_{RC}} = \mathsf{N_{RR}} = \emptyset$, this yields that $\mathcal{W}$ is r-satisfiable w.r.t. ι and $\mathcal{K}$.

Thus, we have by Lemma 5.18 that if such a mapping ι exists, then ϕ is satisfiable w.r.t. $\mathcal{K}$. Conversely, again by Lemma 5.18, we have that if ϕ is satisfiable w.r.t. $\mathcal{K}$, then there is a set $\mathcal{W}' = \{X'_1, \dots, X'_{k'}\} \subseteq 2^{\mathcal{P}_\phi}$ and a mapping $\iota'\colon \{0, \dots, n\} \to \{1, \dots, k'\}$ such that $\mathcal{W}'$ is r-satisfiable w.r.t. ι' and $\mathcal{K}$, and ϕ^{p} is t-satisfiable w.r.t. $\mathcal{W}'$ and ι'. The definition of $\mathcal{W}$ above yields that $\mathcal{W}' \subseteq \mathcal{W}$, and thus $k' \le k$. We define the mapping $\iota\colon \{0, \dots, n\} \to \{1, \dots, k\}$ such that $X_{\iota(i)} = X'_{\iota'(i)}$ for every i, $0 \le i \le n$. Hence, we have that $\mathcal{W}$ is r-satisfiable w.r.t. ι and $\mathcal{K}$. Moreover, it is easy to see that the t-satisfiability of ϕ^{p} w.r.t. $\mathcal{W}'$ and ι' implies that ϕ^{p} is t-satisfiable w.r.t. $\mathcal{W}$ and ι.

Hence, we can check whether ϕ is satisfiable w.r.t. $\mathcal{K}$ using the above decision procedure, which shows that the TCQ-satisfiability problem in $\mathcal{SHQ}$ is in ExpTime w.r.t. combined complexity. Since ExpTime is closed under complement, we obtain that the temporalised query-entailment problem is in ExpTime w.r.t. combined complexity.

For data complexity, we non-deterministically guess a set $\mathcal{W} = \{X_1, \dots, X_k\} \subseteq 2^{\mathcal{P}_\phi}$ and a mapping $\iota\colon \{0, \dots, n\} \to \{1, \dots, k\}$. Note that since $\mathcal{P}_\phi$ does not depend on the ABoxes in $\mathcal{K}$, we have that $\mathcal{W}$ is of constant size w.r.t. the ABoxes and ι is of size linear in n. Thus, we can perform these guesses in time polynomial in the size of the ABoxes. Moreover, for checking whether $\mathcal{W}$ is r-satisfiable w.r.t. ι and $\mathcal{K}$, all conditions of Definition 5.16 can be checked in non-deterministic polynomial time w.r.t. data complexity using Theorem 5.10. By Lemma 5.20, deciding whether ϕ^{p} is t-satisfiable w.r.t. $\mathcal{W}$ and ι can be done in time polynomial in n w.r.t. data complexity. Then, by Lemma 5.18, we obtain that the TCQ-satisfiability problem in $\mathcal{SHQ}$ is in NP w.r.t. data complexity. Thus, we obtain that the temporalised query-entailment problem in $\mathcal{SHQ}$ is in CO-NP w.r.t. data complexity. □

Together with Theorem 5.11 and Corollary 5.12, we obtain that the temporalised query-entailment problem in $\mathcal{SHQ}$ is EXPTIME-complete w.r.t. combined complexity and CO-NP-complete w.r.t. data complexity if neither concept nor role names are allowed to be rigid.

The Case of Rigid Concept and Role Names

In this section, we consider the case where both concept and role names may be rigid, i.e. $\mathsf{N_{RC}} \neq \emptyset$ and $\mathsf{N_{RR}} \neq \emptyset$.

Let us assume in the following that a set $\mathcal{W} = \{X_1, \dots, X_k\} \subseteq 2^{\mathcal{P}_\phi}$, and a mapping $\iota\colon \{0, \dots, n\} \to \{1, \dots, k\}$ is given. Note that if concept and role names may be rigid, the satisfiability checks employed in the previous section (see the proof of Theorem 5.21) for deciding whether $\mathcal{W}$ is r-satisfiable w.r.t. ι and $\mathcal{K}$ are no longer independent from each other. To make sure that the models respect the rigid names, we use a renaming technique similar to the one we used in Section 3.2.2, which was adopted from [BGL12]. The difference here is that we have to introduce more copies of the flexible symbols.

For every i, $1 \leq i \leq k + n + 1$, every *flexible* concept name A occurring in $\mathcal{T}$, and every *flexible* role name r occurring in $\mathcal{T}$ or $\mathcal{R}$, we introduce copies $A^{(i)}$ and $r^{(i)}$. We call $A^{(i)}$ the i-th copy of A, and similarly $r^{(i)}$ the i-th copy of r. The conjunctive query $\alpha^{(i)}$ (the axiom $\beta^{(i)}$) is obtained from a conjunctive query α (an axiom β) by replacing every occurrence of a flexible name by its i-th copy. Similarly, for $1 \leq \ell \leq k$, the conjunction of CQ-literals $\zeta^{(i)}_{X_\ell}$ is obtained from ζ_{X_ℓ} (see Definition 5.16) by replacing each CQ α occurring in ζ_{X_ℓ} by $\alpha^{(i)}$. Finally, we define

$$\begin{aligned}
\zeta_{\mathcal{W},\iota} &:= \bigwedge_{1 \leq i \leq k} \zeta^{(i)}_{X_i} \wedge \bigwedge_{0 \leq i \leq n} \Big(\zeta^{(k+i+1)}_{X_{\iota(i)}} \wedge \bigwedge_{\alpha \in \mathcal{A}_i} \alpha^{(k+i+1)} \Big), \\
\mathcal{T}_{\mathcal{W},\iota} &:= \big\{ \beta^{(i)} \mid \beta \in \mathcal{T} \text{ and } 1 \leq i \leq k + n + 1 \big\}, \text{ and} \\
\mathcal{R}_{\mathcal{W},\iota} &:= \big\{ \gamma^{(i)} \mid \gamma \in \mathcal{R} \text{ and } 1 \leq i \leq k + n + 1 \big\}.
\end{aligned}$$

Note that in the definition of $\zeta_{\mathcal{W},\iota}$ it is essential that the ABoxes do not contain complex concepts, otherwise they could not be viewed as sets of conjunctive queries, and hence $\zeta_{\mathcal{W},\iota}$ would not be a conjunction of CQ-literals.

Lemma 5.22. *The set $\mathcal{W}$ is r-satisfiable w.r.t. ι and $\mathcal{K}$ iff the conjunction of CQ-literals $\zeta_{\mathcal{W},\iota}$ is satisfiable w.r.t. the knowledge base $(\mathcal{T}_{\mathcal{W},\iota}, \mathcal{R}_{\mathcal{W},\iota})$.*

Proof. For the 'only if' direction, let $\mathcal{J}_1 = (\Delta, \cdot^{\mathcal{J}_1}), \dots, \mathcal{J}_k = (\Delta, \cdot^{\mathcal{J}_k})$, and $\mathcal{I}_0 = (\Delta, \cdot^{\mathcal{I}_0}), \dots, \mathcal{I}_n = (\Delta, \cdot^{\mathcal{I}_n})$ be the interpretations required by Definition 5.16 for the r-satisfiability of $\mathcal{W}$ w.r.t. ι and $\mathcal{K}$. We construct the interpretation $\mathcal{J} = (\Delta, \cdot^{\mathcal{J}})$ as follows:

- every individual name and every rigid name is interpreted as in $\mathcal{J}_1$;
- the i-th copy, $1 \leq i \leq k$, of each flexible name is interpreted like the original name in $\mathcal{J}_i$; and
- the i-th copy, $k+1 \leq i \leq k+n+1$, of each flexible name is interpreted like the original name in $\mathcal{I}_{i-k-1}$.

It is easy to verify that $\mathcal{J}$ is a model of $\zeta_{\mathcal{W},\iota}$ and $(\mathcal{T}_{\mathcal{W},\iota}, \mathcal{R}_{\mathcal{W},\iota})$.

For the 'if' direction, let $\mathcal{J}$ be a model of $\zeta_{\mathcal{W},\iota}$ w.r.t. $(\mathcal{T}_{\mathcal{W},\iota}, \mathcal{R}_{\mathcal{W},\iota})$. We obtain the interpretations $\mathcal{J}_1, \dots, \mathcal{J}_k$, and $\mathcal{I}_0, \dots, \mathcal{I}_n$ by the inverse construction to the one above:

- the domain of all these interpretations is the domain of $\mathcal{J}$;
- every individual name and every rigid name is interpreted by these interpretations as in $\mathcal{J}$;
- every flexible name is interpreted in $\mathcal{J}_i$, $1 \leq i \leq k$, as its i-th copy is interpreted in $\mathcal{J}$; and
- every flexible name is interpreted in $\mathcal{I}_i$, $0 \leq i \leq n$, as its $(k+i+1)$-st copy is interpreted in $\mathcal{J}$.

Again, it is easy to verify that these interpretations satisfy the conditions for r-satisfiability of $\mathcal{W}$ w.r.t. ι and $\mathcal{K}$. □

Using this lemma, we can prove the following complexity results.

Theorem 5.23. *If $\mathsf{N_{RC}} \neq \emptyset$ and $\mathsf{N_{RR}} \neq \emptyset$, the temporalised query-entailment problem in $\mathcal{SHQ}$ is*

- *in* 2ExpTime *w.r.t. combined complexity and*
- *in* ExpTime *w.r.t. data complexity.*

Proof. Let ϕ be a Boolean TCQ, and let $\mathcal{K} = ((\mathcal{A}_i)_{0 \leq i \leq n}, \mathcal{T}, \mathcal{R})$ be a temporal $\mathcal{SHQ}$-knowledge base. We again consider the TCQ-satisfiability problem, which has the same complexity as the temporalised query non-entailment problem. We decide whether ϕ is satisfiable w.r.t. $\mathcal{K}$ using Lemma 5.18. For that, let $\mathsf{p}\colon \mathsf{CQ}(\phi) \to \mathcal{P}_\phi$ be a bijection. We first enumerate all sets $\mathcal{W} = \{X_1, \dots, X_k\} \subseteq 2^{\mathcal{P}_\phi}$ and all mappings $\iota\colon \{0, \dots, n\} \to \{1, \dots, k\}$, which can be done in time doubly exponential in the size of ϕ and exponential in n.

For every such pair $(\mathcal{W}, \iota)$, we check t-satisfiability of $\phi^{\mathsf{p}}_{\mathcal{W}}$ w.r.t. $\mathcal{W}$ and ι in time exponential in the size of ϕ^{p} (and thus in time exponential in the size of ϕ), linear in the size of $\mathcal{W}$, and polynomial in n (by Lemma 5.20), and check $\mathcal{W}$ for r-satisfiability w.r.t. ι and $\mathcal{K}$. By Lemma 5.18, ϕ has a model w.r.t. $\mathcal{K}$ iff at least one pair passes both tests.

For the r-satisfiability check, we use Lemma 5.22. We construct the conjunction of CQ-literals $\zeta_{\mathcal{W},\iota}$ and the knowledge base $(\mathcal{T}_{\mathcal{W},\iota}, \mathcal{R}_{\mathcal{W},\iota})$, which can be done in time exponential in the size of ϕ, $\mathcal{T}$, and $\mathcal{R}$, and in time linear in the size of $\mathcal{A}_1, \dots, \mathcal{A}_n$. Moreover, the size of $\zeta_{\mathcal{W},\iota}$ and $(\mathcal{T}_{\mathcal{W},\iota}, \mathcal{R}_{\mathcal{W},\iota})$ is at most exponential in the size of ϕ, $\mathcal{T}$, and $\mathcal{R}$, and linear in the size of $\mathcal{A}_1, \dots, \mathcal{A}_n$.

By Theorem 5.10 we can check whether $\zeta_{\mathcal{W},\iota}$ is satisfiable w.r.t. $(\mathcal{T}_{\mathcal{W},\iota}, \mathcal{R}_{\mathcal{W},\iota})$ in time *doubly exponential* in the size of ϕ, $\mathcal{T}$, and $\mathcal{R}$, and *exponential* in the size of $\mathcal{A}_1, \dots, \mathcal{A}_n$. Using this decision procedure, we can check whether ϕ is satisfiable w.r.t. $\mathcal{K}$. Thus, the TCQ-satisfiability problem in $\mathcal{SHQ}$ is in 2ExpTime w.r.t. combined complexity and in ExpTime w.r.t. data complexity. Since both 2ExpTime and ExpTime are closed under complement, we obtain that the temporalised query-entailment problem is in 2ExpTime w.r.t. combined complexity and in ExpTime w.r.t. data complexity. □

Together with Theorem 5.11 and Corollary 5.12, we obtain that the temporalised query-entailment problem in $\mathcal{SHQ}$ is 2ExpTime-complete w.r.t. combined complexity, and co-NP-hard and in ExpTime w.r.t. data complexity if both concept and role names may be rigid.

Unfortunately, the above approach does not allow us to match the lower bound for data complexity, and thus leaves a gap in the data complexity results. As seen in the proof of Theorem 5.23, the issue is that the size of $\zeta_{\mathcal{W},\iota}$ depends on n. More precisely, recall that constructing $\zeta_{\mathcal{W},\iota}$ involves copying the type $\zeta_{X_{\iota(i)}}$ assigned to the ABox $\mathcal{A}_i$ for every i, $1 \leq i \leq n$. Thus, we introduce linearly many *negated* CQs in $\zeta_{\mathcal{W},\iota}$, and Theorem 5.10 yields only an upper bound of ExpTime for the satisfiability problem. Note that linearly many non-negated CQs in $\zeta_{\mathcal{W},\iota}$ are not problematic, as they can be instantiated and viewed as part of the ABox, as detailed in the proof of Theorem 5.10. However, we can match the lower bound of co-NP for the data complexity in the following special cases.

Lemma 5.24. *If $\mathsf{N}_{\mathsf{RC}} \neq \emptyset$ and $\mathsf{N}_{\mathsf{RR}} \neq \emptyset$, the temporalised query-entailment problem in $\mathcal{SHQ}$ is in* co-NP *w.r.t. data complexity if any of the following conditions apply:*

1. *The number n of the input ABoxes is bounded by a constant.*
2. *The set of individual names allowed to occur in the input ABoxes is fixed.*

Proof. As done in the proof of Theorem 5.21, we decide the TCQ-satisfiability problem as follows. We first non-deterministically guess a set $\mathcal{W} = \{X_1, \dots, X_k\} \subseteq 2^{\mathcal{P}_\phi}$ and a mapping $\iota \colon \{0, \dots, n\} \to \{1, \dots, k\}$ in time polynomial in the size of the input ABoxes. By Lemma 5.20, deciding whether ϕ^{p} is t-satisfiable w.r.t. $\mathcal{W}$ and ι can be done in time polynomial in n w.r.t. data complexity. Thus, due to Lemma 5.18, is suffices to show that in the above mentioned special cases r-satisfiability of $\mathcal{W}$ w.r.t. ι and $\mathcal{K}$ can be checked in non-deterministic polynomial time w.r.t. data complexity. For that, we use again Lemma 5.22.

1. If n is bounded by a constant, then the number of negated CQs in $\zeta_{\mathcal{W},\iota}$ is constant, and thus Theorem 5.10 yields the desired upper bound of NP for the TCQ-satisfiability problem.
2. If the set of individual names is fixed, then the number of possible assertions involving concept names opccurring in the TBox is constant. Note that the concept names occurring only in the ABoxes do not affect the entailment of the TCQ, as they can only occur in *positive* assertions, and can thus always be satisfied by appropriately interpreting the new names. This allows us to restrict the formula $\zeta_{\mathcal{W},\iota}$ to contain at most one copy of $\zeta_{X_{\iota(i)}}$ for each distinct combination of $\zeta_{X_{\iota(i)}}$ and $\mathcal{A}_i$ (ignoring

assertions about names that do not occur in the TBox). Clearly, the satisfiability of each combination of an ABox with such a conjunction of CQ-literals need to be checked only once. Since there are only constantly many such combinations, the modified TCQ $\zeta'_{\mathcal{W},\iota}$ again contains only constantly many negated CQs. As in the previous case, Theorem 5.10 yields again that the TCQ-satisfiability problem is in NP.

Since the TCQ-satisfiability problem has the same complexity as the temporalised query non-entailment problem, we obtain the desired complexity results. □

It is still open, however, where any of these conditions is necessary.

5.2.3 Data Complexity for the Case of Rigid Concept Names

In this section, we consider the case where only concept names are allowed to be rigid, i.e. $\mathsf{N_{RC}} \neq \emptyset$ and $\mathsf{N_{RR}} = \emptyset$. We show that, in this case, the temporalised query-entailment problem is in CO-NP w.r.t. data complexity.

For that, we again first assume that a set $\mathcal{W} = \{X_1, \ldots, X_k\} \subseteq 2^{\mathcal{P}_\phi}$ and a mapping $\iota\colon \{0, \ldots, n\} \to \{1, \ldots, k\}$ are given. We first show how to decide r-satisfiability of $\mathcal{W}$ w.r.t. ι and $\mathcal{K}$ in non-deterministic polynomial time w.r.t. data complexity.

Similar to what we did in the previous sections, we construct conjunctions of CQ-literals which we check for satisfiability. The approach is a mixture of those of employed for the case without rigid names and for the case with rigid concept and role names. More precisely, we combine several satisfiability checks required for r-satisfiability, but we do not go as far as compiling all of them into just one conjunction as done to obtain Lemma 5.22. For that we use the ideas of Section 3.2.3 and of the proof of Lemma 6.3 in [BGL12].

We consider the conjunctions of CQ-literals $\xi_i \wedge \zeta_{\mathcal{W}}$, $0 \leq i \leq n$, and the knowledge base $(\mathcal{T}_{\mathcal{W}}, \mathcal{R}_{\mathcal{W}})$, where

$$\xi_i := \bigwedge_{\alpha \in \mathcal{A}_i} \alpha^{(\iota(i))}, \qquad \zeta_{\mathcal{W}} := \bigwedge_{1 \leq i \leq k} \zeta^{(i)}_{X_i},$$
$$\mathcal{T}_{\mathcal{W}} := \big\{\beta^{(i)} \mid \beta \in \mathcal{T} \text{ and } 1 \leq i \leq k\big\}, \qquad \mathcal{R}_{\mathcal{W}} := \big\{\gamma^{(i)} \mid \gamma \in \mathcal{R} \text{ and } 1 \leq i \leq k\big\}.$$

However, for r-satisfiability we have to make sure that rigid consequences of the form $A(a)$ for a rigid concept name $A \in \mathsf{N_{RC}}$ and an individual name $a \in \mathsf{N_I}$ are shared between all of these conjunctions $\xi_i \wedge \zeta_{\mathcal{W}}$. It suffices to do this for the set $\mathsf{RCon}(\mathcal{T})$ of rigid concept names occurring in $\mathcal{T}$ since those occurring only in ABox-assertions cannot affect the entailment of the TCQ ϕ.

Let $\mathcal{D} = (\mathsf{RCon}(\mathcal{T}), \mathcal{Y})$ with $\mathcal{Y} \subseteq 2^{\mathsf{RCon}(\mathcal{T})}$ be arbitrary, and let τ be a mapping from $\mathsf{Ind}(\phi) \cup \mathsf{Ind}(\mathcal{K})$ to $\mathcal{Y}$. Recall that as in Definition 3.18, the idea is that $\mathcal{D}$ fixes the combinations of rigid concept names that are allowed to occur in the models of $\xi_i \wedge \zeta_{\mathcal{W}}$, $0 \leq i \leq n$. The mapping τ assigns to each individual name occurring in ϕ or $\mathcal{K}$ one such combination. To express this formally, we extend the TBox by the axioms in

$$\mathcal{T}_\tau := \big\{A_{\tau(a)} \equiv C_{\mathsf{RCon}(\mathcal{T}),\tau(a)} \mid a \in \mathsf{Ind}(\phi) \cup \mathsf{Ind}(\mathcal{K})\big\},$$

where $A_{\tau(a)}$ are fresh rigid concept names and, for every $Y \subseteq \mathsf{RCon}(\mathcal{T})$, the concept $C_{\mathsf{RCon}(\mathcal{T}),Y}$ is defined as in Definition 3.18. Correspondingly, we extend the conjunctions $\xi_i \wedge \zeta_{\mathcal{W}}$ by

$$\xi_\tau := \bigwedge_{a\in\mathsf{Ind}(\phi)\cup\mathsf{Ind}(\mathcal{K})} A_{\tau(a)}$$

in order to fix the behaviour of the rigid concept names on the named individuals.

The next lemma states how these notions can be used to characterise r-satisfiability of $\mathcal{W}$ w.r.t. ι and $\mathcal{K}$. Its proof is very similar to the proof of Lemma 3.19.

Lemma 5.25. *If $\mathsf{N_{RC}} \neq \emptyset$ and $\mathsf{N_{RR}} = \emptyset$, then $\mathcal{W}$ is r-satisfiable w.r.t. ι and $\mathcal{K}$ iff there exist a pair $\mathcal{D} = (\mathsf{RCon}(\mathcal{T}), \mathcal{Y})$ with $\mathcal{Y} \subseteq 2^{\mathsf{RCon}(\mathcal{T})}$ and a mapping $\tau \colon \mathsf{Ind}(\phi)\cup\mathsf{Ind}(\mathcal{K}) \to \mathcal{Y}$ such that for every i, $0 \leq i \leq n$, the conjunction of CQ-literals $\xi_i \wedge \zeta_{\mathcal{W}} \wedge \xi_\tau$ has a model w.r.t. $(\mathcal{T}_{\mathcal{W}} \cup \mathcal{T}_\tau, \mathcal{R}_{\mathcal{W}})$ that respects $\mathcal{D}$.*

Proof. For the 'if' direction, assume that $\mathcal{I}_i$, $0 \leq i \leq n$, are the models of $\xi_i \wedge \zeta_{\mathcal{W}} \wedge \xi_\tau$ w.r.t. $(\mathcal{T}_{\mathcal{W}} \cup \mathcal{T}_\tau, \mathcal{R}_{\mathcal{W}})$, respectively, that respect $\mathcal{D}$; see Definition 3.18. Similar to the proof of Lemma 3.19 (and Lemma 6.3 in [BGL12]), we can assume w.l.o.g. that their domains Δ_i are countably infinite and for each $Y \in \mathcal{Y}$ there are countably infinitely many elements $d \in (C_{\mathsf{RCon}(\mathcal{T}),Y})^{\mathcal{I}_i}$, where $C_{\mathsf{RCon}(\mathcal{T}),Y}$ is defined in Definition 3.18. This is a consequence of the Löwenheim-Skolem theorem [Löw15; Sko20] and the fact that the countably infinite disjoint union of $\mathcal{I}_i$ with itself is again a model of $\xi_i \wedge \zeta_{\mathcal{W}} \wedge \xi_\tau$ w.r.t. $(\mathcal{T}_{\mathcal{W}} \cup \mathcal{T}_\tau, \mathcal{R}_{\mathcal{W}})$. The latter follows from the observation that for every CQ ψ, there is a homomorphism of ψ into $\mathcal{I}_i$ iff there is a homomorphism of ψ into the disjoint union of $\mathcal{I}_i$ with itself. One direction is trivial, while whenever there is a homomorphism of ψ into the disjoint union of $\mathcal{I}_i$ with itself, we can construct a homomorphism of ψ into $\mathcal{I}_i$ by replacing the elements in the image of this homomorphism by the corresponding elements of Δ_i. It is easy to see that the resulting homomorphism still satisfies all atoms of the CQ ψ.

Consequently, we can partition the domains Δ_i into the countably infinite sets

$$\Delta_i(Y) := \left\{ d \in \Delta_i \mid d \in (C_{\mathsf{RCon}(\mathcal{T}),Y})^{\mathcal{I}_i} \right\}$$

for $Y \in \mathcal{Y}$. By the assumptions above and the fact that every $\mathcal{I}_i$ satisfies ξ_τ and $\mathcal{T}_\tau$, there are bijections $\pi_i \colon \Delta_0 \to \Delta_i$, $1 \leq i \leq n$, such that

- $\pi_i(\Delta_0(Y)) = \Delta_i(Y)$ for every $Y \in \mathcal{Y}$, and
- $\pi_i(a^{\mathcal{I}_0}) = a^{\mathcal{I}_i}$ for every $a \in \mathsf{Ind}(\phi) \cup \mathsf{Ind}(\mathcal{K})$.

Thus, we can assume in the following that the models $\mathcal{I}_i$, $0 \leq i \leq n$, actually share the same domain Δ and interpret the concept names in $\mathsf{RCon}(\mathcal{T})$ and the individual names occurring in ϕ or $\mathcal{K}$ in the same way. We can now construct the interpretations required by Definition 5.16 by appropriately relating the flexible names and their copies. For every j, $1 \leq j \leq k$, we define $\mathcal{J}_j = (\Delta, \cdot^{\mathcal{J}_j})$ by interpreting the concept names in $\mathsf{RCon}(\mathcal{T})$ and the individual names occurring in ϕ or $\mathcal{K}$ as in $\mathcal{I}_0$, and the flexible names as their j-th copies in $\mathcal{I}_0$. Since $\mathcal{I}_0$ is a model of $\zeta_{\mathcal{W}}$ and $(\mathcal{T}_{\mathcal{W}}, \mathcal{R}_{\mathcal{W}})$, we have that $\mathcal{J}_j$ is a model of ζ_{X_j}, $\mathcal{T}$, and $\mathcal{R}$.

Similarly, for every i, $0 \leq i \leq n$, we define $\mathcal{I}'_i = (\Delta, \cdot^{\mathcal{I}'_i})$ by interpreting the concept names in $\mathsf{RCon}(\mathcal{T})$ and the individual names occurring in ϕ or $\mathcal{K}$ as in $\mathcal{I}_i$, and the flexible names as their $\iota(i)$-th copies in $\mathcal{I}_i$. Since $\mathcal{I}_i$ is a model of ξ_i, $\zeta_{\mathcal{W}}$, and $(\mathcal{T}_{\mathcal{W}}, \mathcal{R}_{\mathcal{W}})$, we obtain that $\mathcal{I}_i$ is a model of $\mathcal{A}_i$, $\zeta_{X_{\iota(i)}}$, $\mathcal{T}$, and $\mathcal{R}$. All these models share the same domain and interpret the rigid concept names in $\mathsf{RCon}(\mathcal{T})$ and the individual names occurring in ϕ or $\mathcal{K}$ in the same way. Note that the interpretation of the names that occur neither $\mathcal{K}$ nor in in ϕ is irrelevant and can be fixed arbitrarily, as long as the UNA is satisfied.

Thus, it remains to consider those rigid concept names A occurring in $(\mathcal{A}_i)_{0\leq i\leq n}$, but not in $\mathcal{T}$. Since they are not constrained by the TBox, it suffices to interpret them in such a way that they satisfy all ABox-assertions. But since these assertions can only occur positively in the ABoxes, the set $\{a^{\mathcal{I}_0} \mid A(a) \in \mathcal{A}_i,\ 0 \leq i \leq n\}$ fulfils this restriction. Thus, the conditions required for r-satisfiability of $\mathcal{W}$ w.r.t. ι and $\mathcal{K}$ by Definition 5.16 are satisfied.

For the 'only if' direction, assume that $\mathcal{J}_j = (\Delta, \cdot^{\mathcal{J}_j})$, $1 \leq j \leq k$, and $\mathcal{I}_i = (\Delta, \cdot^{\mathcal{I}_i})$, $0 \leq i \leq n$, are the interpretations required for r-satisfiability of $\mathcal{W}$ w.r.t. ι and $\mathcal{K}$ by Definition 5.16. It is easy to see that for every i, $0 \leq i \leq n$, one can combine the interpretations $\mathcal{I}_i$, $\mathcal{J}_1, \ldots, \mathcal{J}_k$ to obtain a model $\mathcal{I}'_i$ of $\xi_i \wedge \zeta_{\mathcal{W}}$ w.r.t. $(\mathcal{T}_{\mathcal{W}}, \mathcal{R}_{\mathcal{W}})$ by interpreting the $\iota(i)$-th copy of a flexible name as the original name in $\mathcal{I}_i$, and the j-th copy of a flexible name as the original name in $\mathcal{J}_j$ for each j, $1 \leq j \leq k$, with $j \neq \iota(i)$. Obviously, the interpretations $\mathcal{I}'_0, \ldots, \mathcal{I}'_n$ share the same domain, interpret individual names in the same way, and respect rigid concept names. Thus, for every $Y \subseteq \mathsf{RCon}(\mathcal{T})$, we have that $(C_{\mathsf{RCon}(\phi),Y})^{\mathcal{I}'_0} = (C_{\mathsf{RCon}(\phi),Y})^{\mathcal{I}'_i}$ for every i, $1 \leq i \leq n$. We define $\mathcal{D} := (\mathsf{RCon}(\mathcal{T}), \mathcal{Y})$ with

$$\mathcal{Y} := \{Y \subseteq \mathsf{RCon}(\phi) \mid \text{there is some } d \in \Delta \text{ with } d \in (C_{\mathsf{RCon}(\mathcal{T}),Y})^{\mathcal{I}'_0}\}.$$

By construction of $\mathcal{D}$, we obtain that the interpretations $\mathcal{I}'_i$, $0 \leq i \leq n$, respect $\mathcal{D}$. Moreover, for every $a \in \mathsf{Ind}(\phi) \cup \mathsf{Ind}(\mathcal{K})$, we define $\tau(a) := Y \subseteq \mathsf{RCon}(\mathcal{T})$ iff $a \in (C_{\mathsf{RCon}(\mathcal{T}),Y})^{\mathcal{I}'_0}$, which ensures that the interpretations $\mathcal{I}'_i$ can be extended to models of ξ_τ and $\mathcal{T}_\tau$ by appropriately interpreting the new concept names $A_{\tau(a)}$. Hence, we obtain models of $\xi_i \wedge \zeta_{\mathcal{W}} \wedge \xi_\tau$ w.r.t. $(\mathcal{T}_{\mathcal{W}} \cup \mathcal{T}_\tau, \mathcal{R}_{\mathcal{W}})$ that respect $\mathcal{D}$ as required. □

Using this lemma, we can prove our complexity result.

Theorem 5.26. *If* $\mathsf{N_{RC}} \neq \emptyset$ *and* $\mathsf{N_{RR}} = \emptyset$, *the temporalised query-entailment problem in* $\mathcal{SHQ}$ *is in* CO-NP *w.r.t. data complexity.*

Proof. Let ϕ be a Boolean TCQ, and let $\mathcal{K} = ((\mathcal{A}_i)_{0\leq i\leq n}, \mathcal{T}, \mathcal{R})$ be a temporal $\mathcal{SHQ}$-knowledge base. We again consider the TCQ-satisfiability problem, which has the same complexity as the temporalised query non-entailment problem. We decide whether ϕ is satisfiable w.r.t. $\mathcal{K}$ using Lemma 5.18. For that, let $\mathsf{p}\colon \mathsf{CQ}(\phi) \to \mathcal{P}_\phi$ be a bijection. We first non-deterministically guess a set $\mathcal{W} = \{X_1, \ldots, X_k\} \subseteq 2^{\mathcal{P}_\phi}$ and a mapping $\iota\colon \{0, \ldots, n\} \to \{1, \ldots, k\}$ in time polynomial in the size of the input ABoxes. By Lemma 5.20, deciding whether ϕ^{p} is t-satisfiable w.r.t. $\mathcal{W}$ and ι can be done in time polynomial in n w.r.t. data complexity.

Thus, due to Lemma 5.18, is suffices to show that r-satisfiability of $\mathcal{W}$ w.r.t. ι and $\mathcal{K}$ can be checked in non-deterministic polynomial time w.r.t. data complexity. For that, we use Lemma 5.25. We non-deterministically guess a set $\mathcal{Y} \subseteq 2^{\mathsf{RCon}(\mathcal{T})}$ and a mapping $\tau\colon \mathsf{Ind}(\phi) \cup \mathsf{Ind}(\mathcal{K}) \to \mathcal{Y}$, which can be done in time polynomial in the size of the input ABoxes. Indeed, $\mathcal{Y}$ only depends on $\mathcal{T}$, and τ is of size linear in the size of the input ABoxes. We define $\mathcal{D} := (\mathsf{RCon}(\mathcal{T}), \mathcal{Y})$. Next, we construct for every i, $0 \leq i \leq n$, the conjunction of CQ-literals $\xi_i \wedge \zeta_{\mathcal{W}} \wedge \xi_\tau$ and the knowledge base $(\mathcal{T}_{\mathcal{W}} \cup \mathcal{T}_\tau, \mathcal{R}_{\mathcal{W}})$. Note that ξ_i and ξ_τ are of size polynomial in the size of the input ABoxes, and that the sizes of $\zeta_{\mathcal{W}}$, $\mathcal{T}_{\mathcal{W}}$, $\mathcal{T}_\tau$, and $\mathcal{R}_{\mathcal{W}}$ do not depend on the input ABoxes. Furthermore, only $\zeta_{\mathcal{W}}$ may contain negated CQs, and thus their size does not depend on the size of the input ABoxes. It remains to show that we can check the existence of a model of $\xi_i \wedge \zeta_{\mathcal{W}} \wedge \xi_\tau$ w.r.t. $(\mathcal{T}_{\mathcal{W}} \cup \mathcal{T}_\tau, \mathcal{R}_{\mathcal{W}})$ that respects $\mathcal{D}$ in non-deterministic polynomial time w.r.t. data complexity. For that, observe that the restriction imposed by $\mathcal{D}$ can equivalently be expressed as the conjunction of CQ-literals

$$\zeta_{\mathcal{D}} := \big(\neg\exists x.A_{\mathcal{D}}(x)\big) \wedge \bigwedge_{Y \in \mathcal{Y}} \exists x.A_Y(x),$$

where $A_{\mathcal{D}}$ and A_Y are fresh concept names that are restricted by adding the following axioms to the TBox: $A_Y \equiv C_{\mathsf{RCon}(\mathcal{T}),Y}$ and $C_{\mathsf{RCon}(\mathcal{T}),Y} \sqsubseteq A_Y$ for every $Y \in \mathcal{Y}$, and $A_{\mathcal{D}} \equiv \bigsqcap_{Y \in \mathcal{D}} \neg A_Y$.[4] We denote by $\mathcal{T}'_{\mathcal{W}}$ the resulting extension of the TBox $\mathcal{T}_{\mathcal{W}} \cup \mathcal{T}_\tau$. Thus, it is enough to check whether $\xi_i \wedge \zeta_{\mathcal{W}} \wedge \xi_\tau \wedge \zeta_{\mathcal{D}}$ has a model w.r.t. $(\mathcal{T}'_{\mathcal{W}}, \mathcal{R}_{\mathcal{W}})$. Note that neither $\zeta_{\mathcal{D}}$ nor $\mathcal{T}'_{\mathcal{W}}$ depend on the input ABoxes. Hence, one can see from the proof of Theorem 5.10 that this satisfiability problem can be decided in non-deterministic polynomial time w.r.t. data complexity. Thus, we obtain that the TCQ-satisfiability problem in $\mathcal{SHQ}$ is in NP w.r.t. data complexity, which shows that the temporalised query-entailment problem in $\mathcal{SHQ}$ is in co-NP w.r.t. data complexity.□

Together with Corollary 5.12, this yields that the temporalised query-entailment problem in $\mathcal{SHQ}$ is co-NP-complete w.r.t. data complexity if only concept names are allowed to be rigid.

5.2.4 Combined Complexity for the Case of Rigid Concept Names

In this section, we again consider the case where only concept names are allowed to be rigid, i.e. $\mathsf{N_{RC}} \neq \emptyset$ and $\mathsf{N_{RR}} = \emptyset$. However, we consider the combined complexity of the temporalised query-entailment problem. Unfortunately, the approach used in the previous section does not yield a combined complexity of co-NExpTime. The reason is that the conjunctions of CQ-literals $\zeta_{\mathcal{W}}$ and $\zeta_{\mathcal{D}}$ are of size exponential in the size of ϕ, and thus Theorem 5.10 only yields an upper bound of 2ExpTime. Therefore, we describe a different approach with a combined complexity of co-NExpTime.

As a first step, we rewrite the Boolean TCQ ϕ into a Boolean TCQ ψ of size polynomial in the size of ϕ and the temporal KB $\mathcal{K}$ such that answering ϕ at time n w.r.t. $\mathcal{K}$ is equivalent to answering ψ at time 0 w.r.t. a temporal KB containing only a trivial sequence of ABoxes. This is done by compiling the ABoxes into the query and postponing the query ϕ using the X-operator.

Lemma 5.27. *Let $\mathcal{K} = ((\mathcal{A}_i)_{0 \leq i \leq n}, \mathcal{T}, \mathcal{R})$ be a temporal KB and ϕ be a Boolean TCQ. Then there is a Boolean TCQ ψ of size polynomial in the size of ϕ and $\mathcal{K}$ such that $\mathcal{K} \models \phi$ iff $(\emptyset, \mathcal{T}, \mathcal{R}) \models \psi$.*

Proof. We define the Boolean TCQ

$$\psi := (\gamma_0 \wedge \mathsf{X}\,\gamma_1 \wedge \cdots \wedge \mathsf{X}^n\,\gamma_n) \rightarrow \mathsf{X}^n\,\phi,$$

[4] We did not add all the axioms $A_Y \equiv C_{\mathsf{RCon}(\mathcal{T}),Y}$ earlier since we reuse Lemma 5.25 in the following section about combined complexity, and these additional axioms cause an exponential blow-up in the size of the TBox.

where $\gamma_i := \bigwedge \mathcal{A}_i$ for i, $0 \leq i \leq n$, and X^j abbreviates j nested X-operators. Obviously, the size of ψ is polynomial in the size of ϕ and $\mathcal{K}$. It remains to prove that $\mathcal{K} \models \phi$ iff $\mathcal{K}' := (\emptyset, \mathcal{T}, \mathcal{R}) \models \psi$. We have:

$$\begin{aligned}
&\mathcal{K} \models \phi \\
\textit{iff}\quad &((\mathcal{A}_i)_{0\leq i\leq n}, \mathcal{T}, \mathcal{R}) \models \phi \\
\textit{iff}\quad &\mathfrak{I}, n \models \phi \text{ for every } \mathfrak{I} \text{ with } \mathfrak{I} \models ((\mathcal{A}_i)_{0\leq i\leq n}, \mathcal{T}, \mathcal{R}) \\
\textit{iff}\quad &\mathfrak{I}, n \models \phi \text{ for every } \mathfrak{I} \text{ with } \mathfrak{I} \models (\emptyset, \mathcal{T}, \mathcal{R}) \text{ and } \mathfrak{I}, 0 \models \gamma_0;\ \mathfrak{I}, 1 \models \gamma_1;\ \dots;\ \mathfrak{I}, n \models \gamma_n \\
\textit{iff}\quad &\mathfrak{I}, 0 \models \mathsf{X}^n \phi \text{ for every } \mathfrak{I} \text{ with } \mathfrak{I} \models \mathcal{K}' \text{ and } \mathfrak{I}, 0 \models \gamma_0;\ \mathfrak{I}, 0 \models \mathsf{X}\gamma_1;\ \dots;\ \mathfrak{I}, 0 \models \mathsf{X}^n \gamma_n \\
\textit{iff}\quad &\mathfrak{I}, 0 \models \psi \text{ for every } \mathfrak{I} \text{ with } \mathfrak{I} \models \mathcal{K}' \\
\textit{iff}\quad &\mathcal{K}' \models \psi.
\end{aligned}$$

□

To obtain our complexity result, we can thus focus on deciding whether a Boolean TCQ ϕ has a model w.r.t. a temporal KB $\mathcal{K} = (\emptyset, \mathcal{T}, \mathcal{R})$ containing only one empty ABox, i.e. we have $n = 0$. Note that this compilation approach does not yield a low *data complexity* for the TCQ-satisfiability problem since, after encoding the ABoxes into ϕ, the size of the conjunction of CQ-literals $\zeta_{\mathcal{W}}$ is exponential in the size of the input ABoxes. Moreover, then Lemma 5.20 yields that the t-satisfiability check is exponential in the size of the input ABoxes.

Assume from now on that a set $\mathcal{W} = \{X_1, \dots, X_k\} \subseteq 2^{\mathcal{P}_\phi}$ and a mapping $\iota \colon \{0\} \to \{1, \dots, k\}$ are given. We first show how to decide r-satisfiability of $\mathcal{W}$ w.r.t. ι and $\mathcal{K}$ in non-deterministic exponential time w.r.t. combined complexity.

For that, we use the idea of Lemma 5.25. Since $\xi_0 = \mathsf{true}$, according to this lemma, it suffices to non-deterministically guess a pair $\mathcal{D}$ and a mapping τ such that $\zeta_{\mathcal{W}} \wedge \xi_\tau$ has a model w.r.t. $(\mathcal{T}_{\mathcal{W}} \cup \mathcal{T}_\tau, \mathcal{R}_{\mathcal{W}})$ that respects $\mathcal{D}$. Instead of constructing the conjunction of CQ-literals $\zeta_{\mathcal{D}}$, and then applying Theorem 5.10 directly to this problem, which would yield a complexity of 2ExpTime, we split the problem into separate sub-problems for each ζ_{X_i}.

Lemma 5.28. *If $\mathsf{N_{RC}} \neq \emptyset$ and $\mathsf{N_{RR}} = \emptyset$, then $\mathcal{W}$ is r-satisfiable w.r.t. ι and $\mathcal{K} = (\emptyset, \mathcal{T}, \mathcal{R})$ iff there exist a pair $\mathcal{D} = (\mathsf{RCon}(\mathcal{T}), \mathcal{Y})$ with $\mathcal{Y} \subseteq 2^{\mathsf{RCon}(\mathcal{T})}$ and a mapping $\tau \colon \mathsf{Ind}(\phi) \to \mathcal{Y}$ such that for every i, $1 \leq i \leq k$, the conjunction of CQ-literals $\zeta_{X_i} \wedge \xi_\tau$ has a model w.r.t. $(\mathcal{T} \cup \mathcal{T}_\tau, \mathcal{R})$ that respects $\mathcal{D}$.*

Proof. For the 'if' direction, assume that $\mathcal{I}_i$, $1 \leq i \leq k$, are the models of $\zeta_{X_i} \wedge \xi_\tau$ w.r.t. $(\mathcal{T} \cup \mathcal{T}_\tau, \mathcal{R})$ that respect $\mathcal{D}$. As in the proof of Lemma 5.25, we can ensure that they share the same domain and interpret the rigid concept names in $\mathsf{RCon}(\mathcal{T})$ and the individual names in $\mathsf{Ind}(\phi)$ in the same way. Similar to before, we construct a model $\mathcal{J}$ of $\zeta_{\mathcal{W}} \wedge \xi_\tau$ and $(\mathcal{T}_{\mathcal{W}} \cup \mathcal{T}_\tau, \mathcal{R}_{\mathcal{W}})$ over the shared domain of $\mathcal{I}_1, \dots, \mathcal{I}_k$ as follows: interpret the i-th copy of a flexible name as the original name in $\mathcal{I}_i$, and every rigid name as in $\mathcal{I}_1$. Since the interpretations of the names in $\mathsf{RCon}(\mathcal{T})$ are not changed, $\mathcal{J}$ also respects $\mathcal{D}$. By Lemma 5.25, we obtain that $\mathcal{W}$ is r-satisfiable w.r.t. ι and $\mathcal{K}$.

For the 'only if' direction, assume that $\mathcal{W}$ is r-satisfiable w.r.t. ι and $\mathcal{K}$. Lemma 5.25 yields that there exist a pair $\mathcal{D} = (\mathsf{RCon}(\mathcal{T}), \mathcal{Y})$ with $\mathcal{Y} \subseteq 2^{\mathsf{RCon}(\mathcal{T})}$ and a mapping $\tau \colon \mathsf{Ind}(\phi) \to \mathcal{Y}$ such that $\zeta_{\mathcal{W}} \wedge \xi_\tau$ has a model $\mathcal{J}$ w.r.t. $(\mathcal{T}_{\mathcal{W}} \cup \mathcal{T}_\tau, \mathcal{R}_{\mathcal{W}})$ that respects $\mathcal{D}$. As before, for every i, $1 \leq i \leq k$, we obtain a model $\mathcal{I}_i$ of $\zeta_{X_i} \wedge \xi_\tau$ and $(\mathcal{T} \cup \mathcal{T}_\tau, \mathcal{R})$ over the domain of $\mathcal{J}$ by interpreting the rigid names as in $\mathcal{J}$ and the flexible names as their i-th copies in $\mathcal{J}$. Again, these models still respect $\mathcal{D}$. □

To obtain our complexity result, we show how to decide whether the conjunction of CQ-literals $\zeta_{X_i} \wedge \xi_\tau$ has a model w.r.t. $(\mathcal{T} \cup \mathcal{T}_\tau, \mathcal{R})$ that respects $\mathcal{D}$ in time exponential in the size of ϕ and $\mathcal{K}$. Similar to the proof of Theorem 5.10, we can reduce this problem to a *non-entailment problem for a union of Boolean CQs*: there is a model of $\zeta_{X_i} \wedge \xi_\tau$ w.r.t. $(\mathcal{T} \cup \mathcal{T}_\tau, \mathcal{R})$ that respects $\mathcal{D}$ iff there is a model of $\mathcal{K}_i := (\mathcal{A}_i, \mathcal{T} \cup \mathcal{T}_\tau, \mathcal{R})$ that respects $\mathcal{D}$ and is not a model of ρ_i (written $\mathcal{K}_i \not\models \rho_i$ w.r.t. $\mathcal{D}$), where $\mathcal{A}_i$ is an ABox obtained by instantiating the non-negated CQs of $\zeta_{X_i} \wedge \xi_\tau$ with fresh individual names and ρ_i is a union of CQs constructed from the negated CQs of $\zeta_{X_i} \wedge \xi_\tau$. Since all $\mathcal{K}_i$ and ρ_i are of size polynomial in the size of ϕ and $\mathcal{K}$, it thus suffices to show that we can decide the query non-entailment $\mathcal{K}_i \not\models \rho_i$ w.r.t. $\mathcal{D}$ in time exponential in the size of $\mathcal{K}_i$ and ρ_i.

It is known that $\mathcal{K}_i \not\models \rho_i$ iff there is a *forest model* $\mathcal{I}$ of $\mathcal{K}_i$ such that $\mathcal{I} \not\models \rho_i$ [GHL+08; Lut08a]. We define here forest models for the more general case of *Boolean* $\mathcal{SHQ}^\sqcap$*-knowledge bases* since this will be needed later.

As introduced in Chapter 3, the description logic $\mathcal{SHQ}^\sqcap$ extends $\mathcal{SHQ}$ with *role conjunctions*. Recall that role conjunctions are of the form $r_1 \sqcap \dots \sqcap r_\ell$, $\ell \geq 1$, where $r_1, \dots, r_\ell$ are *simple* role names. Such role conjunctions are allowed to occur in existential restrictions instead of a single role, but not in at-least restrictions or role axioms. An interpretation $\mathcal{I}$ is extended to a role conjunction as follows: $(r_1 \sqcap \dots \sqcap r_\ell)^\mathcal{I} := r_1^\mathcal{I} \cap \dots \cap r_\ell^\mathcal{I}$.

In the following, we denote by $\mathsf{Ind}(\Psi)$ the set of individuals occurring in the Boolean knowledge base $\mathcal{B} = (\Psi, \mathcal{R})$. We are now ready to define forest models over Boolean knowledge bases (for a similar definition, see [GHL+08]).

Definition 5.29 (Forest model). *A* tree *is a non-empty prefix-closed subset of* $\mathbb{N}^*$*, where* $\mathbb{N}^*$ *denotes the set of all finite words over the non-negative integers.*

Let $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ *be an interpretation, and let* $\mathcal{B} = (\Psi, \mathcal{R})$ *be a Boolean knowledge base. We say that* $\mathcal{I}$ *is a* forest base *for* $\mathcal{B}$ *if*

- $\Delta^\mathcal{I} \subseteq \mathsf{Ind}(\Psi) \times \mathbb{N}^*$ *such that for every* $a \in \mathsf{Ind}(\Psi)$*, the set* $\{u \mid (a,u) \in \Delta^\mathcal{I}\}$ *is a tree;*
- *if* $((a,u),(b,v)) \in r^\mathcal{I}$*, then either* $u = v = \varepsilon$*, or* $a = b$ *and* $v = u \cdot c$ *for some* $c \in \mathbb{N}$*, where* $\cdot$ *denotes concatenation; and*
- *for every* $a \in \mathsf{Ind}(\Psi)$*, we have* $a^\mathcal{I} = (a, \varepsilon)$*.*

We call a model $\mathcal{J} = (\Delta^\mathcal{J}, \cdot^\mathcal{J})$ *of* $\mathcal{B}$ *a* forest model *of* $\mathcal{B}$ *if there is a forest base* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ *for* $\mathcal{B}$ *such that* $\Delta^\mathcal{I} = \Delta^\mathcal{J}$*, for each* $A \in \mathsf{N_C}$*, we have* $A^\mathcal{I} = A^\mathcal{J}$*, for each* $a \in \mathsf{N_I}$*, we have* $a^\mathcal{I} = a^\mathcal{J}$*, and for each* $r \in \mathsf{N_R}$*, we have*

$$r^\mathcal{J} = r^\mathcal{I} \cup \bigcup_{\mathcal{R} \models s \sqsubseteq r,\ \mathcal{R} \models \mathsf{trans}(s)} (s^\mathcal{I})^+,$$

where $\cdot^+$ *denotes the transitive closure.* ◇

As an example of a forest model, consider Figure 5.30, where a graphical representation of a forest model is given. It depicts the individual names a, b, and c, which represent the roots (a,ε), (b,ε), and (c,ε) of three trees. Moreover, s is a simple role name, and r is a

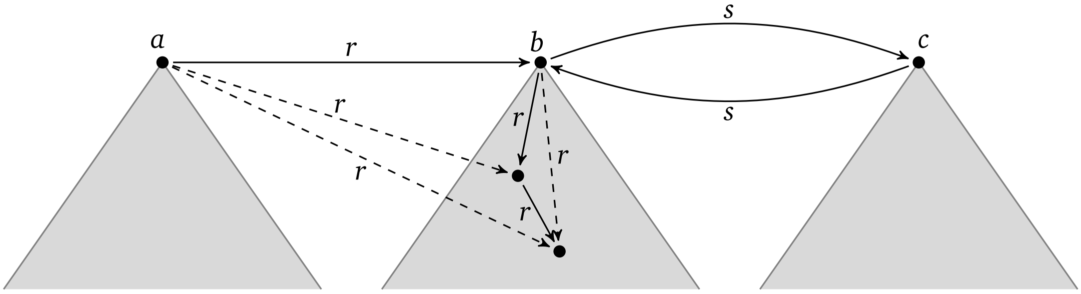

Figure 5.30: An example of a forest model

transitive role name. The solid arrows denote the role connections that are present in the corresponding forest base, and the dashed arrows denote role connection that are introduced due to transitivity.

In the following, we call a model $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ a forest model of a knowledge base $\mathcal{K} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$ if $\mathcal{J}$ is a forest model of the induced Boolean knowledge base $(\bigwedge \mathcal{A} \wedge \bigwedge \mathcal{T}, \mathcal{R})$.

We show now that the restriction to forest models when checking for the consistency of a Boolean $\mathcal{SHQ}^{\sqcap}$-knowledge base w.r.t. a pair $\mathcal{D} = (\mathcal{U}, \mathcal{Y})$ is without loss of generality. First, note that $\mathcal{B} = (\Psi, \mathcal{R})$ has a model that respects $\mathcal{D}$ iff $(\Psi \wedge A(a), \mathcal{R})$ has a model that respects $\mathcal{D}$, where a is a fresh individual name and A is a fresh concept name. In the following, we thus assume without loss of generality that Ψ contains at least one individual name. This is necessary to ensure that there is a non-empty forest base for $\mathcal{B}$. The construction in the proof of the following lemma is very similar to the one in [GHL+08], but we extend the previous result to *Boolean* knowledge bases, and take a pair $\mathcal{D}$ into account.

Lemma 5.31. *Let $\mathcal{B}$ be a Boolean $\mathcal{SHQ}^{\sqcap}$-knowledge base, and let $\mathcal{D} = (\mathcal{U}, \mathcal{Y})$ be a pair such that $\mathcal{U}$ is a set of concept names occurring in $\mathcal{B}$ and $\mathcal{Y} \subseteq 2^{\mathcal{U}}$. Then $\mathcal{B}$ is consistent w.r.t. $\mathcal{D}$ iff it has a forest model that respects $\mathcal{D}$.*

Proof. The 'if' direction is trivial. For the 'only if' direction, assume that $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is a model of $\mathcal{B} = (\Phi, \mathcal{R})$ that respects $\mathcal{D}$. Moreover, we assume that $\Delta^{\mathcal{I}}$ is countable, which is without loss of generality due to the downward Löwenheim-Skolem theorem [Löw15; Sko20]. We can thus assume that $\Delta^{\mathcal{I}} \subseteq \mathbb{N}$. We define now a forest base $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ for $\mathcal{B}$ with domain

$$\Delta^{\mathcal{J}} := \big\{(a, d_1 \dots d_m) \mid a \in \mathsf{Ind}(\Psi),\, m \geq 0,\, d_1, \dots, d_m \in \Delta^{\mathcal{I}}, \text{ and} \\ \text{there is no } b \in \mathsf{Ind}(\Psi) \text{ with } d_1 = b^{\mathcal{I}}\big\}$$

as follows:

- $a^{\mathcal{J}} := (a, \varepsilon)$ for every $a \in \mathsf{Ind}(\Psi)$;
- $b^{\mathcal{J}}$ for each $b \in \mathsf{N_I} \setminus \mathsf{Ind}(\Psi)$ can be fixed arbitrarily, as long as the UNA is satisfied;
- $A^{\mathcal{J}} := \{(a, \varepsilon) \mid a^{\mathcal{I}} \in A^{\mathcal{I}}\} \cup \{(a, d_1 \dots d_m) \mid d_m \in A^{\mathcal{I}}\}$; and
- $r^{\mathcal{J}} := \{((a, \varepsilon), (b, \varepsilon)) \mid (a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}\} \cup$
  $\{((a, \varepsilon), (a, d)) \mid (a^{\mathcal{I}}, d) \in r^{\mathcal{I}}\} \cup$
  $\{((a, d_1 \dots d_m), (a, d_1 \dots d_m d_{m+1})) \mid m > 0,\, (d_m, d_{m+1}) \in r^{\mathcal{I}}\}.$

Obviously, $\mathcal{J}$ satisfies the conditions of a forest base for $\mathcal{B}$. We construct now a forest model $\mathcal{J}' = (\Delta^{\mathcal{J}'}, \cdot^{\mathcal{J}'})$ for $\mathcal{B}$. For that, we define $\Delta^{\mathcal{J}'} := \Delta^{\mathcal{J}}$, for each $A \in \mathsf{N_C}$, $A^{\mathcal{J}'} := A^{\mathcal{J}}$ for each $a \in \mathsf{N_I}$, $a^{\mathcal{J}'} := a^{\mathcal{J}}$, and for each $r \in \mathsf{N_R}$:

$$r^{\mathcal{J}'} := r^{\mathcal{J}} \cup \bigcup_{\mathcal{R} \models s \sqsubseteq r,\ \mathcal{R} \models \mathsf{trans}(s)} (s^{\mathcal{J}})^+.$$

To prove that $\mathcal{J}'$ is indeed a forest model for $\mathcal{B}$, we first show the following claim by structural induction.

Claim 5.32. *For every $(a, d_1 \ldots d_m) \in \Delta^{\mathcal{J}'}$ and every concept C, we have $(a, d_1 \ldots d_m) \in C^{\mathcal{J}'}$ iff either $m = 0$ and $a^{\mathcal{I}} \in C^{\mathcal{I}}$, or $d_m \in C^{\mathcal{I}}$.*

For the base case, where C is a concept name, the claim is directly implied by the definition of $\mathcal{J}'$.

For the case where C is of the form $\neg D$, we have:

$(a, d_1 \ldots d_m) \in (\neg D)^{\mathcal{J}'}$

iff $(a, d_1 \ldots d_m) \notin D^{\mathcal{J}'}$

iff either $m = 0$ and $a^{\mathcal{I}} \notin D^{\mathcal{I}}$, or $d_m \notin D^{\mathcal{I}}$

iff either $m = 0$ and $a^{\mathcal{I}} \in (\neg D)^{\mathcal{I}}$, or $d_m \in (\neg D)^{\mathcal{I}}$.

For the case where C is of the form $D \sqcap E$, we have:

$(a, d_1 \ldots d_m) \in (D \sqcap E)^{\mathcal{J}'}$

iff $(a, d_1 \ldots d_m) \in D^{\mathcal{J}'}$ and $(a, d_1 \ldots d_m) \in E^{\mathcal{J}'}$

iff either $m = 0$ and $a^{\mathcal{I}} \in D^{\mathcal{I}}$ and $a^{\mathcal{I}} \in E^{\mathcal{I}}$, or $d_m \in D^{\mathcal{I}}$ and $d_m \in E^{\mathcal{I}}$

iff either $m = 0$ and $a^{\mathcal{I}} \in (D \sqcap E)^{\mathcal{I}}$, or $d_m \in (D \sqcap E)^{\mathcal{I}}$.

For the case where C is of the form $\exists(r_1 \sqcap \cdots \sqcap r_\ell).D$ with $\ell > 1$, we have that $r_1, \ldots, r_\ell$ are simple role names, and thus $r_1^{\mathcal{J}'} \cap \cdots \cap r_\ell^{\mathcal{J}'} = r_1^{\mathcal{J}} \cap \cdots \cap r_\ell^{\mathcal{J}}$. This yields:

$(a, d_1 \ldots d_m) \in (\exists(r_1 \sqcap \cdots \sqcap r_\ell).D)^{\mathcal{J}'}$

iff either $m = 0$ and

- there is a $(b, \varepsilon) \in D^{\mathcal{J}'}$ such that $((a, \varepsilon), (b, \varepsilon)) \in r_1^{\mathcal{J}} \cap \cdots \cap r_\ell^{\mathcal{J}}$, or
- there is a $(a, d) \in D^{\mathcal{J}'}$ such that $((a, \varepsilon), (a, d)) \in r_1^{\mathcal{J}} \cap \cdots \cap r_\ell^{\mathcal{J}}$;

or there is a $(a, d_1 \ldots d_m d_{m+1}) \in D^{\mathcal{J}'}$ such that $((a, d_1 \ldots d_m), (a, d_1 \ldots d_m d_{m+1}))$ is in $r_1^{\mathcal{J}} \cap \cdots \cap r_\ell^{\mathcal{J}}$

iff either $m = 0$ and there is a $d \in D^{\mathcal{I}}$ such that $(a^{\mathcal{I}}, d) \in r_1^{\mathcal{I}} \cap \cdots \cap r_\ell^{\mathcal{I}}$, or there is a $d \in D^{\mathcal{I}}$ such that $(d_m, d) \in r_1^{\mathcal{I}} \cap \cdots \cap r_\ell^{\mathcal{I}}$

iff either $m = 0$ and $a^{\mathcal{I}} \in (\exists(r_1 \sqcap \cdots \sqcap r_\ell).D)^{\mathcal{I}}$, or $d_m \in (\exists(r_1 \sqcap \cdots \sqcap r_\ell).D)^{\mathcal{I}}$.

For the case where C is of the form $\exists r.D$, we have

$(a, d_1 \ldots d_m) \in (\exists r.D)^{\mathcal{J}'}$

iff there is an $x \in D^{\mathcal{J}'}$ with either $((a, d_1 \dots d_m), x) \in r^{\mathcal{J}}$ or there is an $s \in \mathsf{N_R}$ with $\mathcal{R} \models s \sqsubseteq r$, $\mathcal{R} \models \mathsf{trans}(s)$, and $((a, d_1 \dots d_m), x) \in (s^{\mathcal{J}})^+$

iff either $m = 0$ and

- there is a $(b, \varepsilon) \in D^{\mathcal{J}'}$ with $((a, \varepsilon), (b, \varepsilon)) \in r^{\mathcal{J}}$,
- there is a $(a, d) \in D^{\mathcal{J}'}$ with $((a, \varepsilon), (a, d)) \in r^{\mathcal{J}}$, or
- there is an $s \in \mathsf{N_R}$ with $\mathcal{I} \models s \sqsubseteq r$ and $\mathcal{I} \models \mathsf{trans}(s)$, and a sequence (a_0, ε), $(a_1, \varepsilon), \dots, (a_n, \varepsilon), (a_n, e_1), \dots, (a_n, e_1 \dots e_k)$ of elements of $\Delta^{\mathcal{J}'}$ such that $a_0 = a$, $(a_n, e_1 \dots e_k) \in D^{\mathcal{J}'}$, and each two consecutive elements of this sequence are connected via $s^{\mathcal{J}}$;

or there is a sequence $(a, d_1 \dots d_m)$, $(a, d_1 \dots d_{m+1}), \dots, (a, d_1 \dots d_{m+n})$ of elements of $\Delta^{\mathcal{J}'}$ such that $n \geq 1$, $(a, d_1 \dots d_{m+n}) \in D^{\mathcal{J}'}$, and each two consecutive elements of this sequence are connected via $s^{\mathcal{J}}$, where s is a role name such that either $n = 1$ and $s = r$, or $\mathcal{I} \models s \sqsubseteq r$ and $\mathcal{I} \models \mathsf{trans}(s)$

iff either $m = 0$ and

- there is a $d \in D^{\mathcal{I}}$ such that $(a^{\mathcal{I}}, d) \in r^{\mathcal{I}}$, or
- there is an $s \in \mathsf{N_R}$ with $\mathcal{I} \models s \sqsubseteq r$ and $\mathcal{I} \models \mathsf{trans}(s)$, and an $e_k \in \Delta^{\mathcal{I}}$ such that $(a^{\mathcal{I}}, e_k) \in s^{\mathcal{I}} \subseteq r^{\mathcal{I}}$ and $e_k \in D^{\mathcal{I}}$;

or there is a $d \in D^{\mathcal{I}}$ such that $(d_m, d) \in s^{\mathcal{I}} \subseteq r^{\mathcal{I}}$, where s is a role name such that either $s = r$, or $\mathcal{I} \models s \sqsubseteq r$ and $\mathcal{I} \models \mathsf{trans}(s)$

iff either $m = 0$ and $a^{\mathcal{I}} \in (\exists r.D)^{\mathcal{I}}$, or $d_m \in (\exists r.D)^{\mathcal{I}}$.

For the case where C is of the form $\geq n\, r.D$ for a simple role name r, we again have $r^{\mathcal{J}'} = r^{\mathcal{J}}$, and thus

$(a, d_1 \dots d_m) \in (\geq n\, r.D)^{\mathcal{J}'}$

iff there is a subset $X \subseteq D^{\mathcal{J}'}$ with $|X| = n$ such that $((a, \varepsilon), x) \in r^{\mathcal{J}}$ for every $x \in X$, and either

- $m = 0$ and every $x \in X$ is either of the form (b, ε) or (a, d), or
- every $x \in X$ is of the form $(a, d_1 \dots d_m d_{m+1})$

iff there is a subset $Y \subseteq D^{\mathcal{I}}$ with $|Y| = n$ such that $m = 0$ and $(a^{\mathcal{I}}, y) \in r^{\mathcal{I}}$ for every $y \in Y$, or $(d_m, y) \in r^{\mathcal{I}}$ for every $y \in Y$

iff either $m = 0$ and $a^{\mathcal{I}} \in (\geq n\, r.D)^{\mathcal{I}}$, or $d_m \in (\geq n\, r.D)^{\mathcal{I}}$.

The second equivalence holds since each $r^{\mathcal{I}}$-successor of a named domain element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ is represented by *exactly one* $r^{\mathcal{J}'}$-successor of $(a, \varepsilon) \in \Delta^{\mathcal{J}'}$, which holds due the fact that $\Delta^{\mathcal{J}'}$ does not contain domain elements of the form $(a, b^{\mathcal{I}})$ for $b \in \mathsf{Ind}(\Psi)$. This finishes the proof of Claim 5.32.

It remains only to be shown that $\mathcal{J}'$ is indeed a model of $\mathcal{B}$ that respects $\mathcal{D}$. For this, we prove first the following claim by structural induction.

Claim 5.33. *For every $\psi \in \mathsf{Cl_f}(\Psi)$, we have $\mathcal{J}' \models \psi$ iff $\mathcal{I} \models \psi$.*

For the first base case, assume that ψ is of the form $A(a)$ for some $A \in \mathsf{N_C}$ and $a \in \mathsf{N_I}$. We have $a^{\mathcal{I}} \in A^{\mathcal{I}}$ iff $a^{\mathcal{J}'} = a^{\mathcal{J}} = (a, \varepsilon) \in A^{\mathcal{J}} = A^{\mathcal{J}'}$ by definition.

For the second base case, assume that ψ is of the form $r(a,b)$ for some $r \in \mathsf{N_R}$ and $a, b \in \mathsf{N_I}$. If $\mathcal{I} \models r(a,b)$, then $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$, and thus

$$(a^{\mathcal{J}'}, b^{\mathcal{J}'}) = (a^{\mathcal{J}}, b^{\mathcal{J}}) = ((a,\varepsilon),(b,\varepsilon)) \in r^{\mathcal{J}} \subseteq r^{\mathcal{J}'}.$$

Conversely, if $((a,\varepsilon),(b,\varepsilon)) \in r^{\mathcal{J}'}$, then there is an $s \in \mathsf{N_R}$ and a sequence $(a_0,\varepsilon), \dots, (a_n,\varepsilon)$ of elements of $\Delta^{\mathcal{J}'}$ with $n \geq 1$ such that $a_0 = a$, $a_n = b$, each two consecutive elements of this sequence are connected via $s^{\mathcal{J}}$, and either $n = 1$ and $s = r$, or $\mathcal{R} \models s \sqsubseteq r$ and $\mathcal{R} \models \mathsf{trans}(s)$. By the definition of $s^{\mathcal{J}}$, the properties of s, and since $\mathcal{I} \models \mathcal{R}$, we can infer that $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$, and thus $\mathcal{I} \models r(a,b)$.

For the third base case, assume that ψ is of the form $C \sqsubseteq D$. For the 'if' direction, assume that $\mathcal{I} \models C \sqsubseteq D$ and thus $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. Suppose that there is a $(a, d_1 \dots d_m) \in C^{\mathcal{J}'}$ with $(a, d_1 \dots d_m) \notin D^{\mathcal{J}'}$. By Claim 5.32, either $m = 0$ and we have $a^{\mathcal{I}} \in C^{\mathcal{I}}$ and $a^{\mathcal{I}} \notin D^{\mathcal{I}}$, or $d_m \in C^{\mathcal{I}}$ and $d_m \notin D^{\mathcal{I}}$, which contradicts our assumption that $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.

For the 'only if' direction, assume that $C^{\mathcal{J}'} \subseteq D^{\mathcal{J}'}$. Suppose that there is a $d \in C^{\mathcal{I}}$ with $d \notin D^{\mathcal{I}}$. By the definition of $\Delta^{\mathcal{J}'}$, we have $(a, d'd) \in \Delta^{\mathcal{J}'}$ for any $a \in \mathsf{Ind}(\Psi)$ and $d' \in \Delta^{\mathcal{I}}$ such that there is no $b \in \mathsf{Ind}(\Psi)$ with $d' = b^{\mathcal{I}}$. By Claim 5.32, we obtain $(a, d'd) \in C^{\mathcal{J}'}$ and $(a, d'd) \notin D^{\mathcal{J}'}$, which again yields a contradiction.

For the induction step, assume first that ψ is of the form $\neg\psi'$. We have that $\mathcal{J}' \models \neg\psi'$ iff $\mathcal{J}' \not\models \psi'$ iff $\mathcal{I} \not\models \psi'$ iff $\mathcal{I} \models \neg\psi'$. Assume now that ψ is of the form $\psi_1 \wedge \psi_2$. We have that $\mathcal{J}' \models \psi_1 \wedge \psi_2$ iff $\mathcal{J}' \models \psi_1$ and $\mathcal{J}' \models \psi_2$ iff $\mathcal{I} \models \psi_1$ and $\mathcal{I} \models \psi_2$ iff $\mathcal{I} \models \psi_1 \wedge \psi_2$. This finishes the proof of Claim 5.33.

Since $\Psi \in \mathsf{Cl_f}(\Psi)$, this shows that $\mathcal{J}'$ is indeed a model of Ψ. We show that $\mathcal{J}'$ is also a model of $\mathcal{R}$ in the following claim.

Claim 5.34. *For every $\alpha \in \mathcal{R}$, we have $\mathcal{J}' \models \alpha$.*

Assume first that α is of the form $r \sqsubseteq s$. Since $\mathcal{I} \models \mathcal{R}$, we have $\mathcal{I} \models r \sqsubseteq s$ and thus $r^{\mathcal{I}} \subseteq s^{\mathcal{I}}$. We first show that $r^{\mathcal{J}} \subseteq s^{\mathcal{J}}$. For this, take $(x,y) \in r^{\mathcal{J}}$. There are three cases to consider:

- If $x = (a,\varepsilon)$ and $y = (b,\varepsilon)$ with $a, b \in \mathsf{Ind}(\Psi)$, we have $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$ and thus $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in s^{\mathcal{I}}$. Hence, the definition of $s^{\mathcal{J}}$ yields that $(x,y) \in s^{\mathcal{J}}$.
- If $x = (a,\varepsilon)$ and $y = (a,d)$ with $a \in \mathsf{Ind}(\Psi)$ and $d \in \Delta^{\mathcal{I}}$, we have $(a^{\mathcal{I}}, d) \in r^{\mathcal{I}}$ and thus $(a^{\mathcal{I}}, d) \in s^{\mathcal{I}}$. Again, the definition of $s^{\mathcal{J}}$ yields that $(x,y) \in s^{\mathcal{J}}$.
- If we have $x = (a, d_1 \dots d_m)$ and $y = (a, d_1 \dots d_m d_{m+1})$ with $a \in \mathsf{Ind}(\Psi)$, $m > 0$, and $d_1, \dots, d_{m+1} \in \Delta^{\mathcal{I}}$, we have also $(d_m, d_{m+1}) \in r^{\mathcal{I}}$ and thus $(d_m, d_{m+1}) \in s^{\mathcal{I}}$. Again, the definition of $s^{\mathcal{J}}$ yields that $(x,y) \in s^{\mathcal{J}}$.

To show that $r^{\mathcal{J}'} \subseteq s^{\mathcal{J}'}$, take $(x,y) \in r^{\mathcal{J}'}$. If $(x,y) \in r^{\mathcal{J}}$, we have $(x,y) \in s^{\mathcal{J}}$ and thus $(x,y) \in s^{\mathcal{J}'}$. Otherwise, we have that $(x,y) \in (t^{\mathcal{J}})^+$ with $\mathcal{R} \models t \sqsubseteq r$ and $\mathcal{R} \models \mathsf{trans}(t)$. Since $r \sqsubseteq s \in \mathcal{R}$, we have obviously $\mathcal{R} \models r \sqsubseteq s$. It is easy to see that this implies $\mathcal{R} \models t \sqsubseteq s$. Then the definition of $s^{\mathcal{J}'}$ yields that $(t^{\mathcal{J}})^+ \subseteq s^{\mathcal{J}'}$. Hence $(x,y) \in s^{\mathcal{J}'}$.

Assume now that ψ is of the form $\mathsf{trans}(r)$. Since $\mathcal{I} \models \mathcal{R}$, we have $\mathcal{I} \models \mathsf{trans}(r)$ and thus $r^{\mathcal{I}} \circ r^{\mathcal{I}} \subseteq r^{\mathcal{I}}$. By the same arguments as above, we have for every $t \in \mathsf{N_R}$ with $t^{\mathcal{I}} \subseteq r^{\mathcal{I}}$, that $t^{\mathcal{J}} \subseteq r^{\mathcal{J}}$, and thus $(t^{\mathcal{J}})^+ \subseteq (r^{\mathcal{J}})^+$ since the transitive closure is monotonic. Since $r^{\mathcal{I}} \subseteq r^{\mathcal{I}}$, we have also $\mathcal{I} \models r \sqsubseteq r$. The definition of $r^{\mathcal{J}'}$ yields now that $r^{\mathcal{J}'} = (r^{\mathcal{J}})^+$, and hence $\mathcal{J}'$ is a model of $\mathsf{trans}(r)$. This finishes the proof of Claim 5.34.

Claim 5.33, the fact that $\Psi \in \mathsf{Cl}_\mathsf{f}(\Psi)$, and Claim 5.34 yield that $\mathcal{J}'$ is indeed a model of $\mathcal{B} = (\Psi, \mathcal{R})$. Finally, we show that $\mathcal{J}'$ respects $\mathcal{D} = (\mathcal{U}, \mathcal{Y})$. Since $\mathcal{I}$ respects $\mathcal{D}$, we have

$$\mathcal{Y} = \{Y \subseteq \mathcal{U} \mid \text{there is a } d \in \Delta^\mathcal{I} \text{ with } d \in (C_{\mathcal{U},Y})^\mathcal{I}\},$$

where $C_{\mathcal{U},Y}$ is defined as in Definition 3.18. We now set $\mathcal{D}' := (\mathcal{U}, \mathcal{Y}')$ where

$$\mathcal{Y}' := \{Y \subseteq \mathcal{U} \mid \text{there is an } x \in \Delta^{\mathcal{J}'} \text{ with } x \in (C_{\mathcal{U},Y})^{\mathcal{J}'}\},$$

and show that $\mathcal{D} = \mathcal{D}'$. Since $\mathcal{J}'$ respects $\mathcal{D}'$, this implies that $\mathcal{J}'$ respects $\mathcal{D}$.

For the direction ($\subseteq$), assume that $Y \in \mathcal{Y}$, and thus there is a $d \in (C_{\mathcal{U},Y})^\mathcal{I}$. By Claim 5.32 and the definition of $\Delta^{\mathcal{J}'}$, there is a $(a, d'd) \in (C_{\mathcal{U},Y})^{\mathcal{J}'}$, and hence $Y \in \mathcal{Y}'$. Conversely, for the direction ($\supseteq$), assume that $Y \in \mathcal{Y}'$, i.e. there is a $(a, d_1 \dots d_m) \in (C_{\mathcal{U},Y})^{\mathcal{J}'}$. By Claim 5.32 and the definition of $\Delta^{\mathcal{J}'}$, there is a $d \in (C_{\mathcal{U},Y})^\mathcal{I}$, where for $m = 0$, we can set $d := a^\mathcal{I}$, and for $m > 0$, we can take $d := d_m$. Hence, we have that $Y \in \mathcal{Y}$. □

Moreover, we can extend the aforementioned result about the non-entailment problem for UCQs from [GHL+08; Lut08a] to our setting. In the following, we assume that the UCQ ρ contains only individual names that also occur in the ABox (or Boolean axiom formula). This is without loss of generality, because for any individual name a not occurring in the ABox (or Boolean axiom formula), we can simply add the assertion $A(a)$ to the ABox, where A is a fresh concept name.

Lemma 5.35. *Let $\mathcal{K} := (\mathcal{A}, \mathcal{T}, \mathcal{R})$ be a knowledge base, let ρ be a union of Boolean CQs, and let $\mathcal{D} = (\mathcal{U}, \mathcal{Y})$ be a pair such that U is a set of concept names and $\mathcal{Y} \subseteq 2^\mathcal{U}$. Then, we have $\mathcal{K} \not\models \rho$ w.r.t. $\mathcal{D}$ iff there is a forest model $\mathcal{J}$ of $\mathcal{K}$ that respects $\mathcal{D}$ with $\mathcal{J} \not\models \rho$.*

Proof. The 'if' direction is trivial. For the 'only if' direction, assume that there is a model $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ of $\mathcal{K}$ that respects $\mathcal{D}$ such that $\mathcal{I} \not\models \rho$. As shown in the proof of Lemma 5.31, the model $\mathcal{I}$ can be transformed into a forest model $\mathcal{J}' = (\Delta^{\mathcal{J}'}, \cdot^{\mathcal{J}'})$ of $\mathcal{K}$ that respects $\mathcal{D}$. Assume that $\mathcal{J}$ and $\mathcal{J}'$ are the forest base and the forest model, respectively, obtained from $\mathcal{I}$ as in the proof of Lemma 5.31.

It is left to be shown that $\mathcal{J}' \not\models \rho$. Assume to the contrary that $\mathcal{J}' \models \rho$. Then, there is a Boolean CQ ρ_i in the UCQ ρ such that there is a homomorphism π from ρ_i into $\mathcal{J}'$. We define a homomorphism π' from ρ_i into $\mathcal{I}$ as follows:

- $\pi'(a) := a^\mathcal{I}$ for every individual name $a \in \mathsf{Ind}(\mathcal{A})$; and
- for every $v \in \mathsf{Var}(\rho_i)$, we define

$$\pi'(v) := \begin{cases} a^\mathcal{I} & \text{if } \pi(v) = (a, \varepsilon) \text{ with } a \in \mathsf{Ind}(\mathcal{A}), \text{ and} \\ d_m & \text{if } \pi(v) = (a, d_1 \dots d_m) \text{ with } m > 0. \end{cases}$$

We now show that π' is indeed a homomorphism from ρ_i into $\mathcal{I}$.

Consider first a concept atom $A(a) \in \mathsf{At}(\rho_i)$ with $a \in \mathsf{Ind}(\mathcal{A})$. Since π is a homomorphism from ρ_i into $\mathcal{J}'$, we have $\pi(a) = a^{\mathcal{J}'} = (a, \varepsilon) \in A^{\mathcal{J}'}$. By Claim 5.32, we obtain $a^\mathcal{I} \in A^\mathcal{I}$, and thus $\pi'(a) \in A^\mathcal{I}$.

Similarly, for a concept atom $A(v) \in \mathsf{At}(\rho_i)$ with $v \in \mathsf{Var}(\rho_i)$, we have $\pi(v) \in A^{\mathcal{J}'}$, and thus $\pi'(v) \in A^{\mathcal{I}}$ again by Claim 5.32.

For a role atom $r(a,b) \in \mathsf{At}(\rho_i)$, we can show $(\pi'(a), \pi'(b)) = (a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$ using the same arguments as in the proof of Claim 5.33: Since π is a homomorphism from ρ_i into $\mathcal{J}'$, we have $(\pi(a), \pi(b)) = (a^{\mathcal{J}'}, b^{\mathcal{J}'}) = ((a,\varepsilon),(b,\varepsilon)) \in r^{\mathcal{J}'}$, and thus that there is an $s \in \mathsf{N_R}$ and a sequence $(a_0, \varepsilon), \dots, (a_n, \varepsilon)$ of elements of $\Delta^{\mathcal{J}'}$ with $n \geq 1$ such that $a_0 = a$, $a_n = b$, each two consecutive elements of this sequence are connected via $s^{\mathcal{J}}$, and either $n = 1$ and $s = r$, or $\mathcal{R} \models s \sqsubseteq r$ and $\mathcal{R} \models \mathsf{trans}(s)$. By the definition of $s^{\mathcal{J}}$, the properties of s, and since $\mathcal{I} \models \mathcal{R}$, we can infer that $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$.

If there is a role atom of the form $r(a,v) \in \mathsf{At}(\rho_i)$ with $a \in \mathsf{Ind}(\mathcal{A})$ and $v \in \mathsf{Var}(\rho_i)$, we have $(\pi(a), \pi(v)) = (a^{\mathcal{J}'}, \pi(v)) = ((a,\varepsilon), \pi(v)) \in r^{\mathcal{J}'}$. If $\pi(v) = (b, \varepsilon)$ with $b \in \mathsf{Ind}(\mathcal{A})$, we can argue as in the previous case that $(a^{\mathcal{I}}, b^{\mathcal{I}}) = (\pi'(a), \pi'(b)) \in r^{\mathcal{I}}$. Otherwise there are again two cases to consider. First, if we have $((a,\varepsilon), \pi(v)) \in r^{\mathcal{J}}$, then we have also $(a^{\mathcal{I}}, \pi'(v)) = (\pi'(a), \pi'(v)) \in r^{\mathcal{I}}$ by the definitions of $\mathcal{J}$ and π'. Otherwise, there must be a role name $s \in \mathsf{N_R}$ such that $\mathcal{R} \models s \sqsubseteq r$, $\mathcal{R} \models \mathsf{trans}(s)$, and $((a,\varepsilon), \pi(v)) \in (s^{\mathcal{J}})^+$. This yields that there is a sequence (a_0, ε), (a_1, ε), $\dots$, (a_n, ε), (a_n, e_1), $\dots$, $(a_n, e_1 \dots e_k)$ of elements of $\Delta^{\mathcal{J}'}$ with $n \geq 1$ and $k \geq 1$ such that $a_0 = a$, $\pi(v) = (a_n, e_1 \dots e_k)$, and each two consecutive elements of this sequence are connected via $s^{\mathcal{J}}$. By the definitions of $s^{\mathcal{J}}$ and π', we obtain $(a^{\mathcal{I}}, \pi'(v)) \in s^{\mathcal{I}}$, and thus since $s^{\mathcal{I}} \subseteq r^{\mathcal{I}}$, also $(\pi'(a), \pi'(v)) \in r^{\mathcal{I}}$.

For any role atom $r(v,a) \in \mathsf{At}(\rho_i)$ with $v \in \mathsf{Var}(\rho_i)$ and $a \in \mathsf{Ind}(\mathcal{A})$, we have that $(\pi(v), \pi(a)) = (\pi(v), a^{\mathcal{J}'}) = (\pi(v), (a,\varepsilon)) \in r^{\mathcal{J}'}$. By the definition of $r^{\mathcal{J}'}$, this implies that there is a sequence $(a_0, \varepsilon), \dots, (a_n, \varepsilon)$ of elements of $\Delta^{\mathcal{J}'}$ with $n \geq 1$ such that $a_n = a$, $\pi(v) = (a_0, \varepsilon)$, and each two consecutive elements of this sequence are connected via $s^{\mathcal{J}}$, where s is a role name such that either $n = 1$ and $s = r$, or $\mathcal{R} \models s \sqsubseteq r$ and $\mathcal{R} \models \mathsf{trans}(s)$. By the definitions of $s^{\mathcal{J}}$ and π', the properties of s, and since $\mathcal{I} \models \mathcal{R}$, this yields that $(\pi'(v), \pi'(a)) = (\pi'(v), a^{\mathcal{I}}) = (a_0^{\mathcal{I}}, a_n^{\mathcal{I}}) \in r^{\mathcal{I}}$.

Finally, consider a role atom $r(v,v') \in \mathsf{At}(\rho_i)$ with $v, v' \in \mathsf{Var}(\rho_i)$. Again, since π is a homomorphism from ρ_i into $\mathcal{J}'$, we have $(\pi(v), \pi(v')) \in r^{\mathcal{J}'}$. If $\pi(v) = (a,\varepsilon)$ for some $a \in \mathsf{Ind}(\mathcal{A})$, we can show as in second-last case that $(\pi'(v), \pi'(v')) = (a^{\mathcal{I}}, \pi'(v')) \in r^{\mathcal{I}}$. Otherwise, we have $\pi(v) = (a, d_1 \dots d_m)$ with $m > 0$ and that there is a sequence $(a, d_1 \dots d_m)$, $(a, d_1 \dots d_{m+1})$, $\dots$, $(a, d_1 \dots d_{m+n})$ of elements of the domain $\Delta^{\mathcal{J}'}$ such that we have that $n \geq 1$, $\pi(v') = (a, d_1 \dots d_{m+n})$, and each two consecutive elements of this sequence are connected via $s^{\mathcal{J}}$, where s is a role name such that either $n = 1$ and $s = r$, or $\mathcal{R} \models s \sqsubseteq r$ and $\mathcal{R} \models \mathsf{trans}(s)$. Thus, we obtain $(\pi'(v), \pi'(v')) = (d_m, d_{m+n}) \in s^{\mathcal{I}} \subseteq r^{\mathcal{I}}$ by similar arguments as before.

Hence, π' is a homomorphism from ρ_i into $\mathcal{I}$, and thus $\mathcal{I} \models \rho_i$. But this yields that $\mathcal{I} \models \rho$, which contradicts our assumption that $\mathcal{I} \not\models \rho$. □

Recall that we want to decide the existence of such a forest model in time exponential in the size of $\mathcal{K} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$, and ρ. To achieve this, we further reduce this decision problem following an idea from [Lut08a]. There, the notion of a spoiler is introduced. According to [Lut08a], a *spoiler for $\mathcal{K}$ and ρ* is an $\mathcal{SHQ}^{\sqcap}$-knowledge base $\mathcal{K}' = (\mathcal{A}', \mathcal{T}', \emptyset)$ that states properties that must be satisfied such that ρ is not entailed by $\mathcal{K}$. Note that the ABox $\mathcal{A}'$ of such a spoiler may also contain *negated* assertions. Furthermore, a spoiler may contain role conjunctions. Thus, according to our notation, $\mathcal{K}'$ is a Boolean $\mathcal{SHQ}^{\sqcap}$-knowledge base, i.e. $\mathcal{K}' = (\bigwedge \mathcal{A}' \wedge \bigwedge \mathcal{T}', \emptyset)$. The following proposition is shown in [Lut08a].

Proposition 5.36. *Let $\mathcal{K} := (\mathcal{A}, \mathcal{T}, \mathcal{R})$ be a $\mathcal{SHQ}$-knowledge base, and let ρ be a union of Boolean CQs. Then, we have $\mathcal{K} \not\models \rho$ iff there is a spoiler $\mathcal{K}' = (\bigwedge \mathcal{A}' \wedge \bigwedge \mathcal{T}', \emptyset)$ for $\mathcal{K}$ and ρ such that $(\bigwedge \mathcal{A} \wedge \bigwedge \mathcal{A}' \wedge \bigwedge \mathcal{T} \wedge \bigwedge \mathcal{T}', \mathcal{R})$ is consistent.* ◇

In addition, it is shown in [Lut08a] that all spoilers for $\mathcal{K}$ and ρ can be computed in time exponential in the size of $\mathcal{K}$ and ρ, and that each spoiler is of polynomial size. In the proof of these results, one only has to deal with forest models, which furthermore do not need to be modified. More formally, for any forest model $\mathcal{I}$ of $(\mathcal{A}, \mathcal{T}, \mathcal{R})$ that does not satisfy ρ there is a spoiler $(\bigwedge \mathcal{A}' \wedge \bigwedge \mathcal{T}', \emptyset)$ that also has $\mathcal{I}$ as a model and, conversely, every forest model of the knowledge base $(\mathcal{A}, \mathcal{T}, \mathcal{R})$ that also satisfies a spoiler $(\bigwedge \mathcal{A}' \wedge \bigwedge \mathcal{T}', \emptyset)$ does not satisfy ρ (see the proof of Lemma 3 in [Lut08b]). This yields the following more general result that also takes into account the pair $\mathcal{D}$.

Proposition 5.37. *Let $\mathcal{K} := (\mathcal{A}, \mathcal{T}, \mathcal{R})$ be a $\mathcal{SHQ}$-knowledge base, let ρ be a union of Boolean CQs, and let $\mathcal{D} = (\mathcal{U}, \mathcal{Y})$ be a pair such that $\mathcal{U}$ is a set of concept names and $\mathcal{Y} \subseteq 2^{\mathcal{U}}$. Then, we have $\mathcal{K} \not\models \rho$ w.r.t. $\mathcal{D}$ iff there is a spoiler $\mathcal{K}' = (\bigwedge \mathcal{A}' \wedge \bigwedge \mathcal{T}', \emptyset)$ for $\mathcal{K}$ and ρ such that there is a model of $(\bigwedge \mathcal{A} \wedge \bigwedge \mathcal{A}' \wedge \bigwedge \mathcal{T} \wedge \bigwedge \mathcal{T}', \mathcal{R})$ that respects $\mathcal{D}$.* ◇

It remains to show that the existence of such a model can be checked in time exponential in the size of $(\bigwedge \mathcal{A} \wedge \bigwedge \mathcal{A}' \wedge \bigwedge \mathcal{T} \wedge \bigwedge \mathcal{T}', \mathcal{R})$. But this follows directly from Theorem 3.33, where it is shown that consistency of Boolean $\mathcal{SHOQ}^{\sqcap}$-knowledge base $\mathcal{B}$ w.r.t. a pair $\mathcal{D}$ can be decided in time exponential in the size of $\mathcal{B}$. Thus, we obtain the following theorem.

Theorem 5.38. *Let $\mathcal{K} := (\mathcal{A}, \mathcal{T}, \mathcal{R})$ be a $\mathcal{SHQ}$-knowledge base, let ρ be a union of Boolean CQs, and let $\mathcal{D} = (\mathcal{U}, \mathcal{Y})$ be a pair such that $\mathcal{U}$ is a set of concept names and $\mathcal{Y} \subseteq 2^{\mathcal{U}}$. Then, we can decide whether $\mathcal{K} \not\models \rho$ w.r.t. $\mathcal{D}$ in time exponential in the size of $\mathcal{K}$ and ρ.*

Combining this with the reductions in Lemmas 5.27 and 5.28, we obtain the desired complexity result.

Theorem 5.39. *If $\mathsf{N_{RC}} \neq \emptyset$ and $\mathsf{N_{RR}} = \emptyset$, the temporalised query-entailment problem in $\mathcal{SHQ}$ is in* co-NExpTime *w.r.t. combined complexity.*

Proof. Let ϕ be a Boolean TCQ, and let $\mathcal{K} = ((\mathcal{A}_i)_{0 \leq i \leq n}, \mathcal{T}, \mathcal{R})$ be a temporal $\mathcal{SHQ}$-knowledge base. By Lemma 5.27, we can construct a Boolean TCQ ψ polynomial in the size of ϕ and $\mathcal{K}$ such that $\mathcal{K} \models \phi$ iff $\mathcal{K}' := (\emptyset, \mathcal{T}, \mathcal{R}) \models \psi$. We again consider the TCQ-satisfiability problem, which has the same complexity as the temporalised query non-entailment problem. We decide whether ψ is satisfiable w.r.t. $\mathcal{K}'$ using Lemma 5.18. For that, let $\mathsf{p}\colon \mathsf{CQ}(\psi) \to \mathcal{P}_{\psi}$ be a bijection. We first non-deterministically guess a set $\mathcal{W} = \{X_1, \dots, X_k\} \subseteq 2^{\mathcal{P}_{\psi}}$ and a mapping $\iota\colon \{0\} \to \{1, \dots, k\}$ in time exponential in the size of ψ and $\mathcal{K}'$. By Lemma 5.20, deciding whether ψ^{p} is t-satisfiable w.r.t. $\mathcal{W}$ and ι can be done in time exponential in the size of ψ^{p} (and thus also in time exponential in the size of ψ), linear in the size of $\mathcal{W}$, and polynomial in n.

Thus, due to Lemma 5.18, is suffices to show that r-satisfiability of $\mathcal{W}$ w.r.t. ι and $\mathcal{K}'$ can be checked in non-deterministic exponential time w.r.t. combined complexity. For that, we use Lemma 5.28. We non-deterministically guess a set $\mathcal{Y} \subseteq 2^{\mathsf{RCon}(\mathcal{T})}$ and a mapping $\tau\colon \mathsf{Ind}(\psi) \to \mathcal{Y}$, which can be done in time exponential in the size of ψ and $\mathcal{K}'$ since $\mathcal{Y}$ is of size exponential in $\mathcal{T}$ and τ is of size polynomial in the size of ψ and $\mathcal{T}$. We define

$\mathcal{D} := (\mathsf{RCon}(\mathcal{T}), \mathcal{Y})$. Next, we construct for every i, $1 \leq i \leq k$, the conjunction of CQ-literals $\zeta_{X_i} \wedge \xi_\tau$, the knowledge base $\mathcal{K}_i$ and the Boolean union of CQs ρ_i. Note that the size of each $\zeta_{X_i} \wedge \xi_\tau$, $\mathcal{K}_i$, and ρ_i is polynomial in the size of ψ and $\mathcal{K}'$ and the number k of these conjunctions is exponential in the size of ψ. Thus, it remains to show that we can decide every query non-entailment $\mathcal{K}_i \not\models \rho_i$ w.r.t. $\mathcal{D}$ in time exponential in the size of $\mathcal{K}_i$ and ρ_i, and thus in time exponential in the size of ψ and $\mathcal{K}'$, which we obtain by Theorem 5.38.

Hence, we can check whether ϕ is satisfiable w.r.t. $\mathcal{K}$ using the above decision procedure, which shows that the TCQ-satisfiability problem in $\mathcal{SHQ}$ is in NExpTime w.r.t. combined complexity. Thus, we obtain that the temporalised query-entailment problem is in co-NExpTime w.r.t. combined complexity. □

Together with Theorem 5.11, we obtain that the temporalised query-entailment problem in $\mathcal{SHQ}$ is co-NExpTime-complete w.r.t. combined complexity if only concept names are allowed to be rigid.

5.3 Summary

In this chapter, we have shown all the complexity results that are summarised in Table 5.2 for the proposed temporal query language. More precisely, we considered both the combined complexity and the data complexity of temporalised query entailment for all description logics between $\mathcal{ALC}$ and $\mathcal{SHQ}$ in the settings where (i) neither concept names nor role names are allowed to be rigid, (ii) only concept names may be rigid, and (iii) both concept names and role names may be rigid. It turned out that in Setting (i), the temporalised query-entailment problem is as hard as entailment of conjunctive queries w.r.t. atemporal $\mathcal{ALC}$- and $\mathcal{SHQ}$-knowledge bases, namely co-NP-complete w.r.t. data complexity and ExpTime-complete w.r.t. combined complexity. However, if we allow rigid concept names (but no rigid role names), the picture changes. Whilst the data complexity remains the same as in the atemporal case, the combined complexity of the temporalised query-entailment problem increases to co-NExpTime, i.e. the temporalised query non-entailment problem is as hard as the satisfiability problem in the temporalised description logic $\mathcal{ALC}$-LTL. If we further allow rigid role names, the combined complexity of the temporalised query (non-)entailment problem again increases in accordance with the complexity of the satisfiability problem in $\mathcal{ALC}$-LTL. In fact, all three problems are 2ExpTime-complete. For the data complexity, it is still open whether adding rigid role names results in an increase of the complexity. We have shown an upper bound of ExpTime—which is one exponential better than the combined complexity—, but the only lower bound we have is the trivial one of co-NP.

Further work will include trying to close this gap. Moreover, it would be interesting to find out what effect the addition of inverse roles has on the complexity of query entailment in the temporal case. Given the results for $\mathcal{ALCI}$ and $\mathcal{SHIQ}$ in the atemporal case, where the query entailment problem is 2ExpTime-complete w.r.t. combined complexity [Lut08a] and co-NP-complete w.r.t. data complexity [OCE06], there is the possibility that also in the temporal case, the query entailment problem remains co-NP-complete w.r.t. data complexity and 2ExpTime-complete w.r.t. combined complexity for all three settings considered in this chapter. But showing this will require considerable extensions of the proof techniques employed until now since the presence of inverse roles creates additional problems. We have also left open the complexity of the temporalised query entailment problem for the case where non-simple roles are allowed to occur in the queries. This problem is, however, already 2EXPTIME-hard w.r.t. combined complexity for the description logic $\mathcal{SH}$ [ELO+09] in the atemporal case.

Chapter 6

Verification in Action Formalisms Based on $\mathcal{ALCQIO}$

Action programming languages are successfully applied to modelling the behaviour of autarkic systems, which are often called *agents*. In this area, it is of keen interest to reason about the behaviour of non-terminating action programs as one expects that the agents perform open-ended tasks, which are not supposed to terminate. Since most action programming languages are based on action formalisms that encompass full first-order logic, the problem of verifying properties for such action programs is in general undecidable.

In this chapter, we restrict the setting in two directions to regain decidability. Firstly, we consider action formalisms based on DLs, for which important inference problems become decidable, and secondly, we verify properties of action sequences generated by Büchi-automata instead of considering full-fledged high-level action programming languages.

A first step was done in [BLM10], where the authors show that the problem of verifying properties formulated in a restricted version of the temporalised description logic $\mathcal{ALCO}$-LTL is decidable for the DL-based action formalism introduced in [BLM+05a]. However, the authors consider only acyclic TBoxes instead of general ones. In this chapter, we overcome this problem by enriching the DL-based action formalism with so-called *causal relationships*. Most of the results of this chapter have already been published in [BLL10b; BLL10a; YLL+12].

This chapter is organised as follows. In Section 6.1, we formally define a DL-based action formalism with causal relationships. In Sections 6.2 and 6.3, we then show that important inference problems such as the consistency problem and the projection problem are decidable in this formalism. Then, in Section 6.4, we show how to verify temporal properties in this more expressive action formalism. Lastly, in Section 6.5, we briefly summarise the results of this chapter.

6.1 DL-Based Action Formalisms and Causal Relationships

The situation calculus [Rei01] and the fluent calculus [Thi05b] are popular many-sorted languages for representing action theories. However, for action theories represented in those languages, important inference problems are in general undecidable, since these languages encompass full first-order logic. One possibility to restrict these languages to avoid this source of undecidability is to use a decidable fragment of first-order logic instead of full first-order logic as underlying base logic. Description logics are well-suited for this purpose since their expressive power goes far beyond propositional logic, whilst reasoning in DLs is still decidable.

Basically, an action theory consists of three components: (i) a description of the *initial state*, (ii) a description of the possible *actions*, and (iii) a description of the *domain constraints*. For each action, it is specified what the *pre-conditions* are, which need to be satisfied for an action to be applicable, and it is specified what the *post-conditions* are, i.e. the changes to the current state that the execution of the action causes. The domain constraints formulate general knowledge about the functioning of the domain, in which the actions are executed, and thus restrict the possible states. This is realised in a DL-based action formalism as follows. The initial state is described by an ABox. This description is incomplete due to the open-world assumption. The pre-conditions are ABox assertions that must hold, post-conditions are ABox assertions that are added or removed, and domain constraints are specified using TBox axioms.

The *projection problem* [Rei01] is one of the most basic reasoning problems for action formalisms. Intuitively, it deals with the question whether after applying a sequence of actions to an initial state a certain property holds. In expressive action formalisms such as the situation calculus, this property is specified with a formula of first-order logic, whereas in the case of DL-based action formalisms this property is specified with an ABox assertion.

The first action formalism based on DLs was introduced in [BLM+05a], and the authors have shown that the projection problem and other important inference problems become decidable in this restricted formalism. This action formalism has been further examined and extended in the last years [LLM+06; BLM10; BLL10b; BZ13]. Recently, an action formalism that is based on a DL and the situation calculus was proposed [GS10; SY12]. Both action formalisms, an extension of the one in [BLM+05a] and the one in [GS10; SY12], have been evaluated by implementing their respective approaches to solving the projection problem, and comparing the running times on random testing data for several realistic application domains [YLL+12]. This evaluation is, however, beyond the scope of this thesis.

Before we introduce our DL-based action formalism formally, we recall an important problem, namely the ramification problem, which has to be solved by the action formalism.

6.1.1 The Ramification Problem

The interaction of post-conditions of an action and domain constraints can cause so-called *ramifications*. More precisely, when an action is applied to a state, it might not be enough to make only those changes to the current state that are explicitly stated in the post-conditions of the action that is applied, since it is possible that the resulting state does not satisfy the domain constraints. We call the changes required by the post-conditions of an action *direct effects*, whereas we call the additional changes that one needs to make such that the resulting state after applying the action satisfies also the domain constraints *indirect effects*. The ramification problem deals now with the question how to characterise both the direct and the indirect effects while still solving the *frame problem*, i.e. do not characterise the complete resulting state but only the 'necessary' changes to the current state, which are required by the applied action and the domain constraints [MH69; Rei01].

Example 6.1. *Take a hiring action, which has the direct effect that the person that is hired becomes an employee. Moreover, we have a domain constraint that says that every employee must have a health insurance. If John, for instance, does not have health insurance, then just* *applying the hiring action for John, i.e. hiring John, would result in a state where John is an employee without a health insurance, which violates the domain constraint.* ♢

One approach to solving the ramification problem is to define a semantics for action theories that automatically deals with such indirect effects. This semantics should describe additional changes to the state in order to satisfy the domain constraints, whilst taking care that only 'necessary' changes are made. An example of such an attempt is the possible-models approach (PMA) [Win88; Her96]. Without additional restrictions, however, the PMA and all the other approaches in this direction can lead to unintuitive results. It is not clear how to construct a general semantics that does not suffer from this problem. Consider again Example 6.1, and assume that there are only two insurance companies that offer health insurance: AOK and TK. To satisfy the health-insurance domain constraint, John must get insured by one of them. However, it is unclear how to design a general semantic framework, which is able to decide which one to pick.

A second approach is avoiding the issues raised by the ramification problem rather than solving them. This is actually what is done in [BLM+05a]. There, the domain constraints are given by an acyclic TBox and the post-conditions of the actions are restricted such that only *primitive* concepts and roles can be changed. Recall the definition of the syntax and semantics of acyclic TBoxes; see Definitions 2.5 and 2.6. One can observe that w.r.t. an acyclic TBox, the interpretations of the primitive concepts and roles uniquely determine the interpretations of the defined concepts. Thus, in this restricted action formalism, it is clear what indirect effects changing a primitive concept or role has. The semantics obtained in this way can be seen as an instance of the PMA. It is shown in [BLM+05a] that the use of the PMA in a less restrictive setting, i.e. using general TBoxes to describe the domain constraints or allowing defined concepts to occur in post-conditions, can lead to unintuitive results.

A third approach is letting the user rather than a general semantic machinery decide which the indirect effects of an action are. To resolve the ramifications in Example 6.1, we assume that employers actually are required to enrol new employees with AOK in case they do not already have a health insurance. However, one needs to extend the action formalism such that it allows the user to add such information to the action theory. For DL-based action formalisms, this approach was first employed in [LLM+06], where the formalism for describing the actions is enriched such that the user can make complex statements about the changes to the interpretations of concepts and roles that can be caused by applying a given action. The authors show that important inference problems such as the projection problem stay decidable in this setting, but another important inference problem for action formalisms, namely the *consistency problem*,[1] becomes undecidable. Basically, an action is consistent if, whenever it is applicable to a state, there is a well-defined successor state that can be obtained after applying it.

In this chapter, we realise this third approach in a different way, namely by adapting a method for addressing the ramification problem that has already been employed in the reasoning about actions community [Lin95; Thi97; BDT98; DTB98; LS11; ST13]. Instead of changing the formalism for defining actions directly, we introduce so-called *causal relationships* as an additional component of action theories. In Example 6.1, such a causal relationship would state that, whenever someone becomes a new employee, this person is then insured by AOK, unless this person already had a health insurance.

[1] In [LLM+06], this is actually called strong consistency.

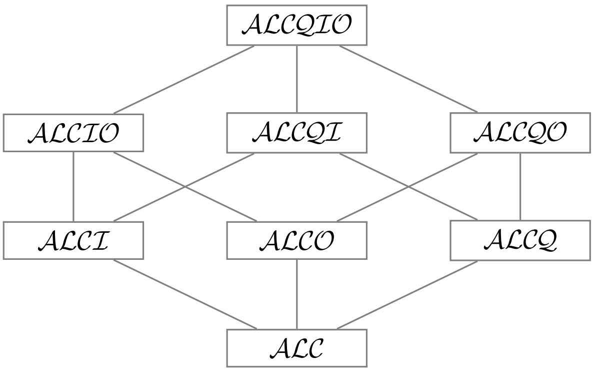

Figure 6.2: The relative expressivity of the DLs between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$

In Section 6.1.2, we formally introduce a DL-based action formalism with causal relationships. This new action formalism has two advantages over the one introduced in [LLM+06]. Firstly, the action formalism in [LLM+06] requires of the user to deal with the ramification problem within every single action description. In our action formalism, however, causal relationships are defined independently of a specific action, as they state general facts about causation. The formal semantics takes then care of how these relationships are translated into the indirect effects of the actions. A second advantage is that the consistency problem in our action formalism is decidable. This advantage is crucial since in the context of the third approach, the *user* is supposed to deal with the ramification problem, which mean in our action formalism that the user needs to define appropriate causal relationships. Testing the consistency of actions might help the user with this task, because it enables the user to check whether all relevant causal relationships have been stated correctly. Coming back to Example 6.1, it is clear that if the user does not specify any causal relationships, the hiring action is inconsistent since its application may result in a state that does not satisfy the health-insurance domain constraint, and thus is not well-defined. If the user, however, adds the causal relationship mentioned above, then the action becomes consistent. We show that in our action formalism the consistency problem is decidable in Section 6.2. After that, in Section 6.3, we show that also the projection problem is decidable in action theories stated in our action formalism.

6.1.2 A DL-Based Action Formalism with Causal Relationships

In principle, the action formalism can be parameterised with any DL. In this chapter, we focus on DLs between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$. The relative expressivity of these DLs is depicted in Figure 6.2. Since most of the notions in this chapter do not depend on the specific DL chosen, we again omit the prefix in the formal definitions and write e.g. ABox instead of $\mathcal{ALCQIO}$-ABox.

For defining the action formalism, we need besides atomic assertions also *negated* atomic assertions, i.e. ABox-literals. The semantics of ABox-literals extends the one of assertions in Definition 2.10 in a straightforward manner.

Definition 6.3 (ABox-literal). *An* ABox-literal *is either an atomic concept assertion $A(a)$, an atomic role assertion $r(a,b)$, a negated atomic concept assertion $\neg A(a)$, or a negated atomic role assertion $\neg r(a,b)$, where $A \in \mathsf{N_C}$, $r \in \mathsf{N_R}$, and $a, b \in \mathsf{N_I}$.*

A generalised ABox-literal *is either an ABox-literal, a concept assertion $C(a)$, or a negated concept assertion $\neg C(a)$, where C is a concept, and $a \in \mathsf{N_I}$.*

The interpretation $\mathcal{I}$ is a model *of a (generalised) ABox-literal of the form $\neg\alpha$ (written $\mathcal{I} \models \neg\alpha$) if $\mathcal{I} \not\models \alpha$, where $\mathcal{I} \models \alpha$ is defined as in Definition 2.10.* ◇

We call the non-negated assertions *positive*, and the negated assertions *negative*. Given a (generalised) ABox-literal α, we denote its negation by $\neg\alpha$. For the ease of presentation, we identify in the following $\neg\neg\beta$ and β for every (generalised) ABox-literal β. Note that finite sets of (generalised) ABox-literals are in general no ABoxes, because of the presence of negative assertions. To close this gap, we introduce the notion of a generalised ABox.

Definition 6.4 (Generalised ABox). *A* generalised ABox *is a finite set of generalised ABox-literals. The interpretation $\mathcal{I}$ is a* model *of the generalised ABox $\mathcal{A}$ (written $\mathcal{I} \models \mathcal{A}$) if it is a model of each generalised ABox-literal in $\mathcal{A}$. We call $\mathcal{A}$* consistent *if it has a model.* ◇

It is now obvious how to define knowledge bases that contain generalised ABoxes instead of classical ones. Recall that in this chapter we consider only DLs between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$, and thus there is no RBox.

Definition 6.5 (Generalised knowledge base). *A* generalised knowledge base *is a pair $\mathcal{K} = (\mathcal{A}, \mathcal{T})$ where $\mathcal{A}$ is a generalised ABox and $\mathcal{T}$ is a TBox.*

The interpretation $\mathcal{I}$ is a model *of $\mathcal{K}$ (written $\mathcal{I} \models \mathcal{K}$) if it is a model of $\mathcal{A}$ and $\mathcal{T}$. We call $\mathcal{K}$* consistent *if it has a model.*

We say that $\mathcal{K}$ entails *a generalised ABox-literal α (written $\mathcal{K} \models \alpha$) if all models of $\mathcal{K}$ are also models of α.* ◇

We are now ready to recall the notion of a DL-action without occlusions,[2] which has first been introduced in [BLM+05a]. In this chapter, we do not allow occlusions in our framework since it is not yet clear how to handle them algorithmically in the presence of causal relationships.

Definition 6.6 (Syntax of DL-actions). *A* DL-action *is a pair $\boldsymbol{a} = (\mathsf{pre}, \mathsf{post})$ where*

- pre *is a finite set of generalised ABox-literals called* pre-conditions*, and*
- post *is a finite set of* conditional post-conditions *of the form α/β and* unconditional post-conditions *of the form β, where α is a generalised ABox-literal and β is an* ABox-literal.

A DL-action is called unconditional *if all its post-conditions are unconditional.* ◇

Basically, a DL-action is applicable in an interpretation if all its pre-conditions are satisfied, and the conditional post-condition α/β requires that β must hold after applying the action if α was satisfied before the application. We can now express the hiring action of Example 6.1 formally.

[2] Occlusions describe which parts of the domain can change arbitrarily when an action is applied. Details about occlusions can be found in [BLM+05a].

Example 6.7. *A DL-action for hiring John would be formalised as*

$$\mathsf{HireJohn} := (\emptyset,\ \{\mathsf{Employee(John)}\}).$$

This action has no pre-conditions and a single unconditional post-condition. Additionally, the domain constraints are described in the following TBox:

$$\begin{aligned}\mathcal{T} := \big\{\ &\{\mathsf{AOK}\} \sqcup \{\mathsf{TK}\} \sqsubseteq \mathsf{HealthInsuranceCompany},\\ &\mathsf{Employee} \sqsubseteq \exists\mathsf{insuredBy}.\mathsf{HealthInsuranceCompany}\ \big\},\end{aligned}$$

where the first GCI states that AOK and TK are health-insurance companies, and the second GCI states that every employee needs to be insured by a health-insurance company. ◇

As sketched in Section 6.1.1, this example can be used to show that just considering the direct effects of the actions is not adequate if the domain constraints are given by a general TBox containing arbitrary GCIs rather than an acyclic TBox as it is done in [BLM+05a]. To be more precise, take a model $\mathcal{I}$ of the TBox $\mathcal{T}$ such that we have $\mathcal{I} \not\models \mathsf{Employee(John)}$ and $\mathcal{I} \not\models (\exists\mathsf{insuredBy}.\mathsf{HealthInsuranceCompany})(\mathsf{John})$. It should be clear that such a model exists. According to the semantics of DL-actions defined in [BLM+05a], after applying the DL-action $\mathsf{HireJohn}$ to $\mathcal{I}$, nothing should change that is not explicitly required to be changed by some post-condition. Hence, if we apply $\mathsf{HireJohn}$ to $\mathcal{I}$ using that semantics, $\mathcal{I}$ is transformed into an interpretation $\mathcal{I}'$ such that the only difference to $\mathcal{I}$ is that $\mathcal{I}' \models \mathsf{Employee(John)}$, i.e. John is now an employee. Since nothing else is allowed to change, we still have $\mathcal{I}' \not\models (\exists\mathsf{insuredBy}.\mathsf{HealthInsuranceCompany})(\mathsf{John})$. This a counterexample to the second GCI of $\mathcal{T}$, and thus $\mathcal{I}'$ is not a model of $\mathcal{T}$. Consequently, even though the DL-action $\mathsf{HireJohn}$ is applicable to $\mathcal{I}$—since the empty set of pre-conditions does not impose any applicability condition—, its application does not result in an interpretation satisfying the domain constraints in $\mathcal{T}$. We call a DL-action where this kind of problem can occur an *inconsistent DL-action*. To achieve consistency of the DL-action $\mathsf{HireJohn}$, we can complement the DL-action with an appropriate causal relationship.

Definition 6.8 (Causal relationship). *A* causal relationship *is of the form $\mathcal{A}_1 \to_{\mathcal{B}} \mathcal{A}_2$ where $\mathcal{A}_1$, $\mathcal{A}_2$ are finite sets of ABox-literals, and $\mathcal{B}$ is a generalised ABox.* ◇

A causal relationship can be read as '$\mathcal{A}_1$ causes $\mathcal{A}_2$ if $\mathcal{B}$ holds'. To be more precise, it means the following: if $\mathcal{B}$ is satisfied *before* the application of a DL-action $\boldsymbol{a}$, and $\mathcal{A}_1$ is *newly* satisfied by its application—i.e. it was not satisfied before, but is satisfied after the application of $\boldsymbol{a}$—, then $\mathcal{A}_2$ must also be satisfied *after* the application of $\boldsymbol{a}$.[3] Therefore, we often call $\mathcal{A}_1$ the *trigger*, $\mathcal{A}_2$ the *consequence set*, and $\mathcal{B}$ the *condition* of a causal relationship.

Example 6.9. *Consider the causal relationship*

$$\{\mathsf{Employee(John)}\} \to_{\mathcal{B}} \{\mathsf{insuredBy(John, AOK)}\}$$

with $\mathcal{B} := \{\neg(\exists\mathsf{insuredBy}.\mathsf{HealthInsuranceCompany})(\mathsf{John})\}$. *This causal relationship indeed adds the appropriate indirect effects to the direct effect of the DL-action* $\mathsf{HireJohn}$. *It states that*

[3] Actually, there are different ways of defining the meaning of causal relationships. We follow here the approach used in [BDT98; DTB98] rather than the one employed by [Lin95; Thi97]. The meaning of causal relationships in [Lin95; Thi97] requires that $\mathcal{B}$ is satisfied *after* the application of $\boldsymbol{a}$ instead of before.

if John becomes newly employed—i.e. he was not an employee before—, and he did not have a health insurance before the application of the DL-action, then he is newly insured with AOK after its application. If on the other hand, John becomes newly employed, but already has a health insurance, then he keeps his old health insurance and is not *newly insured with AOK. In both cases, the domain constraints stated in the TBox $\mathcal{T}$ of Example 6.7 stay satisfied.* ♢

To define the semantics of DL-actions in the presence of causal relationships formally, we need some more notions. DL-actions and causal relationships as they are introduced above can only cause changes to *named* individuals, i.e. state that a named individual does (not) belong to an atomic concept, and similarly that pairs of named individuals are (not) connected via a specific role. Consequently, such *effects* can be described in an obvious way using ABox-literals. Therefore, we sometimes call a finite set of ABox-literals a *set of effects*.

We define the set of *direct effects* using the definition of the semantics of DL-actions introduced in [BLM+05a].

Definition 6.10 (Direct effects). *For a DL-action $\boldsymbol{a} = (\mathsf{pre}, \mathsf{post})$, and an interpretation $\mathcal{I}$, the* set of direct effects of $\boldsymbol{a}$ on $\mathcal{I}$ *is defined as*

$$\mathsf{Dir}(\boldsymbol{a}, \mathcal{I}) := \{\beta \mid \beta \in \mathsf{post}\} \cup \{\beta \mid \alpha/\beta \in \mathsf{post} \text{ and } \mathcal{I} \models \alpha\}. \qquad \diamondsuit$$

Direct effects of a DL-action might cause indirect effects specified by causal relationships. Whether a specific causal relationship is applicable depends both on the interpretation to which the DL-action is applied, and a set of effects computed so far.

Definition 6.11 (Indirect effects). *The causal relationship $\mathcal{A}_1 \rightarrow_{\mathcal{B}} \mathcal{A}_2$ in the finite set of causal relationships* CR *is* applicable to an interpretation $\mathcal{I}$ and a set of effects $\mathcal{E}$ *if*

1. $\mathcal{I} \models \mathcal{B}$,
2. $\mathcal{I} \not\models \mathcal{A}_1$, *and*
3. *for every $\alpha \in \mathcal{A}_1$, either $\alpha \in \mathcal{E}$, or $\mathcal{I} \models \alpha$ and $\neg\alpha \notin \mathcal{E}$.*

The set of indirect effects of CR on $\mathcal{I}$ and $\mathcal{E}$ *is defined as*

$$\mathsf{Indir}(\mathsf{CR}, \mathcal{I}, \mathcal{E}) := \{\beta \mid \beta \in \mathcal{A}_2 \text{ for some } \mathcal{A}_1 \rightarrow_{\mathcal{B}} \mathcal{A}_2 \in \mathsf{CR} \text{ applicable to } \mathcal{I} \text{ and } \mathcal{E}\}. \qquad \diamondsuit$$

According to this definition, the causal relationship $\mathcal{A}_1 \rightarrow_{\mathcal{B}} \mathcal{A}_2 \in \mathsf{CR}$ is applicable if the condition $\mathcal{B}$ is satisfied in $\mathcal{I}$, i.e. before applying the DL-action, (Condition 1), and the trigger $\mathcal{A}_1$ is newly satisfied, i.e. $\mathcal{A}_1$ is not satisfied in $\mathcal{I}$ (Condition 2), but it is satisfied according to the effect set $\mathcal{E}$, i.e. every generalised ABox-literal $\alpha \in \mathcal{A}_1$ is either an effect, or it is satisfied in $\mathcal{I}$, which is not changed by an effect (Condition 3).

Obviously, the indirect effects caused by a causal relationship may again cause causal relationships to be applicable, which cause again indirect effects. Thus, the overall effects of an action are obtained by iteratively adding indirect effects to the direct ones until no new indirect effects can be added.

Definition 6.12 (Effects). *For a DL-action $\boldsymbol{a} = (\mathsf{pre}, \mathsf{post})$, a finite set of causal relationships* CR*, and an interpretation $\mathcal{I}$, the* set of effects of $\boldsymbol{a}$ on $\mathcal{I}$ w.r.t. CR *is defined as*

$$\mathsf{Eff}(\boldsymbol{a}, \mathcal{I}, \mathsf{CR}) := \bigcup_{i \geq 0} \mathsf{Eff}_i(\boldsymbol{a}, \mathcal{I}, \mathsf{CR})$$

where $\mathsf{Eff}_i(\boldsymbol{a},\mathcal{I},\mathsf{CR})$ *is defined inductively as follows:*

- $\mathsf{Eff}_0(\boldsymbol{a},\mathcal{I},\mathsf{CR}) := \mathsf{Dir}(\boldsymbol{a},\mathcal{I})$*; and*
- $\mathsf{Eff}_{i+1}(\boldsymbol{a},\mathcal{I},\mathsf{CR}) := \mathsf{Eff}_i(\boldsymbol{a},\mathcal{I},\mathsf{CR}) \cup \mathsf{Indir}(\mathsf{CR},\mathcal{I},\mathsf{Eff}_i(\boldsymbol{a},\mathcal{I},\mathsf{CR}))$. ◇

Moreover, the set $\mathsf{Eff}(\boldsymbol{a},\mathcal{I},\mathsf{CR})$ can effectively be computed due to the following arguments. Firstly, we have by definition that

$$\mathsf{Eff}_0(\boldsymbol{a},\mathcal{I},\mathsf{CR}) \subseteq \mathsf{Eff}_1(\boldsymbol{a},\mathcal{I},\mathsf{CR}) \subseteq \mathsf{Eff}_2(\boldsymbol{a},\mathcal{I},\mathsf{CR}) \subseteq \ldots,$$

and secondly, since we add only ABox-literals that belong to the consequence set of a causal relationship, and the set CR is moreover finite, there must exist an $n \geq 0$ such that

$$\mathsf{Eff}_n(\boldsymbol{a},\mathcal{I},\mathsf{CR}) = \mathsf{Eff}_{n+1}(\boldsymbol{a},\mathcal{I},\mathsf{CR}) = \mathsf{Eff}_{n+2}(\boldsymbol{a},\mathcal{I},\mathsf{CR}) = \ldots.$$

Thus, $\mathsf{Eff}(\boldsymbol{a},\mathcal{I},\mathsf{CR}) = \mathsf{Eff}_n(\boldsymbol{a},\mathcal{I},\mathsf{CR})$, i.e. we obtain $\mathsf{Eff}(\boldsymbol{a},\mathcal{I},\mathsf{CR})$ after n iterative steps, where n is polynomially bounded by the size of CR.

Note, however, that it could happen that the set $\mathsf{Eff}(\boldsymbol{a},\mathcal{I},\mathsf{CR})$ is *contradictory*, i.e. that there is an ABox-literal α such that $\{\alpha,\neg\alpha\} \subseteq \mathsf{Eff}(\boldsymbol{a},\mathcal{I},\mathsf{CR})$. Then it can, of course, not lead to a well-defined successor interpretation.

We are now ready to formally define the semantics of DL-actions in the presence of causal relationships.

Definition 6.13 (Semantics of DL-actions). *Let* $\boldsymbol{a} = (\mathsf{pre},\mathsf{post})$ *be a DL-action,* CR *a finite set of causal relationships,* $\mathcal{T}$ *a TBox, and* $\mathcal{I} = (\Delta^{\mathcal{I}},\cdot^{\mathcal{I}})$ *and* $\mathcal{I}' = (\Delta^{\mathcal{I}'},\cdot^{\mathcal{I}'})$ *two interpretations. We say that* $\boldsymbol{a}$ is applicable to $\mathcal{I}$ w.r.t. $\mathcal{T}$ *if* $\mathcal{I} \models \mathcal{T}$ *and* $\mathcal{I} \models \mathsf{pre}$. *Moreover,* $\boldsymbol{a}$ transforms $\mathcal{I}$ into $\mathcal{I}'$ w.r.t. $\mathcal{T}$ and CR *(written* $\mathcal{I} \Rightarrow_{\boldsymbol{a}}^{\mathcal{T},\mathsf{CR}} \mathcal{I}'$*) if*

1. $\boldsymbol{a}$ *is applicable to* $\mathcal{I}$ *w.r.t.* $\mathcal{T}$*;*
2. $\Delta^{\mathcal{I}} = \Delta^{\mathcal{I}'}$ *and* $a^{\mathcal{I}} = a^{\mathcal{I}'}$ *for every* $a \in \mathsf{N_I}$*;*
3. $\mathcal{I}' \models \mathcal{T}$*;*
4. $\mathsf{Eff}(\boldsymbol{a},\mathcal{I},\mathsf{CR})$ *is not contradictory;*
5. *for every* $A \in \mathsf{N_C}$*, we have* $A^{\mathcal{I}'} = (A^{\mathcal{I}} \cup A^{+}) \setminus A^{-}$ *where*

$$A^{+} := \{a^{\mathcal{I}} \mid A(a) \in \mathsf{Eff}(\boldsymbol{a},\mathcal{I},\mathsf{CR})\}, \text{ and}$$
$$A^{-} := \{a^{\mathcal{I}} \mid \neg A(a) \in \mathsf{Eff}(\boldsymbol{a},\mathcal{I},\mathsf{CR})\}; \text{ and}$$

6. *for every* $r \in \mathsf{N_R}$*, we have* $r^{\mathcal{I}'} = (r^{\mathcal{I}} \cup r^{+}) \setminus r^{-}$ *where*

$$r^{+} := \{(a^{\mathcal{I}},b^{\mathcal{I}}) \mid r(a,b) \in \mathsf{Eff}(\boldsymbol{a},\mathcal{I},\mathsf{CR})\}, \text{ and}$$
$$r^{-} := \{(a^{\mathcal{I}},b^{\mathcal{I}}) \mid \neg r(a,b) \in \mathsf{Eff}(\boldsymbol{a},\mathcal{I},\mathsf{CR})\}.$$

The finite sequence of DL-actions $\boldsymbol{a}_1,\ldots,\boldsymbol{a}_n$ transforms $\mathcal{I}$ into $\mathcal{I}'$ w.r.t. $\mathcal{T}$ and CR *(written* $\mathcal{I} \Rightarrow_{\boldsymbol{a}_1,\ldots,\boldsymbol{a}_n}^{\mathcal{T},\mathsf{CR}} \mathcal{I}'$*) if there are interpretations* $\mathcal{I}_0,\ldots,\mathcal{I}_n$ *such that* $\mathcal{I}_0 = \mathcal{I}$*,* $\mathcal{I}_n = \mathcal{I}'$*, and* $\mathcal{I}_{i-1} \Rightarrow_{\boldsymbol{a}_i}^{\mathcal{T},\mathsf{CR}} \mathcal{I}_i$ *for every* i*,* $1 \leq i \leq n$*.* ◇

Note that if $\mathcal{T}$ and CR are empty, then this semantics is very similar with the one of DL-actions without occlusions given in [BLM+05a]. However, Condition 1 is not demanded in [BLM+05a], which we do here for convenience. The following lemma is an immediate consequence of this definition.

Lemma 6.14. *Let $\boldsymbol{a}$ be a DL-action, CR a finite set of causal relationships, $\mathcal{T}$ a TBox, and $\mathcal{I}$ and $\mathcal{I}'$ two interpretations. If we have $\mathcal{I} \Rightarrow_{\boldsymbol{a}}^{\mathcal{T},\mathsf{CR}} \mathcal{I}'$, then $\mathcal{I}' \models \mathsf{Eff}(\boldsymbol{a},\mathcal{I},\mathsf{CR})$.*

Proof. Assume that $\mathcal{I} \Rightarrow_{\boldsymbol{a}}^{\mathcal{T},\mathsf{CR}} \mathcal{I}'$. Then, by Definition 6.13, we have that $\mathsf{Eff}(\boldsymbol{a},\mathcal{I},\mathsf{CR})$ is not contradictory. Thus, we have that $A^+ \cap A^- = \emptyset$ and $r^+ \cap r^- = \emptyset$, where A^+, A^-, r^+, and r^- are defined as in Definition 6.13. We prove that for every $\alpha \in \mathsf{Eff}(\boldsymbol{a},\mathcal{I},\mathsf{CR})$, we have $\mathcal{I}' \models \alpha$ by a case distinction.

If α is of the form $A(a)$ for $A \in \mathsf{N_C}$ and $a \in \mathsf{N_I}$, we have that $a^{\mathcal{I}} \in A^+$. Thus, Definition 6.13 yields that $a^{\mathcal{I}'} = a^{\mathcal{I}} \in (A^{\mathcal{I}} \cup A^+) \setminus A^- = A^{\mathcal{I}'}$. If α is of the form $\neg A(a)$ for $A \in \mathsf{N_C}$ and $a \in \mathsf{N_I}$, we have that $a^{\mathcal{I}} \in A^-$. Again by Definition 6.13, we have that $a^{\mathcal{I}'} = a^{\mathcal{I}} \notin (A^{\mathcal{I}} \cup A^+) \setminus A^- = A^{\mathcal{I}'}$. If α is of the form $r(a,b)$ for $r \in \mathsf{N_R}$ and $a,b \in \mathsf{N_I}$, we have that $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^+$. Thus, Definition 6.13 yields that $(a^{\mathcal{I}'}, b^{\mathcal{I}'}) = (a^{\mathcal{I}}, b^{\mathcal{I}}) \in (r^{\mathcal{I}} \cup r^+) \setminus r^- = r^{\mathcal{I}'}$. Finally, if α is of the form $\neg r(a,b)$ for $r \in \mathsf{N_R}$ and $a,b \in \mathsf{N_I}$, we have that $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^-$. Again, Definition 6.13 yields that $(a^{\mathcal{I}'}, b^{\mathcal{I}'}) = (a^{\mathcal{I}}, b^{\mathcal{I}}) \notin (r^{\mathcal{I}} \cup r^+) \setminus r^- = r^{\mathcal{I}'}$. □

It is also important to note that the DL-actions defined here are *deterministic* in the following sense: for every model $\mathcal{I}$ of $\mathcal{T}$, there exists *at most one* interpretation $\mathcal{I}'$ such that $\mathcal{I} \Rightarrow_{\boldsymbol{a}}^{\mathcal{T},\mathsf{CR}} \mathcal{I}'$. There are several reasons why such an interpretation $\mathcal{I}'$ might not exist. Firstly, this is the case if Condition 1 is violated, i.e. $\boldsymbol{a}$ is not applicable to $\mathcal{I}$ w.r.t. $\mathcal{T}$. A second reason is that even if Condition 1 is satisfied, Condition 4 is violated, i.e. the set $\mathsf{Eff}(\boldsymbol{a},\mathcal{I},\mathsf{CR})$ is contradictory. Lastly, it might be the case that Conditions 1 and 4 are satisfied, but the new interpretation induced by $\mathsf{Eff}(\boldsymbol{a},\mathcal{I},\mathsf{CR})$ is not a model of $\mathcal{T}$. If such an $\mathcal{I}'$ does not exist, even if Condition 1 is satisfied, this indicates a modelling error. In fact, the correct modelling of an action theory should ensure that for every applicable DL-action, there is a well-defined successor state.

Definition 6.15 (Consistency problem). *The DL-action $\boldsymbol{a} = (\mathsf{pre},\mathsf{post})$ is* consistent *w.r.t. the TBox $\mathcal{T}$ and the finite set CR of causal relationships if for every interpretation $\mathcal{I}$ such that $\boldsymbol{a}$ is applicable to $\mathcal{I}$ w.r.t. $\mathcal{T}$, there exists an interpretation $\mathcal{I}'$ such that $\mathcal{I} \Rightarrow_{\boldsymbol{a}}^{\mathcal{T},\mathsf{CR}} \mathcal{I}'$.*

The consistency problem *is then to decide whether $\boldsymbol{a}$ is consistent w.r.t. $\mathcal{T}$ and CR.* ◊

As argued above, the DL-action HireJohn of Example 6.7 is *not* consistent w.r.t. the TBox $\mathcal{T}$ defined there and the empty set of causal relationships. However, it becomes consistent if we add the causal relationship of Example 6.9.

We are now ready to define the projection problem formally. Recall that it deals with the question whether for a given (possible incomplete) description of the initial state, a certain property is guaranteed to hold after the execution of a sequence of DL-actions. Our formal definition of this problem is very similar to the one from [BLM+05a], with the difference that we use the 'transforms' relation ($\Rightarrow_{\boldsymbol{a}}^{\mathcal{T},\mathsf{CR}}$) introduced in Definition 6.13, which takes a general TBox and a set of causal relationships into account, instead of the one employed in [BLM+05a].

Definition 6.16 (Projection problem). *Let $\mathcal{A}$ be a generalised ABox, $\mathcal{T}$ be a TBox, CR be a finite set of causal relationships, α be a generalised ABox-literal, and $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$ be a sequence* *of DL-actions such that the DL-action* $\boldsymbol{a}_i$ *is consistent w.r.t.* $\mathcal{T}$ *and* CR *for every* i, $1 \leq i \leq n$. *We say that* α is a consequence of applying $\boldsymbol{a}_1, \dots, \boldsymbol{a}_n$ to $\mathcal{A}$ w.r.t. $\mathcal{T}$ and CR *if for every* $\mathcal{I}$ *and* $\mathcal{I}'$ *with* $\mathcal{I} \models \mathcal{A}$ *and* $\mathcal{I} \Rightarrow^{\mathcal{T},\mathsf{CR}}_{\boldsymbol{a}_1,\dots,\boldsymbol{a}_n} \mathcal{I}'$, *we have* $\mathcal{I}' \models \alpha$.

The projection problem *is then to decide whether* α *is a consequence of applying* $\boldsymbol{a}_1, \dots, \boldsymbol{a}_n$ *to* $\mathcal{A}$ *w.r.t.* $\mathcal{T}$ *and* CR. ♢

Note that in this definition, we consider only DL-actions that are *consistent w.r.t.* $\mathcal{T}$ *and* CR. As argued above, if any DL-action is inconsistent w.r.t. $\mathcal{T}$ and CR, then this indicates a modelling error in the action theory, and this issue should be addressed before starting to ask projection questions. However, it could also happen that not all pre-conditions are guaranteed to be satisfied during the execution of a sequence of DL-actions. The *executability problem* [Rei01], another interesting inference problem for action theories, deals with this question.

Definition 6.17 (Executability problem). *Let* $\mathcal{A}$ *be a generalised ABox,* $\mathcal{T}$ *be a TBox,* CR *be a finite set of causal relationships, and* $\boldsymbol{a}_1, \dots, \boldsymbol{a}_n$ *be a sequence of DL-actions such that the DL-action* $\boldsymbol{a}_i = (\mathsf{pre}_i, \mathsf{post}_i)$ *is consistent w.r.t.* $\mathcal{T}$ *and* CR *for every* i, $1 \leq i \leq n$. *We say that* $\boldsymbol{a}_1, \dots, \boldsymbol{a}_n$ *is* executable *in* $\mathcal{A}$ *w.r.t.* $\mathcal{T}$ *and* CR *if for every model* $\mathcal{I}$ *of* $\mathcal{A}$ *and* $\mathcal{T}$, *we have:*

- $\mathcal{I} \models \mathsf{pre}_1$, *and*
- *for every* i, $1 \leq i < n$, *and all interpretations* $\mathcal{I}'$ *with* $\mathcal{I} \Rightarrow^{\mathcal{T},\mathsf{CR}}_{\boldsymbol{a}_1,\dots,\boldsymbol{a}_i} \mathcal{I}'$, *we have* $\mathcal{I}' \models \mathsf{pre}_{i+1}$.

The executability problem *is then to decide whether* $\boldsymbol{a}_1, \dots, \boldsymbol{a}_n$ *is executable in* $\mathcal{A}$ *w.r.t.* $\mathcal{T}$ *and* CR. ♢

Usually, before one asks for projection questions, one checks whether the sequence of DL-actions is indeed executable. However, Lemma 4 in [BLM+05a] states that for the action formalism defined there, the projection and the executability problem can be reduced to each other in polynomial time. The actual proof is shown in [BLM+05b]; there it is Lemma 11. The arguments can also be used to show that for the action formalism defined here, each executability problem can be reduced to polynomially many projection problems in polynomial time.[4] We repeat these arguments from [BLM+05b] for the sake of completeness.

Theorem 6.18. *The executability problem as introduced in Definition 6.17 can be reduced to polynomially many projection problems as introduced in Definition 6.16 in polynomial time.*

Proof. Take a generalised ABox $\mathcal{A}$, a TBox $\mathcal{T}$, a finite set of causal relationships CR, a generalised ABox-literal α, and a sequence of DL-actions $\boldsymbol{a}_1, \dots, \boldsymbol{a}_n$ such that the DL-action $\boldsymbol{a}_i = (\mathsf{pre}_i, \mathsf{post}_i)$ is consistent w.r.t. $\mathcal{T}$ and CR for every i, $1 \leq i \leq n$. We have that $\boldsymbol{a}_1, \dots, \boldsymbol{a}_n$ is executable in $\mathcal{A}$ w.r.t. $\mathcal{T}$ and CR iff

1. for every $\beta \in \mathsf{pre}_1$, we have that β is a consequence of applying $(\emptyset, \emptyset)$ to $\mathcal{A}$ w.r.t. $\mathcal{T}$ and CR; and
2. for every i, $1 \leq i < n$, and every $\beta \in \mathsf{pre}_{i+1}$, we have that β is a consequence of applying $\boldsymbol{a}_1, \dots, \boldsymbol{a}_i$ to $\mathcal{A}$ w.r.t. $\mathcal{T}$ and CR.

[4] The converse direction, i.e. that the projection problem can be reduced to the executability problem in polynomial time, does not follow from the arguments in [BLM+05b], because we take the pre-condition into account when defining the 'transforms' relation ($\Rightarrow^{\mathcal{T},\mathsf{CR}}_{\boldsymbol{a}}$).

It is easy to see that these are polynomially many projection problems, where each can be constructed in polynomial time. □

For this reason, we can restrict our attention to the consistency and the projection problem for showing decidability and complexity results. We first consider the consistency problem in Section 6.2, and then the projection problem in Section 6.3. Finally, in Section 6.4, we consider the problem of verifying temporal properties.

6.2 Deciding the Consistency Problem

In this section, we first consider the case where the TBox is empty, and develop a solution for this restricted case. After that, we show how this solution can be extended to deal with the general case.

6.2.1 Deciding the Consistency Problem w.r.t. the Empty TBox

In this section, we show that testing consistency of a DL-action w.r.t. the empty TBox and a finite set of causal relationships is decidable and has the same complexity as checking *in*consistency of a generalised ABox.

Given a DL-action $\boldsymbol{a}$ and a finite set of causal relationships CR, we basically consider all possible situations that $\boldsymbol{a}$ could encounter when it is applied to an interpretation. The relevant information is kept in a so-called *action type*.[5] For this, we define for a DL-action $\boldsymbol{a} = (\mathsf{pre}, \mathsf{post})$ and a finite set of causal relationships CR, the set $\mathsf{Cond}(\boldsymbol{a}, \mathsf{CR})$ as the closure under negation ($\neg$) of the set

$$\{\alpha \mid \alpha/\beta \in \mathsf{post}\} \cup \{\alpha \mid \alpha \in \mathcal{A}_1 \cup \mathcal{B} \text{ for some } \mathcal{A}_1 \rightarrow_{\mathcal{B}} \mathcal{A}_2 \in \mathsf{CR}\}.$$

Definition 6.19 (Action type). *An* action type for a DL-action $\boldsymbol{a}$ and a finite set of causal relationships CR *is a generalised ABox* $T \subseteq \mathsf{Cond}(\boldsymbol{a}, \mathsf{CR})$ *such that*

- $\neg\alpha \in T$ *iff* $\alpha \notin T$ *for every* $\neg\alpha \in \mathsf{Cond}(\boldsymbol{a}, \mathsf{CR})$*; and*
- T *is consistent.* ◇

We denote the set of all action types for a DL-action $\boldsymbol{a}$ and a finite set of causal relationships CR by $\mathfrak{T}(\boldsymbol{a}, \mathsf{CR})$. Moreover, for a given interpretation $\mathcal{I}$, there is exactly one action type $T \in \mathfrak{T}(\boldsymbol{a}, \mathsf{CR})$ such that $\mathcal{I} \models T$.

Lemma 6.20. *Let* $\boldsymbol{a}$ *be a DL-action, and* CR *a finite set of causal relationships. For a given interpretation* $\mathcal{I}$*, there is one unique action type* $T \in \mathfrak{T}(\boldsymbol{a}, \mathsf{CR})$ *such that* $\mathcal{I} \models T$.

Proof. Given an interpretation $\mathcal{I}$, we define $T := \{\alpha \in \mathsf{Cond}(\boldsymbol{a}, \mathsf{CR}) \mid \mathcal{I} \models \alpha\}$. We have obviously that $T \in \mathfrak{T}(\boldsymbol{a}, \mathsf{CR})$ and $\mathcal{I} \models T$. It is left to be shown that T is unique. Assume to the contrary that there is a $T' \in \mathfrak{T}(\boldsymbol{a}, \mathsf{CR})$ with $\mathcal{I} \models T'$ and $T \neq T'$. Since T and T' are non-equal action types for $\boldsymbol{a}$ and CR, there is a generalised ABox-literal α with $\alpha \in T$ and $\alpha \notin T'$, and thus $\neg\alpha \in T'$. Since $\mathcal{I} \models T$ and $\mathcal{I} \models T'$, we have that $\mathcal{I} \models \alpha$ and $\mathcal{I} \models \neg\alpha$, i.e. $\mathcal{I} \not\models \alpha$, which is a contradiction. □

[5] In [BLL10b], this is called a *diagram*.

We now continue by defining for an action type $T \in \mathfrak{T}(\boldsymbol{a}, \mathsf{CR})$, a set $\mathsf{Eff}(\boldsymbol{a}, T, \mathsf{CR})$, which describes the set of effects that $\boldsymbol{a}$ has on T w.r.t. CR. We can show that for every interpretation $\mathcal{I}$ with $\mathcal{I} \models T$, we have $\mathsf{Eff}(\boldsymbol{a}, T, \mathsf{CR}) = \mathsf{Eff}(\boldsymbol{a}, \mathcal{I}, \mathsf{CR})$, and thus it is sufficient to know the unique action type $T \in \mathfrak{T}(\boldsymbol{a}, \mathsf{CR})$ with $\mathcal{I} \models T$ to determine the direct and indirect effects of applying $\boldsymbol{a}$ to $\mathcal{I}$ w.r.t. CR.

The definition of the set of *direct effects* on T is very similar to Definition 6.10.

Definition 6.21 (Direct effects on action type). *For a DL-action* $\boldsymbol{a} = (\mathsf{pre}, \mathsf{post})$*, a finite set of causal relationships* CR*, and an action type* $T \in \mathfrak{T}(\boldsymbol{a}, \mathsf{CR})$*, the* set of direct effects of $\boldsymbol{a}$ on T *is defined as*

$$\mathsf{Dir}(\boldsymbol{a}, T) := \{\beta \mid \beta \in \mathsf{post}\} \cup \{\beta \mid \alpha/\beta \in \mathsf{post} \text{ and } \alpha \in T\}. \qquad \diamondsuit$$

The definition of the set of *indirect effects* on T is very similar to Definition 6.11.

Definition 6.22 (Indirect effects on action type). *The causal relationship* $\mathcal{A}_1 \rightarrow_{\mathcal{B}} \mathcal{A}_2$ *in the finite set of causal relationships* CR *is* applicable to an action type T and a set of effects $\mathcal{E}$ *iff*

1. $\mathcal{B} \subseteq T$,
2. $\mathcal{A}_1 \not\subseteq T$, *and*
3. *for every* $\alpha \in \mathcal{A}_1$*, either* $\alpha \in \mathcal{E}$*, or* $\alpha \in T$ *and* $\neg\alpha \notin \mathcal{E}$.

The set of indirect effects of CR on T and $\mathcal{E}$ *is defined as*

$$\mathsf{Indir}(\mathsf{CR}, T, \mathcal{E}) := \{\beta \mid \beta \in \mathcal{A}_2 \text{ for some } \mathcal{A}_1 \rightarrow_{\mathcal{B}} \mathcal{A}_2 \in \mathsf{CR} \text{ applicable to } T \text{ and } \mathcal{E}\}. \qquad \diamondsuit$$

Finally, the set of *effects* on T can be defined similar to what was done for Definition 6.12.

Definition 6.23 (Effects on action type). *For a DL-action* $\boldsymbol{a} = (\mathsf{pre}, \mathsf{post})$*, a finite set of causal relationships* CR*, and an action type* $T \in \mathfrak{T}(\boldsymbol{a}, \mathsf{CR})$*, the* set of effects of $\boldsymbol{a}$ on T w.r.t. CR *is defined as* $\mathsf{Eff}(\boldsymbol{a}, T, \mathsf{CR}) := \bigcup_{i \geq 0} \mathsf{Eff}_i(\boldsymbol{a}, T, \mathsf{CR})$ *where* $\mathsf{Eff}_i(\boldsymbol{a}, T, \mathsf{CR})$ *is defined inductively as follows:*

- $\mathsf{Eff}_0(\boldsymbol{a}, T, \mathsf{CR}) := \mathsf{Dir}(\boldsymbol{a}, T)$*; and*
- $\mathsf{Eff}_{i+1}(\boldsymbol{a}, T, \mathsf{CR}) := \mathsf{Eff}_i(\boldsymbol{a}, T, \mathsf{CR}) \cup \mathsf{Indir}(\mathsf{CR}, T, \mathsf{Eff}_i(\boldsymbol{a}, T, \mathsf{CR}))$. $\diamondsuit$

Again, the set $\mathsf{Eff}(\boldsymbol{a}, T, \mathsf{CR})$ can effectively be computed due to the same arguments that we used above to show that $\mathsf{Eff}(\boldsymbol{a}, \mathcal{I}, \mathsf{CR})$ can be computed effectively. Moreover, it is not hard to see that the set $\mathsf{Eff}(\boldsymbol{a}, T, \mathsf{CR})$ can be computed in time polynomial in the size of $\boldsymbol{a}$, T, and CR. Similar to before, we say that $\mathsf{Eff}(\boldsymbol{a}, T, \mathsf{CR})$ is *contradictory* if there is an ABox-literal α such that $\{\alpha, \neg\alpha\} \subseteq \mathsf{Eff}(\boldsymbol{a}, T, \mathsf{CR})$.

Lemma 6.24. *Let* $\boldsymbol{a}$ *be a DL-action,* CR *be a finite set of causal relationships, and* T *be an action type for* $\boldsymbol{a}$ *and* CR*. Then, for every interpretation* $\mathcal{I}$ *with* $\mathcal{I} \models T$*, we have that* $\mathsf{Eff}(\boldsymbol{a}, \mathcal{I}, \mathsf{CR}) = \mathsf{Eff}(\boldsymbol{a}, T, \mathsf{CR})$.

Proof. Take any interpretation $\mathcal{I}$ with $\mathcal{I} \models T$. We first show the following claim.

Claim 6.25. *Let* $\alpha \in \mathsf{Cond}(\boldsymbol{a}, \mathsf{CR})$*, and* $\mathcal{A} \subseteq \mathsf{Cond}(\boldsymbol{a}, \mathsf{CR})$*. Then, we have*

1. $\mathcal{I} \models \alpha$ *iff* $\alpha \in T$,

2. $\mathcal{I} \models \mathcal{A}$ *iff* $\mathcal{A} \subseteq T$,

3. $\mathsf{Dir}(\boldsymbol{a}, \mathcal{I}) = \mathsf{Dir}(\boldsymbol{a}, T)$*, and*

4. $\mathsf{Indir}(\mathsf{CR}, \mathcal{I}, \mathcal{E}) = \mathsf{Indir}(\mathsf{CR}, T, \mathcal{E})$ *for every set of effects* $\mathcal{E}$.

The 'if' direction of Part 1 of the claim is trivial since $\mathcal{I} \models T$. To prove the 'only if' direction, assume that $\mathcal{I} \models \alpha$, but $\alpha \notin T$. Since $T \in \mathfrak{T}(\boldsymbol{a}, \mathsf{CR})$ and $\alpha \in \mathsf{Cond}(\boldsymbol{a}, \mathsf{CR})$, we have $\neg\alpha \in T$. Then $\mathcal{I} \models T$ yields $\mathcal{I} \models \neg\alpha$, i.e. $\mathcal{I} \not\models \alpha$, which is a contradiction.

To prove Part 2 of the claim, take any $\beta \in \mathcal{A}$. Since $\beta \in \mathsf{Cond}(\boldsymbol{a}, \mathsf{CR})$, we have now by Part 1 of the claim that $\mathcal{I} \models \beta$ iff $\beta \in T$, which finishes this part of the claim.

For Part 3 of the claim, let $\boldsymbol{a} = (\mathsf{pre}, \mathsf{post})$. We have:

$$\begin{aligned}\mathsf{Dir}(\boldsymbol{a}, \mathcal{I}) &= \{\beta \mid \beta \in \mathsf{post}\} \cup \{\beta \mid \alpha/\beta \in \mathsf{post} \text{ and } \mathcal{I} \models \alpha\} && \text{(by Definition 6.10)}\\ &= \{\beta \mid \beta \in \mathsf{post}\} \cup \{\beta \mid \alpha/\beta \in \mathsf{post} \text{ and } \alpha \in T\} && \text{(by Part 1 of the claim)}\\ &= \mathsf{Dir}(\boldsymbol{a}, T) && \text{(by Definition 6.21)}\end{aligned}$$

Finally, to prove Part 4, take any causal relationship $\mathcal{A}_1 \rightarrow_{\mathcal{B}} \mathcal{A}_2 \in \mathsf{CR}$, and any set of effects $\mathcal{E}$. We have:

$\mathcal{A}_1 \rightarrow_{\mathcal{B}} \mathcal{A}_2$ is applicable to $\mathcal{I}$ and $\mathcal{E}$

iff $\mathcal{I} \models \mathcal{B}$, $\mathcal{I} \not\models \mathcal{A}_1$, and for every $\alpha \in \mathcal{A}_1$, either $\alpha \in \mathcal{E}$, or $\mathcal{I} \models \alpha$ and $\neg\alpha \notin \mathcal{E}$
(by Definition 6.11)

iff $\mathcal{B} \subseteq T$, $\mathcal{A}_1 \nsubseteq T$, and for every $\alpha \in \mathcal{A}_1$, either $\alpha \in \mathcal{E}$, or $\alpha \in T$ and $\neg\alpha \notin \mathcal{E}$
(by Parts 1 and 2 of the claim)

iff $\mathcal{A}_1 \rightarrow_{\mathcal{B}} \mathcal{A}_2$ is applicable to T and $\mathcal{E}$
(by Definition 6.22).

This yields using Definitions 6.11 and 6.22:

$$\begin{aligned}\mathsf{Indir}(\mathsf{CR}, \mathcal{I}, \mathcal{E}) &= \{\beta \mid \beta \in \mathcal{A}_2 \text{ for some } \mathcal{A}_1 \rightarrow_{\mathcal{B}} \mathcal{A}_2 \in \mathsf{CR} \text{ applicable to } \mathcal{I} \text{ and } \mathcal{E}\}\\ &= \{\beta \mid \beta \in \mathcal{A}_2 \text{ for some } \mathcal{A}_1 \rightarrow_{\mathcal{B}} \mathcal{A}_2 \in \mathsf{CR} \text{ applicable to } T \text{ and } \mathcal{E}\}\\ &= \mathsf{Indir}(\mathsf{CR}, T, \mathcal{E}).\end{aligned}$$

This finishes the proof the Claim 6.25.

To prove $\mathsf{Eff}(\boldsymbol{a}, \mathcal{I}, \mathsf{CR}) = \mathsf{Eff}(\boldsymbol{a}, T, \mathsf{CR})$, it is enough to prove by induction that we have $\mathsf{Eff}_i(\boldsymbol{a}, \mathcal{I}, \mathsf{CR}) = \mathsf{Eff}_i(\boldsymbol{a}, T, \mathsf{CR})$ for every $i \geq 0$. For $i = 0$, we have by Definitions 6.12 and 6.23, and Part 3 of Claim 6.25:

$$\mathsf{Eff}_0(\boldsymbol{a}, \mathcal{I}, \mathsf{CR}) = \mathsf{Dir}(\boldsymbol{a}, \mathcal{I}) = \mathsf{Dir}(\boldsymbol{a}, T) = \mathsf{Eff}_0(\boldsymbol{a}, T, \mathsf{CR}).$$

For $i > 0$, we have again by Definitions 6.12 and 6.23, Part 4 of Claim 6.25, and the induction hypothesis:

$$\begin{aligned}\mathsf{Eff}_i(\boldsymbol{a},\mathcal{I},\mathsf{CR}) &= \mathsf{Eff}_{i-1}(\boldsymbol{a},\mathcal{I},\mathsf{CR}) \cup \mathsf{Indir}(\mathsf{CR},\mathcal{I},\mathsf{Eff}_{i-1}(\boldsymbol{a},\mathcal{I},\mathsf{CR}))\\ &= \mathsf{Eff}_{i-1}(\boldsymbol{a},T,\mathsf{CR}) \cup \mathsf{Indir}(\mathsf{CR},\mathcal{I},\mathsf{Eff}_{i-1}(\boldsymbol{a},T,\mathsf{CR}))\\ &= \mathsf{Eff}_{i-1}(\boldsymbol{a},T,\mathsf{CR}) \cup \mathsf{Indir}(\mathsf{CR},T,\mathsf{Eff}_{i-1}(\boldsymbol{a},T,\mathsf{CR}))\\ &= \mathsf{Eff}_i(\boldsymbol{a},T,\mathsf{CR}).\end{aligned}$$ □

Using this lemma, we can show that checking which of the sets of effects $\mathsf{Eff}(\boldsymbol{a},T,\mathsf{CR})$, with $T \in \mathfrak{T}(\boldsymbol{a},\mathsf{CR})$, are contradictory is sufficient for deciding whether the DL-action $\boldsymbol{a}$ is consistent w.r.t. the empty TBox and a finite set of causal relationships CR. In fact, there are only two reasons for an interpretation $\mathcal{I}$ that there does not exist an interpretation $\mathcal{I}'$ such that $\mathcal{I} \Rightarrow_{\boldsymbol{a}}^{\emptyset,\mathsf{CR}} \mathcal{I}'$ if the TBox is assumed to be empty: either $\boldsymbol{a}$ is not applicable to $\mathcal{I}$, or the set $\mathsf{Eff}(\boldsymbol{a},\mathcal{I},\mathsf{CR})$ is contradictory. Since for $\boldsymbol{a}$ being consistent, we require the existence of $\mathcal{I}'$ only for interpretations $\mathcal{I}$ such that $\boldsymbol{a}$ is applicable to $\mathcal{I}$, it is enough to consider the action types T that are consistent with the pre-condition of $\boldsymbol{a}$.

Lemma 6.26. *The DL-action $\boldsymbol{a} = (\mathsf{pre},\mathsf{post})$ is consistent w.r.t. the empty TBox and a finite set of causal relationships* CR *iff the set of effects $\mathsf{Eff}(\boldsymbol{a},T,\mathsf{CR})$ is not contradictory for each $T \in \mathfrak{T}(\boldsymbol{a},\mathsf{CR})$ for which $T \cup \mathsf{pre}$ is consistent.*

Proof. For the 'only if' direction, assume to the contrary that there exists an action type $T \in \mathfrak{T}(\boldsymbol{a},\mathsf{CR})$ such that $T \cup \mathsf{pre}$ is consistent, but $\mathsf{Eff}(\boldsymbol{a},T,\mathsf{CR})$ is contradictory. Then, there is an interpretation $\mathcal{I}$ such that $\mathcal{I} \models T \cup \mathsf{pre}$, and thus also that $\mathcal{I} \models T$ and $\mathcal{I} \models \mathsf{pre}$. This yields that $\boldsymbol{a}$ is applicable to $\mathcal{I}$ w.r.t. the empty TBox. However, since $\mathcal{I} \models T$, we have by Lemma 6.24 that $\mathsf{Eff}(\boldsymbol{a},\mathcal{I},\mathsf{CR}) = \mathsf{Eff}(\boldsymbol{a},T,\mathsf{CR})$. Hence, the set of effects $\mathsf{Eff}(\boldsymbol{a},\mathcal{I},\mathsf{CR})$ is contradictory. But then, by Definition 6.13, we have that there is no interpretation $\mathcal{I}'$ such that $\mathcal{I} \Rightarrow_{\boldsymbol{a}}^{\emptyset,\mathsf{CR}} \mathcal{I}'$, which is a contradiction to $\boldsymbol{a}$ being consistent w.r.t. the empty TBox and CR.

For the 'if' direction, assume to the contrary that $\boldsymbol{a}$ is not consistent w.r.t. the empty TBox and CR. Then there exists an interpretation $\mathcal{I}$ with the following two properties: $\boldsymbol{a}$ is applicable to $\mathcal{I}$ w.r.t. the empty TBox, and there is no interpretation $\mathcal{I}'$ with $\mathcal{I} \Rightarrow_{\boldsymbol{a}}^{\emptyset,\mathsf{CR}} \mathcal{I}'$. By Definition 6.13, we have that $\mathsf{Eff}(\boldsymbol{a},\mathcal{I},\mathsf{CR})$ is contradictory. By Lemma 6.20, there is one unique action type $T \in \mathfrak{T}(\boldsymbol{a},\mathsf{CR})$ such that $\mathcal{I} \models T$. Lemma 6.24 yields that $\mathsf{Eff}(\boldsymbol{a},T,\mathsf{CR}) = \mathsf{Eff}(\boldsymbol{a},\mathcal{I},\mathsf{CR})$, and thus we have that $\mathsf{Eff}(\boldsymbol{a},T,\mathsf{CR})$ is contradictory. Moreover, since $\boldsymbol{a}$ is applicable to $\mathcal{I}$ w.r.t. the empty TBox, we have $\mathcal{I} \models \mathsf{pre}$, and thus, together with $\mathcal{I} \models T$, that $T \cup \mathsf{pre}$ is consistent, which yields a contradiction. □

We use this lemma to design a decision procedure for deciding whether a DL-action is consistent w.r.t. the empty TBox and a finite set of causal relationships. The complexity of this problem depends on the DL used.

Theorem 6.27. *The problem of deciding whether a DL-action is consistent w.r.t. the empty TBox and a finite set of causal relationships is*

1. PSPACE-*complete for DLs between $\mathcal{ALC}$ and $\mathcal{ALCQO}$;*
2. PSPACE-*complete for DLs between $\mathcal{ALC}$ and $\mathcal{ALCQI}$;*

3. ExpTime-*complete for* $\mathcal{ALCIO}$*; and*
4. co-NExpTime-*complete for* $\mathcal{ALCQIO}$.

Proof. We first prove the lower bounds. We reduce the ABox-*in*consistency problem, i.e. the problem of deciding whether a given ABox is *in*consistent, to our DL-action consistency problem. Take any ABox $\mathcal{A}$. It is easy to see that $\mathcal{A}$ is *in*consistent iff the DL-action $(\mathcal{A}, \{A(a), \neg A(a)\})$ is consistent w.r.t. the empty TBox and the empty set of causal relationships, where $A \in \mathsf{N_C}$ and $a \in \mathsf{N_I}$ are arbitrary.

We have PSpace-hardness for Parts 1 and 2 of the theorem, since the ABox-consistency problem is PSpace-complete for the description logics $\mathcal{ALC}$ [SS91], $\mathcal{ALCQO}$ [Sch94; BLM+05b], and $\mathcal{ALCQI}$ [Tob01], and PSpace is closed under complement.[6] We obtain ExpTime-hardness for Part 3 of the theorem, because the ABox-consistency problem for $\mathcal{ALCIO}$ is ExpTime-complete [ABM99], and the class ExpTime is closed under complement. Finally, co-NExpTime-hardness for Part 4 of the theorem is obtained, because the ABox-consistency problem for $\mathcal{ALCQIO}$ is NExpTime-complete [Sch94; Tob00; Pra05].[7]

To prove the upper bounds for Parts 1 and 2 of the theorem, we give an NPSpace-decision procedure for deciding whether a DL-action is *in*consistent w.r.t. the empty TBox and a finite set of causal relationships.[8] Given a DL-action $\boldsymbol{a} = (\mathsf{pre}, \mathsf{post})$ and a finite set of causal relationships CR, the decision procedure consists of three steps.

1. Non-deterministically guess an action type $T \in \mathfrak{T}(\boldsymbol{a}, \mathsf{CR})$.
2. Check whether the generalised ABox $T \cup \mathsf{pre}$ is consistent.
3. If Step 2 was successful, compute the set $\mathsf{Eff}(\boldsymbol{a}, T, \mathsf{CR})$, and check whether it is contradictory.

If in Step 3, we obtain a contradictory set of effects, we know by Lemma 6.26 that $\boldsymbol{a}$ is *not* consistent w.r.t. the empty TBox and CR. Otherwise, $\boldsymbol{a}$ is consistent w.r.t. the empty TBox and CR.

Step 1 can be done in PSpace, because the set $\mathfrak{T}(\boldsymbol{a}, \mathsf{CR})$ is of size exponential in the size of $\boldsymbol{a}$ and CR, but each action type $T \in \mathfrak{T}(\boldsymbol{a}, \mathsf{CR})$ is only of polynomial size.

The consistency test in Step 2 can polynomially be reduced to the consistency problem of classical ABoxes [BLM+05b]. Indeed, a generalised ABox $\mathcal{A}$ can be transformed into a classical ABox $\mathcal{A}'$ such that $\mathcal{A}$ is consistent iff $\mathcal{A}'$ is consistent as follows. Obviously, every negative concept assertion $\neg\alpha(a)$ in $\mathcal{A}$ can be replaced by $(\neg\alpha)(a)$ without affecting the consistency of $\mathcal{A}$. Every negative role assertion $\neg r(a, b)$ in $\mathcal{A}$ can be replaced by the two concept assertions $(\neg\exists r.A_b)(a)$ and $A_b(b)$, where A_b is a concept name not occurring in $\mathcal{A}$.

[6] Note that in [BLM+05b], PSpace-completeness of the ABox-consistency problem for $\mathcal{ALCQO}$ is proved only for the case of unary coding of the numbers in the at-least and at-most restrictions. It is conjectured in [BLM+05b], however, that with similar arguments, one obtains PSpace-completeness also for the case of binary coding. For $\mathcal{ALCQI}$, it is proved explicitly that the ABox-consistency problem is PSpace-complete even if the numbers are coded in binary [Tob01].

[7] This is even the case if the number in the at-least and at-most restrictions are coded in binary, because one can reduce the ABox-consistency problem for $\mathcal{ALCQIO}$ to the satisfiability problem for C^2 with counting quantifiers [BLM+05b], which is NExpTime-complete even if the numbers are coded in binary [Pra05].

[8] Recall that Savitch's theorem [Sav70] implies that NPSpace and PSpace coincide, and that PSpace is closed under complement.

It is not hard to verify that $\mathcal{A}$ is consistent iff $\mathcal{A}'$ is consistent. As noted above, the ABox-consistency problem is PSPACE-complete for $\mathcal{ALCQO}$ and $\mathcal{ALCQI}$. Thus, the check whether $T \cup \mathsf{pre}$ is consistent can be done in PSPACE for the DLs $\mathcal{ALCQO}$ and $\mathcal{ALCQI}$.

Step 3 can also be done in PSPACE, because computing the set $\mathsf{Eff}(\boldsymbol{a}, T, \mathsf{CR})$ can be realised by performing the iteration used in the definition of $\mathsf{Eff}(\boldsymbol{a}, T, \mathsf{CR})$. As argued above, this can be done in time polynomial in the size of $\boldsymbol{a}$, T, and CR. Checking whether this set of effects is contradictory is obviously also possible in polynomial time.

To prove Part 4 of the theorem, we proceed similarly. We employ the same decision procedure as above for checking whether the DL-action $\boldsymbol{a}$ is *in*consistent w.r.t. the empty TBox and the finite set of causal relationships CR. Since now the underlying DL is $\mathcal{ALCQIO}$ with the arguments above, this can be done in NEXPTIME. Hence, we obtain that the complement of this problem is in CO-NEXPTIME.

Finally, for Part 3 of the theorem, in order to check whether the DL-action $\boldsymbol{a} = (\mathsf{pre}, \mathsf{post})$ is consistent w.r.t. the empty TBox and the finite set of causal relationships CR, we compute the set $\mathfrak{T}(\boldsymbol{a}, \mathsf{CR})$ explicitly. Now, we check for each $T \in \mathfrak{T}(\boldsymbol{a}, \mathsf{CR})$, whether $T \cup \mathsf{pre}$ is consistent. Then, we compute the set $\mathsf{Eff}(\boldsymbol{a}, T, \mathsf{CR})$, and check whether it is contradictory. If no contradictory set of effects is found, we know that $\boldsymbol{a}$ is consistent w.r.t. the empty TBox and CR. Otherwise, $\boldsymbol{a}$ is inconsistent w.r.t. the empty TBox and CR. Using the arguments from above, this yields an EXPTIME-decision procedure. □

In the next section, we consider the consistency problem for the case where the TBox is general, i.e. a finite set of GCIs.

6.2.2 Deciding the Consistency Problem w.r.t. a General TBox

If the TBox is not assumed to be empty, the picture changes. We can no longer obtain an easy characterisation of consistent DL-actions as for the case where the TBox is assumed to be empty. In this case, the criterion for a DL-action to be consistent w.r.t. the empty TBox and a finite set of causal relationships stated in Lemma 6.26 is a necessary but not a sufficient condition. In fact, it could happen that a not contradictory set of effects induces a successor interpretation that is not a model of the TBox. This is an additional possible reason for a DL-action $\boldsymbol{a}$ to be inconsistent w.r.t. a TBox $\mathcal{T}$ and a finite set of causal relationships CR. Thus, one needs to check additionally for each action type $T \in \mathfrak{T}(\boldsymbol{a}, \mathsf{CR})$, whether for any model $\mathcal{I}$ of $\mathcal{T}$ and T that satisfies the preconditions of $\boldsymbol{a}$, the interpretation $\mathcal{I}'$ obtained from $\mathcal{I}$ by applying the effects in $\mathsf{Eff}(\boldsymbol{a}, T, \mathsf{CR})$ (see Definition 6.13) is a model of $\mathcal{T}$. For this purpose, we define an unconditional DL-action $\boldsymbol{b}_{\boldsymbol{a},T,\mathsf{CR}}$ that has the same effects as $\boldsymbol{a}$ and CR if applied to a model of T. Then, we adapt the approach to solving the projection problem introduced in [BLM+05a] in order to decide whether $\boldsymbol{b}_{\boldsymbol{a},T,\mathsf{CR}}$ transforms models of $\mathcal{T}$ into models of $\mathcal{T}$.

Definition 6.28. *Let $\boldsymbol{a} = (\mathsf{pre}, \mathsf{post})$ be a DL-action, CR a finite set of causal relationships, and $T \in \mathfrak{T}(\boldsymbol{a}, \mathsf{CR})$. The unconditional DL-action $\boldsymbol{b}_{\boldsymbol{a},T,\mathsf{CR}}$ is defined as follows:*

$$\boldsymbol{b}_{\boldsymbol{a},T,\mathsf{CR}} := (\mathsf{pre} \cup T,\ \mathsf{Eff}(\boldsymbol{a}, T, \mathsf{CR})). \qquad \diamondsuit$$

The following lemma is a direct consequence of the definition of the set of effects $\mathsf{Eff}(\boldsymbol{a}, T, \mathsf{CR})$ (see Definition 6.23), the semantics of DL-actions (see Definition 6.13), and Lemma 6.26.

Lemma 6.29. *For every $T \in \mathfrak{T}(\boldsymbol{a}, \mathsf{CR})$, every model $\mathcal{I}$ of T, and every interpretation $\mathcal{I}'$, we have $\mathcal{I} \Rightarrow_{\boldsymbol{a}}^{\emptyset,\mathsf{CR}} \mathcal{I}'$ iff $\mathcal{I} \Rightarrow_{\boldsymbol{b}_{\boldsymbol{a},T,\mathsf{CR}}}^{\emptyset,\emptyset} \mathcal{I}'$.*

Proof. Take any action type $T \in \mathfrak{T}(\boldsymbol{a}, \mathsf{CR})$, and any interpretation $\mathcal{I}$ with $\mathcal{I} \models T$. We have by the construction of $\boldsymbol{b}_{\boldsymbol{a},T,\mathsf{CR}}$ that $\mathsf{Dir}(\boldsymbol{b}_{\boldsymbol{a},T,\mathsf{CR}}, T) = \mathsf{Eff}(\boldsymbol{a}, T, \mathsf{CR})$. Thus, we have also that $\mathsf{Eff}(\boldsymbol{b}_{\boldsymbol{a},T,\mathsf{CR}}, T, \emptyset) = \mathsf{Eff}(\boldsymbol{a}, T, \mathsf{CR})$. By Lemma 6.26, this yields $\mathsf{Eff}(\boldsymbol{b}_{\boldsymbol{a},T,\mathsf{CR}}, \mathcal{I}, \emptyset) = \mathsf{Eff}(\boldsymbol{a}, \mathcal{I}, \mathsf{CR})$. Finally, together with Definition 6.13, we obtain $\mathcal{I} \Rightarrow_{\boldsymbol{a}}^{\emptyset,\mathsf{CR}} \mathcal{I}'$ iff $\mathcal{I} \Rightarrow_{\boldsymbol{b}_{\boldsymbol{a},T,\mathsf{CR}}}^{\emptyset,\emptyset} \mathcal{I}'$. □

The approach to solving the projection problem introduced in [BLM+05a] considers a finite sequence of DL-actions $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$. In this section, however, we are only interested in the special case where $n = 1$. Since we will adopt the same approach also in Section 6.3, where we consider the case $n \geq 1$ to solve the projection problem, we still recall here the relevant notions and results for the general case.

The procedure to solving the projection problem introduced in [BLM+05a] works basically as follows. Firstly, time-stamped copies $A^{(i)}$, $0 \leq i \leq n$, of all relevant concept names in the input, $r^{(i)}$, $0 \leq i \leq n$, of all relevant role names in the input, and new time-stamped concept names $T_C^{(i)}$, $0 \leq i \leq n$, for every relevant concept C in the input are introduced. Whereas in [BLM+05a], not all concept names occurring in the input are relevant, in our setting, the *relevant* role names, concept names, and concepts are precisely the ones occurring in the input of the consistency (or projection) problem.[9] For every generalised ABox-literal α built using a relevant concept C or a relevant role name r (called *relevant generalised ABox-literal* in the following) and every i, $0 \leq i \leq n$, we can then define a time-stamped variant $\alpha^{(i)}$ as follows:

$$
\begin{aligned}
(C(a))^{(i)} &:= T_C^{(i)}(a), & (r(a,b))^{(i)} &:= r^{(i)}(a,b),\\
(\neg C(a))^{(i)} &:= \neg T_C^{(i)}(a), & (\neg r(a,b))^{(i)} &:= \neg r^{(i)}(a,b).
\end{aligned}
$$

Given a generalised ABox $\mathcal{A}$, where each $\alpha \in \mathcal{A}$ is a relevant generalised ABox-literal, we define its time-stamped copy $\mathcal{A}^{(i)}$ as

$$\mathcal{A}^{(i)} := \{\alpha^{(i)} \mid \alpha \in \mathcal{A}\}.$$

Similarly, given a finite set of GCIs $\mathcal{T}$ built from relevant concepts, we define its time-stamped copy $\mathcal{T}^{(i)}$ as

$$\mathcal{T}^{(i)} := \{T_C^{(i)} \sqsubseteq T_D^{(i)} \mid C \sqsubseteq D \in \mathcal{T}\}.$$

Intuitively, given an initial interpretation $\mathcal{I}_0$, the application of $\boldsymbol{b}_1$ to $\mathcal{I}_0$ yields a successor interpretation $\mathcal{I}_1$, the application of $\boldsymbol{b}_2$ to $\mathcal{I}_1$ yields a successor interpretation $\mathcal{I}_2$, and so forth. We can encode the sequence of interpretations $\mathcal{I}_0, \mathcal{I}_1, \ldots, \mathcal{I}_n$ into a single interpretation $\mathcal{J}$ using the time-stamped copies introduced above such that the relevant generalised ABox-literal α holds in $\mathcal{I}_i$ iff its time-stamped copy $\alpha^{(i)}$ holds in $\mathcal{J}$.

[9] Recall that in [BLM+05a] only *acyclic* TBoxes are considered. Additionally, the action formalism there is limited such that for each DL-action $\boldsymbol{a}$, we have that *defined* concept names must not occur in any unconditional post-condition of $\boldsymbol{a}$ or in β for any conditional post-condition α/β of $\boldsymbol{a}$. Intuitively, this is the reason why in [BLM+05a], defined concept names are not relevant, i.e. there is no need to introduce time-stamped copies $A^{(i)}$ for any defined concept name A.

To enforce that $\mathcal{J}$ indeed encodes a sequence of interpretations induced by the application of the sequence of DL-actions $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$, we require it to be a model of the (acyclic) TBox $\mathcal{T}_{\mathsf{red}}$ and the generalised ABox $\mathcal{A}_{\mathsf{red}}$. The construction of $\mathcal{T}_{\mathsf{red}}$ and $\mathcal{A}_{\mathsf{red}}$ is very similar to the one introduced in [BLM+05b] with the difference that we use here a different notion of 'relevant' as explained above. Also, compared to the original construction of $\mathcal{A}_{\mathsf{red}}$ and $\mathcal{T}_{\mathsf{red}}$, the present construction is simpler since we deal only with unconditional DL-actions. Additionally, since we do not consider acyclic TBoxes as domain constraints, we can simplify the construction.

In the following, let $\mathcal{R}$ denote a set of relevant concept names, role names, and concepts, and let Obj denote the set of individual names occurring in the input of the consistency problem. We describe the construction for the case of $\mathcal{ALCQIO}$. The TBox $\mathcal{T}_{\mathsf{red}}$ consists of two parts: a TBox $\mathcal{T}_N$ and a TBox $\mathcal{T}_{\mathsf{sub}}$, i.e. $\mathcal{T}_{\mathsf{red}} := \mathcal{T}_N \cup \mathcal{T}_{\mathsf{sub}}$. As in [BLM+05b], the TBox $\mathcal{T}_N$ introduces a concept name N to capture all named individuals:

$$\mathcal{T}_N := \big\{N \equiv \bigsqcup_{a \in \mathsf{Obj}} \{a\}\big\}.$$

Note that we make use of nominals here. The TBox $\mathcal{T}_{\mathsf{sub}}$ consists of a concept definition of $T_C^{(i)}$ for every concept $C \in \mathcal{R}$ and every i, $0 \leq i \leq n$. The concept definition of $T_C^{(i)}$ is defined inductively on the structure of C as follows:

- $T_A^{(i)} \equiv (N \sqcap A^{(i)}) \sqcup (\neg N \sqcap A^{(0)}) \quad \text{if } A \in \mathsf{N_C}$;
- $T_{\{a\}}^{(i)} \equiv \{a\}$;
- $T_{\neg C_1}^{(i)} \equiv \neg T_{C_1}^{(i)}$;
- $T_{C_1 \sqcap C_2}^{(i)} \equiv T_{C_1}^{(i)} \sqcap T_{C_2}^{(i)}$;
- $T_{\exists r.C_1}^{(i)} \equiv \Big(N \sqcap \big((\exists r^{(0)}.(\neg N \sqcap T_{C_1}^{(i)})) \sqcup (\exists r^{(i)}.(N \sqcap T_{C_1}^{(i)}))\big)\Big) \sqcup (\neg N \sqcap \exists r^{(0)}.T_{C_1}^{(i)})$; and
- $T_{\geq m\, r.C_1}^{(i)} \equiv \Big(N \sqcap \bigsqcup_{0 \leq j \leq m} \big(\geq j\, r^{(i)}.(N \sqcap T_{C_1}^{(i)}) \sqcap \geq (m-j)\, r^{(0)}.(\neg N \sqcap T_{C_1}^{(i)})\big)\Big) \sqcup$
  $(\neg N \sqcap \geq m\, r^{(0)}.T_{C_1}^{(i)})$.

The generalised ABox $\mathcal{A}_{\mathsf{red}}$ also consists of several parts. Let pre_i be the set of pre-conditions of $\boldsymbol{b}_i$ for each i, $1 \leq i \leq n$. The following generalised ABoxes capture the pre-conditions:

$$\mathcal{A}_{\mathsf{pre}}^{(i)} := \big\{\alpha^{(i-1)} \mid \alpha \in \mathsf{pre}_i\big\}.$$

Note that we take $\alpha^{(i-1)}$ since the pre-conditions have to be satisfied *before* the DL-action is applied.

Moreover, let post_i be the set of post-conditions of $\boldsymbol{b}_i$ for each i, $1 \leq i \leq n$. Since all $\boldsymbol{b}_i$ are unconditional, we can define generalised ABoxes capturing the post-conditions as follows:

$$\mathcal{A}_{\mathsf{post}}^{(i)} := \big\{\alpha^{(i)} \mid \alpha \in \mathsf{post}_i\big\}.$$

The ABoxes $\mathcal{A}_{\mathsf{min}}^{(i)}$ ensure a minimisation of changes to the named individuals. For every i, $1 \leq i \leq n$, the ABox $\mathcal{A}_{\mathsf{min}}^{(i)}$ consists of

1. the following assertions for every $a \in \mathsf{Obj}$ and every $A \in \mathsf{N_C}$ occurring in the input:

$$\begin{cases}(A^{(i-1)} \rightarrow A^{(i)})(a) & \text{if } \neg A(a) \notin \mathsf{post}_i\text{, and} \\ (\neg A^{(i-1)} \rightarrow \neg A^{(i)})(a) & \text{if } A(a) \notin \mathsf{post}_i\text{; and}\end{cases}$$

2. the following assertions for every $a, b \in \mathsf{Obj}$ and every $r \in \mathsf{N_R}$ occurring in the input:

$$\begin{cases}(\exists r^{(i-1)}.\{b\} \rightarrow \exists r^{(i)}.\{b\})(a) & \text{if } \neg r(a,b) \notin \mathsf{post}_i\text{, and} \\ (\neg\exists r^{(i-1)}.\{b\} \rightarrow \neg\exists r^{(i)}.\{b\})(a) & \text{if } r(a,b) \notin \mathsf{post}_i.\end{cases}$$

Finally, we can construct the generalised ABox $\mathcal{A}_{\mathsf{red}}$ using the above defined ABoxes and the pre-conditions of the DL-actions:

$$\mathcal{A}_{\mathsf{red}} := \bigcup_{i=1}^{n} \mathcal{A}^{(i)}_{\mathsf{pre}} \cup \bigcup_{i=1}^{n} \mathcal{A}^{(i)}_{\mathsf{post}} \cup \bigcup_{i=1}^{n} \mathcal{A}^{(i)}_{\mathsf{min}}.$$

We now recall the pertinent properties of $\mathcal{T}_{\mathsf{red}}$ and $\mathcal{A}_{\mathsf{red}}$ in the next lemma, whose proof is very similar to the one of Theorem 14 and Lemma 15 in [BLM+05b]. We still present the full proof for the sake of completeness.

Lemma 6.30. *Let $\mathcal{L}$ be a DL between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$ and $\mathcal{LO}$ the DL which extends $\mathcal{L}$ with nominals. Let $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ be a sequence of DL-actions formulated in $\mathcal{L}$, and $\mathcal{R}$ be a set of relevant concept names, role names, and concepts such that $\mathcal{R}$ contains all the concept names, role names, and concepts occurring in $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$.*

Then, there are a generalised $\mathcal{LO}$-ABox $\mathcal{A}_{\mathsf{red}}$ and an (acyclic) $\mathcal{LO}$-TBox $\mathcal{T}_{\mathsf{red}}$ of size polynomial in the size of $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$, and $\mathcal{R}$, such that the following properties hold:

1. *For every sequence of interpretations $\mathcal{I}_0, \ldots, \mathcal{I}_n$ with $\mathcal{I}_i \Rightarrow^{\emptyset,\emptyset}_{\boldsymbol{b}_{i+1}} \mathcal{I}_{i+1}$ for each i, $0 \le i < n$, there exists an interpretation $\mathcal{J}$ such that $\mathcal{J} \models \mathcal{A}_{\mathsf{red}}$, $\mathcal{J} \models \mathcal{T}_{\mathsf{red}}$, and*
   a) *for every i, $0 \le i \le n$, and every relevant generalised ABox-literal α, we have $\mathcal{I}_i \models \alpha$ iff $\mathcal{J} \models \alpha^{(i)}$; and*
   b) *for every i, $0 \le i \le n$, and every relevant concept C, we have $C^{\mathcal{I}_i} = (T^{(i)}_C)^{\mathcal{J}}$.*
2. *For every interpretation $\mathcal{J}$ with $\mathcal{J} \models \mathcal{A}_{\mathsf{red}}$ and $\mathcal{J} \models \mathcal{T}_{\mathsf{red}}$, there exist interpretations $\mathcal{I}_0, \ldots, \mathcal{I}_n$ such that $\mathcal{I}_i \Rightarrow^{\emptyset,\emptyset}_{\boldsymbol{b}_{i+1}} \mathcal{I}_{i+1}$ for every i, $0 \le i < n$, and*
   a) *for every i, $0 \le i \le n$, and every relevant generalised ABox-literal α, we have $\mathcal{I}_i \models \alpha$ iff $\mathcal{J} \models \alpha^{(i)}$; and*
   b) *for every i, $0 \le i \le n$, and every relevant concept C, we have $C^{\mathcal{I}_i} = (T^{(i)}_C)^{\mathcal{J}}$.*

Proof. Let $\mathcal{A}_{\mathsf{red}}$ and $\mathcal{T}_{\mathsf{red}}$ be defined as above. It is easy to see that $\mathcal{A}_{\mathsf{red}}$ and $\mathcal{T}_{\mathsf{red}}$ are of size polynomial in the size of $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$, and $\mathcal{R}$. We first prove Property (1). For that, let $\mathcal{I}_0 = (\Delta^{\mathcal{I}_0}, \cdot^{\mathcal{I}_0}), \ldots, \mathcal{I}_n = (\Delta^{\mathcal{I}_n}, \cdot^{\mathcal{I}_n})$ be a sequence of interpretations with $\mathcal{I}_i \Rightarrow^{\emptyset,\emptyset}_{\boldsymbol{b}_i} \mathcal{I}_{i+1}$ for every i, $0 \le i < n$. Then, Definition 6.13 yields that $\Delta^{\mathcal{I}_0} = \Delta^{\mathcal{I}_1} = \cdots = \Delta^{\mathcal{I}_n}$, and $a^{\mathcal{I}_0} = a^{\mathcal{I}_1} = \cdots = a^{\mathcal{I}_n}$ for every $a \in \mathsf{N_I}$. We define the interpretation $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ as follows:

- $\Delta^{\mathcal{J}} := \Delta^{\mathcal{I}_0}$;
- $a^{\mathcal{J}} := a^{\mathcal{I}_0}$ for every $a \in \mathsf{N_I}$;

- $N^{\mathcal{J}} := \{a^{\mathcal{J}} \mid a \in \mathsf{Obj}\}$;
- $(A^{(i)})^{\mathcal{J}} := A^{\mathcal{I}_i}$ for every $A \in \mathcal{R} \cap \mathsf{N_C}$ and every i, $0 \leq i \leq n$;
- $(r^{(i)})^{\mathcal{J}} := r^{\mathcal{I}_i}$ for every $r \in \mathcal{R} \cap \mathsf{N_R}$ and every i, $0 \leq i \leq n$; and
- $(T_C^{(i)})^{\mathcal{J}} := C^{\mathcal{I}_i}$ for every concept $C \in \mathcal{R}$ and every i, $0 \leq i \leq n$.

We first prove Property (1a). For that, take any i with $0 \leq i \leq n$, and any relevant generalised ABox-literal α. We prove the property by case distinction. Assume that α is of the form $C(a)$ where C is a concept and $a \in \mathsf{N_I}$. We have: $\mathcal{I}_i \models \alpha$ iff $a^{\mathcal{I}_i} \in C^{\mathcal{I}_i}$ iff $a^{\mathcal{J}} \in (T_C^{(i)})^{\mathcal{J}}$ iff $\mathcal{J} \models T_C^{(i)}(a)$ iff $\mathcal{J} \models \alpha^{(i)}$.

Assume now that α is of the form $\neg C(a)$ where C is a concept and $a \in \mathsf{N_I}$. By similar arguments, we have: $\mathcal{I}_i \models \alpha$ iff $a \notin C^{\mathcal{I}_i}$ iff $a^{\mathcal{J}} \notin (T_C^{(i)})^{\mathcal{J}}$ iff $\mathcal{J} \not\models T_C^{(i)}(a)$ iff $\mathcal{J} \models \neg T_C^{(i)}(a)$ iff $\mathcal{J} \models \alpha^{(i)}$.

For the case that α is of the form $r(a,b)$ where $r \in \mathsf{N_R}$ and $a, b \in \mathsf{N_I}$, we have: $\mathcal{I}_i \models \alpha$ iff $(a^{\mathcal{I}_i}, b^{\mathcal{I}_i}) \in r^{\mathcal{I}_i}$ iff $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in (r^{(i)})^{\mathcal{J}}$ iff $\mathcal{J} \models r^{(i)}(a,b)$ iff $\mathcal{J} \models \alpha^{(i)}$.

Finally, assume that α is of the form $\neg r(a,b)$ where $r \in \mathsf{N_R}$ and $a, b \in \mathsf{N_I}$. We have: $\mathcal{I}_i \models \alpha$ iff $(a^{\mathcal{I}_i}, b^{\mathcal{I}_i}) \notin r^{\mathcal{I}_i}$ iff $(a^{\mathcal{J}}, b^{\mathcal{J}}) \notin (r^{(i)})^{\mathcal{J}}$ iff $\mathcal{J} \not\models r^{(i)}(a,b)$ iff $\mathcal{J} \models \neg r^{(i)}(a,b)$ iff $\mathcal{J} \models \alpha^{(i)}$. This finishes the proof of Property (1a).

Property (1b) follows directly from the definition of $\mathcal{J}$. Thus, it is only left to be proved that $\mathcal{J}$ is a model of $\mathcal{A}_{\mathsf{red}}$ and $\mathcal{T}_{\mathsf{red}}$.

We start proving that $\mathcal{J}$ is a model of $\mathcal{A}_{\mathsf{red}}$. Definition 6.13 yields that $\boldsymbol{b}_i$ is applicable to $\mathcal{I}_{i-1}$ w.r.t. the empty TBox, i.e. $\mathcal{I}_{i-1} \models \mathsf{pre}_i$, for every i, $1 \leq i \leq n$. Since pre_i consists of relevant generalised ABox-literals, we have by Property (1a) that $\mathcal{J} \models \mathsf{pre}_i^{(i-1)}$, i.e. $\mathcal{J} \models \mathcal{A}_{\mathsf{pre}}^{(i)}$, for every i, $1 \leq i \leq n$.

Since the set of causal relationships is empty and all DL-actions are unconditional, we have that $\mathsf{Eff}(\boldsymbol{b}_i, \mathcal{I}_{i-1}, \emptyset) = \mathsf{post}_i$ for every i, $1 \leq i \leq n$. We have by Definition 6.13 that post_i is not contradictory for every i, $1 \leq i \leq n$. It is easy to see from Definition 6.13 that $\mathcal{I}_i \models \mathsf{post}_i$ for every i, $1 \leq i \leq n$. Again, since post_i consists of relevant generalised ABox-literals, we have by Property (1a) that $\mathcal{J} \models \mathsf{post}_i^{(i)}$, i.e. $\mathcal{J} \models \mathcal{A}_{\mathsf{post}}^{(i)}$, for every i, $1 \leq i \leq n$.

To show that $\mathcal{J}$ is a model of $\mathcal{A}_{\mathsf{min}}^{(i)}$ for every i, $1 \leq i \leq n$, take any i, $1 \leq i \leq n$, any $a, b \in \mathsf{Obj}$, any relevant $A \in \mathsf{N_C}$, and any relevant $r \in \mathsf{N_R}$. Assume first that $\neg A(a) \notin \mathsf{post}_i$. By the arguments above, we have that $\neg A(a) \notin \mathsf{Eff}(\boldsymbol{b}_i, \mathcal{I}_{i-1}, \emptyset)$. By Definition 6.13, we have that $a^{\mathcal{I}_{i-1}} \notin A^-$, and thus that $a^{\mathcal{I}_{i-1}} \in A^{\mathcal{I}_{i-1}}$ implies that $a^{\mathcal{I}_{i-1}} = a^{\mathcal{I}_i} \in A^{\mathcal{I}_i}$. Hence, $a^{\mathcal{J}} \in (A^{(i-1)})^{\mathcal{J}}$ implies that $a^{\mathcal{J}} \in (A^{(i)})^{\mathcal{J}}$. This is equivalent to $a^{\mathcal{J}} \in (A^{(i-1)} \rightarrow A^{(i)})^{\mathcal{J}}$. Thus, $\mathcal{J} \models (A^{(i-1)} \rightarrow A^{(i)})(a)$.

Assume now that $A(a) \notin \mathsf{post}_i$. By similar arguments, we obtain that $a^{\mathcal{I}_{i-1}} \notin A^+$, and thus that $a^{\mathcal{I}_{i-1}} \notin A^{\mathcal{I}_{i-1}}$ implies that $a^{\mathcal{I}_{i-1}} \notin A^{\mathcal{I}_i}$. Hence, we have that $a^{\mathcal{J}} \notin (A^{(i-1)})^{\mathcal{J}}$ implies that $a^{\mathcal{J}} \notin (A^{(i)})^{\mathcal{J}}$. This is equivalent to $a^{\mathcal{J}} \in (\neg A^{(i-1)} \rightarrow \neg A^{(i)})^{\mathcal{J}}$. Thus, we have $\mathcal{J} \models (\neg A^{(i-1)} \rightarrow \neg A^{(i)})(a)$.

For the case where $\neg r(a,b) \notin \mathsf{post}_i$, we have again by similar arguments as in the previous cases that $(a^{\mathcal{I}_{i-1}}, b^{\mathcal{I}_{i-1}}) \notin r^-$, and thus that $(a^{\mathcal{I}_{i-1}}, b^{\mathcal{I}_{i-1}}) \in r^{\mathcal{I}_{i-1}}$ implies that $(a^{\mathcal{I}_{i-1}}, b^{\mathcal{I}_{i-1}}) \in r^{\mathcal{I}_i}$. Hence, $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in (r^{(i-1)})^{\mathcal{J}}$ implies that $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in (r^{(i)})^{\mathcal{J}}$. It is easy to see that this is equivalent to $a^{\mathcal{J}} \in (\exists r^{(i-1)}.\{b\} \rightarrow \exists r^{(i)}.\{b\})^{\mathcal{J}}$, which yields that we have also that $\mathcal{J} \models (\exists r^{(i-1)}.\{b\} \rightarrow \exists r^{(i)}.\{b\})(a)$.

Finally, assume that $r(a,b) \notin \mathsf{post}_i$. By similar arguments, we have $(a^{\mathcal{I}_{i-1}}, b^{\mathcal{I}_{i-1}}) \notin r^+$, and thus we have that $(a^{\mathcal{I}_{i-1}}, b^{\mathcal{I}_{i-1}}) \notin r^{\mathcal{I}_{i-1}}$ implies $(a^{\mathcal{I}_{i-1}}, b^{\mathcal{I}_{i-1}}) \notin r^{\mathcal{I}_i}$. This yields that $(a^{\mathcal{J}}, b^{\mathcal{J}}) \notin (r^{(i-1)})^{\mathcal{J}}$ implies that $(a^{\mathcal{J}}, b^{\mathcal{J}}) \notin (r^{(i)})^{\mathcal{J}}$. Again, it is easy to see that this is equivalent to $a^{\mathcal{J}} \in (\neg\exists r^{(i-1)}.\{b\} \rightarrow \neg\exists r^{(i)}.\{b\})^{\mathcal{J}}$. Thus, we have also that $\mathcal{J}$ is a model of $(\neg\exists r^{(i-1)}.\{b\} \rightarrow \neg\exists r^{(i)}.\{b\})(a)$.

This finishes the proof that $\mathcal{J} \models \mathcal{A}^{(i)}_{\mathsf{min}}$ for every i, $1 \leq i \leq n$. Thus, we have shown that $\mathcal{J} \models \mathcal{A}_{\mathsf{red}}$. We show now that $\mathcal{J}$ is also a model of $\mathcal{T}_{\mathsf{red}}$. The definition of $N^{\mathcal{J}}$ yields that $\mathcal{J} \models \mathcal{T}_N$. Before we show that $\mathcal{J} \models \mathcal{T}_{\mathsf{sub}}$, we show the following claim.

Claim 6.31. *For every $A \in \mathsf{N_C}$, and every i, $0 \leq i \leq n$, we have that $A^{\mathcal{I}_0} \setminus N^{\mathcal{J}} = A^{\mathcal{I}_i} \setminus N^{\mathcal{J}}$.*

We show this claim by induction on i. For that, take any $A \in \mathsf{N_C}$. For $i = 0$, the claim is trivially satisfied. Assume now that the claim holds for i, i.e. $A^{\mathcal{I}_0} \setminus N^{\mathcal{J}} = A^{\mathcal{I}_i} \setminus N^{\mathcal{J}}$. Thus, it is enough to show $A^{\mathcal{I}_i} \setminus N^{\mathcal{J}} = A^{\mathcal{I}_{i+1}} \setminus N^{\mathcal{J}}$. Since $\mathcal{I}_i \Rightarrow^{\emptyset,\emptyset}_{\boldsymbol{b}_{i+1}} \mathcal{I}_{i+1}$, we have by Definition 6.13 that $A^{\mathcal{I}_{i+1}} = (A^{\mathcal{I}_i} \cup A^+) \setminus A^-$, where $A^+ = \{a^{\mathcal{I}} \mid A(a) \in \mathsf{Eff}(\boldsymbol{b}_{i+1}, \mathcal{I}_i, \emptyset)\}$ and $A^- = \{a^{\mathcal{I}} \mid \neg A(a) \in \mathsf{Eff}(\boldsymbol{b}_{i+1}, \mathcal{I}_i, \emptyset)\}$. Thus, $A^{\mathcal{I}_{i+1}} \setminus N^{\mathcal{J}} = \big((A^{\mathcal{I}_i} \cup A^+) \setminus A^-\big) \setminus N^{\mathcal{J}}$. Obviously, $A^+ \subseteq N^{\mathcal{J}}$ and $A^- \subseteq N^{\mathcal{J}}$, and thus $A^{\mathcal{I}_{i+1}} \setminus N^{\mathcal{J}} = A^{\mathcal{I}_i} \setminus N^{\mathcal{J}}$. This finishes the proof of Claim 6.31.

Very similar arguments can be used to show a similar claim for role names.

Claim 6.32. *For every $r \in \mathsf{N_R}$, and every i, $0 \leq i \leq n$, we have that*

$$r^{\mathcal{I}_0} \setminus (N^{\mathcal{J}} \times N^{\mathcal{J}}) = r^{\mathcal{I}_i} \setminus (N^{\mathcal{J}} \times N^{\mathcal{J}}).$$

To show that $\mathcal{J} \models \mathcal{T}_{\mathsf{sub}}$, we show that for every concept $C \in \mathcal{R}$ and every i, $0 \leq i \leq n$, the concept definition of $T^{(i)}_C$ is satisfied. We prove this by a case distinction.

For the case where $C = A \in \mathsf{N_C}$, we have that $(T^{(i)}_A)^{\mathcal{J}} = A^{\mathcal{I}_i}$ by definition. Obviously, $A^{\mathcal{I}_i} = (N^{\mathcal{J}} \cap A^{\mathcal{I}_i}) \cup (A^{\mathcal{I}_i} \setminus N^{\mathcal{J}})$. Claim 6.31 yields $A^{\mathcal{I}_i} = (N^{\mathcal{J}} \cap A^{\mathcal{I}_i}) \cup (A^{\mathcal{I}_0} \setminus N^{\mathcal{J}})$. Thus, we have $A^{\mathcal{I}_i} = (N^{\mathcal{J}} \cap A^{\mathcal{I}_i}) \cup \big((\Delta^{\mathcal{J}} \setminus N^{\mathcal{J}}) \cap A^{\mathcal{I}_0}\big)$. Hence, we obtain together with the definition of $\mathcal{J}$ that $(T^{(i)}_A)^{\mathcal{J}} = (N^{\mathcal{J}} \cap (A^{(i)})^{\mathcal{J}}) \cup \big((\Delta^{\mathcal{J}} \setminus N^{\mathcal{J}}) \cap (A^{(0)})^{\mathcal{J}}\big)$, which yields that $\mathcal{J}$ is a model of $T^{(i)}_A \equiv (N \sqcap A^{(i)}) \sqcup (\neg N \sqcap A^{(0)})$.

For the case where C is of the form $\{a\}$ with $a \in \mathsf{Obj}$, we have by definition that

$$(T^{(i)}_{\{a\}})^{\mathcal{J}} = \{a\}^{\mathcal{I}_i} = \{a^{\mathcal{I}_i}\} = \{a^{\mathcal{J}}\} = \{a\}^{\mathcal{J}},$$

and thus that $\mathcal{J}$ is a model of $T^{(i)}_{\{a\}} \equiv \{a\}$.

For the case where C is of the form $\neg C_1$, we have again by definition that

$$(T^{(i)}_{\neg C_1})^{\mathcal{J}} = (\neg C_1)^{\mathcal{I}_i} = \Delta^{\mathcal{I}_i} \setminus C_1^{\mathcal{I}_i} = \Delta^{\mathcal{J}} \setminus (T^{(i)}_{C_1})^{\mathcal{J}} = (\neg T^{(i)}_{C_1})^{\mathcal{J}},$$

which yields that $\mathcal{J}$ is a model of $T^{(i)}_{\neg C_1} \equiv \neg T^{(i)}_{C_1}$.

For the case where C is of the form $C_1 \sqcap C_2$, we have analogously that

$$(T^{(i)}_{C_1 \sqcap C_2})^{\mathcal{J}} = (C_1 \sqcap C_2)^{\mathcal{I}_i} = C_1^{\mathcal{I}_i} \cap C_2^{\mathcal{I}_i} = (T^{(i)}_{C_1})^{\mathcal{J}} \cap (T^{(i)}_{C_2})^{\mathcal{J}} = (T^{(i)}_{C_1} \sqcap T^{(i)}_{C_2})^{\mathcal{J}},$$

which yields that $\mathcal{J}$ is a model of $T^{(i)}_{C_1 \sqcap C_2} \equiv T^{(i)}_{C_1} \sqcap T^{(i)}_{C_2}$.

For the case where C is of the form $\exists r.C_1$, we have

$$\begin{aligned}
(T^{(i)}_{\exists r.C_1})^{\mathcal{J}} &= (\exists r.C_1)^{\mathcal{I}_i} \\
&= \{d \in \Delta^{\mathcal{I}_i} \mid \text{there is an } e \in \Delta^{\mathcal{I}_i} \text{ with } (d,e) \in r^{\mathcal{I}_i} \text{ and } e \in C_1^{\mathcal{I}_i}\} \\
&= \{d \in N^{\mathcal{J}} \mid \text{there is an } e \in \Delta^{\mathcal{I}_i} \text{ with } (d,e) \in r^{\mathcal{I}_i} \text{ and } e \in C_1^{\mathcal{I}_i}\} \cup \\
&\quad\ \{d \in \Delta^{\mathcal{I}_i} \setminus N^{\mathcal{J}} \mid \text{there is an } e \in \Delta^{\mathcal{I}_i} \text{ with } (d,e) \in r^{\mathcal{I}_i} \text{ and } e \in C_1^{\mathcal{I}_i}\} \\
&= \{d \in N^{\mathcal{J}} \mid \text{there is an } e \in \Delta^{\mathcal{I}_i} \setminus N^{\mathcal{J}} \text{ with } (d,e) \in r^{\mathcal{I}_i} \text{ and } e \in C_1^{\mathcal{I}_i}\} \cup \\
&\quad\ \{d \in N^{\mathcal{J}} \mid \text{there is an } e \in N^{\mathcal{J}} \text{ with } (d,e) \in r^{\mathcal{I}_i} \text{ and } e \in C_1^{\mathcal{I}_i}\} \cup \\
&\quad\ \{d \in \Delta^{\mathcal{I}_i} \setminus N^{\mathcal{J}} \mid \text{there is an } e \in \Delta^{\mathcal{I}_i} \text{ with } (d,e) \in r^{\mathcal{I}_i} \text{ and } e \in C_1^{\mathcal{I}_i}\} \\
&\overset{*}{=} \{d \in N^{\mathcal{J}} \mid \text{there is an } e \in \Delta^{\mathcal{I}_i} \setminus N^{\mathcal{J}} \text{ with } (d,e) \in r^{\mathcal{I}_0} \text{ and } e \in C_1^{\mathcal{I}_i}\} \cup \\
&\quad\ \{d \in N^{\mathcal{J}} \mid \text{there is an } e \in N^{\mathcal{J}} \text{ with } (d,e) \in r^{\mathcal{I}_i} \text{ and } e \in C_1^{\mathcal{I}_i}\} \cup \\
&\quad\ \{d \in \Delta^{\mathcal{I}_i} \setminus N^{\mathcal{J}} \mid \text{there is an } e \in \Delta^{\mathcal{I}_i} \text{ with } (d,e) \in r^{\mathcal{I}_0} \text{ and } e \in C_1^{\mathcal{I}_i}\} \\
&= \Big(N^{\mathcal{J}} \cap \\
&\quad\ \big(\{d \in \Delta^{\mathcal{J}} \mid \text{there is an } e \in \Delta^{\mathcal{J}} \text{ with } (d,e) \in (r^{(0)})^{\mathcal{J}} \text{ and } e \in (\neg N \sqcap T^{(i)}_{C_1})^{\mathcal{J}}\} \cup \\
&\quad\ \ \{d \in \Delta^{\mathcal{J}} \mid \text{there is an } e \in \Delta^{\mathcal{J}} \text{ with } (d,e) \in (r^{(i)})^{\mathcal{J}} \text{ and } e \in (N \sqcap T^{(i)}_{C_1})^{\mathcal{J}}\}\big)\Big) \cup \\
&\quad\ \Big((\neg N)^{\mathcal{J}} \cap \\
&\quad\ \ \{d \in \Delta^{\mathcal{J}} \mid \text{there is an } e \in \Delta^{\mathcal{J}} \text{ with } (d,e) \in (r^{(0)})^{\mathcal{J}} \text{ and } e \in (T^{(i)}_{C_1})^{\mathcal{J}}\}\Big) \\
&= \Big(\big(N \sqcap \big((\exists r^{(0)}.(\neg N \sqcap T^{(i)}_{C_1})) \sqcup (\exists r^{(i)}.(N \sqcap T^{(i)}_{C_1}))\big)\big) \sqcup (\neg N \sqcap \exists r^{(0)}.T^{(i)}_{C_1})\Big)^{\mathcal{J}}.
\end{aligned}$$

The starred equality $\overset{*}{=}$ holds due to Claim 6.32. This shows that $\mathcal{J}$ is a model of the concept definition of $T^{(i)}_{\exists r.C_1}$.

For the case where C is of the form $\geq m\, r.C_1$, we have by similar arguments that

$$\begin{aligned}
(T^{(i)}_{\geq m\, r.C_1})^{\mathcal{J}} &= (\geq m\, r.C_1)^{\mathcal{I}_i} \\
&= \big\{d \in \Delta^{\mathcal{I}_i} \mid |\{e \in \Delta^{\mathcal{I}_i} \mid (d,e) \in r^{\mathcal{I}_i} \text{ and } e \in C_1^{\mathcal{I}_i}\}| \geq m\big\} \\
&= \big\{d \in N^{\mathcal{J}} \mid |\{e \in \Delta^{\mathcal{I}_i} \mid (d,e) \in r^{\mathcal{I}_i} \text{ and } e \in C_1^{\mathcal{I}_i}\}| \geq m\big\} \cup \\
&\quad\ \big\{d \in \Delta^{\mathcal{I}_i} \setminus N^{\mathcal{J}} \mid |\{e \in \Delta^{\mathcal{I}_i} \mid (d,e) \in r^{\mathcal{I}_i} \text{ and } e \in C_1^{\mathcal{I}_i}\}| \geq m\big\}.
\end{aligned}$$

We have for every $d \in N^{\mathcal{J}}$ that

$d \in (T^{(i)}_{\geq m\, r.C_1})^{\mathcal{J}}$

iff $|\{e \in \Delta^{\mathcal{I}_i} \mid (d,e) \in r^{\mathcal{I}_i} \text{ and } e \in C_1^{\mathcal{I}_i}\}| \geq m$

iff there is a j, $0 \leq j \leq m$, such that $|\{e \in N^{\mathcal{J}} \mid (d,e) \in r^{\mathcal{I}_i} \text{ and } e \in C_1^{\mathcal{I}_i}\}| \geq j$ and $|\{e \in \Delta^{\mathcal{I}_i} \setminus N^{\mathcal{J}} \mid (d,e) \in r^{\mathcal{I}_i} \text{ and } e \in C_1^{\mathcal{I}_i}\}| \geq m - j$

iff there is a j, $0 \leq j \leq m$, such that $|\{e \in N^{\mathcal{J}} \mid (d,e) \in r^{\mathcal{I}_i} \text{ and } e \in C_1^{\mathcal{I}_i}\}| \geq j$ and $|\{e \in \Delta^{\mathcal{I}_i} \setminus N^{\mathcal{J}} \mid (d,e) \in r^{\mathcal{I}_0} \text{ and } e \in C_1^{\mathcal{I}_i}\}| \geq m - j$ by Claim 6.32

iff there is a j, $0 \leq j \leq m$, such that $|\{e \in N^{\mathcal{J}} \mid (d,e) \in (r^{(i)})^{\mathcal{J}}$ and $e \in (T^{(i)}_{C_1})^{\mathcal{J}}\}| \geq j$ and $|\{e \in \Delta^{\mathcal{I}_i} \setminus N^{\mathcal{J}} \mid (d,e) \in (r^{(0)})^{\mathcal{J}}$ and $e \in (T^{(i)}_{C_1})^{\mathcal{J}}\}| \geq m - j$

iff there is a j, $0 \leq j \leq m$, such that we have that $d \in (\geq j\, r^{(i)}.(N \sqcap T^{(i)}_{C_1}))^{\mathcal{J}}$ and $d \in (\geq (m-j)\, r^{(0)}.(\neg N \sqcap T^{(i)}_{C_1}))^{\mathcal{J}}$

iff $d \in \Big(N \sqcap \bigsqcup_{0 \leq j \leq m} \big(\geq j\, r^{(i)}.(N \sqcap T^{(i)}_{C_1}) \sqcap \geq (m-j)\, r^{(0)}.(\neg N \sqcap T^{(i)}_{C_1})\big)\Big)^{\mathcal{J}}$.

Moreover, we have for every $d \in \Delta^{\mathcal{J}} \setminus N^{\mathcal{J}}$ that

$d \in (T^{(i)}_{\geq m\, r.C_1})^{\mathcal{J}}$

iff $|\{e \in \Delta^{\mathcal{I}_i} \mid (d,e) \in r^{\mathcal{I}_i}$ and $e \in C_1^{\mathcal{I}_i}\}| \geq m$

iff $|\{e \in \Delta^{\mathcal{I}_i} \mid (d,e) \in r^{\mathcal{I}_0}$ and $e \in C_1^{\mathcal{I}_i}\}| \geq m$ by Claim 6.32

iff $|\{e \in \Delta^{\mathcal{J}} \mid (d,e) \in (r^{(i)})^{\mathcal{J}}$ and $e \in (T^{(i)}_{C_1})^{\mathcal{J}}\}| \geq m$

iff $d \in (\neg N \sqcap \geq m\, r^{(0)}.T^{(i)}_{C_1})^{\mathcal{J}}$.

Hence, we obtain that $\mathcal{J}$ is a model of the concept definition of $T^{(i)}_{\geq m\, r.C_1}$.

This finishes the proof that $\mathcal{J} \models \mathcal{T}_{\mathsf{sub}}$, and thus we have $\mathcal{J} \models \mathcal{T}_{\mathsf{red}}$, which finishes the proof of Property (1).

To prove Property (2), let $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ be an interpretation with $\mathcal{J} \models \mathcal{A}_{\mathsf{red}}$ and $\mathcal{J} \models \mathcal{T}_{\mathsf{red}}$. We define the interpretations $\mathcal{I}_0 = (\Delta^{\mathcal{I}_0}, \cdot^{\mathcal{I}_0}), \ldots, \mathcal{I}_n = (\Delta^{\mathcal{I}_n}, \cdot^{\mathcal{I}_n})$ as follows:

- $\Delta^{\mathcal{I}_i} := \Delta^{\mathcal{J}}$ for every i, $0 \leq i \leq n$;
- $a^{\mathcal{I}_i} := a^{\mathcal{J}}$ for every $a \in \mathsf{N_I}$ and every i, $0 \leq i \leq n$;
- $A^{\mathcal{I}_i} := (T^{(i)}_A)^{\mathcal{J}}$ for every $A \in \mathcal{R} \cap \mathsf{N_C}$ and every i, $0 \leq i \leq n$; and
- $r^{\mathcal{I}_i} := \big((r^{(i)})^{\mathcal{J}} \cap (N^{\mathcal{J}} \times N^{\mathcal{J}})\big) \cup \Big((r^{(0)})^{\mathcal{J}} \cap \big((\Delta^{\mathcal{J}} \times (\neg N)^{\mathcal{J}}) \cup ((\neg N)^{\mathcal{J}} \times \Delta^{\mathcal{J}})\big)\Big)$ for every $r \in \mathcal{R} \cap \mathsf{N_R}$ and every i, $0 \leq i \leq n$.

The interpretation of concept names and role names that are not contained in $\mathcal{R}$ is irrelevant. We assume in the following without loss of generality that the interpretation of all such names in empty in all interpretations $\mathcal{I}_i$, $0 \leq i \leq n$.

We first show Property (2b). To prove this property, take any i, $0 \leq i \leq n$. We proceed by induction on the structure of C, where we use that $\mathcal{J} \models \mathcal{T}_{\mathsf{red}}$. For the case where $C = A \in \mathsf{N_C}$, we have the claim by definition.

For the case where C is of the form $\{a\}$ with $a \in \mathsf{Obj}$, we have

$$\{a\}^{\mathcal{I}_i} = \{a^{\mathcal{I}_i}\} = \{a^{\mathcal{J}}\} = \{a\}^{\mathcal{J}} = (T^{(i)}_{\{a\}})^{\mathcal{J}}.$$

For the case where C is of the form $\neg C_1$, we have

$$(\neg C_1)^{\mathcal{I}_i} = \Delta^{\mathcal{I}_i} \setminus C_1^{\mathcal{I}_i} = \Delta^{\mathcal{J}} \setminus (T^{(i)}_{C_1})^{\mathcal{J}} = (\neg T^{(i)}_{C_1})^{\mathcal{J}} = (T^{(i)}_{\neg C_1})^{\mathcal{J}}.$$

For the case where C is of the form $C_1 \sqcap C_2$, we have

$$(C_1 \sqcap C_2)^{\mathcal{I}_i} = C_1^{\mathcal{I}_i} \cap C_2^{\mathcal{I}_i} = (T^{(i)}_{C_1})^{\mathcal{J}} \cap (T^{(i)}_{C_2})^{\mathcal{J}} = (T^{(i)}_{C_1} \sqcap T^{(i)}_{C_2})^{\mathcal{J}} = (T^{(i)}_{C_1 \sqcap C_2})^{\mathcal{J}}.$$

For the case where C is of the form $\exists r.C_1$, we have

$d \in (\exists r.C_1)^{\mathcal{I}_i}$

iff $d \in \Delta^{\mathcal{I}_i}$ and there is an $e \in \Delta^{\mathcal{I}_i}$ with $(d,e) \in r^{\mathcal{I}_i}$ and $e \in C_1^{\mathcal{I}_i}$

iff either $d \in N^{\mathcal{J}}$ and there is an $e \in \Delta^{\mathcal{J}}$ with $(d,e) \in r^{\mathcal{I}_i}$ and $e \in C_1^{\mathcal{I}_i}$, or $d \in \Delta^{\mathcal{J}} \setminus N^{\mathcal{J}}$ and there is an $e \in \Delta^{\mathcal{J}}$ with $(d,e) \in r^{\mathcal{I}_i}$ and $e \in C_1^{\mathcal{I}_i}$

iff either $d \in N^{\mathcal{J}}$ and there is an $e \in \Delta^{\mathcal{J}} \setminus N^{\mathcal{J}}$ with $(d,e) \in r^{\mathcal{I}_i}$ and $e \in C_1^{\mathcal{I}_i}$, or $d \in N^{\mathcal{J}}$ and there is an $e \in N^{\mathcal{J}}$ with $(d,e) \in r^{\mathcal{I}_i}$ and $e \in C_1^{\mathcal{I}_i}$, or $d \in \Delta^{\mathcal{J}} \setminus N^{\mathcal{J}}$ and there is an $e \in \Delta^{\mathcal{J}}$ with $(d,e) \in r^{\mathcal{I}_i}$ and $e \in C_1^{\mathcal{I}_i}$

iff either $d \in N^{\mathcal{J}}$ and there is an $e \in \Delta^{\mathcal{J}} \setminus N^{\mathcal{J}}$ with $(d,e) \in (r^{(0)})^{\mathcal{J}}$ and $e \in (T^{(i)}_{C_1})^{\mathcal{J}}$, or $d \in N^{\mathcal{J}}$ and there is an $e \in N^{\mathcal{J}}$ with $(d,e) \in (r^{(i)})^{\mathcal{J}}$ and $e \in (T^{(i)}_{C_1})^{\mathcal{J}}$, or $d \in \Delta^{\mathcal{J}} \setminus N^{\mathcal{J}}$ and there is an $e \in \Delta^{\mathcal{J}}$ with $(d,e) \in (r^{(0)})^{\mathcal{J}}$ and $e \in (T^{(i)}_{C_1})^{\mathcal{J}}$ (by the definition of $r^{\mathcal{I}_i}$ and the induction hypothesis)

iff $d \in \Big(\big(N \sqcap \big((\exists r^{(0)}.(\neg N \sqcap T^{(i)}_{C_1})) \sqcup (\exists r^{(i)}.(N \sqcap T^{(i)}_{C_1}))\big)\big) \sqcup (\neg N \sqcap \exists r^{(0)}.T^{(i)}_{C_1})\Big)^{\mathcal{J}}$

iff $d \in (T^{(i)}_{\exists r.C_1})^{\mathcal{J}}$ (since $\mathcal{J} \models \mathcal{T}_{\mathsf{red}}$).

For the case where C is of the form $\geq m\, r.C_1$, we have by similar arguments as in the previous case that

$d \in (\geq m\, r.C_1)^{\mathcal{I}_i}$

iff $d \in \Delta^{\mathcal{I}_i}$ and $|\{e \in \Delta^{\mathcal{I}_i} \mid (d,e) \in r^{\mathcal{I}_i} \text{ and } e \in C_1^{\mathcal{I}_i}\}| \geq m$

iff either $d \in N^{\mathcal{J}}$ and there is a j, $0 \leq j \leq m$, such that

$$|\{e \in N^{\mathcal{J}} \mid (d,e) \in r^{\mathcal{I}_i} \text{ and } e \in C_1^{\mathcal{I}_i}\}| \geq j$$

and

$$|\{e \in \Delta^{\mathcal{I}_i} \setminus N^{\mathcal{J}} \mid (d,e) \in r^{\mathcal{I}_i} \text{ and } e \in C_1^{\mathcal{I}_i}\}| \geq m - j,$$

or we have $d \in \Delta^{\mathcal{J}} \setminus N^{\mathcal{J}}$ and $|\{e \in \Delta^{\mathcal{I}_i} \mid (d,e) \in r^{\mathcal{I}_i} \text{ and } e \in C_1^{\mathcal{I}_i}\}| \geq m$

iff either $d \in N^{\mathcal{J}}$ and there is a j, $0 \leq j \leq m$, such that

$$|\{e \in N^{\mathcal{J}} \mid (d,e) \in (r^{(i)})^{\mathcal{J}} \text{ and } e \in (T^{(i)}_{C_1})^{\mathcal{J}}\}| \geq j$$

and

$$|\{e \in \Delta^{\mathcal{J}} \setminus N^{\mathcal{J}} \mid (d,e) \in (r^{(0)})^{\mathcal{J}} \text{ and } e \in (T^{(i)}_{C_1})^{\mathcal{J}}\}| \geq m - j,$$

or we have $d \in \Delta^{\mathcal{J}} \setminus N^{\mathcal{J}}$ and $|\{e \in \Delta^{\mathcal{J}} \mid (d,e) \in (r^{(0)})^{\mathcal{J}} \text{ and } e \in (T^{(i)}_{C_1})^{\mathcal{J}}\}| \geq m$ (by the definition of $r^{\mathcal{I}_i}$ and the induction hypothesis) *iff* either $d \in N^{\mathcal{J}}$ and there is a j, $0 \le j \le m$, such that $d \in (\ge j\, r^{(i)}.(N \sqcap T^{(i)}_{C_1}))^{\mathcal{J}}$ and $d \in (\ge(m-j)\, r^{(0)}.(\neg N \sqcap T^{(i)}_{C_1}))^{\mathcal{J}}$, or we have $d \in \Delta^{\mathcal{J}} \setminus N^{\mathcal{J}}$ and $d \in (\ge m\, r^{(0)}.T^{(i)}_{C_1})^{\mathcal{J}}$

iff either $d \in \Big(N \sqcap \bigsqcup_{0 \le j \le m}\big(\ge j\, r^{(i)}.(N \sqcap T^{(i)}_{C_1}) \sqcap \ge(m-j)\, r^{(0)}.(\neg N \sqcap T^{(i)}_{C_1})\big)\Big)^{\mathcal{J}}$ or $d \in (\neg N \sqcap \ge m\, r^{(0)}.T^{(i)}_{C_1})^{\mathcal{J}}$

iff $d \in (T^{(i)}_{\ge m\, r.C_1})^{\mathcal{J}}$ (since $\mathcal{J} \models \mathcal{T}_{\mathsf{red}}$).

This finishes the proof of Property (2b).

We now prove Property (2a). For that, take any i with $0 \le i \le n$, and any relevant generalised ABox-literal α. We prove the property using again a case distinction. Assume that α is of the form $C(a)$ where C is a concept and $a \in \mathsf{N_I}$. We have: $\mathcal{I}_i \models \alpha$ iff $a^{\mathcal{I}_i} \in C^{\mathcal{I}_i}$ iff $a^{\mathcal{J}} \in (T^{(i)}_C)^{\mathcal{J}}$ (by Property (2b)) iff $\mathcal{J} \models \alpha^{(i)}$.

Assume now that α is of the form $\neg C(a)$ where C is a concept and $a \in \mathsf{N_I}$. By similar arguments and Property (2b), we have: $\mathcal{I}_i \models \alpha$ iff $a \notin C^{\mathcal{I}_i}$ iff $a^{\mathcal{J}} \notin (T^{(i)}_C)^{\mathcal{J}}$ iff $\mathcal{J} \not\models T^{(i)}_C(a)$ iff $\mathcal{J} \models \neg T^{(i)}_C(a)$ iff $\mathcal{J} \models \alpha^{(i)}$.

For the case that α is of the form $r(a,b)$ where $r \in \mathsf{N_R}$ and $a, b \in \mathsf{N_I}$, we have: $\mathcal{I}_i \models \alpha$ iff $(a^{\mathcal{I}_i}, b^{\mathcal{I}_i}) \in r^{\mathcal{I}_i}$ iff $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in (r^{(i)})^{\mathcal{J}}$ (by the definition of $r^{\mathcal{I}_i}$) iff $\mathcal{J} \models r^{(i)}(a,b)$ iff $\mathcal{J} \models \alpha^{(i)}$.

Finally, assume that α is of the form $\neg r(a,b)$ where $r \in \mathsf{N_R}$ and $a, b \in \mathsf{N_I}$. We have: $\mathcal{I}_i \models \alpha$ iff $(a^{\mathcal{I}_i}, b^{\mathcal{I}_i}) \notin r^{\mathcal{I}_i}$ iff $(a^{\mathcal{J}}, b^{\mathcal{J}}) \notin (r^{(i)})^{\mathcal{J}}$ (again by the definition of $r^{\mathcal{I}_i}$) iff $\mathcal{J} \not\models r^{(i)}(a,b)$ iff $\mathcal{J} \models \neg r^{(i)}(a,b)$ iff $\mathcal{J} \models \alpha^{(i)}$. This finishes the proof of Property (2a).

Thus, it is only left to be shown that we have also $\mathcal{I}_i \Rightarrow^{\emptyset,\emptyset}_{\boldsymbol{b}_{i+1}} \mathcal{I}_{i+1}$ for every i, $0 \le i < n$. For that, take any i, $0 \le i < n$. We show that the conditions in Definition 6.13 are satisfied.

We start showing that $\boldsymbol{b}_{i+1}$ is applicable to $\mathcal{I}_i$ w.r.t. the empty TBox, i.e. that $\mathcal{I}_i \models \mathsf{pre}_{i+1}$. Since $\mathcal{J} \models \mathcal{A}_{\mathsf{red}}$, we have that $\mathcal{J} \models \mathcal{A}^{(i+1)}_{\mathsf{pre}}$, i.e. $\mathcal{J} \models \alpha^{(i)}$ for every $\alpha \in \mathsf{pre}_{i+1}$. Since all such $\alpha \in \mathsf{pre}_{i+1}$ are relevant generalised ABox-literals, we have by Property (2a) that $\mathcal{I}_i \models \mathsf{pre}_{i+1}$.

Moreover, we have by definition that $\Delta^{\mathcal{I}_i} = \Delta^{\mathcal{J}} = \Delta^{\mathcal{I}_{i+1}}$ and $a^{\mathcal{I}_i} = a^{\mathcal{J}} = a^{\mathcal{I}_{i+1}}$ for every $a \in \mathsf{N_I}$.

We show next that $\mathsf{Eff}(\boldsymbol{b}_{i+1}, \mathcal{I}_i, \emptyset)$ is not contradictory. Since the set of causal relationships is empty and $\boldsymbol{b}_{i+1}$ is unconditional, we have that $\mathsf{Eff}(\boldsymbol{b}_{i+1}, \mathcal{I}_i, \emptyset) = \mathsf{post}_{i+1}$. Since $\mathcal{J} \models \mathcal{A}_{\mathsf{red}}$, we have that $\mathcal{J} \models \mathcal{A}^{(i+1)}_{\mathsf{post}}$, i.e. $\mathcal{J} \models \alpha^{(i+1)}$ for every $\alpha \in \mathsf{post}_{i+1}$. Since all ABox-literals in post_{i+1} are relevant, we have by Property (2a) that $\mathcal{I}_{i+1} \models \mathsf{post}_{i+1}$. Hence, $\mathsf{Eff}(\boldsymbol{b}_{i+1}, \mathcal{I}_i, \emptyset)$ cannot be contradictory.

Let $A \in \mathsf{N_C} \cap \mathcal{R}$, let $A^+ := \{a^{\mathcal{I}_i} \mid A(a) \in \mathsf{Eff}(\boldsymbol{b}_{i+1}, \mathcal{I}_i, \emptyset)\} = \{a^{\mathcal{I}_i} \mid A(a) \in \mathsf{post}_{i+1}\}$, and let $A^- := \{a^{\mathcal{I}_i} \mid \neg A(a) \in \mathsf{Eff}(\boldsymbol{b}_{i+1}, \mathcal{I}_i, \emptyset)\} = \{a^{\mathcal{I}_i} \mid \neg A(a) \in \mathsf{post}_{i+1}\}$. Since $\mathsf{Eff}(\boldsymbol{b}_{i+1}, \mathcal{I}_i, \emptyset)$ is not contradictory, we have that $A^+ \cap A^- = \emptyset$. Moreover, we have by definition that $A^+ \subseteq N^{\mathcal{J}}$ and $A^- \subseteq N^{\mathcal{J}}$. We first show that $A^{\mathcal{I}_{i+1}} \setminus N^{\mathcal{J}} = A^{\mathcal{I}_i} \setminus N^{\mathcal{J}}$. Since $\mathcal{J} \models \mathcal{T}_{\mathsf{sub}}$, we have

$$\begin{aligned}
A^{\mathcal{I}_{i+1}} \setminus N^{\mathcal{J}} &= (T_A^{(i+1)})^{\mathcal{J}} \setminus N^{\mathcal{J}} \\
&= \big((N^{\mathcal{J}} \cap (A^{(i+1)})^{\mathcal{J}}) \cup ((\Delta^{\mathcal{J}} \setminus N^{\mathcal{J}}) \cap (A^{(0)}))^{\mathcal{J}}\big) \setminus N^{\mathcal{J}} \\
&= ((\Delta^{\mathcal{J}} \setminus N^{\mathcal{J}}) \cap (A^{(0)}))^{\mathcal{J}} \setminus N^{\mathcal{J}} \\
&= \big((N^{\mathcal{J}} \cap (A^{(i)})^{\mathcal{J}}) \cup ((\Delta^{\mathcal{J}} \setminus N^{\mathcal{J}}) \cap (A^{(0)}))^{\mathcal{J}}\big) \setminus N^{\mathcal{J}} \\
&= (T_A^{(i)})^{\mathcal{J}} \setminus N^{\mathcal{J}} \\
&= A^{\mathcal{I}_i} \setminus N^{\mathcal{J}}
\end{aligned}$$

Hence, we have for every $d \in \Delta^{\mathcal{J}} \setminus N^{\mathcal{J}}$ that $d \in A^{\mathcal{I}_{i+1}}$ iff $d \in (A^{\mathcal{I}_i} \cup A^+) \setminus A^-$.

Next, we show the following claim.

Claim 6.33. *For every $a^{\mathcal{J}} \in N^{\mathcal{J}}$, and every j, $0 \leq j \leq n$, we have $a^{\mathcal{J}} \in A^{\mathcal{I}_j}$ iff $a^{\mathcal{J}} \in (A^{(j)})^{\mathcal{J}}$.*

Take any $a^{\mathcal{J}} \in N^{\mathcal{J}}$ and any j, $0 \leq j \leq n$. We have

$$\begin{aligned}
&a^{\mathcal{J}} \in A^{\mathcal{I}_j} \\
\textit{iff}\quad &a^{\mathcal{J}} \in (T_A^{(j)})^{\mathcal{J}} \\
\textit{iff}\quad &a^{\mathcal{J}} \in N^{\mathcal{J}} \cap (A^{(j)})^{\mathcal{J}} \text{ or } a^{\mathcal{J}} \in (\Delta^{\mathcal{J}} \setminus N^{\mathcal{J}}) \cap (A^{(0)})^{\mathcal{J}} \text{ since } \mathcal{J} \models \mathcal{T}_{\mathsf{sub}} \\
\textit{iff}\quad &a^{\mathcal{J}} \in (A^{(j)})^{\mathcal{J}} \text{ since } a^{\mathcal{J}} \in N^{\mathcal{J}}.
\end{aligned}$$

This finishes the proof of Claim 6.33.

We prove that for every $a^{\mathcal{J}} \in N^{\mathcal{J}}$, we have $a^{\mathcal{J}} \in A^{\mathcal{I}_{i+1}}$ iff $a^{\mathcal{J}} \in (A^{\mathcal{I}_i} \cup A^+) \setminus A^-$ by a case distinction. For the 'if' direction, it is obvious that we have $a^{\mathcal{J}} \notin A^-$. Now, consider first the case where $a^{\mathcal{J}} \in A^+$. Then, the definition of A^+ yields that $A(a) \in \mathsf{post}_{i+1}$. Hence, we have $a^{\mathcal{J}} \in (T_A^{(i+1)})^{\mathcal{J}}$ since $\mathcal{J} \models \mathcal{A}_{\mathsf{post}}^{(i+1)}$. By the definition of $A^{\mathcal{I}_{i+1}}$, we have $a^{\mathcal{J}} \in A^{\mathcal{I}_{i+1}}$.

Consider now the case where $a^{\mathcal{J}} \notin A^+$, i.e. $a^{\mathcal{J}} \in A^{\mathcal{I}_i} \setminus A^-$. Since $a^{\mathcal{J}} \in A^{\mathcal{I}_i}$, we have by Claim 6.33 that $a^{\mathcal{J}} \in (A^{(i)})^{\mathcal{J}}$. Moreover, we have $\neg A(a) \notin \mathsf{post}_{i+1}$ by the definition of A^- since $a^{\mathcal{J}} \notin A^-$. Since $\mathcal{J} \models \mathcal{A}_{\mathsf{min}}^{(i+1)}$, we have also that $\mathcal{J} \models (A^{(i)} \rightarrow A^{(i+1)})(a)$, i.e. $a^{\mathcal{J}} \in (A^{(i)})^{\mathcal{J}}$ implies $a^{\mathcal{J}} \in (A^{(i+1)})^{\mathcal{J}}$. Since $a^{\mathcal{J}} \in (A^{(i)})^{\mathcal{J}}$, we have $a^{\mathcal{J}} \in (A^{(i+1)})^{\mathcal{J}}$, which yields $a^{\mathcal{J}} \in A^{\mathcal{I}_{i+1}}$ by Claim 6.33.

For the 'only if' direction, assume to the contrary that $a^{\mathcal{J}} \in A^{\mathcal{I}_{i+1}}$, $a^{\mathcal{J}} \notin A^+$, and $a^{\mathcal{J}} \notin A^{\mathcal{I}_i} \setminus A^-$. There are again two cases to consider: either $a^{\mathcal{J}} \in A^-$ or $a^{\mathcal{J}} \notin A^-$. If $a^{\mathcal{J}} \in A^-$, then $\neg A(a) \in \mathsf{post}_{i+1}$ by the definition of A^-. Since $\mathcal{J} \models \mathcal{A}_{\mathsf{post}}^{(i+1)}$, we have that $\mathcal{J} \models (T_{\neg A}^{(i+1)})(a)$, i.e. $a^{\mathcal{J}} \in (T_{\neg A}^{(i+1)})^{\mathcal{J}}$. Since $\mathcal{J} \models \mathcal{T}_{\mathsf{sub}}$, this yields $a^{\mathcal{J}} \in (\neg T_A^{(i+1)})^{\mathcal{J}}$, i.e. $a^{\mathcal{J}} \notin (T_A^{(i+1)})^{\mathcal{J}}$. The definition of $A^{\mathcal{I}_{i+1}}$ yields that $a^{\mathcal{J}} \notin A^{\mathcal{I}_{i+1}}$, which is a contradiction.

Otherwise, if $a^{\mathcal{J}} \notin A^-$, we have $a^{\mathcal{J}} \notin A^{\mathcal{I}_i}$, and thus $a^{\mathcal{J}} \notin (A^{(i)})^{\mathcal{J}}$ by Claim 6.33. Since $a^{\mathcal{J}} \notin A^+$, we have $A(a) \notin \mathsf{post}_{i+1}$ by the definition of A^+. Since $\mathcal{J} \models \mathcal{A}_{\mathsf{min}}^{(i+1)}$, we have also that $\mathcal{J} \models (\neg A^{(i)} \rightarrow \neg A^{(i+1)})(a)$, i.e. $a^{\mathcal{J}} \notin (A^{(i)})^{\mathcal{J}}$ implies $a^{\mathcal{J}} \notin (A^{(i+1)})^{\mathcal{J}}$. Since we have $a^{\mathcal{J}} \notin (A^{(i)})^{\mathcal{J}}$, this yields $a^{\mathcal{J}} \notin (A^{(i+1)})^{\mathcal{J}}$. Thus, by Claim 6.33, we have $a^{\mathcal{J}} \notin A^{\mathcal{I}_{i+1}}$, which again is a contradiction.

Thus, we have shown that $A^{\mathcal{I}_{i+1}} = (A^{\mathcal{I}_i} \cup A^+) \setminus A^-$. Finally, let $r \in \mathsf{N_R} \cap \mathcal{R}$, let

$$r^+ := \{(a^{\mathcal{I}_i}, b^{\mathcal{I}_i}) \mid r(a,b) \in \mathsf{Eff}(\boldsymbol{b}_{i+1}, \mathcal{I}_i, \emptyset)\} = \{(a^{\mathcal{I}_i}, b^{\mathcal{I}_i}) \mid r(a,b) \in \mathsf{post}_{i+1}\},$$

and let

$$r^- := \{(a^{\mathcal{I}_i}, b^{\mathcal{I}_i}) \mid \neg r(a,b) \in \mathsf{Eff}(\boldsymbol{b}_{i+1}, \mathcal{I}_i, \emptyset)\} = \{(a^{\mathcal{I}_i}, b^{\mathcal{I}_i}) \mid \neg r(a,b) \in \mathsf{post}_{i+1}\}.$$

Since $\mathsf{Eff}(\boldsymbol{b}_{i+1}, \mathcal{I}_i, \emptyset)$ is not contradictory, we have that $r^+ \cap r^- = \emptyset$. Moreover, we have by definition that $r^+ \subseteq N^{\mathcal{J}} \times N^{\mathcal{J}}$ and $r^- \subseteq N^{\mathcal{J}} \times N^{\mathcal{J}}$. Similar to before, we first show that $r^{\mathcal{I}_{i+1}} \setminus (N^{\mathcal{J}} \times N^{\mathcal{J}}) = r^{\mathcal{I}_i} \setminus (N^{\mathcal{J}} \times N^{\mathcal{J}})$. By the definitions of $r^{\mathcal{I}_{i+1}}$ and $r^{\mathcal{I}_i}$, we have

$$\begin{aligned} r^{\mathcal{I}_{i+1}} \setminus (N^{\mathcal{J}} \times N^{\mathcal{J}}) &= \big((r^{(0)})^{\mathcal{J}} \cap ((\Delta^{\mathcal{J}} \times (\neg N)^{\mathcal{J}}) \cup ((\neg N)^{\mathcal{J}} \times \Delta^{\mathcal{J}}))\big) \setminus (N^{\mathcal{J}} \times N^{\mathcal{J}}) \\ &= r^{\mathcal{I}_i} \setminus (N^{\mathcal{J}} \times N^{\mathcal{J}}). \end{aligned}$$

Hence, we have for all $d, e \in \Delta^{\mathcal{J}} \setminus N^{\mathcal{J}}$ that $(d,e) \in r^{\mathcal{I}_{i+1}}$ iff $(d,e) \in (r^{\mathcal{I}_i} \cup r^+) \setminus r^-$.

The following claim is also an immediate consequence of the definition of $r^{\mathcal{I}_i}$.

Claim 6.34. *For every $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in N^{\mathcal{J}} \times N^{\mathcal{J}}$, and every j, $0 \leq j \leq n$, we have $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in r^{\mathcal{I}_j}$ iff $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in (r^{(j)})^{\mathcal{J}}$.*

Again by a case distinction, we prove that for every $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in N^{\mathcal{J}} \times N^{\mathcal{J}}$, we have that $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in r^{\mathcal{I}_{i+1}}$ iff $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in (r^{\mathcal{I}_i} \cup r^+) \setminus r^-$. For the 'if' direction, it is obvious that we have $(a^{\mathcal{J}}, b^{\mathcal{J}}) \notin r^-$. Now, consider first the case where $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in r^+$. Then, the definition of r^+ yields that $r(a,b) \in \mathsf{post}_{i+1}$. Hence, we have $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in (r^{(i+1)})^{\mathcal{J}}$ since $\mathcal{J} \models \mathcal{A}^{(i+1)}_{\mathsf{post}}$. By Claim 6.34, we have $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in r^{\mathcal{I}_{i+1}}$.

Consider now the case where $(a^{\mathcal{J}}, b^{\mathcal{J}}) \notin r^+$, i.e. $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in r^{\mathcal{I}_i} \setminus r^-$. Since we have $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in r^{\mathcal{I}_i}$, we have by Claim 6.34 also that $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in (r^{(i)})^{\mathcal{J}}$. This yields that $a^{\mathcal{J}} \in (\exists r^{(i)}.\{b\})^{\mathcal{J}}$. Moreover, we have $\neg r(a,b) \notin \mathsf{post}_{i+1}$ by the definition of r^- since $(a^{\mathcal{J}}, b^{\mathcal{J}}) \notin r^-$. Since $\mathcal{J} \models \mathcal{A}^{(i+1)}_{\mathsf{min}}$, we have also that $\mathcal{J} \models (\exists r^{(i)}.\{b\} \rightarrow \exists r^{(i+1)}.\{b\})(a)$, i.e. $a^{\mathcal{J}} \in (\exists r^{(i)}.\{b\})^{\mathcal{J}}$ implies $a^{\mathcal{J}} \in (\exists r^{(i+1)}.\{b\})^{\mathcal{J}}$. Since $a^{\mathcal{J}} \in (\exists r^{(i)}.\{b\})^{\mathcal{J}}$, we have $a^{\mathcal{J}} \in (\exists r^{(i+1)}.\{b\})^{\mathcal{J}}$, which yields $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in (r^{(i+1)})^{\mathcal{J}}$. By Claim 6.34, we have thus $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in r^{\mathcal{I}_{i+1}}$.

For the 'only if' direction, assume to the contrary that $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in r^{\mathcal{I}_{i+1}}$, $(a^{\mathcal{J}}, b^{\mathcal{J}}) \notin r^+$, and $(a^{\mathcal{J}}, b^{\mathcal{J}}) \notin r^{\mathcal{I}_i} \setminus r^-$. There are again two cases to consider: either $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in r^-$ or $(a^{\mathcal{J}}, b^{\mathcal{J}}) \notin r^-$. If $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in r^-$, then $\neg r(a,b) \in \mathsf{post}_{i+1}$ by the definition of r^-. Since $\mathcal{J} \models \mathcal{A}^{(i+1)}_{\mathsf{post}}$, we have that $\mathcal{J} \models \neg r^{(i+1)}(a,b)$, i.e. $(a^{\mathcal{J}}, b^{\mathcal{J}}) \notin (r^{(i+1)})^{\mathcal{J}}$. By Claim 6.34, we obtain $(a^{\mathcal{J}}, b^{\mathcal{J}}) \notin r^{\mathcal{I}_{i+1}}$, which is a contradiction.

Otherwise, if $(a^{\mathcal{J}}, b^{\mathcal{J}}) \notin r^-$, we have $(a^{\mathcal{J}}, b^{\mathcal{J}}) \notin r^{\mathcal{I}_i}$, and thus $(a^{\mathcal{J}}, b^{\mathcal{J}}) \notin (r^{(i)})^{\mathcal{J}}$ by Claim 6.34. This yields that $a^{\mathcal{J}} \notin (\exists r^{(i)}.\{b\})^{\mathcal{J}}$. Since $(a^{\mathcal{J}}, b^{\mathcal{J}}) \notin r^+$, we have by the definition of r^+ also that $r(a,b) \notin \mathsf{post}_{i+1}$. Moreover, since $\mathcal{J} \models \mathcal{A}^{(i+1)}_{\mathsf{min}}$, we have also that $\mathcal{J} \models (\neg\exists r^{(i)}.\{b\} \rightarrow \neg\exists r^{(i+1)}.\{b\})(a)$, i.e. $a^{\mathcal{J}} \notin (\exists r^{(i)}.\{b\})^{\mathcal{J}}$ implies $a^{\mathcal{J}} \notin (\exists r^{(i+1)}.\{b\})^{\mathcal{J}}$. Since we have $a^{\mathcal{J}} \notin (\exists r^{(i)}.\{b\})^{\mathcal{J}}$, this yields $a^{\mathcal{J}} \notin (\exists r^{(i+1)}.\{b\})^{\mathcal{J}}$. Thus, we have that $(a^{\mathcal{J}}, b^{\mathcal{J}}) \notin (r^{(i+1)})^{\mathcal{J}}$. Hence, by Claim 6.34, we obtain $(a^{\mathcal{J}}, b^{\mathcal{J}}) \notin r^{\mathcal{I}_{i+1}}$, which again is a contradiction.

We have thus shown that $r^{\mathcal{I}_{i+1}} = (r^{\mathcal{I}_i} \cup r^+) \setminus r^-$. Since we have shown that all conditions in Definition 6.13 are satisfied, this finishes the proof that we have $\mathcal{I}_i \Rightarrow^{\emptyset,\emptyset}_{\boldsymbol{b}_{i+1}} \mathcal{I}_{i+1}$ for every i, $0 \leq i < n$. Thus, we have shown Property (2). □

Now, we can come back to the consistency problem for DL-actions. Let $\boldsymbol{a} = (\mathsf{pre}, \mathsf{post})$ be a DL-action, CR a finite set of causal relationships, and $\mathcal{T}$ a general TBox. The set $\mathcal{R}$ of relevant concept names, role names, and concepts consists of the ones occurring in $\boldsymbol{a}$, CR, or $\mathcal{T}$. Given an action type $T \in \mathfrak{T}(\boldsymbol{a}, \mathsf{CR})$, we can compute the set $\mathsf{Eff}(\boldsymbol{a}, T, \mathsf{CR})$, and check whether this set is not contradictory. If this is the case, then we consider the DL-action $\boldsymbol{b}_{\boldsymbol{a},T,\mathsf{CR}}$, and check whether an application of this DL-action transforms models of $\mathcal{T}$ satisfying pre and T into models of $\mathcal{T}$. This check can be realised using the generalised ABox $\mathcal{A}_{\mathsf{red}}$ and the (acyclic) TBox $\mathcal{T}_{\mathsf{red}}$ of Lemma 6.30.

Lemma 6.35. *The DL-action* $\boldsymbol{a} = (\mathsf{pre}, \mathsf{post})$ *is consistent w.r.t.* $\mathcal{T}$ *and* CR iff *the following holds for every* $T \in \mathfrak{T}(\boldsymbol{a}, \mathsf{CR})$*: if* $T \cup \mathsf{pre}$ *is consistent w.r.t.* $\mathcal{T}$*, then*

- $\mathsf{Eff}(\boldsymbol{a}, T, \mathsf{CR})$ *is not contradictory, and*
- *every model of* $\mathcal{A}_{\mathsf{red}}$*,* $\mathcal{T}_{\mathsf{red}}$*,* $T^{(0)}$*, and* $\mathcal{T}^{(0)}$ *is also a model of* $\mathcal{T}^{(1)}$*, where* $\mathcal{A}_{\mathsf{red}}$ *and* $\mathcal{T}_{\mathsf{red}}$ *are constructed using* $\boldsymbol{b}_{\boldsymbol{a},T,\mathsf{CR}}$ *and* $\mathcal{R}$*.*

Proof. For the 'only if' direction, let $T \in \mathfrak{T}(\boldsymbol{a}, \mathsf{CR})$ be an action type such that $T \cup \mathsf{pre}$ is consistent w.r.t. $\mathcal{T}$. Then there exists an interpretation $\mathcal{I}$ such that $\mathcal{I} \models \mathsf{pre}$, $\mathcal{I} \models T$, and $\mathcal{I} \models \mathcal{T}$. Thus, $\boldsymbol{a}$ is applicable to $\mathcal{I}$ w.r.t. $\mathcal{T}$. Since $\boldsymbol{a}$ is consistent w.r.t. $\mathcal{T}$ and CR, there exists an interpretation $\mathcal{I}'$ such that $\mathcal{I} \Rightarrow^{\mathcal{T},\mathsf{CR}}_{\boldsymbol{a}} \mathcal{I}'$. Hence, $\mathsf{Eff}(\boldsymbol{a}, \mathcal{I}, \mathsf{CR})$ is not contradictory, and since $\mathcal{I} \models T$, we have by Lemma 6.24 that $\mathsf{Eff}(\boldsymbol{a}, \mathcal{I}, \mathsf{CR}) = \mathsf{Eff}(\boldsymbol{a}, T, \mathsf{CR})$, and thus $\mathsf{Eff}(\boldsymbol{a}, T, \mathsf{CR})$ is also not contradictory.

Let $\mathcal{J}$ be a model of $\mathcal{A}_{\mathsf{red}}$, $\mathcal{T}_{\mathsf{red}}$, $T^{(0)}$, and $\mathcal{T}^{(0)}$. We need to show that $\mathcal{J} \models \mathcal{T}^{(1)}$. By (2) of Lemma 6.30, there exist interpretations $\mathcal{I}_0$ and $\mathcal{I}_1$ such that $\mathcal{I}_0 \Rightarrow^{\emptyset,\emptyset}_{\boldsymbol{b}_{\boldsymbol{a},T,\mathsf{CR}}} \mathcal{I}_1$. Since $\mathcal{J}$ is a model of $\mathcal{A}_{\mathsf{red}}$, we have in particular that $\mathcal{J} \models \mathsf{pre}^{(0)}$, and thus we obtain again by (2) of Lemma 6.30 that $\mathcal{I}_0 \models T \cup \mathsf{pre}$, and $\mathcal{I}_0 \models \mathcal{T}$. By Lemma 6.29, $\mathcal{I}_0 \Rightarrow^{\emptyset,\emptyset}_{\boldsymbol{b}_{\boldsymbol{a},T,\mathsf{CR}}} \mathcal{I}_1$ implies $\mathcal{I}_0 \Rightarrow^{\emptyset,\mathsf{CR}}_{\boldsymbol{a}} \mathcal{I}_1$. Assume that $\mathcal{I}_1 \not\models \mathcal{T}$. Since the DL-action $\boldsymbol{a}$ is deterministic, we can conclude that there is no interpretation $\mathcal{I}'$ with $\mathcal{I}_0 \Rightarrow^{\mathcal{T},\mathsf{CR}}_{\boldsymbol{a}} \mathcal{I}'$, which is a contradiction to the assumption that $\boldsymbol{a}$ is consistent w.r.t. $\mathcal{T}$ and CR. Thus, we have $\mathcal{I}_1 \models \mathcal{T}$, which together with (2) of Lemma 6.30 yields that $\mathcal{J} \models \mathcal{T}^{(1)}$.

For the 'if' direction, let $\mathcal{I}$ be any interpretation such that $\boldsymbol{a}$ is applicable to $\mathcal{I}$ w.r.t. $\mathcal{T}$. Then, we have that $\mathcal{I} \models \mathsf{pre}$ and $\mathcal{I} \models \mathcal{T}$. By Lemma 6.20, there is one unique action type $T \in \mathfrak{T}(\boldsymbol{a}, \mathsf{CR})$ such that $\mathcal{I} \models T$, and thus that $T \cup \mathsf{pre}$ is consistent w.r.t. $\mathcal{T}$. Then, $\mathsf{Eff}(\boldsymbol{a}, T, \mathsf{CR})$ is not contradictory, which yields that there exists an interpretation $\mathcal{I}'$ such that $\mathcal{I} \Rightarrow^{\emptyset,\emptyset}_{\boldsymbol{b}_{\boldsymbol{a},T,\mathsf{CR}}} \mathcal{I}'$. Thus, by Lemma 6.29, we have $\mathcal{I} \Rightarrow^{\emptyset,\mathsf{CR}}_{\boldsymbol{a}} \mathcal{I}'$. Moreover, by (1) of Lemma 6.30, there exists an interpretation $\mathcal{J}$ such that $\mathcal{J} \models \mathcal{A}_{\mathsf{red}}$, $\mathcal{J} \models \mathcal{T}_{\mathsf{red}}$, $\mathcal{J} \models T^{(0)}$, and $\mathcal{J} \models \mathcal{T}^{(0)}$. Hence, we have also that $\mathcal{J} \models \mathcal{T}^{(1)}$. Again by (1) of Lemma 6.30, we have that $\mathcal{I}' \models \mathcal{T}$, which, together with $\mathcal{I} \Rightarrow^{\emptyset,\mathsf{CR}}_{\boldsymbol{a}} \mathcal{I}'$ and $\mathcal{I} \models \mathcal{T}$, yields that $\mathcal{I} \Rightarrow^{\mathcal{T},\mathsf{CR}}_{\boldsymbol{a}} \mathcal{I}'$. Thus, $\boldsymbol{a}$ is consistent w.r.t. $\mathcal{T}$ and CR. □

This lemma can be used to design a decision procedure for deciding whether a DL-action is consistent w.r.t. a TBox and finite set of causal relationships. Again, the complexity of this problem depends on the DL used.

Theorem 6.36. *The problem of deciding whether a DL-action is consistent w.r.t. a TBox and a finite set of causal relationships is*

1. ExpTime-*complete for the following DLs:* $\mathcal{ALC}$, $\mathcal{ALCO}$, $\mathcal{ALCQ}$, $\mathcal{ALCI}$, $\mathcal{ALCQO}$, *and* $\mathcal{ALCIO}$*;*
2. ExpTime-*hard and in* co-NExpTime *for* $\mathcal{ALCQI}$*; and*
3. co-NExpTime-*hard and in* $\mathrm{P}^{\mathrm{NExpTime}}$ *for* $\mathcal{ALCQIO}$.

Proof. We first prove the lower bounds of the theorem. As in the case where the TBox was assumed to be empty, we reduce the ABox-*in*consistency problem, i.e. the problem of deciding whether a given ABox is *in*consistent, to our DL-action consistency problem. Take any ABox $\mathcal{A}$ and any general TBox $\mathcal{T}$. It is not hard to see that $\mathcal{A}$ is *in*consistent w.r.t. $\mathcal{T}$ iff $(\mathcal{A}, \{A(a), \neg A(a)\})$ is consistent w.r.t. $\mathcal{T}$ and the empty set of causal relationships, where $A \in \mathsf{N_C}$ and $a \in \mathsf{N_I}$ are arbitrary.

Since the ABox-consistency problem w.r.t. a general TBox is ExpTime-hard for $\mathcal{ALC}$ [Sch91], and ExpTime is closed under complement, we have ExpTime-hardness for Parts 1 and 2 of the theorem. Moreover, co-NExpTime-hardness for Part 3 of the theorem is obtained, because the ABox-consistency problem w.r.t. a general TBox is NExpTime-complete [Sch94; Tob00; Pra05].[10]

To prove the upper bounds for Part 1 of the theorem, we give an ExpTime-decision procedure. Given a DL-action $\boldsymbol{a} = (\mathsf{pre}, \mathsf{post})$, a general TBox $\mathcal{T}$, and a finite set of causal relationships CR, do the following for every action type $T \in \mathfrak{T}(\boldsymbol{a}, \mathsf{CR})$:

1. Check whether the generalised ABox $T \cup \mathsf{pre}$ is consistent w.r.t. $\mathcal{T}$;
2. If Step 1 was successful, compute the set $\mathsf{Eff}(\boldsymbol{a}, T, \mathsf{CR})$, and check whether it is contradictory.
3. If Step 1 and Step 2 were successful, compute the DL-action $\boldsymbol{b}_{\boldsymbol{a},T,\mathsf{CR}}$ and the set of relevant symbols $\mathcal{R}$. Using those, compute the generalised ABox $\mathcal{A}_{\mathsf{red}}$, the TBox $\mathcal{T}_{\mathsf{red}}$, $T^{(0)}$, $\mathcal{T}^{(0)}$, and $\mathcal{T}^{(1)}$, and check whether every model of $\mathcal{A}_{\mathsf{red}}$, $\mathcal{T}_{\mathsf{red}}$, $T^{(0)}$, and $\mathcal{T}^{(0)}$ is also a model of $\mathcal{T}^{(1)}$.

If for every such action type either Step 1 is not successful or Step 3 is successful, we know by Lemma 6.35 that $\boldsymbol{a}$ is consistent w.r.t. $\mathcal{T}$ and CR. Otherwise, $\boldsymbol{a}$ is not consistent w.r.t. $\mathcal{T}$ and CR.

First recall that the set $\mathfrak{T}(\boldsymbol{a}, \mathsf{CR})$ is of size exponential in the size of $\boldsymbol{a}$ and CR, but each action type $T \in \mathfrak{T}(\boldsymbol{a}, \mathsf{CR})$ is only of polynomial size. Thus, it is enough to show that Steps 1–3 can be performed in exponential time. Using the arguments in the proof of Theorem 6.27, the consistency check in Step 1 can be polynomially reduced to the consistency problem of classical ABoxes [BLM+05b]. Since for $\mathcal{ALCIO}$ and $\mathcal{ALCQO}$, the ABox consistency problem w.r.t. general TBoxes can be decided in ExpTime [Sch94; Hla04; HS01], Step 1 can be done in exponential time. As argued above, Step 2 can be done in time polynomial in the size of $\boldsymbol{a}$, T, and CR. For Step 3, note that computing $\boldsymbol{b}_{\boldsymbol{a},T,\mathsf{CR}}$, $\mathcal{R}$, $\mathcal{A}_{\mathsf{red}}$, $\mathcal{T}_{\mathsf{red}}$, $T^{(0)}$, $\mathcal{T}^{(0)}$, and $\mathcal{T}^{(1)}$ can be done in time polynomial in the size of $\boldsymbol{a}$, T, CR, and $\mathcal{T}$. The check in Step 3 can be reduced

[10] As noted in the proof of Theorem 6.27, this is even the case if the number in the at-least and at-most restrictions are coded in binary.

to an ABox-*in*consistency problem w.r.t. a general TBox. Indeed, $\mathcal{T}^{(1)}$ can be transformed to an ABox $\mathcal{A}_{\mathcal{T}^{(1)}}$ as follows: for every GCI $C \sqsubseteq D \in \mathcal{T}^{(1)}$, we add $(C \sqcap \neg D)(a)$ to $\mathcal{A}_{\mathcal{T}^{(1)}}$, where $a \in \mathsf{N_I}$ does not occur in the input. It is not hard to see that the check in Step 3 is equivalent to checking whether the generalised ABox $\mathcal{A}_{\mathsf{red}} \cup T^{(0)} \cup \mathcal{A}_{\mathcal{T}^{(1)}}$ is *in*consistent w.r.t. $\mathcal{T}^{(0)} \cup \mathcal{T}_{\mathsf{red}}$. As shown above, the complement of this problem can be decided in ExpTime, and since ExpTime is closed under complement, Step 3 can also be performed in exponential time. Thus, overall, we obtain an ExpTime-decision procedure.

For the upper bound of Part 2 of the theorem, we employ the same decision procedure. In this case, Step 1 can be done in ExpTime, since the ABox-consistency problem w.r.t. a general TBox is ExpTime-complete [Tob01]. In Step 3, however, we deal with the generalised ABox $\mathcal{A}_{\mathsf{red}}$, which contains nominals. Hence, we obtain with the above reduction, an ABox-*in*consistency problem w.r.t. a general TBox in $\mathcal{ALCQIO}$, which can be done in co-NExpTime as argued above. Since Step 2 can still be done in polynomial time, we obtain overall a co-NExpTime-decision procedure.

For the upper bound of Part 3 of the theorem, we proceed similarly. We first non-deterministically guess an action type $T \in \mathfrak{T}(\boldsymbol{\alpha}, \mathsf{CR})$, and then perform Steps 1–3 of the above decision procedure to decide whether $\boldsymbol{\alpha}$ is *in*consistent w.r.t. $\mathcal{T}$ and CR. As argued above, Step 2 employs a NExpTime-check, and Step 3 employs a co-NExpTime-check. Thus, overall, we obtain that deciding whether $\boldsymbol{\alpha}$ is *in*consistent w.r.t. $\mathcal{T}$ and CR can be done in $\mathrm{NP}^{\mathrm{NExpTime}}$, i.e. in NP using a NExpTime-oracle. Thus deciding whether $\boldsymbol{\alpha}$ is consistent w.r.t. $\mathcal{T}$ and CR can be done in $\text{co-NP}^{\mathrm{NExpTime}}$. It follows from a result in [Hem87] that $\text{co-NP}^{\mathrm{NExpTime}} = \mathrm{NP}^{\mathrm{NExpTime}} = \mathrm{P}^{\mathrm{NExpTime}}$, and thus we obtain Part 3 of the theorem. □

Note, however, that is is still open whether the upper bounds for $\mathcal{ALCQI}$ and $\mathcal{ALCQIO}$ are optimal. In the next section, we consider the projection problem for our DL-based action formalism.

6.3 Deciding the Projection Problem

According to Definition 6.16, the input of the projection problem is a finite sequence of DL-actions $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n$, together with a TBox $\mathcal{T}$, a finite set of causal relationships CR, an initial generalised ABox $\mathcal{A}$, and a generalised ABox-literal α such that every DL-action $\boldsymbol{\alpha}_i$ $(1 \leq i \leq n)$ is *consistent* w.r.t. $\mathcal{T}$ and CR. By definition, α is a consequence of applying $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n$ to $\mathcal{A}$ w.r.t. $\mathcal{T}$ and CR iff for all interpretations $\mathcal{I}_0, \ldots, \mathcal{I}_n$ the following holds: if $\mathcal{I}_0 \models \mathcal{A}$ and $\mathcal{I}_0 \Rightarrow^{\mathcal{T},\mathsf{CR}}_{\boldsymbol{\alpha}_1} \mathcal{I}_1, \ldots, \mathcal{I}_{n-1} \Rightarrow^{\mathcal{T},\mathsf{CR}}_{\boldsymbol{\alpha}_n} \mathcal{I}_n$, then $\mathcal{I}_n \models \alpha$.

Our solution of the projection problem w.r.t. $\mathcal{T}$ and CR uses the same ideas as the solution of the consistency sketched in Section 6.2. Firstly, instead of considering interpretations $\mathcal{I}_0, \ldots, \mathcal{I}_{n-1}$, we consider action types $T_0, \ldots, T_{n-1}$, where $T_i \in \mathfrak{T}(\boldsymbol{\alpha}_{i+1}, \mathsf{CR})$ for $0 \leq i < n$.[11] Secondly, we use the original sequence of DL-actions $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n$, the set of causal relationships CR, and the action types $T_0, \ldots, T_{n-1}$ to construct the corresponding sequence of DL-actions $\boldsymbol{b}_{\boldsymbol{\alpha}_1, T_0, \mathsf{CR}}, \ldots, \boldsymbol{b}_{\boldsymbol{\alpha}_n, T_{n-1}, \mathsf{CR}}$. Lemma 6.29 then tells us that for every i, $0 \leq i < n$, and every model $\mathcal{I}$ of T_i, and every interpretation $\mathcal{I}'$, we have $\mathcal{I} \Rightarrow^{\emptyset,\mathsf{CR}}_{\boldsymbol{\alpha}_{i+1}} \mathcal{I}'$ iff $\mathcal{I} \Rightarrow^{\emptyset,\emptyset}_{\boldsymbol{b}_{\boldsymbol{\alpha}_{i+1}, T_i, \mathsf{CR}}} \mathcal{I}'$. Thirdly, we use the sequence of DL-actions $\boldsymbol{b}_{\boldsymbol{\alpha}_1, T_0, \mathsf{CR}}, \ldots, \boldsymbol{b}_{\boldsymbol{\alpha}_n, T_{n-1}, \mathsf{CR}}$, and the set of relevant

[11] Note that it is enough to consider the action types $T_0, \ldots, T_{n-1}$ for $\mathcal{I}_0, \ldots, \mathcal{I}_{n-1}$ since no DL-action is applied to $\mathcal{I}_n$.

concept names, role names, and concepts $\mathcal{R}$ to construct a generalised ABox $\mathcal{A}_{\mathsf{red}}$ and an (acyclic) TBox $\mathcal{T}_{\mathsf{red}}$ such that the properties (1) and (2) of Lemma 6.30 hold. In this setting, the set $\mathcal{R}$ consists of the concept names, role names, and concepts occurring in $\mathcal{A}$, $\mathcal{T}$, $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$, CR, and α. The properties of $\mathcal{A}_{\mathsf{red}}$ and $\mathcal{T}_{\mathsf{red}}$ can be used to express that the initial interpretation $\mathcal{I}_0$ must be a model of $\mathcal{A}$ and that we only consider successor interpretations $\mathcal{I}_i$ that are models of $\mathcal{T}$. In addition, we can then check, whether all this implies that the final interpretation $\mathcal{I}_n$ is a model of α. To be more precise, we can show that the characterisation of the projection problem stated in the next lemma holds.

Lemma 6.37. *The generalised ABox-literal α is a consequence of applying the finite sequence of DL-actions $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$ to a generalised ABox $\mathcal{A}$ w.r.t. a TBox $\mathcal{T}$ and a finite set of causal relationships* CR iff *we have the following for all action types $T_0, \ldots, T_{n-1}$ with $T_i \in \mathfrak{T}(\boldsymbol{a}_{i+1}, \mathsf{CR})$ for every i, $0 \leq i < n$: every model of $\bigcup_{i=0}^{n-1} T_i^{(i)}$, $\bigcup_{i=0}^{n} \mathcal{T}^{(i)}$, $\mathcal{A}^{(0)}$, $\mathcal{A}_{\mathsf{red}}$, and $\mathcal{T}_{\mathsf{red}}$ is also a model of $\alpha^{(n)}$, where $\mathcal{A}_{\mathsf{red}}$ and $\mathcal{T}_{\mathsf{red}}$ are constructed from $\boldsymbol{b}_{\boldsymbol{a}_1, T_0, \mathsf{CR}}, \ldots, \boldsymbol{b}_{\boldsymbol{a}_n, T_{n-1}, \mathsf{CR}}$ and $\mathcal{R}$.*

Proof. For the 'if' direction, consider action types $T_0, \ldots, T_{n-1}$ with $T_i \in \mathfrak{T}(\boldsymbol{a}_{i+1}, \mathsf{CR})$ for every i, $0 \leq i < n$, and let $\mathcal{J}$ be a model of $\bigcup_{i=0}^{n-1} T_i^{(i)}$, $\bigcup_{i=0}^{n} \mathcal{T}^{(i)}$, $\mathcal{A}^{(0)}$, $\mathcal{A}_{\mathsf{red}}$, and $\mathcal{T}_{\mathsf{red}}$. By (2) of Lemma 6.30, there are interpretations $\mathcal{I}_0, \ldots, \mathcal{I}_n$ such that $\mathcal{I}_i \Rightarrow^{\emptyset,\emptyset}_{\boldsymbol{b}_{\boldsymbol{a}_{i+1}, T_i, \mathsf{CR}}} \mathcal{I}_{i+1}$ for every i, $0 \leq i < n$. Additionally, we have by the same lemma that $\mathcal{I}_i \models T_i$ for every i, $0 \leq i < n$, and $\mathcal{I}_0 \models \mathcal{A}$. Using Lemma 6.29, we obtain $\mathcal{I}_i \Rightarrow^{\emptyset,\mathsf{CR}}_{\boldsymbol{a}_{i+1}} \mathcal{I}_{i+1}$ for every i, $0 \leq i < n$. Since (2) of Lemma 6.30 yields that $\mathcal{I}_i \models \mathcal{T}$ for every i, $0 \leq i \leq n$, we have furthermore $\mathcal{I}_i \Rightarrow^{\mathcal{T},\mathsf{CR}}_{\boldsymbol{a}_{i+1}} \mathcal{I}_{i+1}$. Since α is a consequence of applying $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$ to $\mathcal{A}$ w.r.t. $\mathcal{T}$ and CR, we have that $\mathcal{I}_n \models \alpha$, which implies again by (2) of Lemma 6.30 that $\mathcal{J} \models \alpha^{(n)}$.

For the 'only if' direction, let $\mathcal{I}_0, \ldots, \mathcal{I}_n$ be interpretations such that we have $\mathcal{I}_0 \models \mathcal{A}$ and $\mathcal{I}_i \Rightarrow^{\mathcal{T},\mathsf{CR}}_{\boldsymbol{a}_{i+1}} \mathcal{I}_{i+1}$ for every i, $0 \leq i < n$. It is enough to show that $\mathcal{I}_n \models \alpha$. We have obviously that $\mathcal{I}_i \Rightarrow^{\emptyset,\mathsf{CR}}_{\boldsymbol{a}_{i+1}} \mathcal{I}_{i+1}$ for every i, $0 \leq i < n$, and that $\mathcal{I}_i \models \mathcal{T}$ for every i, $0 \leq i \leq n$. By Lemma 6.20, there are unique action types $T_i \in \mathfrak{T}(\boldsymbol{a}_{i+1}, \mathsf{CR})$ such that $\mathcal{I}_i \models T_i$ for every i, $0 \leq i < n$. Then, by Lemma 6.29, we have $\mathcal{I}_i \Rightarrow^{\emptyset,\emptyset}_{\boldsymbol{b}_{\boldsymbol{a}_{i+1}, T_i, \mathsf{CR}}} \mathcal{I}_{i+1}$ for every i, $0 \leq i < n$. Thus, by (1) of Lemma 6.30, there exists an interpretation $\mathcal{J}$ such that $\mathcal{J}$ is a model of $\mathcal{A}_{\mathsf{red}}$, $\mathcal{T}_{\mathsf{red}}$, $\bigcup_{i=0}^{n-1} T_i^{(i)}$, $\bigcup_{i=0}^{n} \mathcal{T}^{(i)}$, and $\mathcal{A}^{(0)}$. Thus, we have also $\mathcal{J} \models \alpha^{(n)}$, which implies again by (1) of Lemma 6.30 that $\mathcal{I}_n \models \alpha$. □

It is easy to see that this lemma directly yields a decision procedure for the projection problem. Again, the exact complexity of this problem depends on the DL used, and the fact whether the TBox is assumed to be acyclic (or empty) or not.

Theorem 6.38. *The projection problem for our action formalism is*

1. ExpTime*-complete for the DLs $\mathcal{ALC}$, $\mathcal{ALCO}$, $\mathcal{ALCI}$, $\mathcal{ALCIO}$, $\mathcal{ALCQ}$, and $\mathcal{ALCQO}$; and*
2. co-NExpTime*-complete for the DLs $\mathcal{ALCQI}$ and $\mathcal{ALCQIO}$.*

Moreover, if the TBox is assumed to be acyclic (or empty), the projection problem is

3. PSpace*-complete for the DLs $\mathcal{ALC}$, $\mathcal{ALCO}$, $\mathcal{ALCQ}$, and $\mathcal{ALCQO}$;*
4. ExpTime*-complete for the DLs $\mathcal{ALCI}$ and $\mathcal{ALCIO}$; and*

5. co-NExpTime-*complete for the DLs $\mathcal{ALCQI}$ and $\mathcal{ALCQIO}$.*

Proof. We first prove the lower bounds of Part 1 of the theorem by reducing the *un*satisfiability problem, i.e. the problem of deciding whether a given concept is *un*satisfiable, to the projection problem. It is not hard to see that a concept C is unsatisfiable w.r.t. a TBox $\mathcal{T}$ iff $\neg C(a)$ is a consequence of applying the DL-action $(\emptyset, \emptyset)$ to the ABox $\emptyset$ w.r.t. $\mathcal{T}$ and the empty set of causal relationships, where $a \in \mathsf{N_I}$ does not occur in C or $\mathcal{T}$. Satisfiability of a concept w.r.t. a general TBox is ExpTime-complete in $\mathcal{ALC}$ [Sch91], and since ExpTime is closed under complement, we obtain the lower bounds of Part 1 of the theorem.

For the remaining lower bounds, we reduce the projection problem of [BLM+05a] to our projection problem. In the case where the TBox is assumed to be empty, there is only one difference between the 'transforms' relation from Definition 6.13 and the one in [BLM+05a]: we demand here that the DL-action is applicable to the interpretation, i.e. a model of the preconditions of the DL-action. Therefore, we have the following: The generalised ABox-literal α is a consequence of applying the DL-action $(\mathsf{pre}, \mathsf{post})$ to a generalised ABox $\mathcal{A}$ w.r.t. the empty TBox (as defined in [BLM+05a]) *iff* α is a consequence of applying $(\emptyset, \mathsf{post})$ to $\mathcal{A}$ w.r.t. the empty TBox and the empty set of causal relationships. The projection problem defined in [BLM+05a] is PSpace-complete for $\mathcal{ALC}$, ExpTime-complete for $\mathcal{ALCI}$, and co-NExpTime-complete for $\mathcal{ALCQI}$, even if the TBox is assumed to be empty and we deal with only one DL-action [BLM+05a]. Thus, we obtain the remaining lower bounds of our theorem.[12]

For the upper bounds of Parts 1 and 4 of the theorem, we give an ExpTime-decision procedure. Given a generalised ABox-literal α, a finite sequence of DL-actions $\boldsymbol{a}_1, \dots, \boldsymbol{a}_n$, a generalised ABox $\mathcal{A}$, a TBox $\mathcal{T}$ and a finite set of causal relationships CR, we do the following for all action types $T_0, \dots, T_{n-1}$ with $T_i \in \mathfrak{T}(\boldsymbol{a}_{i+1}, \mathsf{CR})$ for every i, $0 \leq i < n$:

1. Construct $\boldsymbol{b}_{\boldsymbol{a}_1,T_0,\mathsf{CR}}, \dots, \boldsymbol{b}_{\boldsymbol{a}_n,T_{n-1},\mathsf{CR}}$, $\mathcal{R}$, $\bigcup_{i=0}^{n-1} T_i^{(i)}$, $\bigcup_{i=0}^{n} \mathcal{T}^{(i)}$, $\mathcal{A}^{(0)}$, $\mathcal{A}_{\mathsf{red}}$, $\mathcal{T}_{\mathsf{red}}$, and $\alpha^{(n)}$, where $\mathcal{A}_{\mathsf{red}}$ and $\mathcal{T}_{\mathsf{red}}$ are constructed from $\boldsymbol{b}_{\boldsymbol{a}_1,T_0,\mathsf{CR}}, \dots, \boldsymbol{b}_{\boldsymbol{a}_n,T_{n-1},\mathsf{CR}}$ and $\mathcal{R}$.
2. Check whether every model of $\bigcup_{i=0}^{n-1} T_i^{(i)}$, $\bigcup_{i=0}^{n} \mathcal{T}^{(i)}$, $\mathcal{A}^{(0)}$, $\mathcal{A}_{\mathsf{red}}$, and $\mathcal{T}_{\mathsf{red}}$ is also a model of $\alpha^{(n)}$.

If Step 2 is successful, then we know by Lemma 6.37 that α is a consequence of applying $\boldsymbol{a}_1, \dots, \boldsymbol{a}_n$ to $\mathcal{A}$ w.r.t. $\mathcal{T}$ and CR. Note that there are exponentially many sequences of action types to consider. Also, observe that Step 1 can be done in polynomial time, and thus the input to the reasoning problem of Step 2 is of polynomial size. Using the arguments in the proof of Theorem 6.36, we can reduce this reasoning problem to an ABox-*in*consistency problem. Since the ABox-consistency problem w.r.t. a general TBox is ExpTime-complete in $\mathcal{ALCIO}$ [Sch94; Hla04] and $\mathcal{ALCQO}$ [Sch94; HS01], and ExpTime is closed under complement, we obtain the upper bounds of Parts 1 and 4 of the theorem.

For the upper bounds of Parts 2 and 5 of the theorem, we consider the complement of the projection problem. For that, we first non-deterministically guess a sequence of action types $T_0, \dots, T_{n-1}$, perform Step 1 of the above decision procedure, and then perform the *complement* of Step 2 of the above decision procedure, i.e. we check whether there is a model

[12] We could obtain most of the lower bounds also by a reduction of the unsatisfiability problem as for Part 1 of the theorem. However, we would not get the lower bounds for $\mathcal{ALCI}$ (in the case where the TBox is assumed to be empty) and $\mathcal{ALCQI}$ as the satisfiability problem in $\mathcal{ALCQI}$ w.r.t. the empty TBox is PSpace-complete [Tob01], and the satisfiability problem in $\mathcal{ALCQI}$ w.r.t. a general TBox is ExpTime-complete [Tob01].

of $\bigcup_{i=0}^{n-1} T_i^{(i)}$, $\bigcup_{i=0}^{n} \mathcal{T}^{(i)}$, $\mathcal{A}^{(0)}$, $\mathcal{A}_{\mathsf{red}}$, $\mathcal{T}_{\mathsf{red}}$, and $\neg\alpha^{(n)}$, which is clearly an ABox-consistency problem. Since the ABox-consistency problem w.r.t. a general TBox is NExpTime-complete for $\mathcal{ALCQIO}$ [Sch94; Tob00; Pra05], we obtain a NExpTime-decision procedure for the complement of the projection problem, and thus a co-NExpTime-decision procedure for the projection problem, which proves the upper bounds of Parts 2 and 5 of the theorem.

It is left to show the upper bounds of Part 3 of the theorem. For that, it suffices to give an NPSpace-decision procedure for the complement of the projection problem for the case of $\mathcal{ALCQO}$ w.r.t. an acyclic TBox.[13] We use the same decision procedure as for the upper bounds of Parts 2 and 5 of the theorem. Note that the ABox-consistency problem of Step 2 of the above decision procedure does not contain any GCIs, as $\mathcal{T}_{\mathsf{red}}$ is acyclic. Since the ABox-consistency problem w.r.t. acyclic TBoxes is PSpace-complete for $\mathcal{ALCQO}$ [Sch94; BLM+05b], we obtain an NPSpace-decision procedure for the complement of the projection problem, and thus we obtain the upper bounds of Part 3 of the theorem. □

This finishes the section on the projection problem, and in the next section, we use the results obtained so far for verifying properties of infinite sequences of DL-actions.

6.4 Verification of DL-Actions

In this section, we show how to verify temporal properties in the DL-based action formalism described in Section 6.1. For that, we follow the approach in [BLM10]. The principle idea is that a Büchi-automaton defines infinite sequences of DL-actions that characterise which DL-actions an agent may execute. Then, we verify whether a temporal property is satisfied. This temporal property is encoded in a restricted $\mathcal{L}$-LTL-formula where $\mathcal{L}$ is a DL between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$.[14] The $\mathcal{L}$-LTL-formulas considered in this section are restricted in the following sense: whereas for $\mathcal{SHOQ}$-LTL, formulas could contain GCIs, we do not allow this here (see Chapter 3). This restriction is in accordance with [BLM10], where it was also made. In principle, we do not need this restriction here, but from an application point of view, it makes sense to encode the domain knowledge in a global TBox and assume that it does not change as done in the previous sections for the projection problem.

Moreover, the $\mathcal{L}$-LTL-formulas in this section contain only assertions, whereas in [BLM10] they contain arbitrary generalised ABox-literals. This is, however, not a restriction since the generalised ABox-literal $\neg\alpha$ can be equivalently expressed by the $\mathcal{L}$-LTL-formula $\neg(\alpha)$. In spite of these restrictions, we refer to the formulas considered in this section as $\mathcal{L}$-LTL-formulas for simplicity. We are now ready to introduce the problems that we consider in this section.

Definition 6.39 (Verification problem). *Let $\mathcal{L}$ be a DL between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$. Furthermore, let $\mathcal{A}$ be a generalised $\mathcal{L}$-ABox, $\mathcal{T}$ be an $\mathcal{L}$-TBox,* CR *be a finite set of causal relationships,* $\boldsymbol{A}$ *be a finite set of DL-actions, $\mathcal{N} = (Q, \boldsymbol{A}, \Delta, Q_0, F)$ be a Büchi-automaton, and ϕ be an $\mathcal{L}$-LTL-formula.*

[13] Recall again that Savitch's theorem [Sav70] implies that NPSpace and PSpace coincide, and that PSpace is closed under complement.

[14] Note that in Chapter 3, we introduced $\mathcal{SHOQ}$-LTL and its fragments. So far we have not considered *inverse roles*. It is, however, quite clear how the syntax and semantics of $\mathcal{L}$-LTL is defined for $\mathcal{L}$ being a DL between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$ that involves inverse roles. Therefore, we use the notation from Chapter 3 such as the propositional abstraction etc. also here.

We say that ϕ *is* valid w.r.t. $\mathcal{A}$, $\mathcal{T}$, CR, and $\mathcal{N}$ *if for every infinite DL-action sequence* $\boldsymbol{a}_1\boldsymbol{a}_2\ldots \in L_\omega(\mathcal{N})$*, and every DL-LTL-structure* $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$ *with* $\mathcal{I}_0 \models \mathcal{A}$ *and* $\mathcal{I}_i \Rightarrow^{\mathcal{T},\mathsf{CR}}_{\boldsymbol{a}_{i+1}} \mathcal{I}_{i+1}$ *for each* $i \geq 0$*, we have* $\mathfrak{I}, 0 \models \phi$*.*

Moreover, we say that ϕ *is* satisfiable w.r.t. $\mathcal{A}$, $\mathcal{T}$, CR, and $\mathcal{N}$ *if there is an infinite DL-action sequence* $\boldsymbol{a}_1\boldsymbol{a}_2\ldots \in L_\omega(\mathcal{N})$*, and a DL-LTL-structure* $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$ *with* $\mathcal{I}_0 \models \mathcal{A}$ *and* $\mathcal{I}_i \Rightarrow^{\mathcal{T},\mathsf{CR}}_{\boldsymbol{a}_{i+1}} \mathcal{I}_{i+1}$ *for each* $i \geq 0$ *such that we have* $\mathfrak{I}, 0 \models \phi$*.*

The verification problem (satisfiability problem) *is then to decide whether* ϕ *is valid (satisfiable) w.r.t.* $\mathcal{A}$*,* $\mathcal{T}$*,* CR*, and* $\mathcal{N}$*.* ◇

It is easy to see that ϕ is valid w.r.t. $\mathcal{A}$, $\mathcal{T}$, CR, and $\mathcal{N}$ iff $\neg\phi$ is *un*satisfiable w.r.t. $\mathcal{A}$, $\mathcal{T}$, CR, and $\mathcal{N}$. Conversely, ϕ is satisfiable w.r.t. $\mathcal{A}$, $\mathcal{T}$, CR, and $\mathcal{N}$ iff $\neg\phi$ is *not* valid w.r.t. $\mathcal{A}$, $\mathcal{T}$, CR, and $\mathcal{N}$. Hence, the verification problem and the *un*satisfiability problem have the same complexity. The complexity of these problems is investigated in [BLM10] for the case where the TBox is assumed to be *acyclic* and no causal relationships are available. These complexity results depend on the DL used. The following result is proved in [BLM10; BLM09].

Proposition 6.40. *If the TBox is assumed to be acyclic and the set of causal relationships* CR *is assumed to be empty, the verification problem is*

1. *in* ExpSpace *for the DLs* $\mathcal{ALC}$*,* $\mathcal{ALCO}$*,* $\mathcal{ALCQ}$*, and* $\mathcal{ALCQO}$*;*
2. *in* 2ExpTime *for the DLs* $\mathcal{ALCI}$ *and* $\mathcal{ALCIO}$*; and*
3. *in* co-2NExpTime *for the DLs* $\mathcal{ALCQI}$ *and* $\mathcal{ALCQIO}$*.* ◇

Unfortunately, no tight lower bounds of this problem are known. Using similar ideas, we show now the upper bounds of the verification problem in the case where the TBox is arbitrary and a finite set of causal relationships is present.

From now on, let $\mathcal{L}$ be a description logic between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$, let $\mathcal{A}$ be a generalised $\mathcal{L}$-ABox, let $\mathcal{T}$ be an $\mathcal{L}$-TBox, let CR be a finite set of causal relationships, let $\boldsymbol{A} = \{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n\}$ be a finite set of DL-actions, let $\mathcal{N} = (Q, \boldsymbol{A}, \Delta, Q_0, F)$ be a Büchi-automaton, and let ϕ be an $\mathcal{L}$-LTL-formula. First observe that we can assume without loss of generality that $\mathcal{A}$ is empty by using an argument similar to the one used in the proof of Lemma 5.27, namely, we compile $\mathcal{A}$ into ϕ. More precisely, it is easy to verify that we have that ϕ is valid (satisfiable) w.r.t. $\mathcal{A}$, $\mathcal{T}$, CR, and $\mathcal{N}$ iff $\phi \wedge \bigwedge \mathcal{A}$ is valid (satisfiable) w.r.t. $\emptyset$, $\mathcal{T}$, CR, and $\mathcal{N}$.

We furthermore assume without loss of generality that every axiom occurring in a DL-action of $\boldsymbol{A}$ or a causal relationship of CR also occurs in ϕ. Moreover, we assume without loss of generality that for every $A \in \mathsf{N_C}$, $r \in \mathsf{N_R}$, $a, b \in \mathsf{N_I}$ occurring in ϕ, $\mathcal{T}$, a DL-action of $\boldsymbol{A}$, or a causal relationship of CR, we have that the assertions $A(a)$ and $r(a,b)$ also occur in ϕ. These two assumptions are indeed without loss of generality since for every such axiom α, which does not occur in ϕ, we can define $\phi' := \phi \wedge (\alpha \vee \neg\alpha)$. Obviously, every model of ϕ is also a model of ϕ', and vice versa.

To solve the satisfiability problem, we combine the approach of Chapter 3 with the one of Sections 6.2 and 6.3. We again split the problem in two sub-problems. First note that Lemma 3.13 also holds for the restricted $\mathcal{L}$-LTL-formulas that we consider in this section. Let $\mathsf{p}\colon \mathsf{Ax}(\phi) \to \mathcal{P}_\phi$ be a bijection. Hence, we know that ϕ is satisfiable iff there is a set $\mathcal{W} \subseteq 2^{\mathcal{P}_\phi}$ such that $\mathcal{W}$ is r-satisfiable (see Definition 3.10) and ϕ^{p} is t-satisfiable w.r.t. $\mathcal{W}$ (see Definition 3.11). However, we need to adapt both sub-problems due to the semantics of DL-actions.

We first consider how to decide t-satisfiability w.r.t. a given set $\mathcal{W}$. For that, we assume from now on that $\mathcal{W} = \{X_1, \ldots, X_k\} \subseteq 2^{\mathcal{P}_\phi}$ is given. By Lemma 3.14, we can decide whether ϕ^{p} is t-satisfiable w.r.t. $\mathcal{W}$ in time exponential in the size of ϕ^{p} and linear in the size of $\mathcal{W}$. In the proof of Lemma 3.14, a Büchi-automaton for $\phi^{\mathsf{p}}_{\mathcal{W}}$ is constructed. Let $\mathcal{N}_{\phi^{\mathsf{p}}_{\mathcal{W}}}$ be that Büchi-automaton.

However, $\mathcal{N}_{\phi^{\mathsf{p}}_{\mathcal{W}}}$ may accept a sequence of worlds that does not correspond to a sequence of DL-actions defined by $\mathcal{N}$. Thus, we need to intersect the ω-language accepted by $\mathcal{N}_{\phi^{\mathsf{p}}_{\mathcal{W}}}$ with the one 'generated' by the ω-language accepted by $\mathcal{N}$. For that, we construct a Büchi-automaton $\widehat{\mathcal{N}}$ by considering the effects of the DL-actions occurring in an accepting run of $\mathcal{N}$. In order to be able to enforce that the semantics of DL-actions is respected, it is essential that the worlds contain information about all $a, b \in \mathsf{N_I}$, $A \in \mathsf{N_C}$, and $r \in \mathsf{N_R}$ occurring in the input whether $A(a)$ $(r(a, b))$ holds or not.

To be able to specify the effects of a DL-action in $\widehat{\mathcal{N}}$, we need to keep track of the action type associated with a world. For that, we define for $\sigma \in \Sigma_{\mathcal{P}_\phi}$, the generalised ABox $T_{\sigma,\boldsymbol{a},\mathsf{CR}} \subseteq \mathsf{Cond}(\boldsymbol{a},\mathsf{CR})$ as follows:

- for every positive generalised ABox-literal $\alpha \in \mathsf{Cond}(\boldsymbol{a},\mathsf{CR})$, we have $\alpha \in T_{\sigma,\boldsymbol{a},\mathsf{CR}}$ iff $\mathsf{p}(\alpha) \in \sigma$; and
- for every negative generalised ABox-literal $\neg\alpha \in \mathsf{Cond}(\boldsymbol{a},\mathsf{CR})$, we have $\neg\alpha \in T_{\sigma,\boldsymbol{a},\mathsf{CR}}$ iff $\mathsf{p}(\alpha) \notin \sigma$.

We can now define the Büchi-automaton

$$\widehat{\mathcal{N}} := (Q \times \boldsymbol{A} \times \Sigma_{\mathcal{P}_\phi},\ \Sigma_{\mathcal{P}_\phi},\ \widehat{\Delta},\ Q_0 \times \boldsymbol{A} \times \Sigma_{\mathcal{P}_\phi},\ F \times \boldsymbol{A} \times \Sigma_{\mathcal{P}_\phi}),$$

where we have $((q, \boldsymbol{a}, \sigma),\ \sigma'',\ (q', \boldsymbol{a}', \sigma')) \in \widehat{\Delta}$ iff

- $(q, \boldsymbol{a}, q') \in \Delta$;
- $\sigma = \sigma''$;
- for every positive generalised ABox-literal $\alpha \in \mathsf{pre}$, we have $\mathsf{p}(\alpha) \in \sigma$, and for every negative generalised ABox-literal $\neg\alpha \in \mathsf{pre}$, we have $\mathsf{p}(\alpha) \notin \sigma$, where $\boldsymbol{a} = (\mathsf{pre}, \mathsf{post})$;
- $T_{\sigma,\boldsymbol{a},\mathsf{CR}}$ is an action type for $\boldsymbol{a}$ and CR;
- for every positive ABox-literal $\beta \in \mathsf{Eff}(\boldsymbol{a}, T_{\sigma,\boldsymbol{a},\mathsf{CR}}, \mathsf{CR})$, we have $\mathsf{p}(\beta) \in \sigma'$, and for every negative ABox-literal $\neg\beta \in \mathsf{Eff}(\boldsymbol{a}, T_{\sigma,\boldsymbol{a},\mathsf{CR}}, \mathsf{CR})$, we have $\mathsf{p}(\beta) \notin \sigma'$; and
- for every ABox-literal $\gamma \in \mathsf{Ax}(\phi)$, we have:
  - if $\mathsf{p}(\gamma) \in \sigma$ and $\neg\gamma \notin \mathsf{Eff}(\boldsymbol{a}, T_{\sigma,\boldsymbol{a},\mathsf{CR}}, \mathsf{CR})$, we have $\mathsf{p}(\gamma) \in \sigma'$, and
  - if $\mathsf{p}(\gamma) \notin \sigma$ and $\gamma \notin \mathsf{Eff}(\boldsymbol{a}, T_{\sigma,\boldsymbol{a},\mathsf{CR}}, \mathsf{CR})$, we have $\mathsf{p}(\gamma) \notin \sigma'$.

Let $\widehat{\mathcal{N}}_{\phi^{\mathsf{p}}_{\mathcal{W}}}$ denote the Büchi-automaton that accepts the intersection of the ω-language accepted by the Büchi-automaton $\mathcal{N}_{\phi^{\mathsf{p}}_{\mathcal{W}}}$ and the ω-language accepted by the Büchi-automaton $\widehat{\mathcal{N}}$. Such a Büchi-automaton can be obtained using the standard product construction in time polynomial in the size of the input Büchi-automata, see e.g. [BK08; Tho90].

In order to be sure that ω-words accepted by $\widehat{\mathcal{N}}_{\phi^{\mathsf{p}}_{\mathcal{W}}}$ can indeed be 'lifted' to DL-LTL-structures, we need to check whether $\mathcal{W}$ is r-satisfiable. However, to make sure that the semantics of the DL-actions is satisfied, we need to intertwine this check with another one. Indeed, so far we only dealt with the *named* part of the interpretations in a DL-LTL-structure. Additionally, we need to ensure that the *unnamed* part of the interpretations in a DL-LTL-structure remains unchanged. Recall that the TBox $\mathcal{T}_{\mathsf{red}}$ defined in Section 6.2 was designed to take care of this matter. We use a very similar TBox $\mathcal{T}_{\mathsf{red}}$ also here.

In the following, let again $\mathcal{R}$ denote a set of relevant concept names, role names, and concepts such that $\mathcal{R}$ contains all concept names, role names, and concepts occurring in ϕ. Moreover, let Obj denote the set of individual names occurring in ϕ.

We introduce time-stamped copies $A^{(i)}$ ($r^{(i)}$), $0 \leq i \leq k$, of all concept names $A \in \mathcal{R}$ (role names $r \in \mathcal{R}$), and new time-stamped concept names $T_C^{(i)}$, $1 \leq i \leq k$, of all concepts $C \in \mathcal{R}$. Now, the TBox $\mathcal{T}_{\mathsf{red}}$ again consists of two parts, i.e. $\mathcal{T}_{\mathsf{red}} := \mathcal{T}_N \cup \mathcal{T}_{\mathsf{sub}}$. The TBox $\mathcal{T}_N$ is defined as in Section 6.2, and the TBox $\mathcal{T}_{\mathsf{sub}}$ consists of a concept definition of $T_C^{(i)}$ for every concept $C \in \mathcal{R}$ and every i, $0 \leq i \leq k$, where the concept definition of $T_C^{(i)}$ is defined inductively as in Section 6.2.

Moreover, we define the generalised ABox $\mathcal{A}_{\mathcal{W}}$ as follows:

$$\mathcal{A}_{\mathcal{W}} := \bigcup_{i=1}^{k} \Big(\big\{ (\mathsf{p}^{-1}(p))^{(i)} \mid p \in X_i \big\} \cup \big\{ (\neg \mathsf{p}^{-1}(p))^{(i)} \mid p \in \mathcal{P}_\phi \setminus X_i \big\} \Big),$$

where the time-stamped variant $\alpha^{(i)}$ of an axiom $\alpha \in \mathsf{Ax}(\phi)$ is defined as in Section 6.2.

Additionally, we need to ensure that the TBox $\mathcal{T}$ is respected. For that, we construct copies $\mathcal{T}^{(i)}$, $1 \leq i \leq k$. Now, the following lemma states how the Büchi-automaton $\widehat{\mathcal{N}}_{\phi^{\mathsf{p}}_{\mathcal{W}}}$, the ABox $\mathcal{A}_{\mathcal{W}}$, the TBoxes $\mathcal{T}^{(i)}$, $1 \leq i \leq k$, and the TBox $\mathcal{T}_{\mathsf{red}}$ can be used to solve the satisfiability problem.

Lemma 6.41. *The $\mathcal{L}$-LTL-formula ϕ is satisfiable w.r.t. $\emptyset$, $\mathcal{T}$, CR, and $\mathcal{N}$ iff there is a set $\mathcal{W} = \{X_1, \ldots, X_k\} \subseteq 2^{\mathcal{P}_\phi}$ such that*

- *$L_\omega(\widehat{\mathcal{N}}_{\phi^{\mathsf{p}}_{\mathcal{W}}}) \neq \emptyset$, and*
- *$\mathcal{A}_{\mathcal{W}}$ has a model w.r.t. $\mathcal{T}_{\mathsf{red}} \cup \bigcup_{i=1}^{k} \mathcal{T}^{(i)}$.*

Proof. For the 'only if' direction, assume that there is an infinite sequence of DL-actions $\boldsymbol{a}_1\boldsymbol{a}_2 \ldots \in L_\omega(\mathcal{N})$, and a DL-LTL-structure $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$ with $\mathcal{I}_i \Rightarrow^{\mathcal{T},\mathsf{CR}}_{\boldsymbol{a}_{i+1}} \mathcal{I}_{i+1}$ for each $i \geq 0$ such that we have $\mathfrak{I}, 0 \models \phi$. Let $\mathfrak{I}^{\mathsf{p}} = (w_i)_{i\geq 0}$ be the propositional abstraction of $\mathfrak{I}$ w.r.t. p, and let $\mathcal{W} := \{w_i \mid i \geq 0\} = \{X_1, \ldots, X_k\} \subseteq 2^{\mathcal{P}_\phi}$. By Lemma 3.12, we have that $\mathfrak{I}^{\mathsf{p}}$ is a model of ϕ^{p}. By construction of $\mathcal{W}$, we have also that $\mathfrak{I}^{\mathsf{p}}$ is a model of $\phi^{\mathsf{p}}_{\mathcal{W}}$. Since $\mathcal{N}_{\phi^{\mathsf{p}}_{\mathcal{W}}}$ is a Büchi-automaton for $\phi^{\mathsf{p}}_{\mathcal{W}}$ (which is the one constructed in the proof of Lemma 3.14), we have that $w := w_0w_1 \ldots \in L_\omega(\mathcal{N}_{\phi^{\mathsf{p}}_{\mathcal{W}}})$.

To show that $w \in L_\omega(\widehat{\mathcal{N}}_{\phi^{\mathsf{p}}_{\mathcal{W}}})$, it remains to prove that $w \in L_\omega(\widehat{\mathcal{N}})$ as $\widehat{\mathcal{N}}_{\phi^{\mathsf{p}}_{\mathcal{W}}}$ accepts the intersection of the ω-language accepted by $\mathcal{N}_{\phi^{\mathsf{p}}_{\mathcal{W}}}$ and the ω-language accepted by $\widehat{\mathcal{N}}$. Since $\boldsymbol{a}_1\boldsymbol{a}_2 \ldots \in L_\omega(\mathcal{N})$, there is an accepting run $q_0q_1 \ldots$ of $\mathcal{N}$ on $\boldsymbol{a}_1\boldsymbol{a}_2 \ldots$. Then,

$$(q_0, \boldsymbol{a}_1, w_0)(q_1, \boldsymbol{a}_2, w_1)(q_2, \boldsymbol{a}_3, w_2) \ldots$$

is an accepting run of $\widehat{\mathcal{N}}$ on w due to the following reasons:

- Obviously, we have that for every $i \geq 0$ that $(q_i, \boldsymbol{a}_{i+1}, w_i)$ is a state of $\widehat{\mathcal{N}}$.

- We have for every $i \geq 0$ that $((q_i, \boldsymbol{a}_{i+1}, w_i),\ w_i,\ (q_{i+1}, \boldsymbol{a}_{i+2}, w_{i+1})) \in \widehat{\Delta}$ since the following holds:
  - We have $(q_i, \boldsymbol{a}_{i+1}, q_{i+1}) \in \Delta$ by construction.
  - The condition that w_i is the last component of the tuple $(q_i, \boldsymbol{a}_{i+1}, w_i)$ is also satisfied.
  - Since $\mathcal{I}_i \Rightarrow^{\mathcal{T},\mathsf{CR}}_{\boldsymbol{a}_{i+1}} \mathcal{I}_{i+1}$, we have that $\boldsymbol{a}_{i+1} = (\mathsf{pre}_{i+1}, \mathsf{post}_{i+1})$ is applicable to $\mathcal{I}_i$ w.r.t. $\mathcal{T}$. Thus, $\mathcal{I}_i \models \mathsf{pre}_{i+1}$. Since we assumed that all axioms occurring in pre_{i+1} also occur in ϕ, for every positive generalised ABox-literal $\alpha \in \mathsf{pre}_{i+1}$, we have that $\mathsf{p}(\alpha) \in w_i$. Likewise, for every negative generalised ABox-literal $\neg\alpha \in \mathsf{pre}_{i+1}$, we have $\mathsf{p}(\alpha) \notin w_i$.
  - Since we assumed that all axioms occurring in $\boldsymbol{a}_{i+1}$ and CR also occur in ϕ, we have, by the definition of $T_{w_i,\boldsymbol{a}_{i+1},\mathsf{CR}}$, for every negative generalised ABox-literal $\neg\alpha \in \mathsf{Cond}(\boldsymbol{a}_{i+1}, \mathsf{CR})$, that $\neg\alpha \in T_{w_i,\boldsymbol{a}_{i+1},\mathsf{CR}}$ iff $\mathsf{p}(\alpha) \notin w_i$ iff $\alpha \notin T_{w_i,\boldsymbol{a}_{i+1},\mathsf{CR}}$. Moreover, we have $\mathcal{I}_i \models T_{w_i,\boldsymbol{a}_{i+1},\mathsf{CR}}$, and hence $T_{w_i,\boldsymbol{a}_{i+1},\mathsf{CR}}$ is an action type for $\boldsymbol{a}_{i+1}$ and CR.
  - Since $\mathcal{I}_i \Rightarrow^{\mathcal{T},\mathsf{CR}}_{\boldsymbol{a}_{i+1}} \mathcal{I}_{i+1}$, we have by Lemma 6.14 that $\mathcal{I}_{i+1} \models \mathsf{Eff}(\boldsymbol{a}_{i+1}, \mathcal{I}_i, \mathsf{CR})$. By Lemma 6.24, we obtain further that $\mathsf{Eff}(\boldsymbol{a}_{i+1}, \mathcal{I}_i, \mathsf{CR}) = \mathsf{Eff}(\boldsymbol{a}_{i+1}, T_{w_i,\boldsymbol{a}_{i+1},\mathsf{CR}}, \mathsf{CR})$. Thus, we have that $\mathcal{I}_{i+1} \models \mathsf{Eff}(\boldsymbol{a}_{i+1}, T_{w_i,\boldsymbol{a}_{i+1},\mathsf{CR}}, \mathsf{CR})$. Since we assumed that all axioms occurring in $\boldsymbol{a}_{i+1}$ and CR also occur in ϕ, for every positive ABox-literal $\beta \in \mathsf{Eff}(\boldsymbol{a}_{i+1}, T_{w_i,\boldsymbol{a}_{i+1},\mathsf{CR}}, \mathsf{CR})$, we have that $\mathsf{p}(\beta) \in w_{i+1}$. Likewise, for every negative ABox-literal $\neg\beta \in \mathsf{Eff}(\boldsymbol{a}_{i+1}, T_{w_i,\boldsymbol{a}_{i+1},\mathsf{CR}}, \mathsf{CR})$, we have $\mathsf{p}(\beta) \notin w_{i+1}$.
  - For the last condition, take first any ABox-literal $\gamma \in \mathsf{Ax}(\phi)$ with $\mathsf{p}(\gamma) \in w_i$ and $\neg\gamma \notin \mathsf{Eff}(\boldsymbol{a}_{i+1}, T_{w_i,\boldsymbol{a}_{i+1},\mathsf{CR}}, \mathsf{CR})$. Thus, $\mathcal{I}_i \models \gamma$. We prove $\mathcal{I}_{i+1} \models \gamma$ by a case distinction. If γ is of the form $A(a)$ for $A \in \mathsf{N_C}$ and $a \in \mathsf{N_I}$, we have $a^{\mathcal{I}_i} \in A^{\mathcal{I}_i}$. Since $\mathcal{I}_i \Rightarrow^{\mathcal{T},\mathsf{CR}}_{\boldsymbol{a}_{i+1}} \mathcal{I}_{i+1}$, this yields $a^{\mathcal{I}_i} \in A^{\mathcal{I}_{i+1}}$, and thus $\mathcal{I}_{i+1} \models A(a)$. Otherwise, if γ is of the form $r(a,b)$ for $r \in \mathsf{N_R}$ and $a, b \in \mathsf{N_I}$, we have $(a^{\mathcal{I}_i}, b^{\mathcal{I}_i}) \in r^{\mathcal{I}_i}$. Since $\mathcal{I}_i \Rightarrow^{\mathcal{T},\mathsf{CR}}_{\boldsymbol{a}_{i+1}} \mathcal{I}_{i+1}$, this yields $(a^{\mathcal{I}_i}, b^{\mathcal{I}_i}) \in r^{\mathcal{I}_{i+1}}$, and thus $\mathcal{I}_{i+1} \models r(a,b)$. Overall, we obtain $\mathsf{p}(\gamma) \in w_{i+1}$.

    For the second case, take any ABox-literal $\gamma \in \mathsf{Ax}(\phi)$ with $\mathsf{p}(\gamma) \notin w_i$ and $\gamma \notin \mathsf{Eff}(\boldsymbol{a}_{i+1}, T_{w_i,\boldsymbol{a}_{i+1},\mathsf{CR}}, \mathsf{CR})$. Thus, $\mathcal{I}_i \not\models \gamma$. We prove $\mathcal{I}_{i+1} \not\models \gamma$ again by a case distinction. If γ is of the form $A(a)$ for $A \in \mathsf{N_C}$ and $a \in \mathsf{N_I}$, we have $a^{\mathcal{I}_i} \notin A^{\mathcal{I}_i}$. Since $\mathcal{I}_i \Rightarrow^{\mathcal{T},\mathsf{CR}}_{\boldsymbol{a}_{i+1}} \mathcal{I}_{i+1}$, this yields $a^{\mathcal{I}_i} \notin A^{\mathcal{I}_{i+1}}$, and thus $\mathcal{I}_{i+1} \not\models A(a)$. Otherwise, if γ is of the form $r(a,b)$ for $r \in \mathsf{N_R}$ and $a, b \in \mathsf{N_I}$, we have $(a^{\mathcal{I}_i}, b^{\mathcal{I}_i}) \notin r^{\mathcal{I}_i}$. Since $\mathcal{I}_i \Rightarrow^{\mathcal{T},\mathsf{CR}}_{\boldsymbol{a}_{i+1}} \mathcal{I}_{i+1}$, this yields $(a^{\mathcal{I}_i}, b^{\mathcal{I}_i}) \notin r^{\mathcal{I}_{i+1}}$, and thus $\mathcal{I}_{i+1} \not\models r(a,b)$. Overall, we obtain $\mathsf{p}(\gamma) \notin w_{i+1}$.
- Since $q_0 \in Q_0$, we have that $(q_0, \boldsymbol{a}_1, w_0)$ is an initial state of $\widehat{\mathcal{N}}$.
- Since $q_0 q_1 \ldots$ is an accepting run of $\mathcal{N}$ on $\boldsymbol{a}_1\boldsymbol{a}_2\ldots$, there are infinitely many $j \geq 0$ such that $q_j \in F$. The definition of the final states of $\widehat{\mathcal{N}}$ yields now that the above run is accepting.

It is left to be shown that $\mathcal{A}_{\mathcal{W}}$ has a model w.r.t. $\mathcal{T}_{\mathsf{red}} \cup \bigcup_{i=1}^{k} \mathcal{T}^{(i)}$. We have for every $i \geq 0$ that there is an index $\nu_i \in \{1, \ldots, k\}$ such that $\mathcal{I}_i$ induces the set X_{ν_i}, i.e.

$$X_{\nu_i} = \{\mathsf{p}(\alpha) \mid \alpha \in \mathsf{Ax}(\phi) \text{ and } \mathcal{I}_i \models \alpha\},$$

and, conversely, for every $\nu \in \{1,\dots,k\}$, there is an index $i \geq 0$ such that $\nu = \nu_i$. Let $\ell_1,\dots,\ell_k$ be such that $\nu_{\ell_1} = 1, \dots, \nu_{\ell_k} = k$. Note that Definition 6.13 yields that the domain $\Delta^{\mathcal{I}_0}$ of $\mathcal{I}_0$ is equal to the domain $\Delta^{\mathcal{I}_i}$ of $\mathcal{I}_i$ for every $i \geq 0$. Moreover, $a^{\mathcal{I}_0} = a^{\mathcal{I}_i}$ for every $a \in \mathsf{N_I}$ and every $i \geq 0$. Now, we define the interpretation $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ as follows:

- $\Delta^{\mathcal{J}} := \Delta^{\mathcal{I}_0}$;
- $a^{\mathcal{J}} := a^{\mathcal{I}_0}$ for every $a \in \mathsf{N_I}$;
- $N^{\mathcal{J}} := \{a^{\mathcal{J}} \mid a \in \mathsf{Obj}\}$;
- $(A^{(0)})^{\mathcal{J}} := A^{\mathcal{I}_0}$ for every $A \in \mathcal{R} \cap \mathsf{N_C}$;
- $(A^{(i)})^{\mathcal{J}} := A^{\mathcal{I}_{\ell_i}}$ for every $A \in \mathcal{R} \cap \mathsf{N_C}$ and every i, $1 \leq i \leq k$;
- $(r^{(0)})^{\mathcal{J}} := r^{\mathcal{I}_0}$ for every $r \in \mathcal{R} \cap \mathsf{N_R}$;
- $(r^{(i)})^{\mathcal{J}} := r^{\mathcal{I}_{\ell_i}}$ for every $r \in \mathcal{R} \cap \mathsf{N_R}$ and every i, $1 \leq i \leq k$; and
- $(T_C^{(i)})^{\mathcal{J}} := C^{\mathcal{I}_{\ell_i}}$ for every concept $C \in \mathcal{R}$ and every i, $1 \leq i \leq k$.

The definition of $N^{\mathcal{J}}$ yields that $\mathcal{J} \models \mathcal{T}_N$. Moreover, we have that $\mathcal{J} \models \mathcal{T}_{\mathsf{sub}}$, which can be shown using arguments that are very similar to the ones used to prove Property (1) of Lemma 6.30. Hence, $\mathcal{J} \models \mathcal{T}_{\mathsf{red}}$. Moreover, since $\mathcal{I}_i \Rightarrow_{a_{i+1}}^{\mathcal{T},\mathsf{CR}} \mathcal{I}_{i+1}$ for every $i \geq 0$, Definition 6.13 yields that $\mathcal{I}_i \models \mathcal{T}$ for every $i \geq 0$. Thus, we have $\mathcal{I}_{\ell_i} \models \mathcal{T}$ for every i, $1 \leq i \leq k$. Take any $C \sqsubseteq D \in \mathcal{T}$ and any i, $1 \leq i \leq k$. We have $\mathcal{I}_{\ell_i} \models C \sqsubseteq D$, and thus by the definition of $\mathcal{J}$ that $\mathcal{J} \models T_C^{(i)} \sqsubseteq T_D^{(i)}$. Hence, $\mathcal{J} \models \mathcal{T}^{(i)}$. Overall, we obtain that $\mathcal{J} \models \mathcal{T}_{\mathsf{red}} \cup \bigcup_{i=1}^{k} \mathcal{T}^{(i)}$.

It is only left to show that $\mathcal{J} \models \mathcal{A}_{\mathcal{W}}$. Note that we have for every relevant generalised ABox-literal α and every i, $1 \leq i \leq k$, that $\mathcal{I}_{\ell_i} \models \alpha$ iff $\mathcal{J} \models \alpha^{(i)}$. This can be shown using arguments that are very similar to the ones used to prove Property (1a) of Lemma 6.30. Since we have for i, $1 \leq i \leq k$, that

$$X_i = \big\{\mathsf{p}(\alpha) \mid \alpha \in \mathsf{Ax}(\phi) \text{ and } \mathcal{I}_{\ell_i} \models \alpha\big\},$$

this yields that $\mathcal{J}$ is a model of the generalised ABox

$$\big\{(\mathsf{p}^{-1}(p))^{(i)} \mid p \in X_i\big\} \cup \big\{(\neg\mathsf{p}^{-1}(p))^{(i)} \mid p \in \mathcal{P}_\phi \setminus X_i\big\}$$

for every i, $1 \leq i \leq k$. Hence, we have that $\mathcal{J} \models \mathcal{A}_{\mathcal{W}}$.

For the 'if' direction, assume that there is a set $\mathcal{W} = \{X_1,\dots,X_k\} \subseteq 2^{\mathcal{P}_\phi}$ such that $L_\omega(\widehat{\mathcal{N}}_{\phi^{\mathsf{p}}_{\mathcal{W}}}) \neq \emptyset$ and $\mathcal{A}_{\mathcal{W}}$ has a model w.r.t. $\mathcal{T}_{\mathsf{red}} \cup \bigcup_{i=1}^{k} \mathcal{T}^{(i)}$. Hence, there is an ω-word $w = w_0 w_1 \ldots \in L_\omega(\widehat{\mathcal{N}}_{\phi^{\mathsf{p}}_{\mathcal{W}}})$ and a model $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ of $\mathcal{A}_{\mathcal{W}}$ and $\mathcal{T}_{\mathsf{red}} \cup \bigcup_{i=1}^{k} \mathcal{T}^{(i)}$.

We define interpretations $\mathcal{J}_i = (\Delta^{\mathcal{J}_i}, \cdot^{\mathcal{J}_i})$, $1 \leq i \leq k$, as follows:

- $\Delta^{\mathcal{J}_i} := \Delta^{\mathcal{J}}$;
- $a^{\mathcal{J}_i} := a^{\mathcal{J}}$ for every $a \in \mathsf{N_I}$;
- $A^{\mathcal{J}_i} := (T_A^{(i)})^{\mathcal{J}}$ for every $A \in \mathcal{R} \cap \mathsf{N_C}$;
- $r^{\mathcal{J}_i} := \big((r^{(i)})^{\mathcal{J}} \cap (N^{\mathcal{J}} \times N^{\mathcal{J}})\big) \cup \Big((r^{(0)})^{\mathcal{J}} \cap \big((\Delta^{\mathcal{J}} \times (\neg N)^{\mathcal{J}}) \cup ((\neg N)^{\mathcal{J}} \times \Delta^{\mathcal{J}})\big)\Big)$ for every $r \in \mathcal{R} \cap \mathsf{N_R}$.

The interpretation of concept names and role names that are not contained in $\mathcal{R}$ is irrelevant. We assume in the following without loss of generality that the interpretation of all such names in empty in all interpretations $\mathcal{J}_i$, $1 \leq i \leq k$.

One can show that for every relevant generalised ABox-literal α and every i, $1 \leq i \leq k$, we have $\mathcal{J}_i \models \alpha$ iff $\mathcal{J} \models \alpha^{(i)}$. Moreover, one can show that for every relevant concept C and every i, $1 \leq i \leq k$, we have $C^{\mathcal{J}_i} = (T_C^{(i)})^{\mathcal{J}}$. These two claims can be shown using arguments very similar to the ones used to prove Properties (2a) and (2b) of Lemma 6.30. Since $\mathcal{J} \models \mathcal{A}_{\mathcal{W}}$, this yields that every $\mathcal{J}_i$, $1 \leq i \leq k$, satisfies exactly the axioms specified by the propositional variables in X_i. Moreover, since $\mathcal{J} \models \bigcup_{i=1}^{k} \mathcal{T}^{(i)}$, we have that $\mathcal{J} \models T_C^{(i)} \sqsubseteq T_D^{(i)}$ for every $C \sqsubseteq D \in \mathcal{T}$ and every i, $1 \leq i \leq k$. Thus, every $\mathcal{J}_i$, $1 \leq i \leq k$, is a model of $\mathcal{T}$.

Since $\widehat{\mathcal{N}}_{\phi^{\mathsf{p}}_{\mathcal{W}}}$ is defined to accept the intersection of the ω-language accepted by $\mathcal{N}_{\phi^{\mathsf{p}}_{\mathcal{W}}}$ and the ω-language accepted by $\widehat{\mathcal{N}}$, we have that $w \in L_\omega(\mathcal{N}_{\phi^{\mathsf{p}}_{\mathcal{W}}})$ and $w \in L_\omega(\widehat{\mathcal{N}})$. Moreover, since $\mathcal{N}_{\phi^{\mathsf{p}}_{\mathcal{W}}}$ is a Büchi-automaton for $\phi^{\mathsf{p}}_{\mathcal{W}}$ (which is the one constructed in the proof of Lemma 3.14), we have that the propositional LTL-structure $\mathfrak{W} := (w_i)_{i \geq 0}$ is a model of $\phi^{\mathsf{p}}_{\mathcal{W}}$. By the definition of $\phi^{\mathsf{p}}_{\mathcal{W}}$, we have that for every world w_i, there is exactly one index $\nu_i \in \{1, \ldots, k\}$ such that w_i satisfies

$$\bigwedge_{p \in X_{\nu_i}} p \wedge \bigwedge_{p \in \mathcal{P}_\phi \setminus X_{\nu_i}} \neg p.$$

We now define a DL-LTL-structure $\mathfrak{I} := (\mathcal{I}_i)_{i \geq 0}$ as follows. We set $\mathcal{I}_i := \mathcal{J}_{\nu_i}$ for $i \geq 0$. With the above arguments, we have that every $\mathcal{I}_i$ satisfies exactly the axioms specified by the propositional variables in X_{ν_i}. Thus, we obtain that for every $i \geq 0$ and every $\alpha \in \mathsf{Ax}(\phi)$, we have $\mathcal{I}_i \models \alpha$ iff $\mathsf{p}(\alpha) \in X_{\nu_i}$ iff $\mathsf{p}(\alpha) \in w_i$. Moreover, since $\mathfrak{W}$ is a model of $\phi^{\mathsf{p}}_{\mathcal{W}}$, we obtain that $\mathfrak{I}$ is a model of ϕ.

Since $w \in L_\omega(\widehat{\mathcal{N}})$, there is an accepting run of $\widehat{\mathcal{N}}$ on w. The definition of $\widehat{\mathcal{N}}$ yields that this run is of the form

$$(q_0, \boldsymbol{a}_1, w_0)(q_1, \boldsymbol{a}_2, w_1)(q_2, \boldsymbol{a}_3, w_2)\ldots.$$

The definition of $\widehat{\mathcal{N}}$ yields moreover that $q_0 q_1 q_2 \ldots$ is an accepting run of $\mathcal{N}$ on $\boldsymbol{a}_1 \boldsymbol{a}_2 \ldots$. Thus, $\boldsymbol{a}_1 \boldsymbol{a}_2 \ldots \in L_\omega(\mathcal{N})$.

It is only left to show that for every $i \geq 0$, we have that $\mathcal{I}_i \Rightarrow^{\mathcal{T},\mathsf{CR}}_{\boldsymbol{a}_{i+1}} \mathcal{I}_{i+1}$. For that, let $i \geq 0$ be arbitrary, and let $\boldsymbol{a}_{i+1} = (\mathsf{pre}_{i+1}, \mathsf{post}_{i+1})$. First note that by the definition of $\mathcal{I}_i$ and the arguments above, we have that $\mathcal{I}_i \models \mathcal{T}$ and $\mathcal{I}_{i+1} \models \mathcal{T}$. The definition of $\widehat{\Delta}$ yields that for every positive generalised ABox-literal $\alpha \in \mathsf{pre}_{i+1}$, we have $\mathsf{p}(\alpha) \in w_i$, and thus $\mathcal{I}_i \models \alpha$. Moreover, we have for every negative generalised ABox-literal $\neg\alpha \in \mathsf{pre}_{i+1}$ that $\mathsf{p}(\alpha) \notin w_i$, and thus $\mathcal{I}_i \models \neg\alpha$. Thus, $\mathcal{I}_i \models \mathsf{pre}_{i+1}$, and hence $\boldsymbol{a}_{i+1}$ is applicable to $\mathcal{I}_i$ w.r.t. $\mathcal{T}$.

Furthermore, we have that the domains of $\mathcal{I}_i$ and $\mathcal{I}_{i+1}$ coincide by definition, and we have $a^{\mathcal{I}_i} = a^{\mathcal{I}_{i+1}}$ for every $a \in \mathsf{N_I}$, again by definition.

We show now that $\mathsf{Eff}(\boldsymbol{a}_{i+1}, \mathcal{I}_i, \mathsf{CR}) = \mathsf{Eff}(\boldsymbol{a}_{i+1}, T_{w_i, \boldsymbol{a}_{i+1}, \mathsf{CR}}, \mathsf{CR})$. Take the action type $T := \{\alpha \in \mathsf{Cond}(\boldsymbol{a}_{i+1}, \mathsf{CR}) \mid \mathcal{I}_i \models \alpha\}$. Obviously, T is an action type for $\boldsymbol{a}_{i+1}$ and CR with $\mathcal{I}_i \models T$. By Lemma 6.24, we obtain $\mathsf{Eff}(\boldsymbol{a}_{i+1}, \mathcal{I}_i, \mathsf{CR}) = \mathsf{Eff}(\boldsymbol{a}_{i+1}, T, \mathsf{CR})$. Thus, it is enough to show that $T_{w_i, \boldsymbol{a}_{i+1}, \mathsf{CR}} = T$. For every positive generalised ABox-literal $\alpha \in \mathsf{Cond}(\boldsymbol{a}_{i+1}, \mathsf{CR})$, we have $\alpha \in T_{w_i, \boldsymbol{a}_{i+1}, \mathsf{CR}}$ iff $\mathsf{p}(\alpha) \in w_i$ iff $\mathcal{I}_i \models \alpha$ iff $\alpha \in T$. For every negative generalised ABox-literal $\neg\alpha \in \mathsf{Cond}(\boldsymbol{a}_{i+1}, \mathsf{CR})$, we have $\neg\alpha \in T_{w_i, \boldsymbol{a}_{i+1}, \mathsf{CR}}$ iff $\mathsf{p}(\alpha) \notin w_i$ iff $\mathcal{I}_i \not\models \alpha$ iff $\mathcal{I}_i \models \neg\alpha$ iff $\neg\alpha \in T$.

Moreover, $\mathsf{Eff}(\boldsymbol{a}_{i+1}, T_{w_i,\boldsymbol{a}_{i+1},\mathsf{CR}}, \mathsf{CR})$ is not contradictory due to the following arguments. Suppose we have $\{\beta, \neg\beta\} \subseteq \mathsf{Eff}(\boldsymbol{a}_{i+1}, T_{w_i,\boldsymbol{a}_{i+1},\mathsf{CR}}, \mathsf{CR})$ for some positive ABox-literal β. The definition of $\widehat{\Delta}$ yields that $\mathsf{p}(\beta) \in w_i$ and $\mathsf{p}(\beta) \notin w_i$, which is a contradiction.

Let $A \in \mathsf{N_C} \cap \mathcal{R}$, let

$$A^+ := \{a^{\mathcal{I}_i} \mid A(a) \in \mathsf{Eff}(\boldsymbol{a}_{i+1}, T_{w_i,\boldsymbol{a}_{i+1},\mathsf{CR}}, \mathsf{CR})\},$$

and let

$$A^- := \{a^{\mathcal{I}_i} \mid \neg A(a) \in \mathsf{Eff}(\boldsymbol{a}_{i+1}, T_{w_i,\boldsymbol{a}_{i+1},\mathsf{CR}}, \mathsf{CR})\}.$$

Since $\mathsf{Eff}(\boldsymbol{a}_{i+1}, T_{w_i,\boldsymbol{a}_{i+1},\mathsf{CR}}, \mathsf{CR})$ is not contradictory, we have that $A^+ \cap A^- = \emptyset$. Moreover, we have by definition that $A^+ \subseteq N^{\mathcal{J}}$ and $A^- \subseteq N^{\mathcal{J}}$. We first show that $A^{\mathcal{I}_{i+1}} \setminus N^{\mathcal{J}} = A^{\mathcal{I}_i} \setminus N^{\mathcal{J}}$. Since $\mathcal{J} \models \mathcal{T}_{\mathsf{sub}}$, we have

$$\begin{aligned} A^{\mathcal{I}_{i+1}} \setminus N^{\mathcal{J}} &= (T_A^{(\nu_{i+1})})^{\mathcal{J}} \setminus N^{\mathcal{J}} \\ &= \big((N^{\mathcal{J}} \cap (A^{(\nu_{i+1})})^{\mathcal{J}}) \cup ((\Delta^{\mathcal{J}} \setminus N^{\mathcal{J}}) \cap (A^{(0)}))^{\mathcal{J}}\big) \setminus N^{\mathcal{J}} \\ &= ((\Delta^{\mathcal{J}} \setminus N^{\mathcal{J}}) \cap (A^{(0)}))^{\mathcal{J}} \setminus N^{\mathcal{J}} \\ &= \big((N^{\mathcal{J}} \cap (A^{(\nu_i)})^{\mathcal{J}}) \cup ((\Delta^{\mathcal{J}} \setminus N^{\mathcal{J}}) \cap (A^{(0)}))^{\mathcal{J}}\big) \setminus N^{\mathcal{J}} \\ &= (T_A^{(\nu_i)})^{\mathcal{J}} \setminus N^{\mathcal{J}} \\ &= A^{\mathcal{I}_i} \setminus N^{\mathcal{J}} \end{aligned}$$

Hence, we have for every $d \in \Delta^{\mathcal{J}} \setminus N^{\mathcal{J}}$ that $d \in A^{\mathcal{I}_{i+1}}$ iff $d \in (A^{\mathcal{I}_i} \cup A^+) \setminus A^-$. We prove that for every $a^{\mathcal{J}} \in N^{\mathcal{J}}$, we have $a^{\mathcal{J}} \in A^{\mathcal{I}_{i+1}}$ iff $a^{\mathcal{J}} \in (A^{\mathcal{I}_i} \cup A^+) \setminus A^-$ by a case distinction. For the 'if' direction, it is obvious that we have $a^{\mathcal{J}} \notin A^-$. We consider first the case where $a^{\mathcal{J}} \in A^+$, and thus $A(a) \in \mathsf{Eff}(\boldsymbol{a}_{i+1}, T_{w_i,\boldsymbol{a}_{i+1},\mathsf{CR}}, \mathsf{CR})$. The definition of $\widehat{\Delta}$ yields that $\mathsf{p}(A(a)) \in w_{i+1}$, and thus $\mathcal{I}_{i+1} \models A(a)$, i.e. $a^{\mathcal{J}} \in A^{\mathcal{I}_{i+1}}$.

Consider now the case where $a^{\mathcal{J}} \notin A^+$, i.e. $a^{\mathcal{J}} \in A^{\mathcal{I}_i} \setminus A^-$. Since $a^{\mathcal{J}} \in A^{\mathcal{I}_i}$, we have $\mathcal{I}_i \models A(a)$, and thus $\mathsf{p}(A(a)) \in w_i$. Moreover, we have $\neg A(a) \notin \mathsf{Eff}(\boldsymbol{a}_{i+1}, T_{w_i,\boldsymbol{a}_{i+1},\mathsf{CR}}, \mathsf{CR})$ by the definition of A^-. Again, the definition of $\widehat{\Delta}$ yields that $\mathsf{p}(A(a)) \in w_{i+1}$, and thus $\mathcal{I}_{i+1} \models A(a)$, i.e. $a^{\mathcal{J}} \in A^{\mathcal{I}_{i+1}}$.

For the 'only if' direction, assume to the contrary that $a^{\mathcal{J}} \in A^{\mathcal{I}_{i+1}}$, $a^{\mathcal{J}} \notin A^+$, and $a^{\mathcal{J}} \notin A^{\mathcal{I}_i} \setminus A^-$. There are again two cases to consider: either $a^{\mathcal{J}} \in A^-$ or $a^{\mathcal{J}} \notin A^-$. If $a^{\mathcal{J}} \in A^-$, then $\neg A(a) \in \mathsf{Eff}(\boldsymbol{a}_{i+1}, T_{w_i,\boldsymbol{a}_{i+1},\mathsf{CR}}, \mathsf{CR})$. Again, the definition of $\widehat{\Delta}$ yields that $\mathsf{p}(A(a)) \notin w_{i+1}$, and thus $\mathcal{I}_{i+1} \not\models A(a)$, i.e. we have $a^{\mathcal{J}} \notin A^{\mathcal{I}_{i+1}}$, which is a contradiction.

Otherwise, if $a^{\mathcal{J}} \notin A^-$, we have $a^{\mathcal{J}} \notin A^{\mathcal{I}_i}$, and thus $\mathcal{I}_i \not\models A(a)$, which yields $\mathsf{p}(A(a)) \notin w_i$. Since $a^{\mathcal{J}} \notin A^+$, we have by the definition of A^+ that $A(a) \notin \mathsf{Eff}(\boldsymbol{a}_{i+1}, T_{w_i,\boldsymbol{a}_{i+1},\mathsf{CR}}, \mathsf{CR})$. Again, the definition of $\widehat{\Delta}$ yields that $\mathsf{p}(A(a)) \notin w_{i+1}$, and thus $\mathcal{I}_{i+1} \not\models A(a)$, i.e. we have $a^{\mathcal{J}} \notin A^{\mathcal{I}_{i+1}}$, which is again a contradiction.

Thus, we have shown that $A^{\mathcal{I}_{i+1}} = (A^{\mathcal{I}_i} \cup A^+) \setminus A^-$. Finally, let $r \in \mathsf{N_R} \cap \mathcal{R}$, let

$$r^+ := \{(a^{\mathcal{I}_i}, b^{\mathcal{I}_i}) \mid r(a,b) \in \mathsf{Eff}(\boldsymbol{a}_{i+1}, T_{w_i,\boldsymbol{a}_{i+1},\mathsf{CR}}, \mathsf{CR})\},$$

and let

$$r^- := \{(a^{\mathcal{I}_i}, b^{\mathcal{I}_i}) \mid \neg r(a,b) \in \mathsf{Eff}(\boldsymbol{a}_{i+1}, T_{w_i,\boldsymbol{a}_{i+1},\mathsf{CR}}, \mathsf{CR})\}.$$

Since $\mathsf{Eff}(\boldsymbol{a}_{i+1}, T_{w_i,\boldsymbol{a}_{i+1},\mathsf{CR}}, \mathsf{CR})$ is not contradictory, we have that $r^+ \cap r^- = \emptyset$. Moreover, we have by definition that $r^+ \subseteq N^{\mathcal{J}} \times N^{\mathcal{J}}$ and $r^- \subseteq N^{\mathcal{J}} \times N^{\mathcal{J}}$. Similar to before, we first show that $r^{\mathcal{I}_{i+1}} \setminus (N^{\mathcal{J}} \times N^{\mathcal{J}}) = r^{\mathcal{I}_i} \setminus (N^{\mathcal{J}} \times N^{\mathcal{J}})$. By the definitions of $r^{\mathcal{I}_{i+1}}$ and $r^{\mathcal{I}_i}$, we have

$$\begin{aligned} r^{\mathcal{I}_{i+1}} \setminus (N^{\mathcal{J}} \times N^{\mathcal{J}}) &= \big((r^{(0)})^{\mathcal{J}} \cap ((\Delta^{\mathcal{J}} \times (\neg N)^{\mathcal{J}}) \cup ((\neg N)^{\mathcal{J}} \times \Delta^{\mathcal{J}}))\big) \setminus (N^{\mathcal{J}} \times N^{\mathcal{J}}) \\ &= r^{\mathcal{I}_i} \setminus (N^{\mathcal{J}} \times N^{\mathcal{J}}). \end{aligned}$$

Hence, we have for all $d, e \in \Delta^{\mathcal{J}} \setminus N^{\mathcal{J}}$ that $(d,e) \in r^{\mathcal{I}_{i+1}}$ iff $(d,e) \in (r^{\mathcal{I}_i} \cup r^+) \setminus r^-$. Using very similar arguments as above, we prove that for every $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in N^{\mathcal{J}} \times N^{\mathcal{J}}$, we have $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in r^{\mathcal{I}_{i+1}}$ iff $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in (r^{\mathcal{I}_i} \cup r^+) \setminus r^-$. For the 'if' direction, it is obvious that we have $(a^{\mathcal{J}}, b^{\mathcal{J}}) \notin r^-$. We consider first the case where $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in r^+$, and thus $r(a,b) \in \mathsf{Eff}(\boldsymbol{a}_{i+1}, T_{w_i,\boldsymbol{a}_{i+1},\mathsf{CR}}, \mathsf{CR})$. The definition of $\widehat{\Delta}$ yields that $\mathsf{p}(r(a,b)) \in w_{i+1}$, and thus $\mathcal{I}_{i+1} \models r(a,b)$, i.e. we have $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in A^{\mathcal{I}_{i+1}}$.

Consider now the case where $(a^{\mathcal{J}}, b^{\mathcal{J}}) \notin r^+$, i.e. $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in r^{\mathcal{I}_i} \setminus r^-$. Since $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in r^{\mathcal{I}_i}$, we have $\mathcal{I}_i \models r(a,b)$, and thus $\mathsf{p}(r(a,b)) \in w_i$. Moreover, we have by the definition of r^- that $\neg r(a,b) \notin \mathsf{Eff}(\boldsymbol{a}_{i+1}, T_{w_i,\boldsymbol{a}_{i+1},\mathsf{CR}}, \mathsf{CR})$. Again, the definition of $\widehat{\Delta}$ yields that $\mathsf{p}(r(a,b)) \in w_{i+1}$, and thus $\mathcal{I}_{i+1} \models r(a,b)$, i.e. we have $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in r^{\mathcal{I}_{i+1}}$.

For the 'only if' direction, assume to the contrary that $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in r^{\mathcal{I}_{i+1}}$, $(a^{\mathcal{J}}, b^{\mathcal{J}}) \notin r^+$, and $(a^{\mathcal{J}}, b^{\mathcal{J}}) \notin r^{\mathcal{I}_i} \setminus r^-$. There are again two cases to consider: either $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in r^-$ or $(a^{\mathcal{J}}, b^{\mathcal{J}}) \notin r^-$. If $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in r^-$, then $\neg r(a,b) \in \mathsf{Eff}(\boldsymbol{a}_{i+1}, T_{w_i,\boldsymbol{a}_{i+1},\mathsf{CR}}, \mathsf{CR})$. Again, the definition of $\widehat{\Delta}$ yields that $\mathsf{p}(r(a,b)) \notin w_{i+1}$, and thus $\mathcal{I}_{i+1} \not\models r(a,b)$, i.e. we have $(a^{\mathcal{J}}, b^{\mathcal{J}}) \notin r^{\mathcal{I}_{i+1}}$, which is a contradiction.

Otherwise, if $(a^{\mathcal{J}}, b^{\mathcal{J}}) \notin r^-$, we have $(a^{\mathcal{J}}, b^{\mathcal{J}}) \notin r^{\mathcal{I}_i}$, and thus $\mathcal{I}_i \not\models r(a,b)$, which yields $\mathsf{p}(r(a,b)) \notin w_i$. Since $(a^{\mathcal{J}}, b^{\mathcal{J}}) \notin r^+$, we have $r(a,b) \notin \mathsf{Eff}(\boldsymbol{a}_{i+1}, T_{w_i,\boldsymbol{a}_{i+1},\mathsf{CR}}, \mathsf{CR})$ by the definition of r^+. Again, the definition of $\widehat{\Delta}$ yields that $\mathsf{p}(r(a,b)) \notin w_{i+1}$, and thus $\mathcal{I}_{i+1} \not\models r(a,b)$, i.e. we have $(a^{\mathcal{J}}, b^{\mathcal{J}}) \notin r^{\mathcal{I}_{i+1}}$, which is again a contradiction.

We have thus shown that $r^{\mathcal{I}_{i+1}} = (r^{\mathcal{I}_i} \cup r^+) \setminus r^-$. Since we have shown that all conditions in Definition 6.13 are satisfied, this finishes the proof that we have $\mathcal{I}_i \Rightarrow^{\mathcal{T},\mathsf{CR}}_{\boldsymbol{a}_{i+1}} \mathcal{I}_{i+1}$ for every $i \geq 0$. □

Using this lemma, we can prove our complexity result.

Theorem 6.42. *The verification problem for our action formalism is*

1. *in* 2ExpTime *for the DLs $\mathcal{ALC}$, $\mathcal{ALCO}$, $\mathcal{ALCI}$, $\mathcal{ALCIO}$, $\mathcal{ALCQ}$, and $\mathcal{ALCQO}$; and*
2. *in* co-2NExpTime *for the DLs $\mathcal{ALCQI}$ and $\mathcal{ALCQIO}$.*

Moreover, if the TBox is assumed to be acyclic (or empty), the verification problem is

3. *in* ExpSpace *for the DLs $\mathcal{ALC}$, $\mathcal{ALCO}$, $\mathcal{ALCQ}$, and $\mathcal{ALCQO}$;*
4. *in* 2ExpTime *for the DLs $\mathcal{ALCI}$ and $\mathcal{ALCIO}$; and*
5. *in* co-2NExpTime *for the DLs $\mathcal{ALCQI}$ and $\mathcal{ALCQIO}$.*

Proof. Let $\mathcal{L}$ be a DL between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$. Furthermore, let $\mathcal{A}$ be a generalised $\mathcal{L}$-ABox, $\mathcal{T}$ be an $\mathcal{L}$-TBox, CR be a finite set of causal relationships, $\boldsymbol{A}$ be a finite set of DL-actions, $\mathcal{N} = (Q, \boldsymbol{A}, \Delta, Q_0, F)$ be a Büchi-automaton, ϕ be an $\mathcal{L}$-LTL-formula, and $\mathsf{p}\colon \mathsf{Ax}(\phi) \to \mathcal{P}_\phi$ be a bijection. As argued above, the following assumptions are without loss of generality:

- $\mathcal{A}$ is empty;
- every axiom occurring in a DL-action of $\boldsymbol{A}$ or a causal relationship of CR also occurs in ϕ; and
- for every $A \in \mathsf{N_C}$, $r \in \mathsf{N_R}$, $a, b \in \mathsf{N_I}$ occurring in ϕ, $\mathcal{T}$, a DL-action of $\boldsymbol{A}$, or a causal relationship of CR, we have that the assertions $A(a)$ and $r(a, b)$ also occur in ϕ.

Moreover, we have argued that the verification problem and the *un*satisfiability problem have the same complexity.

The satisfiability problem can be decided using the characterisation of Lemma 6.41. Consider the following decision procedure. For every set $\mathcal{W} \subseteq 2^{\mathcal{P}_\phi}$, do the following:

1. Construct the Büchi-automaton $\widehat{\mathcal{N}}_{\phi^{\mathsf{p}}_{\mathcal{W}}}$, and check it for non-emptiness.
2. Construct the generalised ABox $\mathcal{A}_{\mathcal{W}}$, the TBox $\mathcal{T}_{\mathsf{red}}$, and the TBox $\bigcup_{i=1}^{|\mathcal{W}|} \mathcal{T}^{(i)}$, and check whether $\mathcal{A}_{\mathcal{W}}$ has a model w.r.t. $\mathcal{T}_{\mathsf{red}} \cup \bigcup_{i=1}^{|\mathcal{W}|} \mathcal{T}^{(i)}$.

If both steps are successful for any set $\mathcal{W} \subseteq 2^{\mathcal{P}_\phi}$, i.e. we have $L_\omega(\widehat{\mathcal{N}}_{\phi^{\mathsf{p}}_{\mathcal{W}}}) \neq \emptyset$ and $\mathcal{A}_{\mathcal{W}}$ has a model w.r.t. $\mathcal{T}_{\mathsf{red}} \cup \bigcup_{i=1}^{|\mathcal{W}|} \mathcal{T}^{(i)}$, we know by Lemma 6.41 that ϕ is satisfiable w.r.t. $\mathcal{A}$, $\mathcal{T}$, CR, and $\mathcal{N}$. Otherwise, ϕ is unsatisfiable w.r.t. $\mathcal{A}$, $\mathcal{T}$, CR, and $\mathcal{N}$.

First note that there are doubly exponentially many sets $\mathcal{W} \subseteq 2^{\mathcal{P}_\phi}$, and each of these sets is of size exponential in the size of ϕ. Thus, all sets $\mathcal{W}$ can be enumerated in exponential space (and doubly exponential time). We first show that Step 1 can be performed in exponential space (and doubly exponential time).

In the proof of Lemma 3.14, we have seen that the Büchi-automaton $\mathcal{N}_{\phi^{\mathsf{p}}_{\mathcal{W}}}$ can be constructed in time exponential in the size of ϕ^{p} (and thus in time exponential in the size of ϕ) and linear in the size of $\mathcal{W}$. Moreover, the Büchi-automaton $\widehat{\mathcal{N}}$ is clearly of size exponential in the size of ϕ, since $\Sigma_{\mathcal{P}_\phi}$ is of size exponential in the size of ϕ. Furthermore, $\widehat{\mathcal{N}}$ can be constructed in time doubly exponential in the size of ϕ, $\boldsymbol{A}$, and CR. Indeed, for every $\sigma \in \Sigma_{\mathcal{P}_\phi}$ and $\boldsymbol{a} \in \boldsymbol{A}$, the generalised ABox $T_{\sigma,\boldsymbol{a},\mathsf{CR}}$ is of polynomial size, and can be constructed in polynomial time. Checking whether $T_{\sigma,\boldsymbol{a},\mathsf{CR}}$ is an action type for $\boldsymbol{a}$ and CR involves checking the two conditions of Definition 6.19. The first condition is obviously satisfied by construction, and the second condition is satisfied if $T_{\sigma,\boldsymbol{a},\mathsf{CR}}$ is consistent. Since the ABox-consistency problem for $\mathcal{ALCQIO}$ is NExpTime-complete [Sch94; Tob00; Pra05],[15] we can perform this check in doubly exponential time. Overall, $\widehat{\mathcal{N}}$ can be constructed in exponential space (and doubly exponential time).

As noted above, the Büchi-automaton $\widehat{\mathcal{N}}_{\phi^{\mathsf{p}}_{\mathcal{W}}}$, which accepts the intersection of the ω-language accepted by $\mathcal{N}_{\phi^{\mathsf{p}}_{\mathcal{W}}}$ and the ω-language accepted by $\widehat{\mathcal{N}}$, can be obtained using the standard product construction in time polynomial in the size of the input Büchi-automata, see e.g. [BK08; Tho90]. Thus, $\widehat{\mathcal{N}}_{\phi^{\mathsf{p}}_{\mathcal{W}}}$ is of size exponential in the size of ϕ, $\boldsymbol{A}$, and CR. Since the emptiness problem for Büchi-automata can be solved in time polynomial in the size of the Büchi-automaton [VW94], we obtain that Step 1 above can be performed in exponential space (and doubly exponential time) for each DL between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$.

For Step 2, note that the ABox $\mathcal{A}_{\mathcal{W}}$, the TBox $\mathcal{T}_{\mathsf{red}}$, and the TBox $\bigcup_{i=1}^{|\mathcal{W}|} \mathcal{T}^{(i)}$ can be constructed in exponential time, and is of size exponential in the size of the input. Since for

[15] As noted in the proof of Theorem 6.27, this is even the case if the number in the at-least and at-most restrictions are coded in binary.

$\mathcal{ALCIO}$ and $\mathcal{ALCQO}$, the ABox-consistency problem (w.r.t. general TBoxes) can be decided in ExpTime [Sch94; Hla04; HS01], Step 2 can be performed in doubly exponential time for all DLs that are fragments of $\mathcal{ALCIO}$ or $\mathcal{ALCQO}$. Since ExpTime is closed under complement, we obtain overall Parts 1 and 4 of the theorem.

For Parts 2 and 5 of the theorem, note since $\mathcal{T}_{\mathsf{red}}$ contains nominals, and for $\mathcal{ALCQIO}$, the ABox-consistency problem (w.r.t. general TBoxes) is NExpTime-complete [Sch94; Tob00; Pra05], we obtain that the satisfiability problem for the DLs $\mathcal{ALCQI}$ and $\mathcal{ALCQIO}$ is in 2NExpTime. Hence, we obtain that the verification problem is in co-2NExpTime for those DLs.

Finally, for Part 3 of the theorem, note that the TBox $\mathcal{T}_{\mathsf{red}} \cup \bigcup_{i=1}^{|\mathcal{W}|} \mathcal{T}^{(i)}$ is acyclic. Since for $\mathcal{ALCQO}$, the ABox-consistency problem w.r.t. acyclic TBoxes is PSpace-complete [Sch94; BLM+05b], we obtain that Step 2 can be performed in exponential space for any fragment of $\mathcal{ALCQO}$. Since ExpSpace is closed under complement, we obtain overall Part 3 of the theorem. □

If we compare these complexity results with the ones of [BLM10] (see Proposition 6.40) where only acyclic TBoxes and no causal relationships were considered, we observe the following. Allowing general TBoxes and causal relationships does not result in an increase of the complexity upper bounds for the verification problem for the description logics $\mathcal{ALCI}$, $\mathcal{ALCIO}$, $\mathcal{ALCQI}$, and $\mathcal{ALCQIO}$. For $\mathcal{ALC}$, $\mathcal{ALCO}$, $\mathcal{ALCQ}$, and $\mathcal{ALCQO}$, however, the complexity upper bound increases from ExpSpace to 2ExpTime if general TBoxes are allowed. The main reason for that is that the ABox-consistency problem in those DLs is harder if general TBoxes are considered. Unfortunately, we do not know for any of these shown upper bounds of the verification problem whether they are tight.

6.5 Summary

We will now sum up the main results of this chapter. In this chapter, we have proposed to use causal relationships to deal with the ramification problem for DL-based action formalisms. In Sections 6.2 and 6.3, we have shown, for our more expressive action formalism, that important inference problems for action formalisms such as the consistency problema and the projection problem are decidable in the setting with and without domain knowledge, which is described with a general TBox instead of only an acylcic one, for the DLs considered in [BLM+05a]. Moreover, we have derived complexity results from the decision procedures. What differs from DL to DL is the complexity of the basic inference problems in the respective DL (extended with nominals). Except for two cases, we obtain the matching hardness results by a reduction from such a basic inference problem. Finally, in Section 6.4, we considered the verification problem for our more expressive action formalism. There, a Büchi-automaton defines infinite sequences of DL-actions that an agent may execute. We have shown how to verify whether temporal properties are satisfied for all such sequences, and again we have derived complexity results. The complexity results obtained in this chapter are listed in Table 6.43.

Regarding future work, one interesting question is whether our approaches to deciding the consistency, projection, and verification problem can be extended to DL-actions with so-called occlusions [BLM+05a]. Basically, occlusions are sets of axioms that are allowed to change

Table 6.43: The complexity of the inference problems considered in this chapter for all DLs between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$

DL	*GCIs?*	*Consistency problem* (Theorems 6.27, 6.36)	*Projection problem* (Theorem 6.38)	*Verification problem* (Theorem 6.42)
$\mathcal{ALC}[\mathcal{Q}][\mathcal{O}]$	✗	PSpace-c.	PSpace-c.	in ExpSpace
	✓	ExpTime-c.	ExpTime-c.	in 2ExpTime
$\mathcal{ALCI}$	✗	PSpace-c.	ExpTime-c.	in 2ExpTime
	✓	ExpTime-c.	ExpTime-c.	in 2ExpTime
$\mathcal{ALCIO}$	✗	ExpTime-c.	ExpTime-c.	in 2ExpTime
	✓	ExpTime-c.	ExpTime-c.	in 2ExpTime
$\mathcal{ALCQI}$	✗	PSpace-c.	co-NExpTime-c.	in co-2NExpTime
	✓	ExpTime-hard / in co-NExpTime	co-NExpTime-c.	in co-2NExpTime
$\mathcal{ALCQIO}$	✗	co-NExpTime-c.	co-NExpTime-c.	in co-2NExpTime
	✓	co-NExpTime-hard / in $\text{P}^{\text{NExpTime}}$	co-NExpTime-c.	in co-2NExpTime

$\mathcal{ALC}[\mathcal{Q}][\mathcal{O}]$ is short for any DL between $\mathcal{ALC}$ and $\mathcal{ALCQO}$, and ‘c.’ is short for ‘complete’.

arbitrarily. Thus, occlusions allow the user to specify statements about the possible changes to the interpretations of concepts and roles that can be caused by applying a given DL-action. Note that such DL-actions are non-deterministic, i.e. their application to an interpretation may yield several possible successor interpretations. Consequently, such a DL-action may still be consistent although some of the successors interpretations are not models of the TBox (see the proof of Lemma 6.35). Thus, consistency can no longer be characterised by an analog of Lemma 6.35.

When defining our semantics for DL-actions in the presence of causal relationships, we followed the approach used in [BDT98; DTB98] rather than the one employed by [Lin95; Thi97]. In our health insurance example (see Examples 6.7 and 6.9), this was actually the appropriate semantics, but there may also be examples where it would be better to use the other semantics. Thus, it would be interesting to see whether our approach for deciding the consistency, the projection, and the verification problem can be adapted to deal with the semantics of [Lin95; Thi97].

Instead of trying to decide the projection problem directly, one can also follow the progression approach: given a DL-action and a (possibly incomplete) description of the current state, this approach tries to compute a description of the possible successor states. Projection then boils down to computing consequences of this successor description. For DL-based action theories, progression has been investigated in [LLM+11]. It would be interesting to see whether the results obtained there can be extended to the DL-based action theories with causal relationships considered in the present chapter.

In this chapter, we followed the approach for obtaining decidability results for action theories introduced in [BLM+05a], which is based on the idea of restricting the base logic to a decidable DL. In the literature, other ways of restricting the base logic to achieve this goal have been considered. For example, in [LL09] the authors consider so-called local effect actions[16] and restrict the base logic to so-called 'proper$^+$ knowledge bases' [LL05]. They show that, in this setting, progression is efficiently computable, which implies that the projection problem is efficiently decidable. It would be interesting to see whether this result can be extended to actions theories with causal relationships.

Moreover, it is interesting to see whether the results of this chapter can be used to verifying properties of action sequences 'generated' by restricted forms of high-level action programming languages such as GOLOG [LRL+97] and FLUX [Thi05a]. A first step in that direction has been done in [BZ13].

[16] Note that our DL-based actions are local effect actions.

Chapter 7

Conclusions

In this chapter, we first provide a brief summary in Section 7.1 about what was achieved in the present thesis. Then, in Section 7.2, we mention some future work.

7.1 Main Results

In this thesis, we have shown how to verify properties of dynamical systems whose behaviour can be partially observed. Whereas we assumed that we do not have a complete description of neither the system itself nor the current state of the system, we assumed that we have access to some background knowledge that encodes basic knowledge about the functioning of the observed system. Moreover, since the systems' states may have a complex internal structure, the expressive power of the formalism for representing the observations should go beyond propositional logic. We used description logics and extensions of them as one way to address these requirements.

After introducing the temporalised description logic $\mathcal{SHOQ}$-LTL in Chapter 3, which extends propositional LTL by allowing $\mathcal{SHOQ}$-axioms to occur in place of propositional variables, we have shown complexity results for the satisfiability problem in that temporalised DL. We have considered the problem in three different settings:

(i) neither concept names nor role names are allowed to be rigid,

(ii) only concept names are allowed to be rigid, and

(iii) both concept and role names are allowed to be rigid.

We have shown that the complexity is the same as in the less expressive temporalised description logic $\mathcal{ALC}$-LTL [BGL12], namely ExpTime-complete in Setting (i), NExpTime-complete in Setting (ii), and 2ExpTime-complete in Setting (iii). Table 3.4 mentions the respective theorems that state those results. Moreover, we have shown in this chapter that the consistency problem for (an extension of) Boolean $\mathcal{SHOQ}$-knowledge bases (w.r.t. some side condition) can be decided in exponential time.

Using these results, we considered in the following three different contexts. Firstly, in Chapter 4, we have shown how to perform runtime verification using $\mathcal{SHOQ}$-LTL. In this chapter, we provided a construction for monitors that runs in doubly exponential time, and produces monitors of doubly exponential size, even in the most complex case where both rigid concept names and rigid role names are allowed, i.e. in Setting (iii). For that we have shown how to construct Büchi-automata for $\mathcal{SHOQ}$-LTL-formulas using results from Chapter 3. Moreover, we have shown that this doubly exponential blow-up in the construction of the monitor cannot be avoided as it already occurs for propositional LTL. Finally, we have considered the related decision problems of liveness and monitorability and have shown some complexity results for them. Our results are only tight in Setting (iii). In this setting, both problems are 2EXPTIME-complete. For the other cases, a gap remains: both problems are EXPTIME-hard and in 2EXPTIME in Setting (i), and CO-NEXPTIME-hard and in 2EXPTIME in Setting (ii). However, the exact complexity of these problems are not even known for propositional LTL.

Secondly, in Chapter 5, we have considered a temporal version of ontology-based data access. More precisely, we proved complexity results for query entailment in a temporal query language that extends propositional LTL by allowing conjunctive queries in place of propositional variables. Moreover, background knowledge is encoded in a TBox that is formulated in a description logic between $\mathcal{ALC}$ and $\mathcal{SHQ}$. We considered both the data complexity and the combined complexity for this problem for the three settings above. In Setting (i), temporalised query entailment is CO-NP-complete w.r.t. data complexity and EXPTIME-complete w.r.t. combined complexity. In Setting (ii), the problem is CO-NP-complete w.r.t. data complexity and CO-NEXPTIME-complete w.r.t. combined complexity. Finally, in Setting (iii), the problem is CO-NP-hard and in EXPTIME w.r.t. data complexity and 2EXPTIME-complete w.r.t. combined complexity. For showing these results, some results of Chapter 3 are used. The obtained results are summarised in Table 5.2, where also the respective theorems and corollaries that state these results are listed.

Thirdly, in Chapter 6, we have considered an action formalism based on any description logic between $\mathcal{ALC}$ and $\mathcal{ALCQIO}$ that is capable of treating ramifications that arise naturally if domain constraints are encoded in general TBoxes. For this, we have extended the DL-based action formalism introduced in [BLM+05a] (which could deal only with acyclic TBoxes) with causal relationships. We have shown that important inference problems such as the consistency problem and the projection problem are decidable in our new formalism, and continue the work of [BLM10] by generalising the verification problem. We have derived a number of complexity results from the obtained decision procedures. Depending on the base DL, the complexity results range from PSPACE-complete to CO-NEXPTIME-hard and in $\text{P}^{\text{NEXPTIME}}$ for the consistency problem, and from PSPACE-complete to CO-NEXPTIME-complete for the projection problem. For the verification problem, the complexity ranges from in EXPSPACE to in CO-2NEXPTIME, and it is unknown whether these bounds are tight. The obtained results are summarised in Table 6.43.

7.2 Future Work

Some more technical directions for future work have been already mentioned at the end of each chapter. These include tightening some complexity bounds and considering slight extensions of the approaches introduced.

We now give some more general remarks on future research. Throughout the thesis, we have considered extensions of propositional LTL for specifying temporal properties. It would be interesting to consider also different temporal formalisms. Especially in the area of DL-based action formalisms, it is worthwile to examine the verification problem for extensions of the temporal logic CTL [CE82] and its extension CTL* [EH86] that encompasses propositional LTL. Furthermore, it might make sense to consider *real-time extensions* of temporal logics; see e.g. [Koy90; AH93; AFH96; RS99].

With respect to temporal query languages, it is interesting to consider light-weight description logics such as members of the *DL-Lite*-family [CDL+05; ACK+09; CDL+09] in our context since they allow first-order rewritability, i.e. query answering can be reduced to classical database reasoning. It is interesting to see whether the approaches considered in the present thesis work also for *DL-Lite*. First steps in that direction have been done in [BLT13b; BLT13a; BLT13c].

Additionally, it makes sense to consider extensions of the presented approaches that are capable of dealing with faulty sensor information. For instance, temporal extensions of (decidable) *probabilistic description logics* [Luk08; LS10] and (decidable) *fuzzy description logics* [Str01; BS09; BDG+12; BP13; BP14] may be useful. Moreover, one may want to deal with concrete numerical values, and thus it is interesting to consider also temporal extensions of description logics with *concrete domains* [BH91; Lut02; BS03; Lut04; LAH+05; LM07]. Concrete domains allow to use concrete values such as numbers or strings within concepts. Description logics with concrete domains allow furthermore to use predicates on such concrete values. Thus, one can, for instance, compare concrete values.

Moreover, it would be interesting to combine some of the above mentioned extensions. It is very challenging, however, to find a useful combination that remains decidable.

Bibliography

[AHV95] Serge Abiteboul, Richard Hull, and Victor Vianu: *Foundations of Databases*. Addison-Wesley, 1995 (cited on pages 7, 8, 83, 86).

[AS85] Bowen Alpern and Fred B. Schneider: 'Defining Liveness'. In *Information Processing Letters* **21**(4): 1985, pages 181–185 (cited on page 61).

[AFH96] Rajeev Alur, Tomás Feder, and Thomas A. Henzinger: 'The Benefits of Relaxing Punctuality'. In *Journal of the ACM* **43**(1): 1996, pages 116–146 (cited on page 166).

[AH93] Rajeev Alur and Thomas A. Henzinger: 'Real-Time Logics: Complexity and Expressiveness'. In *Information and Computation* **104**(1): 1993, pages 35–77 (cited on page 166).

[ABM99] Carlos Areces, Patrick Blackburn, and Maarten Marx: 'A Road-Map on Complexity for Hybrid Logics'. In *Proc. of the 13th Int. Workshop on Computer Science Logic (CSL 1999) and the 8th Annual Conf. of the EACSL*, Madrid, Spain. Edited by Jörg Flum and Mario Rodríguez-Artalejo. Volume 1683. Lecture Notes in Computer Science. Springer-Verlag, Sept. 1999, pages 307–321 (cited on page 133).

[ACK+09] Alessandro Artale, Diego Calvanese, Roman Kontchakov, and Michael Zakharyaschev: 'The *DL-Lite* Family and Relations'. In *Journal of Artificial Intelligence Research* **36**: 2009, pages 1–69 (cited on pages 2, 167).

[AF00] Alessandro Artale and Enrico Franconi: 'A survey of temporal extensions of description logics'. In *Annals of Mathematics and Artificial Intelligence* **30**(1–4): 2000, pages 171–210 (cited on page 3).

[AF05] Alessandro Artale and Enrico Franconi: 'Temporal Description Logics'. In *Handbook of Temporal Reasoning in Artificial Intelligence*. Edited by Michael Fischer, Dov M. Gabbay, and Lluís Vila. Elsevier, 2005. Chapter 12, pages 375–388 (cited on page 3).

[AFW+02] Alessandro Artale, Enrico Franconi, Frank Wolter, and Michael Zakharyaschev: 'A Temporal Description Logic for Reasoning over Conceptual Schemas and Queries'. In *Proc. of the 8th European Conf. on Logics in Artificial Intelligence (JELIA 2002)*, Cosenza, Italy. Edited by Sergio Felsca, Sergio Greco, Nicola Leone, and Giovambattista Ianni. Volume 2424. Lecture Notes in Computer Science. Springer-Verlag, Sept. 2002, pages 98–110 (cited on pages 84, 85).

[AKL+07] Alessandro Artale, Roman Kontchakov, Carsten Lutz, Frank Wolter, and Michael Zakharyaschev: 'Temporalising Tractable Description Logics'. In *Proc. of the 14th Int. Symp. on Temporal Representation and Reasoning (TIME 2007)*, Alicante, Spain. Edited by Valentin Goranko and X. Sean Wang. IEEE Press, June 2007, pages 11–22 (cited on pages 5, 84).

[AKR+10] Alessandro Artale, Roman Kontchakov, Vladislav Ryzhikov, and Michael Zakharyaschev: 'Past and Future of *DL-Lite*'. In *Proc. of the 24th AAAI Conf. on Artificial Intelligence (AAAI 2010)*, Atlanta, GA, USA. Edited by Maria Fox and David Poole. AAAI Press, July 2010, pages 243–248 (cited on page 5).

[AKW+13] Alessandro Artale, Roman Kontchakov, Frank Wolter, and Michael Zakharyaschev: 'Temporal Description Logic for Ontology-Based Data Access'. In *Proc. of the 23rd Int. Joint Conf. on Artificial Intelligence (IJCAI 2013)*, Beijing, China. Edited by Francesca Rossi. AAAI Press, Aug. 2013, pages 711–717 (cited on pages 84, 85).

[Baa03] Franz Baader: 'Terminological Cycles in a Description Logic with Existential Restrictions'. In *Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI 2003)*, Acapulco, Mexico. Edited by Georg Gottlob and Toby Walsh. Morgan Kaufmann, Los Altos, Aug. 2003, pages 325–330 (cited on page 2).

[BBB+09] Franz Baader, Andreas Bauer, Peter Baumgartner, Anne Cregan, Alfredo Gabaldon, Krystian Ji, Kevin Lee, David Rajaratnam, and Rolf Schwitter: 'A Novel Architecture for Situation Awareness Systems'. In *Proc. of the 18th Int. Conf. on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX 2009)*, Oslo, Norway. Edited by Martin Giese and Arild Waaler. Volume 5607. Lecture Notes in Computer Science. Springer-Verlag, July 2009, pages 77–92 (cited on page 1).

[BBL09] Franz Baader, Andreas Bauer, and Marcel Lippmann: 'Runtime Verification Using a Temporal Description Logic'. In *Proc. of the 7th Int. Symp. on Frontiers of Combining Systems (FroCoS 2009)*, Trento, Italy. Edited by Silvio Ghilardi and Roberto Sebastiani. Volume 5749. Lecture Notes in Computer Science. Springer-Verlag, Sept. 2009, pages 149–164 (cited on pages 6, 7, 11, 63).

[BBL13a] Franz Baader, Stefan Borgwardt, and Marcel Lippmann: 'On the Complexity of Temporal Query Answering'. LTCS-Report 13-01. Chair of Automata Theory, Institute of Theoretical Computer Science, Technische Universität Dresden, Mar. 2013. URL: `http://lat.inf.tu-dresden.de/research/reports.html` (cited on pages 10, 11, 41, 83, 85).

[BBL13b] Franz Baader, Stefan Borgwardt, and Marcel Lippmann: 'Temporalizing Ontology-Based Data Access'. In *Proc. of the 24th Int. Conf. on Automated Deduction (CADE-24)*, Lake Placid, NY, USA. Edited by Maria Paola Bonacina. Volume 7898. Lecture Notes in Artificial Intelligence. Springer-Verlag, June 2013, pages 330–344 (cited on pages 10, 11, 41, 83, 85, 90).

[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 2).

[BBL08] Franz Baader, Sebastian Brandt, and Carsten Lutz: 'Pushing the $\mathcal{EL}$ Envelope Further'. In *Proc. of the 5th Workshop on OWL: Experiences and Directions (OWLED 2008)*, Washington, DC, USA. Edited by Kendall Clark and Peter F. Patel-Schneider. Volume 496. CEUR Workshop Proceedings. `CEUR-WS.org`, Apr. 2008 (cited on page 2).

[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 1, 2, 13, 87, 91).

[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 page 171).

[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 5–7, 10, 29, 31, 32, 34, 36, 37, 39, 41, 76–78, 80, 83, 85, 92–94, 101, 104, 105, 165).

[BH91] Franz Baader and Philipp Hanschke: 'A Scheme for Integrating Concrete Domains into Concept Languages'. 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 452–457 (cited on page 167).

[BL14] Franz Baader and Marcel Lippmann: 'Runtime Verification Using a Temporal Description Logic Revisited'. LTCS-Report 14-01. Chair of Automata Theory, Institute of Theoretical Computer Science, Technische Universität Dresden, Mar. 2014. URL: `http://lat.inf.tu-dresden.de/research/reports.html` (cited on pages 11, 29, 57).

[BLL10a] Franz Baader, Marcel Lippmann, and Hongkai Liu: 'Adding Causal Relationships to DL-based Action Formalisms'. LTCS-Report 10-01. Chair of Automata Theory, Institute of Theoretical Computer Science, Technische Universität Dresden, Feb. 2010. URL: `http://lat.inf.tu-dresden.de/research/reports.html` (cited on pages 12, 119).

[BLL10b] Franz Baader, Marcel Lippmann, and Hongkai Liu: 'Using Causal Relationships to Deal with the Ramification Problem in Action Formalisms Based on Description Logics'. In *Proc. of the 17th Int. Conf. on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR-17)*, Yogyakarta, Indonesia. Edited by Christian G. Fermüller and Andrei Voronkov. Volume 6397. Lecture Notes in Computer Science. Springer-Verlag, Oct. 2010, pages 82–96 (cited on pages 12, 119, 120, 129).

[BLM09] Franz Baader, Hongkai Liu, and Anees ul Mehdi: 'Integrate Action Formalisms into Linear Temporal Description Logics'. LTCS-Report 09-03. Chair of Automata Theory, Institute of Theoretical Computer Science, Technische Universität Dresden, 2009. URL: `http://lat.inf.tu-dresden.de/research/reports.html` (cited on page 152).

[BLM10] Franz Baader, Hongkai Liu, and Anees ul Mehdi: 'Verifying Properties of Infinite Sequences of Description Logic Actions'. In *Proc. of the 19th European Conf. on Artificial Intelligence (ECAI 2010)*, Lisbon, Portugal. Edited by Helder Coelho, Rudi Studer, and Michael Wooldridge. Volume 215. Frontiers in Artificial Intelligence and Applications. IOS Press, Aug. 2010, pages 53–58 (cited on pages 10, 12, 119, 120, 151, 152, 161, 166).

[BLM+05a] Franz Baader, Carsten Lutz, Maja Miličić, Ulrike Sattler, and Frank Wolter: 'Integrating Description Logics and Action Formalisms: First Results'. In *Proc. of the 20th Nat. Conf. on Artificial Intelligence (AAAI 2005) and the 17th Innovative Applications of Artificial Intelligence Conference (IAAI 2005)*, Pittsburgh, PA, USA. Edited by Manuela M. Veloso and Subbarao Kambhampati. AAAI Press/The MIT Press, July 2005, pages 572–577 (cited on pages 10, 11, 119–121, 123–125, 127, 128, 134, 135, 150, 161, 163, 166).

[BLM+05b] Franz Baader, Carsten Lutz, Maja Miličić, Ulrike Sattler, and Frank Wolter: 'Integrating Description Logics and Action Formalisms for Reasoning about Web Services'. LTCS-Report 05-02. Chair of Automata Theory, Institute of Theoretical Computer Science, Technische Universität Dresden, 2005. URL: `http://lat.inf.tu-dresden.de/research/reports.html` (cited on pages 128, 133, 136, 137, 147, 151, 161).

[BS03] Franz Baader and Ulrike Sattler: 'Description Logics with Aggregates and Concrete Domains'. In *Information Systems* **28**(8): 2003, pages 979–1004 (cited on page 167).

[BZ13] Franz Baader and Benjamin Zarrieß: 'Verification of Golog Programs over Description Logic Actions'. In *Proc. of the 9th Int. Symp. on Frontiers of Combining Systems (FroCoS 2013)*, Nancy, France. Edited by Pascal Fontaine, Christophe Ringeissen, and Renate A. Schmidt. Volume 8152. Lecture Notes in Computer Science. Springer-Verlag, Sept. 2013, pages 181–196 (cited on pages 120, 163).

[BK08] Christel Baier and Joost-Pieter Katoen: *Principles of Model Checking*. Cambridge, MA, USA: The MIT Press, 2008 (cited on pages 1, 4, 22, 26, 153, 160).

[BGH+04] Howard Barringer, Allen Goldberg, Klaus Havelund, and Koushik Sen: 'Program Monitoring with LTL in EAGLE'. In *Proc. of the 18th Int. Parallel and Distributed Processing Symp. (IPDPS 2004)*, Santa Fe, NM, USA. IEEE Computer Society, Apr. 2004 (cited on page 6).

[BRH07] Howard Barringer, David E. Rydeheard, and Klaus Havelund: 'Rule Systems for Run-Time Monitoring: From EAGLE to RULER'. In *Revised Selected Papers of the 7th Int. Workshop on Runtime Verification (RV 2007)*, Vancouver, Canada. Edited by Oleg Sokolsky and Serdar Tasiran. Volume 4839. Lecture Notes in Computer Science. Springer-Verlag, Mar. 2007, pages 111–125 (cited on page 6).

[Bau10] Andreas Bauer: 'Monitorability of ω-regular languages'. In *Computing Research Repository* **abs/1006.3638**: 2010. URL: `http://arxiv.org/abs/1006.3638` (cited on pages 60, 61).

[BLS06] Andreas Bauer, Martin Leucker, and Christian Schallhart: 'Monitoring of Real-Time Properties'. In *Proc. of the 26th Int. Conf. on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2006)*, Kolkata, India. Edited by S. Arun-Kumar and Naveen Garg. Volume 4337. Lecture Notes in Computer Science. Springer-Verlag, Dec. 2006, pages 260–272 (cited on page 173).

[BLS10] Andreas Bauer, Martin Leucker, and Christian Schallhart: 'Comparing LTL Semantics for Runtime Verification'. In *Journal of Logic and Computation* **20**(3): 2010, pages 651–674 (cited on pages 6, 57).

[BLS11] Andreas Bauer, Martin Leucker, and Christian Schallhart: 'Runtime Verification for LTL and TLTL'. In *ACM Transactions on Software Engineering and Methodology* **20**(4): 2011. This is an extended version of [BLS06], 14:1–14:64 (cited on pages 6, 57, 58).

[BHW+04] Sebastian Bauer, Ian M. Hodkinson, Frank Wolter, and Michael Zakharyaschev: 'On Non-local Propositional and Weak Monodic Quantified CTL'. In *Journal of Logic and Computation* **14**(1): 2004, pages 3–22 (cited on page 4).

[BDT98] Kristof van Belleghem, Marc Denecker, and Daniele Theseider-Dupré: 'A Constructive Approach to the Ramification Problem'. In *Proc. of the ESSLLI 1998 Workshop on Reasoning about Actions: Foundations and Applications*, Saarbrücken, Germany. Edited by Guiseppe De Giacomo and Daniele Nardi. 1998, pages 1–17 (cited on pages 121, 124, 162).

[BBW07] Patrick Blackburn, Johann van Benthem, and Frank Wolter, editors: *Handbook of Modal Logic*. Volume 3. Studies in Logic and Practical Reasoning. Elsevier, 2007 (cited on page 3).

[BDG+12] Fernando Bobillo, Miguel Delgado, Juan Gómez-Romero, and Umberto Straccia: 'Joining Gödel and Zadeh Fuzzy Logics in Fuzzy Description Logics'. In *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* **20**(4): 2012, pages 475–508 (cited on page 167).

[BS09] Fernando Bobillo and Umberto Straccia: 'Fuzzy description logics with general t-norms and datatypes'. In *Fuzzy Sets and Systems* **160**(23): 2009, pages 3382–3402 (cited on page 167).

[BLT13a] Stefan Borgwardt, Marcel Lippmann, and Veronika Thost: 'Temporal Query Answering in *DL-Lite*'. In *Proc. of the 26th Int. Workshop on Description Logics (DL 2013)*, Ulm, Germany. Edited by Thomas Eiter, Birte Glimm, Yevgeny Kazakov, and Markus Krötzsch. Volume 1014. CEUR Workshop Proceedings. `CEUR-WS.org`, July 2013 (cited on pages 84, 85, 167).

[BLT13b] Stefan Borgwardt, Marcel Lippmann, and Veronika Thost: 'Temporal Query Answering in the Description Logic *DL-Lite*'. In *Proc. of the 9th Int. Symp. on Frontiers of Combining Systems (FroCoS 2013)*, Nancy, France. Edited by Pascal Fontaine, Christophe Ringeissen, and Renate A. Schmidt. Volume 8152. Lecture Notes in Computer Science. Springer-Verlag, Sept. 2013, pages 165–180 (cited on pages 84, 85, 167).

[BLT13c] Stefan Borgwardt, Marcel Lippmann, and Veronika Thost: 'Temporal Query Answering w.r.t. *DL-Lite*-Ontologies'. LTCS-Report 13-05. Chair of Automata Theory, Institute of Theoretical Computer Science, Technische Universität Dresden, Apr. 2013. URL: `http://lat.inf.tu-dresden.de/research/reports.html` (cited on pages 84, 85, 167).

[BP13] Stefan Borgwardt and Rafael Peñaloza: 'The Complexity of Lattice-Based Fuzzy Description Logics'. In *Journal on Data Semantics* **2**(1): 2013, pages 1–19 (cited on page 167).

[BP14] Stefan Borgwardt and Rafael Peñaloza: 'Consistency Reasoning in Lattice-Based Fuzzy Description Logics'. In *International Journal of Approximate Reasoning*: 2014. In press (cited on page 167).

[Bra04] Sebastian Brandt: 'Polynomial Time Reasoning in a Description Logic with Existential Restrictions, GCI Axioms, and—What Else?' In *Proc. of the 16th European Conf. on Artificial Intelligence (ECAI 2004)*, Valencia, Spain. Edited by Ramón López de Mántaras and Lorenza Saitta. IOS Press, Aug. 2004, pages 298–302 (cited on page 2).

[Büc62] Julius Richard Büchi: 'On a Decision Method in Restricted Second Order Arithmetic'. In *International Congress on Logic, Methodology and Philosophy of Science*. Stanford University Press, 1962, pages 1–11 (cited on page 21).

[CDL+09] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, Mariano Rodriguez-Muro, and Riccardo Rosati: 'Ontologies and Databases: The *DL-Lite* Approach'. In *Reasoning Web: Semantic Technologies for Information Systems, 5th Int. Summer School 2009*. Edited by Sergio Tessaris, Enrico Franconi, Thomas Eiter, Claudio Gutierrez, Siegfried Handschuh, Marie-Christine Rousset, and Renate A. Schmidt. Volume 5689. Lecture Notes in Computer Science. Springer-Verlag, Aug. 2009, pages 255–356 (cited on pages 2, 9, 84, 167).

[CDL+05] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati: '*DL-Lite*: Tractable Description Logics for Ontologies'. In *Proc. of the 20th Nat. Conf. on Artificial Intelligence (AAAI 2005) and the 17th Innovative Applications of Artificial Intelligence Conference (IAAI 2005)*, Pittsburgh, PA, USA. Edited by Manuela M. Veloso and Subbarao Kambhampati. AAAI Press/The MIT Press, July 2005, pages 602–607 (cited on pages 2, 167).

[CDL+06] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati: 'Data Complexity of Query Answering in Description Logics'. In *Proc. of the 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2006)*, Lake District, UK. Edited by Patrick Doherty, John Mylopoulos, and Christopher A. Welty. AAAI Press, June 2006, pages 260–270 (cited on pages 8, 91).

[CDL98] Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini: 'On the Decidability of Query Containment under Constraints'. In *Proc. of the 17th ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems (PODS 1998)*, Seattle, WA, USA. Edited by Alberto O. Mendelzon and Jan Paredaens. ACM Press, June 1998, pages 149–158 (cited on page 8).

[CM77] Ashok K. Chandra and Philip M. Merlin: 'Optimal Implementation of Conjunctive Queries in Relational Data Bases'. In *Proc. of the 9th Annual ACM Symp. on Theory of Computing (STOC 1977)*, Boulder, CO, USA. Edited by John E. Hopcroft, Emily P. Friedman, and Michael A. Harrison. ACM Press, May 1977, pages 77–90 (cited on pages 8, 87).

[Cho95] Jan Chomicki: 'Efficient Checking of Temporal Integrity Constraints Using Bounded History Encoding'. In *ACM Transactions on Database Systems* **20**(2): 1995, pages 148–186 (cited on page 84).

[CT05] Jan Chomicki and David Toman: 'Temporal Databases'. In *Handbook of Temporal Reasoning in Artificial Intelligence*. Edited by Michael Fischer, Dov M. Gabbay, and Lluís Vila. Elsevier, 2005. Chapter 14, pages 429–467 (cited on page 84).

[CE82] Edmund M. Clarke and E. Allen Emerson: 'Design and Synthesis of Synchronization Skeletons Using Branching-Time Temporal Logic'. In *Proc. of the Workshop on Logic of Programs (LP 1981)*, Yorktown Heights, NY, USA. Edited by Dexter Kozen. Volume 131. Lecture Notes in Computer Science. Springer-Verlag, 1982, pages 52–71 (cited on pages 4, 166).

[CGP99] Edmund M. Clarke, Orna Grumberg, and Doron A. Peled: *Model Checking*. Cambridge, MA, USA: The MIT Press, 1999 (cited on page 1).

[CM04] Séverine Colin and Leonardo Mariani: 'Run-Time Verification'. In *Model-Based Testing of Reactive Systems, Advanced Lectures*. Edited by Manfred Broy, Bengt Jonsson, Joost-Pieter Katoen, Martin Leucker, and Alexander Pretschner. Volume 3472. Lecture Notes in Computer Science. Springer-Verlag, 2004, pages 525–555 (cited on page 6).

[dR05] Marcelo d'Amorim and Grigore Roşu: 'Efficient Monitoring of ω-Languages'. In *Proc. of the 17th Int. Conf. on Computer Aided Verification (CAV 2005)*, Edinburgh, Scotland, UK. Edited by Kousha Etessami and Sriram K. Rajamani. Volume 3576. Lecture Notes in Computer Science. Springer-Verlag, July 2005, pages 364–378 (cited on page 6).

[Dan84] Ryszard Danecki: 'Nondeterministic Propositional Dynamic Logic with Intersection is Decidable'. In *Proc. of the 5th Symp. on Computation Theory (SCT 1984)*, Zaborów, Poland. Edited by Andrzej Skowron. Volume 208. Lecture Notes in Computer Science. Springer-Verlag, Dec. 1984, pages 34–53 (cited on page 41).

[DM00] Giuseppe De Giacomo and Fabio Massacci: 'Combining Deduction and Model Checking into Tableaux and Algorithms for Converse-PDL'. In *Information and Computation* **162**(1–2): 2000, pages 117–137 (cited on page 41).

[DEF+99] Stefan Decker, Michael Erdmann, Dieter Fensel, and Rudi Studer: 'Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information'. In *Proc. of the 8th Working Conf. on Database Semantics (DS-8)*, Rotorua, New Zealand. Edited by Robert Meersman, Zahir Tari, and Scott M. Stevens. Volume 138. IFIP Conference Proceedings. Kluwer, Jan. 1999, pages 351–369 (cited on pages 7, 83).

[DTB98] Marc Denecker, Daniele Theseider-Dupré, and Kristof van Belleghem: 'An inductive definition approach to ramifications'. In *Linkoping Electronic Articles in Computer and Information Science* **3**(7): Jan. 1998, pages 1–43 (cited on pages 121, 124, 162).

[DLN+94] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Andrea Schaerf: 'Deduction in Concept Languages: From Subsumption to Instance Checking'. In *Journal of Logic and Computation* **4**(4): 1994, pages 423–452 (cited on pages 87, 91).

[ELO+09] Thomas Eiter, Carsten Lutz, Magdalena Ortiz, and Mantas Šimkus: 'Query Answering in Description Logics with Transitive Roles'. In *Proc. of the 21st Int. Joint Conf. on Artificial Intelligence (IJCAI 2009)*, Pasadena, CA, USA. Edited by Craig Boutilier. AAAI Press, July 2009, pages 759–764 (cited on pages 87, 91, 118).

[Eme90] E. Allen Emerson: 'Temporal and Modal Logic'. In *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics*. Edited by Jan van Leeuwen. Elsevier/The MIT Press, 1990, pages 995–1072 (cited on page 4).

[EH86] E. Allen Emerson and Joseph Y. Halpern: '"Sometimes" and "Not Never" Revisited: On Branching versus Linear Time Temporal Logic'. In *Journal of the ACM* **33**(1): 1986, pages 151–178 (cited on pages 4, 166).

[End95] Mica R. Endsley: 'Toward a Theory of Situation Awareness in Dynamic Systems'. In *Human Factors* **37**(1): 1995, pages 32–64 (cited on page 1).

[FFM09] Yliès Falcone, Jean-Claude Fernandez, and Laurent Mounier: 'Runtime Verification of Safety-Progress Properties'. In *Selected Papers of the 9th Int. Workshop on Runtime Verification (RV 2009)*, Grenoble, France. Edited by Saddek Bensalem and Doron A. Peled. Volume 5779. Lecture Notes in Computer Science. Springer-Verlag, June 2009, pages 40–59 (cited on page 60).

[FT03] Enrico Franconi and David Toman: 'Fixpoint Extensions of Temporal Description Logics'. In *Proc. of the 16th Int. Workshop on Description Logics (DL 2003)*, Rome, Italy. Edited by Diego Calvanese, Giuseppe De Giacomo, and Enrico Franconi. Volume 81. CEUR Workshop Proceedings. `CEUR-WS.org`, Sept. 2003 (cited on page 4).

[FT11] Enrico Franconi and David Toman: 'Fixpoints in Temporal Description Logics'. In *Proc. of the 22nd Int. Joint Conf. on Artificial Intelligence (IJCAI 2011)*, Barcelona, Spain. Edited by Toby Walsh. AAAI Press, July 2011, pages 875–880 (cited on page 4).

[Gab89] Dov M. Gabbay: 'The Declarative Past and Imperative Future: Executable Temporal Logic for Interactive Systems'. In *Proc. of the 1987 Coll. on Temporal Logic in Specification*. Edited by Behnam Banieqbal, Howard Barringer, and Amir Pnueli. Volume 398. Lecture Notes in Computer Science. Springer-Verlag, 1989, pages 409–448 (cited on page 21).

[GHR94] Dov M. Gabbay, Ian M. Hodkinson, and Mark Reynolds: *Temporal Logic: Mathematical Foundations and Computational Aspects*. Volume 1. Clarendon Press, Oxford, 1994 (cited on page 4).

[GKW+03] Dov M. Gabbay, Ágnes Kurucz, Frank Wolter, and Michael Zakharyaschev: *Many-Dimensional Modal Logics: Theory and Applications*. Elsevier Science, 2003 (cited on pages 3–6, 19, 41).

[GPS+80] Dov M. Gabbay, Amir Pnueli, Saharon Shelah, and Jonathan Stavi: 'On the Temporal Analysis of Fairness'. In *Proc. of the 7th ACM Symp. on Principles of Programming Languages (POPL 1980)*. ACM Press, 1980, pages 163–173 (cited on page 21).

[GO01] Paul Gastin and Denis Oddoux: 'Fast LTL to Büchi Automata Translation'. In *Proc. of the 13th Int. Conf. on Computer Aided Verification (CAV 2001)*, Paris, France. Edited by Gérard Berry, Hubert Comon, and Alain Finkel. Volume 2102. Lecture Notes in Computer Science. Springer-Verlag, 2001, pages 53–65 (cited on page 26).

[GO03] Paul Gastin and Denis Oddoux: 'LTL with Past and Two-Way Very-Weak Alternating Automata'. In *Proc. of the 28th Int. Symp. on Mathematical Foundations of Computer Science (MFCS 2003)*, Bratislava, Slovakia. Edited by Branislav Rovan and Peter Vojtás. Volume 2747. Lecture Notes in Computer Science. Springer-Verlag, 2003, pages 439–448 (cited on page 26).

[GPV+96] Rob Gerth, Doron A. Peled, Moshe Y. Vardi, and Pierre Wolper: 'Simple On-the-fly Automatic Verification of Linear Temporal Logic'. In *Proc. of the 15th IFIP WG6.1 Int. Symp. on Protocol Specification, Testing and Verification XV (PSTV 1995)*, Warsaw, Poland. Edited by Piotr Dembinski and Marek Sredniawa. Volume 38. IFIP Conference Proceedings. Chapman & Hall, Ltd., 1996, pages 3–18 (cited on pages 22, 26).

[Gli07] Birte Glimm: 'Querying Description Logic Knowledge Bases'. PhD thesis. Manchester, UK: The University of Manchester, 2007 (cited on page 41).

[GHL+08] Birte Glimm, Ian Horrocks, Carsten Lutz, and Ulrike Sattler: 'Conjunctive Query Answering for the Description Logic $\mathcal{SHIQ}$'. In *Journal of Artificial Intelligence Research* **31**: 2008, pages 157–204 (cited on pages 85, 87, 92, 109, 110, 114).

[GHS08] Birte Glimm, Ian Horrocks, and Ulrike Sattler: 'Unions of Conjunctive Queries in SHOQ'. 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 252–262 (cited on page 41).

[GK08] Birte Glimm and Yevgeny Kazakov: 'Role Conjunctions in Expressive Description Logics'. In *Proc. of the 15th Int. Conf. on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR-15)*, Doha, Qatar. Edited by Iliano Cervesato, Helmut Veith, and Andrei Voronkov. Volume 5330. Lecture Notes in Computer Science. Springer-Verlag, Nov. 2008, pages 391–405 (cited on page 41).

[GS10] Yilan Gu and Mikhail Soutchanski: 'A Description Logic Based Situation Calculus'. In *Annals of Mathematics and Artificial Intelligence* **58**(1-2): 2010, pages 3–83 (cited on page 120).

[GHV05] Claudio Gutiérrez, Carlos A. Hurtado, and Alejandro A. Vaisman: 'Temporal RDF'. In *Proc. of the 2nd European Semantic Web Conference (ESWC 2005)*. Edited by Asunción Gómez-Pérez and Jérôme Euzenat. Volume 3532. Lecture Notes in Computer Science. Springer-Verlag, 2005, pages 93–107 (cited on page 84).

[GJL12] Víctor Gutiérrez-Basulto, Jean Christoph Jung, and Carsten Lutz: 'Complexity of Branching Temporal Description Logics'. In *Proc. of the 20th European Conf. on Artificial Intelligence (ECAI 2012)*, Montpellier, France. Edited by Luc De Raedt, Christian Bessiére, Didier Dubois, Patrick Doherty, Paolo Frasconi, Fredrik Heintz, and Peter J. F. Lucas. Volume 242. Frontiers in Artificial Intelligence and Applications. IOS Press, Aug. 2012, pages 390–395 (cited on page 4).

[GK12] Víctor Gutiérrez-Basulto and Szymon Klarman: 'Towards a Unifying Approach to Representing and Querying Temporal Data in Description Logics'. In *Proc. of the 6th Int. Conf. on Web Reasoning and Rule Systems (RR 2012)*, Vienna, Austria. Edited by Markus Krötzsch and Umberto Straccia. Volume 7497. Lecture Notes in Computer Science. Springer-Verlag, Sept. 2012, pages 90–105 (cited on pages 84, 85).

[HR04] Klaus Havelund and Grigore Roşu: 'Efficient Monitoring of Safety Properties'. In *International Journal on Software Tools for Technology Transfer* **6**(2): 2004, pages 158–173 (cited on page 6).

[Hem87] Lane Adrian Hemachandra: 'The Strong Exponential Hierarchy Collapses'. In *Proc. of the 19th Annual ACM Symp. on Theory of Computing (STOC 1987)*, New York City, NY, USA. ACM Press, May 1987, pages 110–122 (cited on page 148).

[Her96] Andreas Herzig: 'The PMA Revisited'. In *Proc. of the 5th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 1996)*. 1996, pages 40–50 (cited on page 121).

[Hla04] Jan Hladik: 'A Tableau System for the Description Logic $\mathcal{SHIO}$'. In *Contributions to the Doctoral Programme of the 2nd Int. Joint Conf. on Automated Reasoning (IJCAR 2004)*, Cork, Ireland. Edited by Ulrike Sattler. Volume 106. `CEUR-WS.org`, July 2004 (cited on pages 147, 150, 161).

[HWZ02] Ian M. Hodkinson, Frank Wolter, and Michael Zakharyaschev: 'Decidable and Undecidable Fragments of First-Order Branching Temporal Logics'. In *Proc. of the 17th Annual IEEE Symp. on Logic in Computer Science (LICS 2002)*, Copenhagen, Denmark. IEEE Computer Society, July 2002, pages 393–402 (cited on page 4).

[HKS06] Ian Horrocks, Oliver Kurz, and Ulrike Sattler: 'The Even More Irresistible $\mathcal{SROIQ}$'. In *Proc. of the 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2006)*, Lake District, UK. Edited by Patrick Doherty, John Mylopoulos, and Christopher A. Welty. AAAI Press, June 2006, pages 57–67 (cited on page 2).

[HPH03] Ian Horrocks, Peter F. Patel-Schneider, and Frank van Harmelen: 'From SHIQ and RDF to OWL: The Making of a Web Ontology Language'. In *Journal of Web Semantics* **1**(1): 2003, pages 7–26 (cited on page 3).

[HS01] Ian Horrocks and Ulrike Sattler: 'Ontology Reasoning in the $\mathcal{SHOQ}(D)$ Description Logic'. In *Proc. of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI 2001)*, Seattle, WA, USA. Edited by Bernhard Nebel. Morgan Kaufmann, Los Altos, Aug. 2001, pages 199–204 (cited on pages 18, 147, 150, 161).

[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 pages 18, 42, 53, 91).

[Koy90] Ron Koymans: 'Specifying Real-Time Properties with Metric Temporal Logic'. In *Real-Time Systems* **2**(4): 1990, pages 255–299 (cited on page 166).

[KR10] Orna Kupferman and Adin Rosenberg: 'The Blowup in Translating LTL to Deterministic Automata'. In *Revised Selected and Invited Papers of the 6th Int. Workshop on Model Checking and Artifical Intelligence (MoChArt 2010)*, Atlanta, GA, USA. Edited by Ron van der Meyden and Jan-Geory Smaus. Volume 6572. Lecture Notes in Computer Science. Springer-Verlag, July 2010, pages 85–94 (cited on pages 58, 59).

[KV01] Orna Kupferman and Moshe Y. Vardi: 'Model Checking of Safety Properties'. In *Formal Methods in System Design* **19**(3): 2001, pages 291–314 (cited on page 58).

[LMS02] François Laroussinie, Nicolas Markey, and Philippe Schnoebelen: 'Temporal Logic With Forgettable Past'. In *Proc. of the 17th Annual IEEE Symp. on Logic in Computer Science (LICS 2002)*, Copenhagen, Denmark. IEEE Computer Society, July 2002, pages 383–392 (cited on page 21).

[LRL+97] Hector J. Levesque, Raymond Reiter, Yves Lespérance, Fangzhen Lin, and Richard B. Scherl: 'GOLOG: A logic programming language for dynamic domains'. In *The Journal of Logic Programming* **31**(1–3): 1997, pages 59–83 (cited on pages 9, 163).

[LPZ85] Orna Lichtenstein, Amir Pnueli, and Lenore D. Zuck: 'The Glory of the Past'. In *Proc. of the Conf. on Logics of Programs*. Edited by Rohit Parikh. Volume 193. Lecture Notes in Computer Science. Springer-Verlag, 1985, pages 196–218 (cited on pages 21, 22, 26, 36, 60).

[Lin95] Fangzhen Lin: 'Embracing Causality in Specifying the Indirect Effects of Actions'. In *Proc. of the 14th Int. Joint Conf. on Artificial Intelligence (IJCAI 1995)*. 1995, pages 1985–1993 (cited on pages 121, 124, 162).

[LS11] Fangzhen Lin and Mikhail Soutchanski: 'Causal Theories of Actions Revisited'. 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, pages 235–240 (cited on page 121).

[Lip09] Marcel Lippmann: 'Runtime Verification Using Temporal Description Logics'. Diploma Thesis. Dresden, Germany: Technische Universität Dresden, Apr. 2009 (cited on page 63).

[LLM+06] Hongkai Liu, Carsten Lutz, Maja Miličić, and Frank Wolter: 'Reasoning about Actions using Description Logics with General TBoxes'. In *Proc. of the 10th European Conf. on Logics in Artificial Intelligence (JELIA 2006)*. Edited by Michael Fisher, Wiebe van der Hoek, Boris Konev, and Alexei Lisitsa. Volume 4160. Lecture Notes in Artificial Intelligence. Springer-Verlag, 2006, pages 266–279 (cited on pages 120–122).

[LLM+11] Hongkai Liu, Carsten Lutz, Maja Miličić, and Frank Wolter: 'Foundations of Instance Level Updates in Expressive Description Logics'. In *Artificial Intelligence* **175**(18): 2011, pages 2170–2197 (cited on page 162).

[LL09] Yongmei Liu and Gerhard Lakemeyer: 'On First-Order Definability and Computability of Progression for Local-Effect Actions and Beyond'. In *Proc. of the 21st Int. Joint Conf. on Artificial Intelligence (IJCAI 2009)*, Pasadena, CA, USA. Edited by Craig Boutilier. AAAI Press, July 2009, pages 860–866 (cited on page 163).

[LL05] Yongmei Liu and Hector J. Levesque: 'Tractable Reasoning in First-Order Knowledge Bases with Disjunctive Information'. In *Proc. of the 20th Nat. Conf. on Artificial Intelligence (AAAI 2005) and the 17th Innovative Applications of Artificial Intelligence Conference (IAAI 2005)*, Pittsburgh, PA, USA. Edited by Manuela M. Veloso and Subbarao Kambhampati. AAAI Press/The MIT Press, July 2005, pages 639–644 (cited on page 163).

[Löw15] Leopold Löwenheim: 'Über Möglichkeiten im Relativkalkül'. In German. In *Mathematische Annalen* **76**(4): 1915, pages 447–470 (cited on pages 36, 39, 65, 100, 105, 110).

[Luk08] Thomas Lukasiewicz: 'Expressive probabilistic description logics'. In *Artificial Intelligence* **172**(6-7): 2008, pages 852–883 (cited on page 167).

[Lut01] Carsten Lutz: 'Interval-based Temporal Reasoning with General TBoxes'. In *Proc. of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI 2001)*, Seattle, WA, USA. Edited by Bernhard Nebel. Morgan Kaufmann, Los Altos, Aug. 2001, pages 89–94 (cited on page 4).

[Lut02] 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 167).

[Lut04] Carsten Lutz: 'NExpTime-complete Description Logics with Concrete Domains'. In *ACM Transactions on Computational Logic* **5**(4): 2004, pages 669–705 (cited on page 167).

[Lut08a] Carsten Lutz: 'The Complexity of Conjunctive Query Answering in Expressive Description Logics'. In *Proc. of the 4th Int. Joint Conf. on Automated Reasoning (IJCAR 2008)*, Sydney, Australia. Edited by Alessandro Armando, Peter Baumgartner, and Gilles Dowek. Volume 5195. Lecture Notes in Computer Science. Springer-Verlag, Aug. 2008, pages 179–193 (cited on pages 8, 41, 85, 87, 92, 109, 114–117).

[Lut08b] Carsten Lutz: 'Two Upper Bounds for Conjunctive Query Answering in $\mathcal{SHIQ}$'. 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 116).

[LAH+05] Carsten Lutz, Carlos Areces, Ian Horrocks, and Ulrike Sattler: 'Keys, Nominals, and Concrete Domains'. In *Journal of Artificial Intelligence Research* **23**: 2005, pages 667–726 (cited on page 167).

[LM07] Carsten Lutz and Maja Miličić: 'A Tableau Algorithm for DLs with Concrete Domains and GCIs'. In *Journal of Automated Reasoning* **38**(1-3): 2007, pages 227–259 (cited on page 167).

[LS10] Carsten Lutz and Lutz Schröder: 'Probabilistic Description Logics for Subjective Uncertainty'. In *Proc. of the 12th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2010)*, Toronto, Canada. Edited by Fangzhen Lin, Ulrike Sattler, and Miroslaw Truszczynski. AAAI Press, May 2010, pages 393–403 (cited on page 167).

[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 3, 5).

[Mar04] Nicolas Markey: 'Past is for free: On the complexity of verifying linear temporal properties with past'. In *Acta Informatica* **40**(6–7): 2004, pages 431–458 (cited on page 21).

[Mas01] Fabio Massacci: 'Decision Procedures for Expressive Description Logics with Intersection, Composition, Converse of Roles and Role Identity'. In *Proc. of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI 2001)*, Seattle, WA, USA. Edited by Bernhard Nebel. Morgan Kaufmann, Los Altos, Aug. 2001, pages 193–198 (cited on page 41).

[MH69] John McCarthy and Patrick J. Hayes: 'Some Philosophical Problems from the Standpoint of Artificial Intelligence'. In *Machine Intelligence* **4**: 1969, pages 463–502 (cited on page 120).

[Min81] Marvin Minsky: 'A Framework for Representing Knowledge'. In *Mind Design*: 1981. Edited by John Haugeland (cited on page 2).

[Mot12] Boris Motik: 'Representing and Querying Validity Time in RDF and OWL: A Logic-Based Approach'. In *Journal of Web Semantics* **12–13**: Apr. 2012, pages 3–21 (cited on page 84).

[Mul63] David E. Muller: 'Infinite sequences and finite machines'. In *Proc. of the 4th Annual Symp. on Switching Circuit Theory and Logical Design (SWCT 1963)*. IEEE Computer Society, 1963, pages 3–16 (cited on page 21).

[NV07] Sumit Nain and Moshe Y. Vardi: 'Branching vs. Linear Time: Semantical Perspective'. In *Proc. of the 5th Int. Symp. on Automated Technology for Verification and Analysis (ATVA 2007)*, Tokyo, Japan. Edited by Kedar S. Namjoshi, Tomohiro Yoneda, Teruo Higashino, and Yoshio Okamura. Volume 4762. Lecture Notes in Computer Science. Springer-Verlag, Oct. 2007, pages 19–34 (cited on page 4).

[OCE06] Magdalena Ortiz, Diego Calvanese, and Thomas Eiter: 'Characterizing Data Complexity for Conjunctive Query Answering in Expressive Description Logics'. In *Proc. of the 21st Nat. Conf. on Artificial Intelligence (AAAI 2006) and the 18th Innovative Applications of Artificial Intelligence Conference (IAAI 2006)*, Boston, MA, USA. AAAI Press, July 2006, pages 275–280 (cited on pages 85, 92, 117).

[Pap81] Christos H. Papadimitriou: 'On the Complexity of Integer Programming'. In *Journal of the ACM* **28**(4): 1981, pages 765–768 (cited on pages 44, 54).

[Pnu77] Amir Pnueli: 'The Temporal Logic of Programs'. In *Proc. of the 18th Annual Symp. on Foundations of Computer Science (FOCS 1977)*, Providence, RI, USA. IEEE Computer Society Press, 1977, pages 46–57 (cited on pages 4, 6, 19, 21, 83).

[PZ06] Amir Pnueli and Aleksandr Zaks: 'PSL Model Checking and Run-Time Verification Via Testers'. In *Proc. of the 14th Int. Symp. on Formal Methods (FM 2006)*, Hamilton, Canada. Edited by Jayadev Misra, Tobias Nipkow, and Emil Sekerinski. Volume 4085. Lecture Notes in Computer Science. Springer-Verlag, Aug. 2006, pages 573–586 (cited on page 60).

[PCD+08] Antonella Poggi, Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati: 'Linking Data to Ontologies'. In *Journal on Data Semantics* **X**: 2008, pages 133–173 (cited on pages 7, 83).

[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 18, 133, 147, 151, 160, 161).

[Qui67] M. Ross Quillian: 'Word concepts: A theory and simulation of some basic semantic capabilities'. In *Behavioral Science* **12**(5): 1967, pages 410–430 (cited on page 2).

[Rab69] Michael O. Rabin: 'Decidability of Second-Order Theories and Automata on Infinite Trees'. In *Transactions of the AMS* **141**: 1969, pages 1–35 (cited on page 22).

[RS99] Jean-François Raskin and Pierre-Yves Schobbens: 'The Logic of Event Clocks – Decidability, Complexity and Expressiveness'. In *Journal of Automata, Languages and Combinatorics* **4**(3): 1999, pages 247–286 (cited on page 166).

[Rei01] Raymond Reiter: *Knowledge in Action: Logical Foundations for Describing and Implementing Dynamical Systems*. The MIT Press, Bradford Books, 2001 (cited on pages 9, 119, 120, 128).

[Roş12] Grigore Roşu: 'On Safety Properties and Their Monitoring'. In *Scientific Annals of Computer Science* **22**(2): 2012, pages 327–365 (cited on page 6).

[RH05] Grigore Roşu and Klaus Havelund: 'Rewriting-Based Techniques for Runtime Verification'. In *Automated Software Engineering* **12**(2): 2005, pages 151–197 (cited on page 6).

[RG10] Sebastian Rudolph and Birte Glimm: 'Nominals, Inverses, Counting, and Conjunctive Queries or: Why Infinity is your Friend!' In *Journal of Artificial Intelligence Research* **39**(1): 2010, pages 429–481 (cited on page 90).

[Sav70] Walter J. Savitch: 'Relationships between nondeterministic and deterministic tape complexities'. In *Journal of Computer and System Sciences* **4**(2): 1970, pages 177–192 (cited on pages 133, 151).

[Sch93a] Andrea Schaerf: 'On the Complexity of the Instance Checking Problem in Concept Languages with Existential Quantification'. In *Journal of Intelligent Information Systems* **2**(3): 1993, pages 265–278 (cited on pages 87, 91).

[Sch94] Andrea Schaerf: ‘Reasoning with individuals in concept languages’. In *Data & Knowledge Engineering* **13**(2): 1994, pages 141–176 (cited on pages 18, 133, 147, 150, 151, 160, 161).

[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 18, 147, 150).

[Sch93b] Klaus Schild: ‘Combining Terminological Logics with Tense Logic’. In *Proc. of the 6th Portuguese Conf. on Artificial Intelligence (EPIA 1993)*, Porto, Portugal. Edited by Miguel Filgueiras and Luís Damas. Volume 727. Lecture Notes in Computer Science. Springer-Verlag, Oct. 1993, pages 105–120 (cited on pages 4, 5).

[SS91] Manfred Schmidt-Schauß and Gert Smolka: ‘Attributive Concept Descriptions with Complements’. In *Artificial Intelligence* **48**(1): 1991, pages 1–26 (cited on pages 2, 5, 17, 133).

[SC85] A. Prasad Sistla and Edmund M. Clarke: ‘The Complexity of Proposional Linear Temporal Logics’. In *Journal of the ACM* **32**(3): 1985, pages 733–749 (cited on pages 21, 26, 36, 60).

[Sko20] Thoralf Skolem: ‘Logisch-kombinatorische Untersuchungen über die Erfüllbarkeit oder Beweisbarkeit mathematischer Sätze nebst einem Theorem über dichte Mengen’. In German. In *Videnskapsselskapets skrifter I, Matematisk-naturvidenskabelig klasse, Videnskabsakademiet i Kristiania* **4**: 1920, pages 1–36 (cited on pages 36, 39, 65, 100, 105, 110).

[SY12] Mikhail Soutchanski and Wael Yehia: ‘Towards an Expressive Decidable Logical Action Theory’. In *Proc. of the 25th Int. Workshop on Description Logics (DL 2012)*, Rome, Italy. Edited by Yevgeny Kazakov, Domenico Lembo, and Frank Wolter. Volume 846. CEUR Workshop Proceedings. `CEUR-WS.org`, June 2012 (cited on page 120).

[Str01] Umberto Straccia: ‘Reasoning within Fuzzy Description Logics’. In *Journal of Artificial Intelligence Research* **14**: 2001, pages 137–166 (cited on page 167).

[ST13] Hannes Straß and Michael Thielscher: ‘A general first-order solution to the ramification problem with cycles’. In *Journal of Applied Logic* **11**(3): 2013, pages 289–308 (cited on page 121).

[Str82] Robert S. Streett: ‘Propositional Dynamic Logic of Looping and Converse Is Elementarily Decidable’. In *Information and Control* **54**(1–2): 1982, pages 121–141 (cited on page 22).

[SBS+07] Boontawee Suntisrivaraporn, Franz Baader, Stefan Schulz, and Kent Spackman: ‘Replacing SEP-Triplets in SNOMED CT using Tractable Description Logic Operators’. In *Proc. of the 11th Conf. on Artificial Intelligence in Medicine (AIME 2007)*. Edited by Riccardo Bellazzi, Ameen Abu-Hanna, and Jim Hunter. Volume 4594. Lecture Notes in Computer Science. Springer-Verlag, 2007, pages 287–291 (cited on page 3).

[Tes01] Sergio Tessaris: 'Questions and Answers: Reasoning and Querying in Description Logic'. PhD thesis. Manchester, UK: The University of Manchester, 2001 (cited on page 90).

[Thi97] Michael Thielscher: 'Ramification and Causality'. In *Artificial Intelligence* **89**(1–2): 1997, pages 317–364 (cited on pages 121, 124, 162).

[Thi05a] Michael Thielscher: 'FLUX: A logic programming method for reasoning agents'. In *Theory and Practice of Logic Programming* **5**(4–5): 2005, pages 533–565 (cited on pages 9, 163).

[Thi05b] Michael Thielscher: *Reasoning Robots: The Art and Science of Programming Robotic Agents*. Volume 33. Applied Logic Series. Springer-Verlag, 2005 (cited on pages 9, 119).

[Tho90] Wolfgang Thomas: 'Automata on Infinite Objects'. In *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics*. Edited by Jan van Leeuwen. Elsevier/The MIT Press, 1990, pages 133–192 (cited on pages 153, 160).

[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 18, 133, 147, 151, 160, 161).

[Tob01] Stephan Tobies: 'Complexity Results and Practical Algorithms for Logics in Knowledge Representation'. PhD thesis. Aachen, Germany: RWTH Aachen, 2001 (cited on pages 18, 41, 133, 148, 150).

[UW01] Ulrich Ultes-Nitsche and Pierre Wolper: 'Checking Properties within Fairness and Behavior Abstractions'. In *Computing Research Repository* **cs.LO/0101017**: 2001. URL: `http://arxiv.org/abs/cs.LO/0101017` (cited on page 61).

[Var01] Moshe Y. Vardi: 'Branching vs. Linear Time: Final Showdown'. In *Proc. of the 7th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2001)*, Genova, Italy. Edited by Tiziana Margaria and Wang Yi. Volume 2031. Lecture Notes in Computer Science. Springer-Verlag, Apr. 2001, pages 1–22 (cited on page 4).

[VW94] Moshe Y. Vardi and Pierre Wolper: 'Reasoning About Infinite Computations'. In *Information and Computation* **155**(1): 1994, pages 1–37 (cited on pages 22, 36, 66, 67, 74, 85, 99, 160).

[Win88] Marianne Winslett: 'Reasoning about Action Using a Possible Models Approach'. In *Proc. of the 7th Nat. Conf. on Artificial Intelligence (AAAI 1988)*. 1988, pages 89–93 (cited on page 121).

[WVS83] Pierre Wolper, Moshe Y. Vardi, and A. Prasad Sistla: 'Reasoning About Infinite Computation Paths'. In *Proc. of the 24th Annual Symp. on Foundations of Computer Science (FOCS 1983)*, Tucson, AZ, USA. IEEE Computer Society Press, 1983, pages 185–194 (cited on pages 22, 85).

[WZ00] Frank Wolter and Michael Zakharyaschev: ‘Temporalizing Description Logics’. In *Proc. of the 2nd Int. Workshop on Frontiers of Combining Systems (FroCoS 1998)*, Amsterdam, The Netherlands. Edited by Dov Gabbay and Maarten de Rijke. Volume 7. Studies in Logic and Computation. Research Studies Press/Wiley, 2000, pages 379–402 (cited on page 4).

[YLL+12] Wael Yehia, Hongkai Liu, Marcel Lippmann, Franz Baader, and Mikhail Soutchanski: ‘Experimental Results on Solving the Projection Problem in Action Formalisms Based on Description Logics’. In *Proc. of the 25th Int. Workshop on Description Logics (DL 2012)*, Rome, Italy. Edited by Yevgeny Kazakov, Domenico Lembo, and Frank Wolter. Volume 846. CEUR Workshop Proceedings. `CEUR-WS.org`, June 2012 (cited on pages 12, 119, 120).