Technische Universität Dresden
Institute for Theoretical Computer Science
Chair for Automata Theory

LTCS–Report

On the Complexity of Temporal Query Answering

Franz Baader Stefan Borgwardt Marcel Lippmann

LTCS-Report 13-01

Postal Address:
Lehrstuhl für Automatentheorie
Institut für Theoretische Informatik
TU Dresden
01062 Dresden

http://lat.inf.tu-dresden.de

Visiting Address:
Nöthnitzer Str. 46
Dresden

On the Complexity of Temporal Query Answering

Franz Baader, Stefan Borgwardt, and Marcel Lippmann

Institute of Theoretical Computer Science
Technische Universität Dresden, Germany
{baader,stefborg,lippmann}@tcs.inf.tu-dresden.de

Abstract

Ontology-based data access (OBDA) generalizes query answering in databases towards deduction since (i) the fact base is not assumed to contain complete knowledge (i.e., there is no closed world assumption), and (ii) the interpretation of the predicates occurring in the queries is constrained by axioms of an ontology. OBDA has been investigated in detail for the case where the ontology is expressed by an appropriate Description Logic (DL) and the queries are conjunctive queries. Motivated by situation awareness applications, we investigate an extension of OBDA to the temporal case. As query language we consider an extension of the well-known propositional temporal logic LTL where conjunctive queries can occur in place of propositional variables, and as ontology language we use the prototypical expressive DL $\mathcal{ALC}$. For the resulting instance of temporalized OBDA, we investigate both data complexity and combined complexity of the query entailment problem.

Contents

1 Introduction

Situation awareness tools [BBB+09, End95] try to help the user to detect certain situations within a running system. Here “system” is seen in a broad sense: it may be a computer system, air traffic observed by radar, or a patient in an intensive care unit. From an abstract point of view, the system is observed by certain “sensors” (e.g., heart-rate and blood pressure monitors for a patient), and the results of sensing are stored in a fact base. Based on the information available in the fact base, the situation awareness tool is supposed to detect certain predefined situations (e.g., heart-rate very high and blood pressure low), which require a reaction (e.g., fetch a doctor or give medication).

In a simple setting, one could realize such a tool by using standard database techniques: the information obtained from the sensors is stored in a relational database, and the situations to be recognized are specified by queries in an appropriate query language (e.g., conjunctive queries [AHV95]). However, in general we cannot assume that the sensors provide us with a complete description of the current state of the system, and thus the closed world assumption (CWA) employed by database systems (where facts not occurring in the database are assumed to be false) is not appropriate (since there may be facts for which it is not known whether they are true or false). In addition, though one usually does not have a complete specification of the working of the system (e.g., a complete biological model of a human patient), one has some knowledge about how the system works. This knowledge can be used to formulate constraints on the interpretation of the predicates used in the queries, which may cause more answers to be found.

Ontology-based data access [DEFS99, PCDG+08] addresses these requirements. The fact base is viewed to be a Description Logic ABox (which is not interpreted with the CWA), and an ontology, also formulated in an appropriate DL, constrains the interpretations of unary and binary predicates, called concepts and roles in the DL community. As an example, assume that the ABox $\mathcal{A}$ contains the following assertions about the patient Bob:

$$\begin{array}{l} \mathit{systolic\_pressure}(\mathit{BOB}, \mathit{P1}), \quad \mathit{High\_pressure}(P1), \\ \qquad \mathit{history}(\mathit{BOB}, H1), \quad \mathit{Hypertension}(H1), \quad \mathit{Male}(\mathit{BOB}) \end{array}$$

which say that Bob has high blood pressure (obtained from sensor data), and is male and has a history of hypertension (obtained from the patient records). In addition, we have an ontology that says 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 for a heart attack:

$$\begin{array}{rcl} \exists \mathit{systolic\_pressure}.\mathit{High\_pressure} & \sqsubseteq & \exists \mathit{finding}.\mathit{Hypertension} \\ \exists \mathit{finding}.\mathit{Hypertension} \sqcap \exists \mathit{history}.\mathit{Hypertension} & \sqsubseteq & \exists \mathit{risk}.\mathit{Myocardial\_infarction} \end{array}$$

The situation we want to recognize for a given patient x is whether this patient is a male person that is at risk for a heart attack. This situation can be described by the conjunctive query $\exists y.\mathit{risk}(x, y) \wedge \mathit{Myocardial\_infarction}(y) \wedge \mathit{Male}(x)$. Given the information in the ABox and the axioms in the ontology, we can derive that Bob satisfies this query, i.e., he is a *certain answer* of the query. Obviously, without the ontology this answer could not be derived.

The complexity of OBDA, i.e., the complexity of checking whether a given tuple of individuals is a certain answer of a conjunctive query in an ABox w.r.t. an ontology, has been investigated in detail for cases where the ontology is expressed in an appropriate DL 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 query and ontology are usually relatively small, whereas the size of the data may be huge. In the database setting (where there is no ontology and CWA is used), answering conjunctive queries 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 instance, for the well-known DL $\mathcal{ALC}$, OBDA is ExpTime-complete w.r.t. combined complexity and co-NP-complete w.r.t. data complexity [CDL98, Lut08a, CDL$^+$06]. For this reason, more light-weight DLs have been developed, for which the data complexity of OBDA is still in AC^0 and for which computing certain answers can be reduced to answering conjunctive queries in the database setting [CDL$^+$09].

Unfortunately, OBDA as described until now is not sufficient to achieve situation awareness. The reason is that the situations we want to recognize may depend on states of the system at different time points. For example, assume that we want to find male patients that have a history of hypertension, i.e., patients that are male and at some previous time point had hypertension.[1] In order to express this kind of temporal queries, we propose to extend the well-known propositional temporal logic LTL [Pnu77] by allowing the use of conjunctive queries in place of propositional variables. For example, male patients with a history of hypertension can then be described by the query

$$\mathit{Male}(x) \wedge \bigcirc^- \Diamond^- (\exists y.\mathit{finding}(x, y) \wedge \mathit{Hypertension}(y)),$$

where $\bigcirc^-$ stands for "previous" and $\Diamond^-$ stands for "sometime in the past." The query language obtained this way extends the temporal description logic $\mathcal{ALC}$-LTL introduced and investigated in [BGL12]. In $\mathcal{ALC}$-LTL, only concept and role assertions (i.e., very restricted conjunctive queries without variables and existential quantification) can be used in place of propositional variables. As in [BGL12], we

[1] Whereas in the previous example we have assumed that a history of hypertension was explicitly noted in the patient records, we now want to derive this information from previously stored information about blood pressure, etc.

also consider rigid concepts and roles, i.e., concepts and roles whose interpretation does not change over time. 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.

Our overall setting for recognizing situations will thus be the following. In addition to a global ontology $\mathcal{T}$ (which describes properties of the system that hold at every time point, using the DL $\mathcal{ALC}$), 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 recognized is expressed by a temporal conjunctive query, as introduced above, which is evaluated w.r.t. the current time point n. We will investigate both the combined and the data complexity of this temporal extension of OBDA in three different settings: (i) both concepts and roles may be rigid; (ii) only concepts may be rigid; and (iii) neither concepts nor roles are allowed to be rigid. For the combined complexity, the obtained complexity results are identical to the ones for $\mathcal{ALC}$-LTL, though the upper bounds are considerably harder to show. For the data complexity, the results for the settings (ii) and (iii) coincides with the one for atemporal OBDA (co-NP-complete). For the setting (i), we can show that the data complexity is in ExpTime (in contrast to 2-ExpTime-completeness for the combined complexity), but we do not have a matching lower bound.

2 Preliminaries

In this section, we present the preliminaries that we need in this report.

2.1 Description Logics

Description Logics (DLs) are a family of knowledge representation formalisms (for an introduction, see [BCM$^+$03]). While in principle our temporal query language can be parameterized with any DL, in this report we focus on $\mathcal{ALC}$ [SS91] and its extension with role conjunctions $\mathcal{ALC}^\cap$ as prototypical expressive DLs.

The syntax of $\mathcal{ALC}^\cap$ is defined as follows.

Definition 2.1 (syntax of $\mathcal{ALC}^\cap$)**.** *Let N_C, N_R, and N_I, respectively, be non-empty, pairwise disjoint sets of* concept names*,* role names*, and* individual names*. The set of* concept descriptions *(or* concepts*) is the smallest set such that*

- *all concept names $A \in N_C$ are concepts, and*
- *if C, D are concepts, and $r \in N_R$, then $\neg C$ (negation), $C \sqcap D$ (conjunction), and $\exists(r_1 \cap \dots \cap r_\ell).C$ (existential restriction) are also concepts.*

A general concept inclusion (GCI) *is of the form* $C \sqsubseteq D$*, where* C, D *are concepts, and an* assertion *is of the form* $C(a)$ *or* $(r_1 \cap \cdots \cap r_\ell)(a, b)$ *with* $\ell > 0$*, where* C *is a concept,* $r_1, \ldots, r_\ell \in N_R$*, and* $a, b \in N_I$*. We call both GCIs and assertions* axioms*.*

A Boolean combination of axioms is called a Boolean knowledge base*, i.e.,*

- *every axiom is a Boolean knowledge base and*
- *if* $\mathcal{B}_1, \mathcal{B}_2$ *are Boolean knowledge bases, then so are* $\neg\mathcal{B}_1$ *and* $\mathcal{B}_1 \wedge \mathcal{B}_2$*.*

A TBox *(or* ontology*) is a finite set of GCIs and an* ABox *is a finite set of assertions.*

We denote by $\mathsf{Ind}(\mathcal{B})$ the set of individual names that occur in the Boolean knowledge base $\mathcal{B}$. As usual, we use the concept $C \sqcup D$ (disjunction) as an abbreviation for the concept $\neg(\neg C \sqcap \neg D)$, the concept $\forall(r_1 \cap \cdots \cap r_\ell).C$ (value restriction) as an abbreviation for $\neg(\exists(r_1 \cap \cdots \cap r_\ell).\neg C)$, the concept $\top$ (top) as abbreviation for an arbitrary (but fixed) tautology such as $A \sqcup \neg A$ for $A \in N_{\mathrm{C}}$, and the concept $\bot$ (bottom) as abbreviation for $\neg\top$.

The semantics of $\mathcal{ALC}^\cap$ is defined in a model-theoretic way.

Definition 2.2 (semantics of $\mathcal{ALC}^\cap$)**.** *An interpretation is a pair* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$*, where* $\Delta^\mathcal{I}$ *is a non-empty set (called* domain*), and* $\cdot^\mathcal{I}$ *is a function that assigns to every* $A \in N_C$ *a set* $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$*, to every* $r \in N_R$ *a binary relation* $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$*, and to every* $a \in N_I$ *an element* $a^\mathcal{I} \in \Delta^\mathcal{I}$*.*

This function is extended to concept descriptions as follows:

- $(\neg C)^\mathcal{I} := \Delta^\mathcal{I} \setminus C^\mathcal{I}$*;*
- $(C \sqcap D)^\mathcal{I} := C^\mathcal{I} \cap D^\mathcal{I}$*; and*
- $(\exists(r_1 \cap \cdots \cap r_\ell).C)^\mathcal{I} := \{d \in \Delta^\mathcal{I} \mid$ *there is an* $e \in \Delta^\mathcal{I}$ *with* $(d, e) \in r_1^\mathcal{I} \cap \cdots \cap r_\ell^\mathcal{I}$ *and* $e \in C^\mathcal{I}\}$.

The interpretation $\mathcal{I}$ *is a* model *of the axiom* α *if*

- $C^\mathcal{I} \subseteq D^\mathcal{I}$ *if* $\alpha = C \sqsubseteq D$*;*
- $a^\mathcal{I} \in C^\mathcal{I}$ *if* $\alpha = C(a)$*; and*
- $(a^\mathcal{I}, b^\mathcal{I}) \in r_1^\mathcal{I} \cap \cdots \cap r_\ell^\mathcal{I}$ *if* $\alpha = (r_1 \cap \cdots \cap r_\ell)(a, b)$*.*

We write $\mathcal{I} \models \alpha$ if $\mathcal{I}$ is a model of the axiom α, $\mathcal{I} \models \mathcal{T}$ if $\mathcal{I}$ is a model of all GCIs in the TBox $\mathcal{T}$, and $\mathcal{I} \models \mathcal{A}$ if $\mathcal{I}$ is a model of all assertions in the ABox $\mathcal{A}$.

The notion of a model is extended to Boolean $\mathcal{ALC}^{\cap}$-knowledge bases as follows: $\mathcal{I} \models \neg\mathcal{B}$ iff $\mathcal{I} \not\models \mathcal{B}$, and $\mathcal{I} \models \mathcal{B}_1 \wedge \mathcal{B}_2$ iff $\mathcal{I} \models \mathcal{B}_1$ and $\mathcal{I} \models \mathcal{B}_2$. We say that the Boolean $\mathcal{ALC}^{\cap}$-knowledge base $\mathcal{B}$ is consistent *iff it has a model.*

We assume that all interpretations $\mathcal{I}$ satisfy the *unique name assumption (UNA)*, i.e., for all $a, b \in N_\mathrm{I}$ with $a \neq b$ we have $a^\mathcal{I} \neq b^\mathcal{I}$.

The syntax and semantics of the DL $\mathcal{ALC}$ is obtained from $\mathcal{ALC}^{\cap}$ by restricting the variable ℓ to $\ell = 1$ in the above definitions, i.e., role conjunctions are disallowed.

2.2 Temporal Conjunctive Queries

We now introduce a temporal query language that generalizes a subset of first-order queries called conjunctive queries [AHV95, CM77] and the temporal DL $\mathcal{ALC}$-LTL [BGL12]. In this section, we focus on the DL $\mathcal{ALC}$, but in principle, the temporal query language can be defined using any other DL.

In the following, we assume (as in [BGL12]) that a subset of the concept and role names is designated as being rigid. The intuition is that the interpretation of the rigid names is not allowed to change over time. Let N_RC denote the *rigid concept names*, and N_RR the *rigid role names* with $N_\mathrm{RC} \subseteq N_\mathrm{C}$ and $N_\mathrm{RR} \subseteq N_\mathrm{R}$. We sometimes call the names in $N_\mathrm{C} \setminus N_\mathrm{RC}$ and $N_\mathrm{R} \setminus N_\mathrm{RR}$ *flexible.* All individual names are implicitly assumed to be rigid, i.e., an individual always keeps its name.

Definition 2.3. *A* temporal knowledge base *(TKB) $\mathcal{K} = \langle(\mathcal{A}_i)_{0\leq i\leq n}, \mathcal{T}\rangle$ consists of a finite sequence of ABoxes $\mathcal{A}_i$ and an TBox $\mathcal{T}$, where the ABoxes $\mathcal{A}_i$ can only contain concept names that also occur in $\mathcal{T}$.*[2]

*Let $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$ be an infinite sequence of interpretations $\mathcal{I}_i = (\Delta, \cdot^{\mathcal{I}_i})$ over a fixed non-empty domain Δ (*constant domain assumption*). Then $\mathfrak{I}$ is a* model *of $\mathcal{K}$ (written $\mathfrak{I} \models \mathcal{K}$) if*

- $\mathcal{I}_i \models \mathcal{A}_i$ *for all $i, 0 \leq i \leq n$,*
- $\mathcal{I}_i \models \mathcal{T}$ *for all $i \geq 0$, and*
- $\mathfrak{I}$ *respects rigid names, i.e., $x^{\mathcal{I}_i} = x^{\mathcal{I}_j}$ for all $x \in N_I \cup N_{RC} \cup N_{RR}$ and all $i, j \geq 0$.*

We denote by $\mathsf{Ind}(\mathcal{K})$ the set of all individual names occurring in the TKB $\mathcal{K}$. As query language, we use a temporal extension of conjunctive queries.

[2]This restriction is motivated by the intuition that the TBox $\mathcal{T}$ contains all concepts relevant for a knowledge domain, while the ABoxes $\mathcal{A}_i$ contain observations of the real world that are formulated using the terminology given by $\mathcal{T}$.

Definition 2.4. *Let N_V be a set of variables. A* conjunctive query (CQ) *is of the form $\phi = \exists y_1, \ldots, y_m.\psi$, where $y_1, \ldots, y_m \in N_V$ and ψ is a (possibly empty) finite conjunction of* atoms *of the form*

- *$A(z)$ for $A \in N_C$ and $z \in N_V \cup N_I$* (concept atom)*; or*
- *$r(z_1, z_2)$ for $r \in N_R$ and $z_1, z_2 \in N_V \cup N_I$* (role atom)*.*

The empty conjunction is denoted by true*.* Temporal conjunctive queries (TCQs) *are built from CQs as follows:*

- *each CQ 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),*
  - *$\bigcirc\phi_1$ (next), $\bigcirc^-\phi_1$ (previous),*
  - *$\phi_1 \mathsf{U} \phi_2$ (until), and $\phi_1 \mathsf{S} \phi_2$ (since).*

We denote the set of individuals occurring in a TCQ 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 a TCQ ϕ with $\mathsf{FVar}(\phi) = \emptyset$ a *Boolean TCQ*.

As usual, we use the following abbreviations: $\phi_1 \vee \phi_2$ (disjunction) for $\neg(\neg\phi_1 \wedge \neg\phi_2)$, $\Diamond\phi$ (eventually) for $\mathsf{true}\mathsf{U}\phi$, $\Box\phi$ (always) for $\neg\Diamond\neg\phi$, and analogously for the past: $\Diamond^-\phi$ for $\mathsf{true}\mathsf{S}\phi$, and $\Box^-\phi$ for $\neg\Diamond^-\neg\phi$.

A *union of CQs* is a disjunction of CQs.

For our purposes, it is sufficient to define the semantics of CQs and TCQs only for Boolean queries. As usual, it is given using the notion of homomorphisms [CM77].

Definition 2.5. *Let $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$ be an interpretation and ψ be a Boolean CQ. A mapping $\pi\colon \mathsf{Var}(\psi) \cup \mathsf{Ind}(\psi) \to \Delta$ is a* homomorphism *of ψ into $\mathcal{I}$ if*

- *$\pi(a) = a^{\mathcal{I}}$ for all $a \in \mathsf{Ind}(\psi)$;*
- *$\pi(z) \in A^{\mathcal{I}}$ for all concept atoms $A(z)$ in ψ; and*
- *$(\pi(z_1), \pi(z_2)) \in r^{\mathcal{I}}$ for all role atoms $r(z_1, z_2)$ in ψ.*

We say that $\mathcal{I}$ is a model *of ψ (written $\mathcal{I} \models \psi$) if there is such a homomorphism.*

Let now ϕ be a Boolean TCQ. For an infinite sequence of interpretations $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$ and $i \geq 0$, we define $\mathfrak{I}, i \models \phi$ by induction on the structure of ϕ:

$$
\begin{array}{lll}
\mathfrak{I}, i \models \exists y_1, \ldots, y_m.\psi & \text{iff} & \mathcal{I}_i \models \exists y_1, \ldots, y_m.\psi \\
\mathfrak{I}, i \models \neg\phi_1 & \text{iff} & \mathfrak{I}, i \not\models \phi_1 \\
\mathfrak{I}, i \models \phi_1 \wedge \phi_2 & \text{iff} & \mathfrak{I}, i \models \phi_1 \text{ and } \mathfrak{I}, i \models \phi_2 \\
\mathfrak{I}, i \models \bigcirc\phi_1 & \text{iff} & \mathfrak{I}, i+1 \models \phi_1 \\
\mathfrak{I}, i \models \bigcirc^-\phi_1 & \text{iff} & i > 0 \text{ and } \mathfrak{I}, i-1 \models \phi_1 \\
\mathfrak{I}, i \models \phi_1 \mathsf{U} \phi_2 & \text{iff} & \text{there is some } k \geq i \text{ such that } \mathfrak{I}, k \models \phi_2 \\
 & & \text{and } \mathfrak{I}, j \models \phi_1 \text{ for all } j,\ i \leq j < k \\
\mathfrak{I}, i \models \phi_1 \mathsf{S} \phi_2 & \text{iff} & \text{there is some } k,\ 0 \leq k \leq i \text{ such that } \mathfrak{I}, k \models \phi_2 \\
 & & \text{and } \mathfrak{I}, j \models \phi_1 \text{ for all } j,\ k < j \leq i
\end{array}
$$

Given a TKB $\mathcal{K} = \langle (\mathcal{A}_i)_{0\leq i\leq n}, \mathcal{T}\rangle$, 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. $\mathcal{K}$.*

It should be noted that Boolean TCQs generalize $\mathcal{ALC}$-LTL formulae as introduced in [BGL12]. More precisely, every TCQ that contains only assertions instead of general CQs and contains no past operators ($\bigcirc^-$ or S) is an $\mathcal{ALC}$-LTL formula. $\mathcal{ALC}$-LTL formulae may additionally contain local GCIs $C \sqsubseteq D$. Such a GCI can, however, be expressed by the TCQ $\neg\exists x.A(x)$ if we add the (global) GCIs $A \sqsubseteq C \sqcap \neg D$, $C \sqcap \neg D \sqsubseteq A$ to the TBox. Thus, TCQs together with a global TBox can express all $\mathcal{ALC}$-LTL formulae. TCQs are more expressive than $\mathcal{ALC}$-LTL formulae since CQs like $\exists y.r(y, y)$, which says that there is a loop in the model without naming the individual which has the loop, can clearly not be expressed in $\mathcal{ALC}$.

Before defining the main inference problem for TCQs to be investigated in this report, we introduce some notation that will be used later on.

The *propositional abstraction* $\widehat{\phi}$ of a TCQ ϕ is built by replacing each CQ occurring in ϕ by a propositional variable such that there is a 1–1 relationship between the CQs $\alpha_1, \ldots, \alpha_m$ occurring in ϕ and the propositional variables $p_1, \ldots, p_m$ occurring in $\widehat{\phi}$. The formula $\widehat{\phi}$ obtained this way is a propositional LTL-formula [Pnu77].

Definition 2.6. *Let $\{p_1, \ldots, p_m\}$ be a finite set of* propositional variables. *An* LTL-formula *ϕ is built from these variables using the constructors* negation *($\neg\phi$),* conjunction *($\phi \wedge \phi'$),* next *($\bigcirc\phi$),* previous *($\bigcirc^-\phi$),* until *($\phi\mathsf{U}\phi'$), and* since *($\phi\mathsf{S}\phi'$).*

An LTL-structure *is an infinite sequence $\mathfrak{J} = (w_i)_{i\geq 0}$ of* worlds *$w_i \subseteq \{p_1, \ldots, p_m\}$. The propositional variable p_j is* satisfied *by $\mathfrak{J}$ at time point $i \geq 0$ (written $\mathfrak{J}, i \models p_j$) iff $p_j \in w_i$. The satisfaction of a complex propositional LTL-formula by an LTL-structure is defined as in Definition 2.5.*

Note that what we introduced above would usually be called Past-LTL, as LTL is normally defined using only the operators $\bigcirc$ and U [Pnu77].

A *CQ-literal* is a Boolean CQ ψ or a negated Boolean CQ $\neg\psi$. We will often deal with conjunctions ϕ of CQ-literals. Since such a formula ϕ contains no temporal operators, the satisfaction of ϕ by an infinite sequence of interpretations $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$ at time point i only depends on the interpretation $\mathcal{I}_i$. For simplicity, we then often write $\mathcal{I}_i \models \phi$ instead of $\mathfrak{I}, i \models \phi$. By the same argument, we use this notation also for unions of CQs. In this context, it is sufficient to deal with *classical* knowledge bases $\mathcal{K} = \langle \mathcal{A}, \mathcal{T} \rangle$, i.e., temporal knowledge bases with only one ABox, and we similarly write $\mathcal{I}_0 \models \mathcal{K}$ instead of $\mathfrak{I}, 0 \models \mathcal{K}$.

A simplifying assumption we make in the remainder of this report 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 [RG10].

Definition 2.7. *A Boolean CQ ϕ is called* 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 all i, $1 \leq i < n$, there is $r \in N_R$ 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 is connected.*

It follows from a result in [Tes01], that we can assume Boolean TCQs to contain only connected CQs without loss of generality. Indeed, if a Boolean TCQ ϕ contains a CQ ψ that is not connected, we can replace ψ by the conjunction $\psi_1 \wedge \dots \wedge \psi_n$, 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 the following we always assume that Boolean TCQs contain only connected CQs.

3 The Entailment Problem

We are now ready to introduce the central reasoning problems of this report, i.e., the problem of finding so-called certain answers to TCQs and the corresponding decision problems.

Definition 3.1. *Let ϕ be a TCQ and $\mathcal{K} = \langle (\mathcal{A}_i)_{0\leq i\leq n}, \mathcal{T} \rangle$ a temporal knowledge base. The mapping $\mathfrak{a}\colon \mathsf{FVar}(\phi) \to \mathsf{Ind}(\mathcal{K})$ is a* certain answer *to ϕ w.r.t. $\mathcal{K}$ if for every $\mathfrak{I} \models \mathcal{K}$, we have $\mathfrak{I}, n \models \mathfrak{a}(\phi)$, where $\mathfrak{a}(\phi)$ denotes the Boolean TCQ that is obtained from ϕ by replacing the free variables according to $\mathfrak{a}$.*

The corresponding decision problem is the recognition problem*, i.e., given $\mathfrak{a}$, ϕ, and $\mathcal{K}$, to check whether $\mathfrak{a}$ is a certain answer to ϕ w.r.t. $\mathcal{K}$. The* (query) entailment problem *is to decide for a Boolean TCQ ϕ and a temporal knowledge base $\mathcal{K} = \langle (\mathcal{A}_i)_{0\leq i\leq n}, \mathcal{T} \rangle$ whether every model $\mathfrak{I}$ of $\mathcal{K}$ satisfies $\mathfrak{I}, n \models \phi$ (written $\mathcal{K} \models \phi$).*

Note that, for a TCQ ϕ, a temporal knowledge base $\mathcal{K}$, and $i \geq 0$, one can compute all certain answers by enumerating all mappings $\mathfrak{a}\colon \mathsf{FVar}(\phi) \to \mathsf{Ind}(\mathcal{K})$ and then solving the recognition problem for each $\mathfrak{a}$. Since there are $|\mathsf{Ind}(\mathcal{K})|^{|\mathsf{FVar}(\phi)|}$ such mappings, in order to compute the set of certain answers, we have to solve the recognition problem exponentially often.

As described in the introduction, in a situation awareness tool we want to solve the recognition problem for temporal knowledge bases $\mathcal{K} = \langle (\mathcal{A}_i)_{0 \leq i \leq n}, \mathcal{T} \rangle$ and TCQs. The intuition is that the ABoxes $\mathcal{A}_i$ describe our observations about the system's states at time points $i = 0, \ldots, n$, where n is the current time point, and the TCQ describes the situation we want to recognize at time point n for a given instantiation of the free variables in the query (e.g., a certain patient).

Obviously, the entailment problem is a special case of the recognition problem, where $\mathfrak{a}$ is the empty mapping. Conversely, the recognition problem for $\mathfrak{a}$, ϕ, and $\mathcal{K}$ is the same as the entailment problem for $\mathfrak{a}(\phi)$ and $\mathcal{K}$. Thus, these two problems have the same complexity.

Therefore, it is sufficient to analyze the complexity of the *entailment problem*. We consider two kinds of complexity measures: combined complexity and data complexity. For the *combined complexity*, all parts of the input, i.e., the TCQ ϕ and the temporal knowledge base $\mathcal{K}$, are taken into account. For the *data complexity*, the TCQ ϕ and the TBox $\mathcal{T}$ are assumed to be constant, and the complexity is measured only w.r.t. the data, i.e., the sequence of ABoxes. As usual when investigating the data complexity of OBDA [CDL+09], we assume that the ABoxes occurring in a temporal knowledge base and the query contain only concept and role names that also occur in the global TBox.

It turns out that it is actually easier to analyze the complexity of the complement of this problem, i.e., *non-entailment* $\mathcal{K} \not\models \phi$. This problem has the same complexity as the *satisfiability problem*. 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 $\mathcal{K} \not\models \neg\phi$.

We first analyze the (atemporal) special case of the satisfiability problem where ϕ is a conjunction of CQ-literals. The following result will turn out to be useful also for analyzing the general case.

Theorem 3.2. *Let $\mathcal{K} = \langle \mathcal{A}, \mathcal{T} \rangle$ be a knowledge base and ϕ be a conjunction of CQ-literals. Then deciding whether ϕ has a model w.r.t. $\mathcal{K}$ is* ExpTime*-complete w.r.t. combined complexity and* NP*-complete w.r.t. data complexity.*

Proof. For the lower bound for combined complexity, we reduce the ExpTime-hard concept satisfiability problem for $\mathcal{ALC}$ w.r.t. TBoxes [Sch91]. Consider a concept C and a TBox $\mathcal{T}$. Let $\mathcal{T}' := \mathcal{T} \cup \{A \sqsubseteq C, C \sqsubseteq A\}$, where A does not occur in $\mathcal{T}$, and let $\phi' := \exists x.A(x)$. Obviously, C is satisfiable w.r.t. $\mathcal{T}$ iff there is an interpretation $\mathcal{I}$ with $\mathcal{I} \models \langle \emptyset, \mathcal{T}' \rangle$ and $\mathcal{I} \models \phi'$.

For the remaining lower bound, we know that already for a Boolean conjunctive query ψ the query entailment problem is co-NP-hard w.r.t. data complexity [CDL+06]. This problem is obviously a special case of the complement of our problem.

To check whether there is an interpretation $\mathcal{I}$ with $\mathcal{I} \models \mathcal{K}$ and $\mathcal{I} \models \phi$, we reduce this problem to a query non-entailment problem of known complexity. Let

$$\phi = \chi_1 \wedge \ldots \wedge \chi_\ell \wedge \neg\rho_1 \wedge \ldots \wedge \neg\rho_m$$

for Boolean CQs $\chi_1, \ldots, \chi_\ell$, $\rho_1, \ldots, \rho_m$. First, we instantiate the non-negated CQs $\chi_1, \ldots, \chi_\ell$ by omitting the existential quantifiers and replacing the variables by fresh individual names. The set $\mathcal{A}'$ of all resulting atoms can thus be viewed as an additional ABox that restricts the interpretation $\mathcal{I}$.

We now show that the existence of an interpretation $\mathcal{I}$ with $\mathcal{I} \models \mathcal{K}$ and $\mathcal{I} \models \phi$ is equivalent to the existence of an interpretation $\mathcal{I}'$ with $\mathcal{I}' \models \langle \mathcal{A} \cup \mathcal{A}', \mathcal{T} \rangle$ and $\mathcal{I}' \models \neg\rho_1 \wedge \ldots \wedge \neg\rho_m$.

The "if" direction is easy to see. For the "only if" direction, assume that $\mathcal{I} \models \mathcal{K}$ and $\mathcal{I} \models \phi$. We extend $\mathcal{I}$ to a model $\mathcal{I}'$ that additionally satisfies the assertions in $\mathcal{A}'$. The idea is that we can define the interpretation of the fresh individual names in $\mathcal{A}'$ according to the homomorphisms that must exist from the non-negated CQs in ϕ into $\mathcal{I}$. Assume now that two of these individual names a, a' are then interpreted in $\mathcal{I}'$ by the same individual $x \in \Delta^{\mathcal{I}'}$, thus violating the UNA. We can introduce a fresh copy x' of x into $\mathcal{I}'$ and interpret the concept and role names as for x, such that we have $x' \in A^{\mathcal{I}'}$ iff $x \in A^{\mathcal{I}'}$ for any $A \in N_{\mathrm{C}}$, and $(x', y) \in r^{\mathcal{I}'}$ iff $(x, y) \in r^{\mathcal{I}'}$ as well as $(y, x') \in r^{\mathcal{I}'}$ iff $(y, x) \in r^{\mathcal{I}'}$ and $(x', x') \in r^{\mathcal{I}'}$ iff $(x, x) \in r^{\mathcal{I}'}$ for any $r \in N_{\mathrm{R}}$ and $y \in \Delta^{\mathcal{I}'} \setminus \{x, x'\}$. We also change the interpretation of a to x' instead of x. The resulting interpretation is still a model of the original knowledge base $\langle \mathcal{A}, \mathcal{T} \rangle$ and the instantiated atoms in $\mathcal{A}'$. Note also that there can still be no homomorphism from any of the CQs $\rho_1, \ldots, \rho_m$ into $\mathcal{I}'$ since they cannot contain a and a' and or distinguish between unnamed individuals satisfying the same concept names and having the same role connections. After we have done this construction for all pairs of fresh individual names violating the UNA, we have constructed a model of $\langle \mathcal{A} \cup \mathcal{A}', \mathcal{T} \rangle$ and $\neg\rho_1 \wedge \cdots \wedge \neg\rho_m$.

The above problem is thus equivalent to finding an interpretation $\mathcal{I}$ with $\mathcal{I} \models \langle \mathcal{A} \cup \mathcal{A}', \mathcal{T} \rangle$ and $\mathcal{I} \not\models \rho$, where $\rho = \rho_1 \vee \cdots \vee \rho_m$ is the union of Boolean CQs that results from negating the conjunction of all negated CQs in ϕ. This is the same as asking whether the knowledge base $\langle \mathcal{A} \cup \mathcal{A}', \mathcal{T} \rangle$ does not entail the union of conjunctive queries ρ.

The complexity of this kind of entailment problems is known: it is ExpTime-complete w.r.t. combined complexity [CDL98, Lut08a] and co-NP-complete w.r.t. data complexity [OCE06]. □

In the remainder of this report, we will present several constructions, most of

	Data complexity	*Combined complexity*
$N_{\mathrm{RC}} = N_{\mathrm{RR}} = \emptyset$	co-NP-complete (Cor. 3.4 and Thm. 3.14)	ExpTime-complete (Thms. 3.5 and 3.14)
$N_{\mathrm{RC}} \neq \emptyset$, $N_{\mathrm{RR}} = \emptyset$	co-NP-complete (Cor. 3.4 and Thm. 4.2)	co-NExpTime-complete (Thms. 3.5 and 5.8)
$N_{\mathrm{RR}} \neq \emptyset$	co-NP-hard/in ExpTime (Cor. 3.4 and Thm. 3.16)	2-ExpTime-complete (Thms. 3.5 and 3.16)

Table 3.3: The complexity of the entailment problem.

which use the above theorem, to derive the complexity results shown in Table 3.3 for the entailment problem in general. 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.

3.1 Lower Bounds for the Entailment Problem

For data complexity, we obtain the lower bounds as a corollary of Theorem 3.2.

Corollary 3.4. *The entailment problem is co-*NP*-hard w.r.t. data complexity.*

Proof. Theorem 3.2 states that for conjunctions of CQ-literals ϕ and atemporal knowledge bases $\mathcal{K}$, deciding whether ϕ has a model w.r.t. $\mathcal{K}$ is NP-complete w.r.t. data complexity. Since ϕ 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 entailment problem in all the cases in Table 3.3. □

For the combined complexity, we get the lower bounds by a simple reduction of the satisfiability problem of the temporal DL $\mathcal{ALC}$-LTL [BGL12].

Theorem 3.5. *The entailment problem w.r.t. combined complexity is*

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

Proof. The satisfiability problem of the temporal DL $\mathcal{ALC}$-LTL is ExpTime-complete without rigid concept and role names, NExpTime-complete w.r.t. rigid concept names, and 2-ExpTime-complete w.r.t. rigid concept and role names (see [BGL12]).

Let ϕ be an $\mathcal{ALC}$-LTL formula, let $C_1 \sqsubseteq D_1, \ldots, C_p \sqsubseteq D_p$ be all GCIs occurring in ϕ, and let $E_1(a_1), \ldots, E_m(a_m)$ be all concept assertions occurring in ϕ. Let ψ be the Boolean TCQ obtained from ϕ by replacing each $C_i \sqsubseteq D_i$ with $\neg(\exists x.A_i(x))$ and each E_j with B_j, where A_i, B_j are assumed to not occur in ϕ, for i, j, $1 \leq i \leq p$, $1 \leq j \leq m$. Furthermore, we define

$$\begin{aligned}\mathcal{T} := \{A_i \sqsubseteq C_i \sqcap \neg D_i \mid 1 \leq i \leq p\} \cup \{C_i \sqcap \neg D_i \sqsubseteq A_i \mid 1 \leq i \leq p\} \cup \\ \{B_j \sqsubseteq E_j \mid 1 \leq j \leq m\} \cup \{E_j \sqsubseteq B_j \mid 1 \leq j \leq m\}.\end{aligned}$$

It is easy to see that ϕ is satisfiable iff $\langle \emptyset, \mathcal{T} \rangle \not\models \neg\psi$. We have thus reduced the satisfiability problem of $\mathcal{ALC}$-LTL to the *non-entailment* problem, which yields the claimed lower bounds. □

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 we have from Theorem 3.5. For the latter, unfortunately we cannot match the lower bound of co-NP in the case where we have rigid role names. While the results need to deal with CQs in an appropriate way, the basic ideas to prove them are similar to those presented for $\mathcal{ALC}$-LTL in [BGL12].

3.2 Upper Bounds for the Entailment Problem

We now describe an approach to solving the satisfiability (and thus the non-entailment problem) in general to obtain the upper bounds of Table 3.3. The basic idea is to reduce the problem to two separate satisfiability problems, similar to what was done for $\mathcal{ALC}$-LTL in Lemma 4.3 of [BGL12].

Let $\mathcal{K} = \langle (\mathcal{A}_i)_{0 \leq i \leq n}, \mathcal{T} \rangle$ be a TKB and ϕ be a Boolean TCQ, for which we want to decide whether ϕ has a model w.r.t. $\mathcal{K}$. Recall that the propositional abstraction $\widehat{\phi}$ of ϕ contains the propositional variables $p_1, \ldots, p_m$ in place of the CQs $\alpha_1, \ldots, \alpha_m$ occurring in ϕ. We assume in the following that α_i was replaced by p_i for all i, $1 \leq i \leq m$. We now consider a set $\mathcal{S} \subseteq 2^{\{p_1, \ldots, p_m\}}$, which intuitively specifies the worlds that are allowed to occur in an LTL-structure satisfying $\widehat{\phi}$. To express this restriction, we define the propositional LTL-formula

$$\widehat{\phi}_{\mathcal{S}} := \widehat{\phi} \wedge \Box^- \Box \left(\bigvee_{X \in \mathcal{S}} \left(\bigwedge_{p \in X} p \wedge \bigwedge_{p \notin X} \neg p \right) \right).$$ [3]

An obvious connection between ϕ and $\widehat{\phi}_{\mathcal{S}}$ is formalized in the next lemma.

Lemma 3.6. *If ϕ has a model w.r.t. $\mathcal{K}$, then there is a set $\mathcal{S} \subseteq 2^{\{p_1, \ldots, p_m\}}$ and a propositional LTL-structure that satisfies $\widehat{\phi}_{\mathcal{S}}$ at time point n.*

[3] Note that a formula $\Box^- \Box \psi$ is satisfied iff ψ holds at all time points.

Proof. Let $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$ be a sequence of interpretations that respects rigid names, is a model of $\mathcal{K}$, and satisfies $\mathfrak{I}, n \models \phi$. For each interpretation $\mathcal{I}_i$ of $\mathfrak{I}$, we set

$$X_i := \{p_j \mid 1 \leq j \leq m \text{ and } \mathcal{I}_i \text{ satisfies } \alpha_j\},$$

and then consider the set $\mathcal{S} := \{X_i \mid i \geq 0\}$ induced by $\mathfrak{I}$. The *propositional abstraction* $\widehat{\mathfrak{I}} = (w_i)_{i\geq 0}$ of $\mathfrak{I}$ is now defined by $w_i := X_i$ for all $i \geq 0$. It is easy to check that the fact that $\mathfrak{I}$ satisfies ϕ at time point n implies that $\widehat{\mathfrak{I}}$ satisfies $\widehat{\phi}_{\mathcal{S}}$ at time point n. □

However, guessing a set $\mathcal{S}$ and then testing whether the induced LTL-formula $\widehat{\phi}_{\mathcal{S}}$ is satisfiable at time point n is not sufficient for checking whether ϕ has a model w.r.t. $\mathcal{K}$. We must also check whether the guessed set $\mathcal{S}$ can indeed be induced by some sequence of interpretations that is a model of $\mathcal{K}$. The following definition introduces a condition that need to be satisfied for this to hold.

Definition 3.7. *Given a set $\mathcal{S} = \{X_1, \ldots, X_k\} \subseteq 2^{\{p_1,\ldots,p_m\}}$ and a mapping $\iota\colon \{0, \ldots, n\} \to \{1, \ldots, k\}$, we say that $\mathcal{S}$ is* r-consistent *w.r.t. ι and $\mathcal{K}$ if there exist interpretations $\mathcal{J}_1, \ldots, \mathcal{J}_k$, $\mathcal{I}_0, \ldots, \mathcal{I}_n$ such that*

- *the interpretations share the same domain and respect rigid names;*[4]
- *the interpretations are models of $\mathcal{T}$;*
- *for i, $0 \leq i \leq k$, $\mathcal{J}_i$ is a model of $\chi_i := \bigwedge_{p_j\in X_i} \alpha_j \wedge \bigwedge_{p_j\notin X_i} \neg\alpha_j$; and*
- *for i, $0 \leq i \leq n$, $\mathcal{I}_i$ is a model of $\mathcal{A}_i$ and $\chi_{\iota(i)}$.*

The intuition underlying this definition is the following. The existence of the interpretation $\mathcal{J}_i$ ($1 \leq i \leq k$) ensures that the conjunction χ_i of the CQ-literals specified by X_i is consistent. In fact, a set $\mathcal{S}$ containing a set X_i for which this does not hold cannot be induced by a sequence of interpretations. The interpretations $\mathcal{I}_i$ ($0 \leq i \leq n$) are supposed to constitute the first $n+1$ interpretations in such a sequence. In addition to inducing a set $X_{\iota(i)} \in \mathcal{S}$ and thus satisfying the corresponding conjunction $\chi_{\iota(i)}$, the interpretation $\mathcal{I}_i$ must thus also satisfy the ABox $\mathcal{A}_i$. The first and the second condition ensure that a sequence of interpretations built from $\mathcal{J}_1, \ldots, \mathcal{J}_k$, $\mathcal{I}_0, \ldots, \mathcal{I}_n$ respects rigid names and satisfies the global TBox $\mathcal{T}$. Note that we can use Theorem 3.2 to check whether interpretations satisfying the last three conditions of Definition 3.7 exist. As we will see below, the difficulty lies in ensuring that they also satisfy the first condition.

Satisfaction of the temporal structure of ϕ by a sequence of interpretations built this way is ensured by testing $\widehat{\phi}_{\mathcal{S}}$ for satisfiability, which can basically be done using algorithms for testing satisfiability in propositional LTL [VW94].

[4]This is defined analogously to the case of sequences of interpretations (Definition 2.3).

Lemma 3.8. *The TCQ ϕ has a model w.r.t. the TKB $\mathcal{K}$ iff there is a set $\mathcal{S} = \{X_1, \ldots, X_k\} \subseteq 2^{\{p_1,\ldots,p_m\}}$ and a mapping $\iota\colon \{0, \ldots, n\} \to \{1, \ldots, k\}$ such that*

- *$\mathcal{S}$ is r-consistent w.r.t. ι and $\mathcal{K}$ and*
- *there is an LTL-structure $\mathfrak{J} = (w_i)_{i\geq 0}$ such that $\mathfrak{J}, n \models \widehat{\phi}_{\mathcal{S}}$ and $w_i = X_{\iota(i)}$ for all i, $0 \leq i \leq n$.*

Proof. For the "only if" direction, assume that ϕ has a model w.r.t. $\mathcal{K}$. Thus, there is a sequence of interpretations $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$ of $\mathcal{ALC}$-interpretations with $\mathfrak{I} \models \mathcal{K}$ and $\mathfrak{I}, n \models \phi$. Recall that we have already seen in Lemma 3.6 that $\mathfrak{I}$ induces a set $\mathcal{S} \subseteq 2^{\{p_1,\ldots,p_m\}}$ such that $\widehat{\phi}_{\mathcal{S}}$ is satisfiable at time point n. Let $\mathcal{S} = \{X_1, \ldots, X_k\}$. For each $i \geq 0$, there is an index $\nu_i \in \{1, \ldots, k\}$ such that $\mathcal{I}_i$ induces the set X_{ν_i}, i.e.,

$$X_{\nu_i} = \{p_j \mid 1 \leq j \leq m \text{ and } \mathcal{I}_i \text{ satisfies } \alpha_j\},$$

and, conversely, for each $\nu \in \{1, \ldots, k\}$, there is an index $i \geq 0$ such that $\nu = \nu_i$. We define the mapping ι as follows: $\iota(i) = \nu_i$ for all i, $0 \leq i \leq n$. Let $\widehat{\mathfrak{I}} = (w_i)_{i\geq 0}$ be the propositional abstraction of $\mathfrak{I}$. As argued in Lemma 3.6, $\widehat{\mathfrak{I}}$ is a model of $\widehat{\phi}_{\mathcal{S}}$ at time point n. By definition of ι, X_{ν_i} and $\widehat{\mathfrak{I}}$, we also have $w_i = X_{\iota(i)}$ for all i, $0 \leq i \leq n$.

For i, $1 \leq i \leq k$, the interpretation $\mathcal{J}_i$ is obtained as follows. Let $\ell_1, \ldots, \ell_k$ be such that $\nu_{\ell_1} = 1, \ldots, \nu_{\ell_k} = k$. Now, if we set $\mathcal{J}_i := \mathcal{I}_{\ell_i}$, then it is clear that $\mathcal{J}_i$ is a model of χ_i. It is now easy to see that the interpretations $\mathcal{J}_1, \ldots, \mathcal{J}_k, \mathcal{I}_0, \ldots, \mathcal{I}_n$ satisfy the conditions for r-consistency of $\mathcal{S}$ w.r.t. ι and $\mathcal{K}$.

To show the "if" direction, assume that there is a set $\mathcal{S} = \{X_1, \ldots, X_k\}$, a mapping $\iota\colon \{0, \ldots, n\} \to \{1, \ldots, k\}$, and an LTL-structure $\mathfrak{J} = (w_i)_{i\geq 0}$ such that $\mathfrak{J}$ is a model of $\widehat{\phi}_{\mathcal{S}}$ at time point n and $w_i = X_{\iota(i)}$ for all i, $0 \leq i \leq n$, and $\mathcal{S}$ is r-consistent w.r.t. ι and $\mathcal{K}$. Let $\mathcal{J}_1, \ldots, \mathcal{J}_k, \mathcal{I}_0, \ldots, \mathcal{I}_n$ be the models of $\mathcal{T}$ with the properties of Definition 3.7.

By the definition of $\widehat{\phi}_{\mathcal{S}}$, for every world w_i, there is exactly one index $\nu_i \in \{1, \ldots, k\}$ such that w_i satisfies

$$\bigwedge_{p \in X_{\nu_i}} p \wedge \bigwedge_{p \notin X_{\nu_i}} \neg p.$$

Since 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 sequence of $\mathcal{ALC}$-interpretations respecting rigid names as follows: $\mathfrak{I} := (\mathcal{I}_i)_{i\geq 0}$ where $\mathcal{I}_i := \mathcal{J}_{\nu_i}$ for $i > n$. By Definition 3.7, each $\mathcal{I}_i$ satisfies exactly the CQs specified by the propositional variables in X_{ν_i}. Since $\mathfrak{J}, n \models \widehat{\phi}_{\mathcal{S}}$, this means that $\mathfrak{I}, n \models \phi$. It also follows directly from Definition 3.7 that $\mathfrak{I} \models \mathcal{K}$. Hence, we have that ϕ has model w.r.t. $\mathcal{K}$. □

Since the overall complexity of the satisfiability problem depends on which symbols are allowed to be rigid, we obtain the set $\mathcal{S}$ and the function ι either by enumeration, guessing, or direct construction. Given $\mathcal{S}$ and ι, it remains to check the two conditions of the lemma. This means that, in order to decide satisfiability of ϕ w.r.t. $\mathcal{K}$, we only need to solve the above two satisfiability problems in $\mathcal{ALC}$ and LTL, respectively, similar to what was done for deciding satisfiability in $\mathcal{ALC}$-LTL [BGL12]. For the r-consistency test, we need to use different constructions depending on which symbols are allowed to be rigid. Using these constructions, we obtain the complexity results for the entailment problem shown in Table 3.3. The details can be found in later sections. First, we focus on the second condition of Lemma 3.8.

3.2.1 An Automaton for LTL-Satisfiabilty

For the second condition of Lemma 3.8, we construct a generalized Büchi automaton similar to the standard construction for satisfiability of LTL-formulae [WVS83, VW94]. Emptiness of this automaton is equivalent to satisfiability of $\widehat{\phi}_{\mathcal{S}}$.

Definition 3.9. *A* generalized Büchi automaton $\mathcal{G} = (Q, \Sigma, \Delta, Q_0, \mathcal{F})$ *consists of a finite set of states Q, a finite input alphabet Σ, a transition relation $\Delta \subseteq Q \times \Sigma \times Q$, a set $Q_0 \subseteq Q$ of initial states, and a set of sets of final states $\mathcal{F} \subseteq 2^Q$.*

Given an infinite word $w = \sigma_0\sigma_1\sigma_2 \ldots \in \Sigma^\omega$, a run *of $\mathcal{G}$ on w is an infinite 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 all $i \geq 0$. 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 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 generalized Büchi automata is the problem of deciding, given a generalized Büchi automaton $\mathcal{G}$, whether $L_\omega(\mathcal{G}) = \emptyset$ or not.*

We use *generalized* Büchi automata rather than normal ones (where $|\mathcal{F}| = 1$) since this allows for a simpler construction below. It is well-known that a generalized Büchi automaton can be transformed into an equivalent normal one in polynomial time [GPVW96, BK08]. Together with the fact that the emptiness problem for normal Büchi automata can be solved in polynomial time [VW94], this yields a polynomial time bound for the complexity of the emptiness problem for generalized Büchi automata.

To define our automaton, we need the notion of a *type for $\widehat{\phi}$*.

Definition 3.10. *A* sub-literal *of $\widehat{\phi}$ is a sub-formula of $\widehat{\phi}$ or its negation. A set T of sub-literals of $\widehat{\phi}$ is a type for $\widehat{\phi}$ iff the following properties are satisfied:*

1. *for every sub-formula ψ of $\widehat{\phi}$, we have $\psi \in T$ iff $\neg\psi \notin T$;*
2. *for every sub-formula $\psi_1 \wedge \psi_2$ of $\widehat{\phi}$, we have $\psi_1 \wedge \psi_2 \in T$ iff $\{\psi_1, \psi_2\} \subseteq T$;*

We denote the set of all types for $\widehat{\phi}$ by $\mathfrak{T}$. We further define the set $\mathfrak{T}|_{\mathcal{S}} \subseteq \mathfrak{T}$ that contains all types T for $\widehat{\phi}$ for which $T \cap \{p_1, \ldots, p_m\} \in \mathcal{S}$.

The reason that we use the types for $\widehat{\phi}$ and not for $\widehat{\phi}_{\mathcal{S}}$ is that the latter formula is exponentially larger than the former. To avoid this exponential blowup in the automaton, we check the additional condition of $\widehat{\phi}_{\mathcal{S}}$, namely that each world must occur in the set $\mathcal{S}$, by restricting the first component of the state set of the automaton to $\mathfrak{T}|_{\mathcal{S}}$.

Another difference to the standard construction is the additional condition that $w_i = X_{\iota(i)}$ should hold for i, $0 \leq i \leq n$. We check this by attaching a counter from $\{0, \ldots, n+1\}$ to the states of the 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. At $i = n$, we ensure that $\widehat{\phi}_{\mathcal{S}}$ is satisfied.

Definition 3.11. *The generalized Büchi-automaton $\mathcal{G} = (Q, \Sigma, \Delta, Q_0, \mathcal{F})$ is defined as follows:*

- $Q := \mathfrak{T}|_{\mathcal{S}} \times \{0, \ldots, n+1\}$*;*
- $\Sigma := 2^{\{p_1, \ldots, p_m\}}$*;*
- $\Delta \subseteq Q \times \Sigma \times Q$ *is defined as follows:* $((T, k), \sigma, (T', k')) \in \Delta$ *iff*
  - $\sigma = T \cap \{p_1, \ldots, p_m\}$*;*
  - $\bigcirc\psi \in T$ *iff* $\psi \in T'$*;*
  - $\bigcirc^{-}\psi \in T'$ *iff* $\psi \in T$*;*
  - $\psi_1 \mathsf{U} \psi_2 \in T$ *iff (i)* $\psi_2 \in T$ *or (ii)* $\psi_1 \in T$ *and* $\psi_1 \mathsf{U} \psi_2 \in T'$*;*
  - $\psi_1 \mathsf{S} \psi_2 \in T'$ *iff (i)* $\psi_2 \in T'$ *or (ii)* $\psi_1 \in T'$ *and* $\psi_1 \mathsf{S} \psi_2 \in T$*;*
  - $k < n+1$ *implies* $\sigma = X_{\iota(k)}$*;*
  - $k = n$ *implies* $\widehat{\phi} \in T$*; and*
  - $k' = \begin{cases} k+1 & \text{if } k < n+1, \text{ and} \\ k & \text{otherwise.} \end{cases}$
- $Q_0 := \{(T, 0) \mid \psi_1 \mathsf{S} \psi_2 \in T \Rightarrow \psi_2 \in T, \text{ and } \bigcirc^{-} \psi \notin T\}$*; and*
- $\mathcal{F} := \{F_{\psi_1 \mathsf{U} \psi_2} \times \{n+1\} \mid \psi_1 \mathsf{U} \psi_2 \text{ is a sub-formula of } \phi\}$*, where* $F_{\psi_1 \mathsf{U} \psi_2} := \{T \in \mathfrak{T} \mid \psi_1 \mathsf{U} \psi_2 \in T \Rightarrow \psi_2 \in T\}$.

We now show that this automaton accepts exactly those sequences of worlds that satisfy the conditions imposed in Lemma 3.8.

Lemma 3.12. *For every infinite word $w = w_0w_1 \ldots \in \Sigma^\omega$, we have $w \in L_\omega(\mathcal{G})$ iff the LTL structure $\mathfrak{J} := (w_i)_{i\geq 0}$ satisfies $\mathfrak{J}, n \models \widehat{\phi}_\mathcal{S}$ and $w_i = X_{\iota(i)}$ for all i, $0 \leq i \leq n$.*

Proof. ($\Longleftarrow$) Assume that the LTL structure $\mathfrak{J} := (w_i)_{i\geq 0}$ is a model of $\widehat{\phi}_\mathcal{S}$ at time point n and $w_i = X_{\iota(i)}$ for all i, $0 \leq i \leq n$.

If we define $S_i := \{\psi \mid \mathfrak{J}, i \models \psi$, and ψ is a sub-literal of $\widehat{\phi}\}$ for $i \geq 0$, then

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

is a run on $\mathcal{G}$:

- We have $(S_i, k) \in Q$ for all $i \geq 0$ and k, $0 \leq k \leq n + 1$:
  - For every sub-formula ψ of $\widehat{\phi}_\mathcal{S}$, we have either $\mathfrak{J}, i \models \psi$ or $\mathfrak{J}, i \models \neg\psi$. Thus, we have $\psi \in S_i$ iff $\neg\psi \notin S_i$.
  - For every sub-formula $\psi_1 \wedge \psi_2$ of $\widehat{\phi}_\mathcal{S}$, we have $\mathfrak{J}, i \models \psi_1 \wedge \psi_2$ iff $\mathfrak{J}, i \models \psi_1$ and $\mathfrak{J}, i \models \psi_2$. Thus, we have $\psi_1 \wedge \psi_2 \in S_i$ iff $\{\psi_1, \psi_2\} \subseteq S_i$.
  - For each world w_i, $i \geq 0$, we have $w_i \in \mathcal{S}$ since $\mathfrak{J}$ satisfies $\widehat{\phi}_\mathcal{S}$. Thus, we have $S_i \cap \{p_1, \ldots, p_m\} = w_i \in \mathcal{S}$ for all $i \geq 0$.
- We have for every sub-formula $\bigcirc^-\psi$ of $\widehat{\phi}_\mathcal{S}$ that $\mathfrak{J}, 0 \not\models \bigcirc^-\psi$, and thus $\bigcirc^-\psi \notin S_0$. Additionally, we have for every $\psi_1 \mathsf{S} \psi_2 \in S_0$, since $\mathfrak{J}, 0 \models \psi_1 \mathsf{S} \psi_2$ also $\mathfrak{J}, 0 \models \psi_2$. This implies that $(S_0, 0) \in Q_0$.
- We have for all i, $0 \leq i \leq n$,

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

  and for all $i \geq n + 1$,

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

  since:

  - by the definition of S_i, we have $w_i = S_i \cap \{p_1, \ldots, p_m\}$;
  - for every sub-formula $\bigcirc\psi$ of $\widehat{\phi}_\mathcal{S}$, we have $\bigcirc\psi \in S_i$ iff $\mathfrak{J}, i \models \bigcirc\psi$ iff $\mathfrak{J}, i + 1 \models \psi$ iff $\psi \in S_{i+1}$;
  - for every sub-formula $\bigcirc^-\psi$ of $\widehat{\phi}_\mathcal{S}$, we have $\bigcirc^-\psi \in S_{i+1}$ iff $\mathfrak{J}, i+1 \models \bigcirc^-\psi$ iff $\mathfrak{J}, i \models \psi$ iff $\psi \in S_i$;
  - for every sub-formula $\psi_1 \mathsf{U} \psi_2$ of $\widehat{\phi}_\mathcal{S}$, we have $\psi_1 \mathsf{U} \psi_2 \in S_i$ iff $\mathfrak{J}, i \models \psi_1 \mathsf{U} \psi_2$ iff (i) $\mathfrak{J}, i \models \psi_2$ or (ii) $\mathfrak{J}, i \models \psi_1$ and $\mathfrak{J}, i + 1 \models \psi_1 \mathsf{U} \psi_2$ iff (i) $\psi_2 \in S_i$ or (ii) $\psi_1 \in S_i$ and $\psi_1 \mathsf{U} \psi_2 \in S_{i+1}$;

- for every sub-formula $\psi_1 \mathsf{S} \psi_2$ of $\widehat{\phi}_{\mathcal{S}}$, we have $\psi_1 \mathsf{S} \psi_2 \in S_{i+1}$ iff $\mathfrak{J}, i+1 \models \psi_1 \mathsf{S} \psi_2$ iff (i) $\mathfrak{J}, i+1 \models \psi_2$ or (ii) $\mathfrak{J}, i+1 \models \psi_1$ and $\mathfrak{J}, i \models \psi_1 \mathsf{S} \psi_2$ iff (i) $\psi_2 \in S_{i+1}$ or (ii) $\psi_1 \in S_{i+1}$ and $\psi_1 \mathsf{S} \psi_2 \in S_i$;
- $i < n+1$ implies $w_i = X_{\iota(i)}$ by assumption;
- for $i = n-1$ we have $\mathfrak{J}, n \models \widehat{\phi}_{\mathcal{S}}$, and thus $\widehat{\phi}_{\mathcal{S}} \in S_n = S_{i+1}$;
- the condition for incrementing the second component of a state (until $n+1$ is reached) is obviously also satisfied.

Moreover, the above run is accepting. We prove this by contradiction. Suppose for some sub-formula $\psi_1 \mathsf{U} \psi_2$, the set $\{i \geq 0 \mid S_i \in F_{\psi_1 \mathsf{U} \psi_2}\}$ is finite. Then there exists a $k \geq 0$ such that $S_\ell \notin F_{\psi_1 \mathsf{U} \psi_2}$ for all $\ell \geq k$. This means $\psi_1 \mathsf{U} \psi_2 \in S_\ell$ and $\psi_2 \notin S_\ell$ for all $\ell \geq k$. Hence, $\mathfrak{J}, k \models \psi_1 \mathsf{U} \psi_2$ and $\mathfrak{J}, \ell \not\models \psi_2$ for all $\ell \geq k$. This contradicts the semantics of U.

($\Longrightarrow$) Assume that $w \in L_\omega(\mathcal{G})$. Let

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

be an accepting run of $\mathcal{G}$ on $w = w_0 w_1 \ldots$.

It is left to be shown that the LTL structure $\mathfrak{J} := (w_i)_{i \geq 0}$ is a model of $\widehat{\phi}_{\mathcal{S}}$ at time point n and that $w_i = X_{\iota(i)}$ for all i, $0 \leq i \leq n$. We get the latter immediately from the definition of Δ, i.e., $i < n+1$ implies that $w_i = X_{\iota(i)}$. For the former, observe that for each $i \geq 0$ we have $w_i = S_i \cap \{p_1, \ldots, p_m\} \in \mathcal{S}$ by definition of the state set Q. Thus, the conjunct

$$\Box^- \Box \left(\bigvee_{X \in \mathcal{S}} \left(\bigwedge_{p \in X} p \wedge \bigwedge_{p \notin X} \neg p \right) \right)$$

of $\widehat{\phi}_{\mathcal{S}}$ is clearly satisfied by $\mathfrak{J}$ (at any time point).

Furthermore, we have that $\widehat{\phi} \in S_n$ again by the definition of Δ, and thus it is now enough to show that $\psi \in S_i$ iff $\mathfrak{J}, i \models \psi$ for each $i \geq 0$. This can be shown by induction on the structure of ψ.

- If ψ is a propositional variable, we have $\psi \in S_i$ iff $\psi \in w_i$ iff $w_i \models \psi$ iff $\mathfrak{J}, i \models \psi$.
- If $\psi = \neg\chi$, we have $\neg\chi \in S_i$ iff $\chi \notin S_i$ iff $\mathfrak{J}, i \not\models \chi$ iff $\mathfrak{J}, i \models \neg\chi$.
- 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 $\mathfrak{J}, i \models \chi_1$ and $\mathfrak{J}, i \models \chi_2$ iff $\mathfrak{J}, i \models \chi_1 \wedge \chi_2$.
- If $\psi = \bigcirc\chi$, we have $\bigcirc\chi \in S_i$ iff $\chi \in S_{i+1}$ iff $\mathfrak{J}, i+1 \models \chi$ iff $\mathfrak{J}, i \models \bigcirc\chi$.

- If $\psi = \bigcirc^{-}\chi$, we have $\bigcirc^{-}\chi \in S_i$ iff $i > 0$ and $\chi \in S_{i-1}$ iff $i > 0$ and $\mathfrak{J}, i-1 \models \chi$ iff $\mathfrak{J}, i \models \bigcirc^{-}\chi$. 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 $\mathfrak{J}, i \models \chi_1 \mathsf{U} \chi_2$ as follows.

  ($\Longleftarrow$) Assume $\mathfrak{J}, i \models \chi_1 \mathsf{U} \chi_2$. Then there exists a $k \geq i$ such that $\mathfrak{J}, k \models \chi_2$ and $\mathfrak{J}, \ell \models \chi_1$ for all ℓ, $i \leq \ell < k$. We show by induction on j that $\chi_1 \mathsf{U} \chi_2 \in S_{k-j}$ for $j \leq k - i$.

  For $j = 0$, we have: $\mathfrak{J}, k \models \chi_2$ 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: $\mathfrak{J}, k - j \models \chi_1$ 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 $\chi_1 \mathsf{U} \chi_2 \in S_{k-j}$.

  ($\Longrightarrow$) Assume $\chi_1 \mathsf{U} \chi_2 \in S_i$. Since states of $F_{\chi_1 \mathsf{U} \chi_2}$ occur infinitely often among $S_0, S_1, S_2 \ldots$, there is a $k \geq i$ 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 all ℓ, $i \leq \ell < k$.

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

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

  $(*)$ and $(**)$ yield that $\mathfrak{J}, i \models \chi_1 \mathsf{U} \chi_2$ 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 $\mathfrak{J}, i \models \chi_1 \mathsf{S} \chi_2$ as follows.

  ($\Longleftarrow$) Assume $\mathfrak{J}, i \models \chi_1 \mathsf{S} \chi_2$. Then there exists a k, $0 \leq k \leq i$ such that $\mathfrak{J}, k \models \chi_2$ and $\mathfrak{J}, \ell \models \chi_1$ for all ℓ, $k < \ell \leq i$. We show by induction on j that $\chi_1 \mathsf{S} \chi_2 \in S_{k+j}$ for $j \leq i - k$.

  For $j = 0$, we have: $\mathfrak{J}, k \models \chi_2$ 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: $\mathfrak{J}, k + j \models \chi_1$ $\chi_1 \in S_{k+j}$ 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 $\chi_1 \mathsf{S} \chi_2 \in S_{k+j}$.

  ($\Longrightarrow$) Assume $\chi_1 \mathsf{S} \chi_2 \in S_i$. There are two cases: 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 $\mathfrak{J}, 0 \models \chi_2$, and thus $\mathfrak{J}, 0 \models \chi_1 \mathsf{S} \chi_2$.

  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 $\mathfrak{J}, i \models \chi_1 \mathsf{S} \chi_2$. 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: $\mathfrak{J}, i - 1 \models \chi_1 \mathsf{S} \chi_2$. Thus, there is a k, $0 \leq k \leq i - 1$, such that $\mathfrak{J}, k \models \chi_2$ and $\mathfrak{J}, j \models \chi_1$ for all j, $k < j \leq i - 1$. Since we have by the outer induction hypothesis also that $\mathfrak{J}, i \models \chi_1$, it follows that there is a k, $0 \leq k \leq i$, such that $\mathfrak{J}, k \models \chi_2$ and $\mathfrak{J}, j \models \chi_1$ for all j, $k < j \leq i$. Hence, $\mathfrak{J}, i \models \chi_1 \mathsf{S} \chi_2$. □

This yields that $L_\omega(\mathcal{G}) \neq \emptyset$ iff there is an LTL-structure $\mathfrak{J} = (w_i)_{i \geq 0}$ such that $\mathfrak{J}, n \models \widehat{\phi}_{\mathcal{S}}$ and $w_i = X_{\iota(i)}$ for all i, $0 \leq i \leq n$. We can thus decide the latter problem by testing $\mathcal{G}$ for emptiness, which yields the following complexity results.

Lemma 3.13. *Given a set $\mathcal{S} = \{X_1, \dots, X_k\} \subseteq 2^{\{p_1, \dots, p_m\}}$ and a mapping $\iota \colon \{0, \dots, n\} \to \{1, \dots, k\}$, the problem of deciding the existence of an LTL-structure $\mathfrak{J} = (w_i)_{i \geq 0}$ such that $\mathfrak{J}, n \models \widehat{\phi}_{\mathcal{S}}$ and $w_i = X_{\iota(i)}$ for all i, $0 \leq i \leq n$, is*

- *in* ExpTime *w.r.t. combined complexity and*
- *in* P *w.r.t. data complexity.*

Proof. For combined complexity, there are exponentially many types for $\widehat{\phi}$ and exponentially many input symbols in $2^{\{p_1, \dots, p_m\}}$. The set $\mathcal{F}$ contains linearly many sets of size at most exponential, while the size of Q_0 and Δ is bounded polynomially in the size of Q (which is exponential). Since all conditions that need to be checked to construct the components of $\mathcal{G}$ can be checked in exponential time, and the size of $\mathcal{G}$ is exponential in the size of $\mathcal{K}$ and ϕ, the emptiness test can be done in ExpTime.

For data complexity, the size of $\mathcal{G}$ is polynomial in n because of the following reasons: the size of $\mathfrak{T}|_{\mathcal{S}}$ is constant since the size of $\mathcal{S}$ is constant, and thus the size of Q is linear in n. The size of Σ is constant. Obviously, then the size of Δ is polynomial in n. The size of Q_0 is linear in n, because $Q_0 \subseteq Q$. The size of $\mathcal{F}$ is also linear in n, because each set $F_{\psi_1 \mathsf{U} \psi_2}$ is of constant size, and the number of such sets does not depend on n. Obviously, $\mathcal{G}$ can also be constructed in time polynomial in n. The data complexity of the emptiness test is thus in P. □

However, the complexity of the entailment problem also depends on the complexity of the r-consistency test for $\mathcal{S}$. In the following sections, we will establish some results as to this complexity in the cases without rigid names, and with rigid concept and role names. The most interesting (and most complex) case without rigid role names, but with rigid concept names, is considered in Section 4 for data complexity and in Section 5 for combined complexity.

3.2.2 The Case Without Rigid Names

To check r-consistency of a set $\mathcal{S} = \{X_1, \dots, X_k\} \subseteq 2^{\{p_1, \dots, p_m\}}$ w.r.t. a mapping $\iota \colon \{0, \dots, n\} \to \{1, \dots, k\}$ and $\mathcal{K}$ without rigid names, it clearly suffices to check the consistency of the following conjunctions of CQ-literals w.r.t. the TBox $\mathcal{T}$ individually:

- for each i, $1 \leq i \leq k$, the conjunction χ_i; and
- for each i, $0 \leq i \leq n$, the conjunction $\chi_{\iota(i)} \wedge \bigwedge_{\alpha \in \mathcal{A}_i} \alpha$.[5]

Each of these conjunctions of CQ-literals is of polynomial size in the size of $\mathcal{K}$ and ϕ. We can now use Theorem 3.2 to establish the complexity of the entailment problem without rigid names.

Theorem 3.14. *If $N_{RC} = N_{RR} = \emptyset$, then the entailment problem is*

- *in* ExpTime *w.r.t. combined complexity and*
- *in co-*NP *w.r.t. data complexity.*

Proof. For combined complexity, note that we do not need to guess the set $\mathcal{S}$. Since the r-consistency test imposes no dependency between the sets $X \in \mathcal{S}$, it suffices to define $\mathcal{S}$ as the set of *all* sets X_i that pass the consistency test of the corresponding conjunction χ_i. Since there are exponentially many such sets, but each of them is of polynomial size, by Theorem 3.2 we only have to do exponentially many ExpTime-tests to construct $\mathcal{S}$. We can further enumerate all possible mappings ι in exponential time and check for each ι the consistency of the conjunctions $\chi_{\iota(i)} \wedge \bigwedge_{\alpha \in \mathcal{A}_i} \alpha$ again in ExpTime. For each ι that passes these tests, we can check the existence of the required LTL-structure in ExpTime by Lemma 3.13. Lemma 3.8 now yields a total complexity of ExpTime for the non-entailment problem, and therefore also for the entailment problem.

For data complexity, note that since $\mathcal{S}$ is of constant size w.r.t. the ABoxes and ι is linear in n, guessing $\mathcal{S}$ and ι can be done in NP. Since the LTL-satisfiability test can be done in P (Lemma 3.13) and the consistency tests for r-consistency of $\mathcal{S}$ can be done in NP (Theorem 3.2), by Lemma 3.8 the non-entailment problem is also in NP. □

3.2.3 The Case With Rigid Role Names

If the sets N_{RC} and N_{RR} are allowed to be non-empty, the consistency tests for the r-consistency of $\mathcal{S}$ are not independent anymore. To make sure that the models respect the rigid symbols, we use a renaming techique similar to the one used in [BGL12] that works by introducing enough copies of the flexible symbols.

[5] We can assume that all of these models have the same domain since their domains can be assumed to be countably infinite by the Löwenheim-Skolem theorem. We can further assume that all individual names are interpreted by the same domain elements in all models.

For every i, $1 \leq i \leq k+n+1$, and every *flexible* concept name A (every *flexible* role name r) occurring in ϕ or in $\mathcal{T}$, we introduce a copy $A^{(i)}$ ($r^{(i)}$). We call $A^{(i)}$ ($r^{(i)}$) the i-th copy of A (r). The conjunctive query $\alpha^{(i)}$ (the GCI $\beta^{(i)}$) is obtained from a CQ α (a GCI 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 $\chi_\ell^{(i)}$ is obtained from χ_ℓ (see Definition 3.7) by replacing each CQ α_j by $\alpha_j^{(i)}$. Finally, we define

$$\chi_{\mathcal{S},\iota} := \bigwedge_{1\leq i\leq k} \chi_i^{(i)} \wedge \bigwedge_{0\leq i\leq n} \left(\chi_{\iota(i)}^{(k+i+1)} \wedge \bigwedge_{\alpha\in\mathcal{A}_i} \alpha^{(k+i+1)} \right) \text{ and}$$

$$\mathcal{T}_{\mathcal{S},\iota} := \{\beta^{(i)} \mid \beta \in \mathcal{T} \text{ and } 1 \leq i \leq k+n+1\}.$$

Note that here it is essential that the ABoxes do not contain complex concepts.

Lemma 3.15. *The set $\mathcal{S}$ is r-consistent w.r.t. ι and $\mathcal{T}$ iff the conjunction of CQ-literals $\chi_{\mathcal{S},\iota}$ has a model w.r.t. $\mathcal{T}_{\mathcal{S},\iota}$.*

Proof. Let $\mathcal{J}_1, \ldots, \mathcal{J}_k, \mathcal{I}_0, \ldots, \mathcal{I}_n$ be the interpretations required by Definition 3.7 for the r-consistency of $\mathcal{S}$ w.r.t. ι and $\mathcal{K}$. We construct the interpretation $\mathcal{J}$ as follows:

- the domain of $\mathcal{J}$ is the shared domain of the above interpretations;
- the rigid names are interpreted as in the above interpretations;
- 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 $\chi_{\mathcal{S},\iota}$ and $\mathcal{T}_{\mathcal{S},\iota}$.

For the other direction, let $\mathcal{J}$ be a model of $\chi_{\mathcal{S},\iota}$ w.r.t. $\mathcal{T}_{\mathcal{S},\iota}$. We obtain the interpretations $\mathcal{J}_1, \ldots, \mathcal{J}_k$, $\mathcal{I}_0, \ldots, \mathcal{I}_n$ by the inverse construction to the one above:

- the domain of all these interpretations is the domain of $\mathcal{J}$;
- the rigid names are 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 it $k+i+1$-st copy is interpreted in $\mathcal{J}$.

Again, it is easy to verify that these interpretations satisfy the conditions in Definition 3.7. □

Unfortunately, the data complexity of this approach does not allow us to match the lower bound of co-NP for the entailment problem we have from Corollary 3.4. However, for the combined complexity we obtain containment in 2-ExpTime.

Theorem 3.16. *If $N_{RR} \neq \emptyset$, then the entailment problem is*

- *in 2-*ExpTime *w.r.t. combined complexity and*
- *in* ExpTime *w.r.t. data complexity.*

Proof. We give a deterministic algorithm for the complement problem, i.e., one that checks whether a TCQ ϕ has a model w.r.t. a TKB $\mathcal{K}$. Since deterministic complexity classes are closed under complementation, this is enough to obtain the complexity results of the theorem. The algorithm works as follows. First, we enumerate all possible sets $\mathcal{S}$ and mappings ι, which can be done in 2-ExpTime w.r.t. combined complexity and in ExpTime w.r.t. data complexity since $\mathcal{S}$ is constant in this case. For each of these double-exponentially many pairs $(\mathcal{S}, \iota)$, we then test the LTL-satisfiability required in Lemma 3.8 in exponential time (see Lemma 3.13) and test $\mathcal{S}$ for r-consistency w.r.t. ι and $\mathcal{K}$. Then ϕ has a model w.r.t. $\mathcal{I}$ iff iff at least one pair passes both tests.

For the combined complexity of the r-consistency test, observe that the conjunction of CQ-literals $\chi_{\mathcal{S},\iota}$ is of exponential size in the size of ϕ and $\mathcal{K}$. By Theorem 3.2, the overall combined complexity of the r-consistency test is thus in 2-ExpTime, which shows that the above satisfiability problem (and hence the entailment problem) can be solved in 2-ExpTime.

For the data complexity of the r-consistency test, observe that $\chi_{\mathcal{S},\iota}$ is of linear size in the size of the input ABoxes. Unfortunately, by copying each of the types $\chi_{\iota(i)}$ assigned to the ABoxes, we have introduced linearly many *negated* CQs, which is why Theorem 3.2 only yields an ExpTime upper bound for the data complexity.[6] □

However, we can match the lower bound of co-NP for the data complexity in the following special cases.

Lemma 3.17. *If $N_{RR} \neq \emptyset$, then the entailment problem 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.*

[6]Linearly many non-negated CQs in $\chi_{\mathcal{S},\iota}$ are not problematic, as they can be instantiated and viewed as part of the ABox, as detailed in the proof of Theorem 3.2.

2. *The set of individual names allowed to occur in the ABoxes is fixed.*

Proof. As in the proof of Theorem 3.14, we can guess the set $\mathcal{S}$ and the mapping ι in NP and do the LTL-satisfiability test in P. Thus, it suffices to show that in the above-mentioned special cases r-consistency of $\mathcal{S}$ can be tested in NP.

1. If n is bounded by a constant, then the number of negated CQs in $\chi_{\mathcal{S},\iota}$ is constant, and thus Theorem 3.2 yields the desired NP upper bound.

2. If the set of individual names is fixed, then the number of possible different ABoxes is constant. We thus do not need to introduce n copies of formulae $\chi_{\iota(i)}$ in $\chi_{\mathcal{S},\iota}$, but need at most one copy for each distinct combination of $\chi_{\iota(i)}$ and $\mathcal{A}_i$—clearly, consistency of each combination of an ABox with a type needs to be checked only once. Since there are only constantly many such combinations, the modified TCQ $\chi'_{\mathcal{S},\iota}$ again contains only constantly many negated CQs. As in the previous case, Theorem 3.2 yields the result. □

4 Data Complexity for the Case of Rigid Concept Names

To obtain an upper bound for the data complexity of the non-entailment problem in the case where $N_{\mathrm{RC}} \neq \emptyset$ and $N_{\mathrm{RR}} = \emptyset$, we consider the conditions of Lemma 3.8 in more detail. First, note that, since $\mathcal{S} \subseteq 2^{\{p_1,\dots,p_m\}}$ is of constant size w.r.t. the input ABoxes and $\iota\colon \{0,\dots,n\} \to \{1,\dots,k\}$ is of size linear in n (the number of ABoxes), guessing $\mathcal{S}$ and ι can be done in NP. Additionally, according to Lemma 3.13, LTL-satisfiability can be tested in P.

We now show that the r-consistency of $\mathcal{S}$ w.r.t. ι and $\mathcal{K}$ can be checked in NP, which yields the desired data complexity of co-NP for the entailment problem.

Similar to the previous sections, we construct conjunctions of CQ-literals of which we want to check consistency. The approach is a mixture of those of Sections 3.2.2 and 3.2.3, as we combine several consistency tests required for r-consistency, but do not go as far as compiling all of them into just one conjunction. More precisely, we consider the conjunctions of CQ-literals $\gamma_i \wedge \chi_{\mathcal{S}}$, $0 \leq i \leq n$, w.r.t. the TBox $\mathcal{T}_{\mathcal{S}}$, where

$$\gamma_i := \bigwedge_{\alpha\in\mathcal{A}_i} \alpha^{(\iota(i))}, \quad \chi_{\mathcal{S}} := \bigwedge_{1\leq i\leq k} \chi_i^{(i)}, \quad \mathcal{T}_{\mathcal{S}} := \{\beta^{(i)} \mid \beta \in \mathcal{T} \text{ and } 1 \leq i \leq k\}.$$

One can see from the proof of Theorem 3.2 that this problem can be decided in NP in the size of the input ABoxes. The main reason is that the negated CQs do not depend on the input ABoxes. In fact, negated CQs only occur in $\chi_{\mathcal{S}}$, which only depends on the query ϕ.

However, for r-consistency we have to make sure that rigid consequences of the form $A(a)$ for a rigid concept name $A \in N_{\mathrm{RC}}$ and an individual name $a \in N_{\mathrm{I}}$ are shared between all of these conjunctions $\gamma_i \wedge \chi_{\mathcal{S}}$. Let $\mathsf{RCon}(\mathcal{T})$ denote the rigid concept names occurring in $\mathcal{T}$. Similar to what was done in Lemma 6.3 of [BGL12], we now guess a set $\mathcal{D} \subseteq 2^{\mathsf{RCon}(\mathcal{T})}$ and a mapping $\tau\colon \mathsf{Ind}(\phi) \cup \mathsf{Ind}(\mathcal{K}) \to \mathcal{D}$. The idea is that $\mathcal{D}$ fixes the combinations of rigid concept names that occur in the models of $\gamma_i \wedge \chi_{\mathcal{S}}$ and τ assigns to each individual name one such combination. Note that $\mathcal{D}$ only depends on $\mathcal{T}$ and τ is of size linear in the size of the input ABoxes, which is why we can guess $\mathcal{D}$ and τ in NP w.r.t. data complexity. We now define

$$\chi_\tau := \bigwedge_{a \in \mathsf{Ind}(\phi) \cup \mathsf{Ind}(\mathcal{K})} \left(\bigwedge_{A \in \tau(a)} A(a) \quad \wedge \bigwedge_{A \in \mathsf{RCon}(\mathcal{T}) \setminus \tau(a)} A'(a) \right),$$

where A' is a rigid concept name that is equivalent to $\neg A$ in $\mathcal{T}$.[7] Note that χ_τ is of polynomial size w.r.t. the size of the input ABoxes.

We need one more notation to formulate the main lemma of this section. We say that an interpretation $\mathcal{I}$ *respects* $\mathcal{D}$ if

$$\mathcal{D} = \{Y \subseteq \mathsf{RCon}(\mathcal{T}) \mid \text{there is a } d \in \Delta^{\mathcal{I}} \text{ such that } d \in (C_Y)^{\mathcal{I}}\},$$

where $C_Y := \bigsqcap_{A \in Y} A \ \sqcap \ \bigsqcap_{A \in \mathsf{RCon}(\mathcal{T}) \setminus Y} \neg A$.

Lemma 4.1. *If $N_{RC} \neq \emptyset$ and $N_{RR} = \emptyset$, then $\mathcal{S}$ is r-consistent w.r.t. ι and $\mathcal{K}$ iff there exist $\mathcal{D} \subseteq 2^{\mathsf{RCon}(\mathcal{T})}$ and $\tau\colon \mathsf{Ind}(\phi) \cup \mathsf{Ind}(\mathcal{K}) \to \mathcal{D}$ such that each of the conjunctions $\gamma_i \wedge \chi_{\mathcal{S}} \wedge \chi_\tau$, $0 \leq i \leq n$, has a model w.r.t. $\mathcal{T}_{\mathcal{S}}$ that respects $\mathcal{D}$.*

Proof. For the "if" direction, assume that $\mathcal{I}_i$ are the required models for $\gamma_i \wedge \chi_{\mathcal{S}} \wedge \chi_\tau$, for $0 \leq i \leq n$. 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 $Y \in \mathcal{D}$ there are countably infinitely many elements $d \in (C_Y)^{\mathcal{I}_i}$. This is a consequence of the Löwenheim-Skolem theorem and the fact that the countably infinite disjoint union of $\mathcal{I}_i$ with itself is again a model of $\gamma_i \wedge \chi_{\mathcal{S}} \wedge \chi_\tau$. The latter follows from the observation that for any CQ there is a homomorphism into $\mathcal{I}_i$ iff there is a homomorphism into the disjoint union of $\mathcal{I}_i$ with itself. One direction is trivial, while whenever there is a homomorphism into the disjoint union, we can construct a homomorphism into $\mathcal{I}_i$ by renaming the elements in the image of this homomorphism to the corresponding elements of Δ_i. It is easy to see that the resulting homomorphism still satisfies all atoms of the CQ.

Consequently, we can partition the domains Δ_i into the countably infinite sets $\Delta_i(Y) := \{d \in \Delta_i \mid d \in (C_Y)^{\mathcal{I}_i}\}$ for $Y \in \mathcal{D}$. By the assumptions above and the

[7] We can assume w.l.o.g. that for each rigid concept name in $\mathcal{T}$, there is a rigid concept name equivalent to its negation in $\mathcal{T}$. We can introduce them if needed while multiplying the size of the TBox by at most 2. We cannot include $\neg A(a)$ in χ_τ since this could result in polynomially many negated CQs in the size of the ABoxes.

fact that all $\mathcal{I}_i$ satisfy χ_τ, there are bijections $\pi_i\colon \Delta_0 \to \Delta_i$, $1 \le i, j \le n$, such that

- $\pi_i(\Delta_0(Y)) = \Delta_i(Y)$ for all $Y \in \mathcal{D}$ and
- $\pi_i(a^{\mathcal{I}_0}) = a^{\mathcal{I}_i}$ for all $a \in \mathsf{Ind}(\phi) \cup \mathsf{Ind}(\mathcal{K})$.

We can now construct the models required by Definition 3.7 from the models $\mathcal{I}_i$ by appropriately relating the flexible names and their copies. For example, interpreting the rigid concept names as in $\mathcal{I}_i$ and the flexible names as their $\iota(i)$-th copies in $\mathcal{I}_i$ yields a model of $\chi_{\iota(i)}$ w.r.t. $\langle \mathcal{A}_i, \mathcal{T} \rangle$, and similarly for the models of χ_j and $\mathcal{T}$ for $1 \le j \le k$. These models share the same domain and respect the rigid names in $\mathsf{RCon}(\mathcal{T})$ and $\mathsf{Ind}(\phi) \cup \mathsf{Ind}(\mathcal{K})$. Note that the interpretation of the names in $N_{\mathrm{RC}} \setminus \mathsf{RCon}(\mathcal{T})$ and $N_{\mathrm{I}} \setminus (\mathsf{Ind}(\phi) \cup \mathsf{Ind}(\mathcal{K}))$ is irrelevant and can be fixed arbitrarily, as long as the UNA is satisfied.

For the "only if" direction, it is easy to see that one can combine the interpretations $\mathcal{I}_i$, $\mathcal{J}_1, \dots, \mathcal{J}_k$ from Definition 3.7 to a model $\mathcal{I}'_i$ of $\gamma_i \wedge \chi_\mathcal{S}$ w.r.t. $\mathcal{T}_\mathcal{S}$ by interpreting the j-th copy of a flexible name as the original name in $\mathcal{J}_j$. For $a \in \mathsf{Ind}(\phi) \cup \mathsf{Ind}(\mathcal{K})$, we define $\tau(a) := Y \subseteq \mathsf{RCon}(\mathcal{T})$ iff $a \in (C_Y)^{\mathcal{I}_0}$. Furthermore, we let $\mathcal{D}$ contain all those sets $Y \subseteq \mathsf{RCon}(\mathcal{T})$ such that there is a $d \in (C_Y)^{\mathcal{I}'_i}$ for some $0 \le i \le n$. To obtain models of $\gamma_i \wedge \chi_\mathcal{S} \wedge \chi_\tau$ w.r.t. $\mathcal{T}_\mathcal{S}$ that respect $\mathcal{D}$, we still need to ensure that all $Y \in \mathcal{D}$ are represented in each of the models $\mathcal{I}'_i$. To do this, we construct the disjoint union $\mathcal{I}''_i$ of $\mathcal{I}'_i$ with all other $\mathcal{I}'_j$ for $0 \le j \le n$, $i \ne j$. It remains to show that this interpretation is still a model of $\mathcal{T}_\mathcal{S}$ and the conjunction $\gamma_i \wedge \chi_\mathcal{S} \wedge \chi_\tau$. This can be seen as follows. For the non-negated CQs in this conjunction, clearly there is a homomorphism into $\mathcal{I}''_i$ if there is one into $\mathcal{I}'_i$. For the negated CQs, which only occur in the shared conjunction $\chi_\mathcal{S}$, it is essential that they are connected (see Definition 2.7). Given this assumption, the non-existence of a homomorphism into any of the components of $\mathcal{I}''_i$ clearly implies the non-existence of a homomorphism into their disjoint union $\mathcal{I}''_i$.[8] □

It remains to show that we can check the existence of a model of $\gamma_i \wedge \chi_\mathcal{S} \wedge \chi_\tau$ w.r.t. $\mathcal{T}_\mathcal{S}$ that respects $\mathcal{D}$ in nondeterministic polynomial time. For this, observe that the restriction imposed by $\mathcal{D}$ can equivalently be expressed as

$$\chi_\mathcal{D} := (\neg \exists x. A_\mathcal{D}(x)) \wedge \bigwedge_{Y \in \mathcal{D}} \exists x. A_Y(x),$$

where A_Y and $A_\mathcal{D}$ are fresh concept names that are restricted by adding the GCIs $A_Y \sqsubseteq C_Y$, $C_Y \sqsubseteq A_Y$ for each $Y \in \mathcal{D}$, and $A_\mathcal{D} \sqsubseteq \bigsqcap_{Y \in \mathcal{D}} \neg A_Y$, $\bigsqcap_{Y \in \mathcal{D}} \neg A_Y \sqsubseteq A_\mathcal{D}$ to

[8] With unconnected negated CQs, the problem is that two interpretations $\mathcal{I}'_i$, $\mathcal{I}'_j$, $i \ne j$, might each satisfy only a part of the CQ such that the disjoint union of both satisfies the whole CQ. With connected CQs, this problem does not appear since the elements of the two disjoint domains are not connected by roles.

$\mathcal{T}_\mathcal{S}$. We call the resulting TBox $\mathcal{T}'_\mathcal{S}$. Since $\chi_\mathcal{D}$ and $\mathcal{T}'_\mathcal{S}$ do not depend on the input ABoxes, by Theorem 3.2 we can check the consistency of $\gamma_i \wedge \chi_\mathcal{S} \wedge \chi_\mathcal{T} \wedge \chi_\mathcal{D}$ w.r.t. $\mathcal{T}'_\mathcal{S}$ in NP w.r.t. data complexity, which yields the desired complexity result for the entailment problem.

Theorem 4.2. *If $N_{RC} \neq \emptyset$ and $N_{RR} = \emptyset$, then the entailment problem is in co-*NP *w.r.t. data complexity.*

5 Combined Complexity for the Case of Rigid Concept Names

Unfortunately, the approach used in the previous section does not yield a *combined complexity* of co-NExpTime. The reason is that the conjunctions $\chi_\mathcal{S}$ and $\chi_\mathcal{D}$ are of exponential size in the size of ϕ, and thus Theorem 3.2 only yields an upper bound of 2-ExpTime. In this section, 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 linear size in the size of ϕ and $\mathcal{K}$ such that answering ϕ at time point n is equivalent to answering ψ at time point 0 w.r.t. a trivial sequence of ABoxes. This is done by compiling the ABoxes into the query and postponing the query ϕ using the $\bigcirc$-operator.

Lemma 5.1. *Let ϕ be a Boolean TCQ and $\mathcal{K} = \langle (\mathcal{A}_i)_{0\leq i\leq n}, \mathcal{T}\rangle$ be a temporal knowledge base. Then there is a Boolean TCQ ψ such that $\mathcal{K} \models \phi$ iff $\langle \emptyset, \mathcal{T}\rangle \models \psi$ and the size of ψ is linear in the size of ϕ and $\mathcal{K}$.*

Proof. We define the Boolean TCQ

$$\psi := (\mathcal{A}_0 \wedge \bigcirc\mathcal{A}_1 \wedge \ldots \wedge \bigcirc^n\mathcal{A}_n) \to \bigcirc^n\phi,$$

where $\bigcirc^j$ means $\bigcirc\ldots\bigcirc$ (j-times). Obviously, the size of ψ is linear in the size of ϕ and $\mathcal{K}$. We further define $\mathcal{K}' := \langle\emptyset, \mathcal{T}\rangle$.

It is left to prove that $\mathcal{K} \models \phi$ iff $\mathcal{K}' \models \psi$. We have:

$$\begin{aligned}
&\mathcal{K} \models \phi\\
\textit{iff}\quad &\langle (\mathcal{A}_i)_{0\leq i\leq n}, \mathcal{T}\rangle \models \phi\\
\textit{iff}\quad &\mathfrak{I}, n \models \phi \text{ for all } \mathfrak{I} \models \langle (\mathcal{A}_i)_{0\leq i\leq n}, \mathcal{T}\rangle\\
\textit{iff}\quad &\mathfrak{I}, n \models \phi \text{ for all } \mathfrak{I} \models \langle \emptyset, \mathcal{T}\rangle \text{ with } \mathfrak{I}, 0 \models \mathcal{A}_0;\ \mathfrak{I}, 1 \models \mathcal{A}_1;\ \ldots;\ \mathfrak{I}, n \models \mathcal{A}_n\\
\textit{iff}\quad &\mathfrak{I}, 0 \models \bigcirc^n\phi \text{ for all } \mathfrak{I} \models \mathcal{K}' \text{ with } \mathfrak{I}, 0 \models \mathcal{A}_0;\ \mathfrak{I}, 0 \models \bigcirc\mathcal{A}_1;\ \ldots;\ \mathfrak{I}, 0 \models \bigcirc^n\mathcal{A}_n
\end{aligned}$$

iff $\mathfrak{I}, 0 \models \psi$ for all $\mathfrak{I} \models \mathcal{K}'$

iff $\mathcal{K}' \models \psi$. □

We can thus focus on deciding whether a Boolean TCQ ϕ has a model w.r.t. a TKB $\mathcal{K} = \langle \emptyset, \mathcal{T} \rangle$ that has only one empty ABox in the sequence. Note that this compilation approach does not allow us to obtain a low *data complexity* for the entailment problem since after encoding the ABoxes into ϕ the size of $\chi_{\mathcal{S}}$ as well as that of the generalized Büchi automaton $\mathcal{G}$ are exponential in the size of the ABoxes (cf. Sections 3.2.1 and 4).

We now again analyze how to check the two conditions in Lemma 3.8, this time with the goal of obtaining a combined complexity of NExpTime. First, observe that guessing $\mathcal{S} = \{X_1, \ldots, X_k\} \subseteq 2^{\{p_1, \ldots, p_m\}}$ can be done in nondeterministic exponential time in the size of ϕ. Furthermore, by Lemma 3.13, the LTL-satisfiability test required by the second condition can be realized in ExpTime. It remains to determine the complexity of testing r-consistency of $\mathcal{S}$ w.r.t. $\mathcal{K} = \langle \emptyset, \mathcal{T} \rangle$. Similarly to the approach used in the previous section and to the proof of Lemma 6.3 in [BGL12], we start by guessing a set $\mathcal{D} \subseteq 2^{\mathsf{RCon}(\mathcal{T})}$ and a mapping $\tau \colon \mathsf{Ind}(\phi) \to \mathcal{D}$. Since $\mathcal{D}$ is of size exponential in $\mathcal{T}$ and τ is of size polynomial in the size of ϕ and $\mathcal{T}$, guessing $\mathcal{D}$ and τ can also be done in NExpTime. By Lemma 4.1, it suffices to test whether $\chi_{\mathcal{S}} \wedge \chi_\tau$ has a model w.r.t. $\mathcal{T}_{\mathcal{S}}$ that respects $\mathcal{D}$. Instead of applying Theorem 3.2 directly to this problem, which would yield a complexity of 2-ExpTime, we split the problem into separate sub-problems for each component χ_i of $\chi_{\mathcal{S}}$. The correctness of this approach is stated in the next lemma. For the special case of $\mathcal{ALC}$-LTL, this was shown in Lemma 6.3 in [BGL12]. The proof for the general case is very similar to the proof of Lemma 4.1 above.

Lemma 5.2. *If $N_{RC} \neq \emptyset$ and $N_{RR} = \emptyset$, then $\mathcal{S}$ is r-consistent w.r.t. $\mathcal{K} = \langle \emptyset, \mathcal{T} \rangle$ iff there exist $\mathcal{D} \subseteq 2^{\mathsf{RCon}(\mathcal{T})}$ and $\tau \colon \mathsf{Ind}(\phi) \to \mathcal{D}$ such that each of the conjunctions $\widehat{\chi}_i := \chi_i \wedge \chi_\tau$, $1 \leq i \leq k$, has a model w.r.t. $\mathcal{K}$ that respects $\mathcal{D}$.*

Note that the size of each $\widehat{\chi}_i$ is polynomial in the size of ϕ and $\mathcal{T}$ and the number k of these conjunctions is exponential in the size of ϕ. Thus, it is enough to show that the existence of a model of $\widehat{\chi}_i$ w.r.t. $\mathcal{K}$ that respects $\mathcal{D}$ can be checked in exponential time in the size of ϕ and $\mathcal{T}$. Similar to the proof of Theorem 3.2, we can reduce this problem to a non-entailment problem for a union of Boolean CQs: there is an interpretation that is a model of $\widehat{\chi}_i$ and $\mathcal{T}$ and respects $\mathcal{D}$ iff there is a model of $\langle \mathcal{A}, \mathcal{T} \rangle$ that respects $\mathcal{D}$ and is not a model of ρ (written $\langle \mathcal{A}, \mathcal{T} \rangle \not\models \rho$ w.r.t. $\mathcal{D}$), where $\mathcal{A}$ is an ABox obtained by instantiating the non-negated CQs of $\widehat{\chi}_i$ with fresh individual names and ρ is a union of CQs constructed from the negated CQs of $\widehat{\chi}_i$.

It thus suffices to show that we can decide query non-entailment $\langle \mathcal{A}, \mathcal{T} \rangle \not\models \rho$ w.r.t. $\mathcal{D}$ in time exponential in the size of $\mathcal{A}$, $\mathcal{T}$, and ρ.

It is known that $\langle \mathcal{A}, \mathcal{T} \rangle \not\models \rho$ iff there is a *forest model* $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{T}$ such that $\mathcal{I} \not\models \rho$ (see [GHLS08, Lut08a] where this result is shown for $\mathcal{SHIQ}$- and $\mathcal{SHQ}$-knowledge bases). We present here a slightly different definition of forest models that is subsumed by the definition of forest models in [GHLS08, Lut08a]. We also define forest models for the more general case of Boolean $\mathcal{ALC}^\cap$-knowledge bases since this will be needed in the proof of Lemma 5.6.

Definition 5.3. *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.*

A model $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ *of a Boolean* $\mathcal{ALC}^\cap$*-knowledge base* $\mathcal{B}$ *is called a* forest model *if*

- $\Delta^\mathcal{I} \subseteq \mathsf{Ind}(\mathcal{B}) \times \mathbb{N}^*$ *such that for all* $a \in \mathsf{Ind}(\mathcal{B})$, *we have that* $\{u \mid (a, u) \in \Delta^\mathcal{I}\}$ *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;*

- *for every* $a \in \mathsf{Ind}(\mathcal{B})$, *we have* $a^\mathcal{I} = (a, \varepsilon)$; *and*

- *for every* $a \in \mathsf{Ind}(\mathcal{B})$, *there is an element* $(a, u) \in \Delta^\mathcal{I}$ *with* $u \neq \varepsilon$ *such that for every* $\mathcal{ALC}^\cap$*-concept description* C, *we have* $a^\mathcal{I} \in C^\mathcal{I}$ *iff* $(a, u) \in C^\mathcal{I}$.

The last condition is required for technical purposes in the proof of Lemma 5.6. We now show that the restriction to forest models in the consistency of a Boolean $\mathcal{ALC}^\cap$-knowledge base is without loss of generality.

Lemma 5.4. *Let* $\mathcal{B}$ *be a Boolean* $\mathcal{ALC}^\cap$*-knowledge base, let* $A_1, \ldots, A_k$ *be concept names occurring in* $\mathcal{B}$, *and let* $\mathcal{D} \subseteq 2^{\{A_1,\ldots,A_k\}}$. $\mathcal{B}$ *has a model that respects* $\mathcal{D}$ *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 $\mathcal{B}$ that respects $\mathcal{D}$. Moreover, we assume that $\Delta^\mathcal{I}$ is countable, which is w.l.o.g. due to the downward Löwenheim-Skolem theorem. We can thus assume w.l.o.g. that $\Delta^\mathcal{I} \subseteq \mathbb{N}$.

We define now a forest model $\mathcal{J} = (\Delta^\mathcal{J}, \cdot^\mathcal{J})$ with domain

$$\Delta^\mathcal{J} := \left\{(a, d_1 \ldots d_m) \mid a \in N_\mathrm{I},\, m \geq 0,\, d_1, \ldots, d_m \in \Delta^\mathcal{I}\right\}$$

as follows:

- $a^\mathcal{J} := (a, \varepsilon)$;

- $A^\mathcal{J} := \{(a, \varepsilon) \mid a^\mathcal{I} \in A^\mathcal{I}\} \cup \{(a, d_0 \ldots d_m) \mid d_m \in A^\mathcal{I}\}$; and

- $r^{\mathcal{J}} := \{((a,\varepsilon),(b,\varepsilon)) \mid a,b \in \mathsf{Ind}(\mathcal{B}),\ (a^{\mathcal{I}},b^{\mathcal{I}}) \in r^{\mathcal{I}}\} \cup$
  $\{((a,\varepsilon),(a,d)) \mid a \in \mathsf{Ind}(\mathcal{B}),\ (a^{\mathcal{I}},d) \in r^{\mathcal{I}}\} \cup$
  $\{((a,d_1 \dots d_m),(a,d_1 \dots d_m d_{m+1})) \mid a \in \mathsf{Ind}(\mathcal{B}),\ m > 0,\ (d_m,d_{m+1}) \in r^{\mathcal{I}}\}$.

Obviously, the conditions for forest models are satisfied. In particular, we have that $a^{\mathcal{J}}$ satisfies exactly the same $\mathcal{ALC}^{\cap}$-concept descriptions as $(a,a^{\mathcal{I}})$. Thus, it is only left to be shown that $\mathcal{J}$ is indeed a model of $\mathcal{B}$ that respects $\mathcal{D}$. We first show by structural induction that $(a,d_1 \dots d_m) \in C^{\mathcal{J}}$ iff either $m = 0$ and $a^{\mathcal{I}} \in C^{\mathcal{I}}$, or $d_m \in C^{\mathcal{I}}$. We assume w.l.o.g. that C is built using only $\exists$, $\sqcap$, and $\neg$.

For the base case, C being a concept name, the claim is directly implied by the definition.

For the case where C is of the form $\neg D$, we have

$(a,d_1 \dots d_m) \in (\neg D)^{\mathcal{J}}$

iff $(a,d_1 \dots d_m) \notin D^{\mathcal{J}}$

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

$(a,d_1 \dots d_m) \in (D \sqcap E)^{\mathcal{J}}$

iff $(a,d_1 \dots d_m) \in D^{\mathcal{J}}$ and $(a,d_1 \dots d_m) \in E^{\mathcal{J}}$

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 $d_m \in (D \sqcap E)^{\mathcal{I}}$.

Finally, for the case where C is of the form $\exists(r_1 \cap \dots \cap r_\ell).D$, we have

$(a,d_1 \dots d_m) \in (\exists(r_1 \cap \dots \cap r_\ell).D)^{\mathcal{J}}$

iff either $m = 0$ and

- there is a $(b,\varepsilon) \in \Delta^{\mathcal{J}}$ with $((a,\varepsilon),(b,\varepsilon)) \in r_1^{\mathcal{J}} \cap \dots \cap r_\ell^{\mathcal{J}}$ and $(b,\varepsilon) \in D^{\mathcal{J}}$, or
- there is a $(a,d) \in \Delta^{\mathcal{J}}$ with $((a,\varepsilon),(a,d)) \in r_1^{\mathcal{J}} \cap \dots \cap r_\ell^{\mathcal{J}}$ and $(a,d) \in D^{\mathcal{J}}$;

or there is a domain element $(a,d_1 \dots d_m d_{m+1}) \in \Delta^{\mathcal{J}}$ such that the pair $((a,d_1 \dots d_m),(a,d_1 \dots d_m d_{m+1}))$ is in $(r_1^{\mathcal{J}} \cap \dots \cap r_\ell^{\mathcal{J}})$ and we have that $(a,d_1 \dots d_m d_{m+1}) \in D^{\mathcal{J}}$

iff either $m = 0$ and there is a $d \in \Delta^{\mathcal{I}}$ such that $(a^{\mathcal{I}}, d) \in r_1^{\mathcal{I}} \cap \dots \cap r_\ell^{\mathcal{I}}$ and $d \in D^{\mathcal{I}}$, or there is a $d \in \Delta^{\mathcal{I}}$ such that $(d_m, d) \in r_1^{\mathcal{I}} \cap \dots \cap r_\ell^{\mathcal{I}}$ and $d \in D^{\mathcal{I}}$

iff either $m = 0$ and $a^{\mathcal{I}} \in (\exists(r_1 \cap \dots \cap r_\ell).D)^{\mathcal{I}}$, or $d_m \in (\exists(r_1 \cap \dots \cap r_\ell).D)^{\mathcal{I}}$.

This finishes the proof of the above claim. We show now for all subformulae $\mathcal{B}'$ of $\mathcal{B}$ that $\mathcal{J}$ is a model of $\mathcal{B}'$ iff $\mathcal{I}$ is a model of $\mathcal{B}'$ by an induction on the structure of $\mathcal{B}'$. Again, we can assume w.l.o.g. that $\mathcal{B}'$ contains just $\wedge$ and $\neg$.

For the first base case, assume that $\mathcal{B}'$ is of the form $C(a)$ for some $\mathcal{ALC}^{\cap}$-concept description C and some $a \in N_{\mathrm{I}}$. We have $a^{\mathcal{I}} \in C^{\mathcal{I}}$ iff $a^{\mathcal{J}} = (a, \varepsilon) \in C^{\mathcal{J}}$ by the above claim, which finishes this case.

For the second base case, assume that $\mathcal{B}'$ is of the form $r(a, b)$ for $a, b \in N_{\mathrm{I}}$ and $r \in N_{\mathrm{R}}$. We have $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$ iff $(a^{\mathcal{J}}, b^{\mathcal{J}}) = ((a, \varepsilon), (b, \varepsilon)) \in r^{\mathcal{J}}$ by the definition of $r^{\mathcal{J}}$, which finishes this case.

For the third base case, assume that $\mathcal{B}'$ is of the form $C \sqsubseteq D$. For the "if" direction, assume that $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. Thus, there is no $d \in C^{\mathcal{I}}$ with $d \notin D^{\mathcal{I}}$. Suppose that there is a $(a, d_1 \dots d_m) \in C^{\mathcal{J}}$ with $(a, d_1 \dots d_m) \notin D^{\mathcal{J}}$. Then, by the above claim, 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 yields a contradiction. For the "only if" direction, assume that $C^{\mathcal{J}} \subseteq D^{\mathcal{J}}$. Thus, there is no $(a, d_1 \dots d_m) \in C^{\mathcal{J}}$ with $(a, d_1 \dots d_m) \notin D^{\mathcal{J}}$. 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 $(a, d) \in \Delta^{\mathcal{J}}$ for any $a \in N_{\mathrm{I}}$. By the above claim, we have that $(a, d) \in C^{\mathcal{J}}$ and $(a, d) \notin D^{\mathcal{J}}$, which yields again a contradiction.

For the induction step, assume first that $\mathcal{B}'$ is of the form $\neg\mathcal{B}''$. We have that $\mathcal{I} \models \mathcal{B}'$ iff $\mathcal{I} \not\models \mathcal{B}''$ iff $\mathcal{J} \not\models \mathcal{B}''$ iff $\mathcal{J} \models \mathcal{B}'$. Assume now that $\mathcal{B}'$ is of the form $\mathcal{B}_1 \wedge \mathcal{B}_2$. We have that $\mathcal{I} \models \mathcal{B}'$ iff $\mathcal{I} \models \mathcal{B}_1$ and $\mathcal{I} \models \mathcal{B}_2$ iff $\mathcal{J} \models \mathcal{B}_1$ and $\mathcal{J} \models \mathcal{B}_2$ iff $\mathcal{J} \models \mathcal{B}'$.

This finishes the proof that $\mathcal{J}$ is a model of $\mathcal{B}$. Moreover, $\mathcal{J}$ respects $\mathcal{D}$ due to the following reasons. We have that

$$\mathcal{D} = \{Y \subseteq \{A_1, \dots, A_k\} \mid \text{there is a } d \in \Delta^{\mathcal{I}} \text{ with } d \in (C_Y)^{\mathcal{I}}\}.$$

Define

$$\begin{aligned}\mathcal{D}' = \{Y \subseteq \{A_1, \dots, A_k\} \mid {} & \text{there is a } (a, d_1 \dots d_m) \in \Delta^{\mathcal{J}} \text{ with} \\ & (a, d_1 \dots d_m) \in (C_Y)^{\mathcal{J}}, \text{ where } a \in N_{\mathrm{I}},\ m \geq 0, \text{ and } d_1, \dots, d_m \in \Delta^{\mathcal{I}}\}.\end{aligned}$$

We show now $\mathcal{D} = \mathcal{D}'$. For the direction ($\subseteq$), assume that $Y \in \mathcal{D}$. For every $d \in \Delta^{\mathcal{I}}$, there is a $(a, d) \in \Delta^{\mathcal{J}}$ by the definition of $\Delta^{\mathcal{J}}$. We have also that $d \in (C_Y)^{\mathcal{I}}$ iff $(a, d) \in (C_Y)^{\mathcal{J}}$. Hence, $Y \in \mathcal{D}'$. For the direction ($\supseteq$), assume that $Y \in \mathcal{D}'$. By construction, for every $(a, d_1 \dots d_m) \in \Delta^{\mathcal{J}}$, there is a $d \in \Delta^{\mathcal{I}}$, where for $m = 0$, we have $d = a^{\mathcal{I}}$, and for $m > 0$, we have $d = d_m$. We have also that $d \in (C_Y)^{\mathcal{I}}$ iff $(a, d_1 \dots d_m) \in (C_Y)^{\mathcal{J}}$. Hence, $Y \in \mathcal{D}$. Since $\mathcal{I}$ respects $\mathcal{D}$, this implies that $\mathcal{J}$ respects $\mathcal{D}$. $\square$

We can now extend the mentioned result about non-entailment of unions of Boolean CQs from [GHLS08, Lut08a] to our setting.

Lemma 5.5. *There is a model $\mathcal{I}$ of $\langle \mathcal{A}, \mathcal{T} \rangle$ that respects $\mathcal{D}$ such that $\mathcal{I} \not\models \rho$ iff there is a forest model $\mathcal{J}$ of $\mathcal{A}$ and $\mathcal{T}$ that respects $\mathcal{D}$ such that $\mathcal{J} \not\models \rho$.*

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 $\langle \mathcal{A}, \mathcal{T} \rangle$ that respects $\mathcal{D}$ such that $\mathcal{I} \not\models \rho$. As shown in the proof of Lemma 5.4, $\mathcal{I}$ can be transformed into a forest model $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ that respects $\mathcal{D}$ by unravelling. It is left to show that then $\mathcal{J} \not\models \rho$. Assume to the contrary that $\mathcal{J} \models \rho$. Then there is a Boolean CQ ρ_i in the union of Boolean CQs 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 all individual names a occurring in the input; and for all $v \in \mathsf{Var}(\rho_i)$, we define $\pi'(v) := a^{\mathcal{I}}$ if $\pi(v) = (a, \varepsilon)$ for $a \in N_{\mathrm{I}}$, and $\pi'(v) = d_m$ if $\pi(v) = (a, d_1 \dots d_m)$ with $m > 0$. It is not hard to verify that π' is indeed a homomorphism from ρ_i into $\mathcal{I}$. Hence, $\mathcal{I} \models \rho_i$, and thus $\mathcal{I} \models \rho$, which is a contradiction. □

Recall that we want to decide the existence of such a forest model in time exponential in the size of $\mathcal{A}$, $\mathcal{T}$, and ρ. To this purpose, we further reduce this problem following an idea from [Lut08a]. There, the notion of a *spoiler* is introduced. A spoiler is an $\mathcal{ALC}^{\cap}$-knowledge base that states properties that must be satisfied such that a query is not entailed by a knowledge base.

It is shown that $\langle \mathcal{A}, \mathcal{T} \rangle \not\models \rho$ iff there is a spoiler $\langle \mathcal{A}', \mathcal{T}' \rangle$ for $\langle \mathcal{A}, \mathcal{T} \rangle$ such that $\langle \mathcal{A} \cup \mathcal{A}', \mathcal{T} \cup \mathcal{T}' \rangle$ is consistent. Additionally, all spoilers can be computed in time exponential in the size of $\langle \mathcal{A}, \mathcal{T} \rangle$ and ρ, and 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[9] $\mathcal{I}$ of $\langle \mathcal{A}, \mathcal{T} \rangle$ that does not satisfy ρ there is a spoiler $\langle \mathcal{A}', \mathcal{T}' \rangle$ that also has $\mathcal{I}$ as a model and, conversely, every forest model of the knowledge base $\langle \mathcal{A}, \mathcal{T} \rangle$ that also satisfies a spoiler $\langle \mathcal{A}', \mathcal{T}' \rangle$ does not satisfy ρ (see Lemma 3 in [Lut08b]).

Thus, it is clear that we have $\langle \mathcal{A}, \mathcal{T} \rangle \not\models \rho$ w.r.t. $\mathcal{D}$ iff there is a spoiler $\langle \mathcal{A}', \mathcal{T}' \rangle$ for $\langle \mathcal{A}, \mathcal{T} \rangle$ such that there is a model of $\langle \mathcal{A} \cup \mathcal{A}', \mathcal{T} \cup \mathcal{T}' \rangle$ that respects $\mathcal{D}$. It now remains to show that the existence of such a model can be checked in exponential time in the size of $\langle \mathcal{A} \cup \mathcal{A}', \mathcal{T} \cup \mathcal{T}' \rangle$, and therefore in exponential time in the size of ϕ and $\mathcal{T}$.

Note that $\langle \mathcal{A} \cup \mathcal{A}', \mathcal{T} \cup \mathcal{T}' \rangle$ can be seen as a special form of a Boolean $\mathcal{ALC}^{\cap}$-knowledge base that contains only conjunctions of axioms, and thus it suffices to show the result for arbitrary Boolean $\mathcal{ALC}^{\cap}$-knowledge bases $\mathcal{B}$. Furthermore, a Boolean $\mathcal{ALC}^{\cap}$-knowledge base $\mathcal{B}$ has a model that respects such a set $\mathcal{D}$ iff the

[9]Recall that such a forest model is also a forest model as defined in [Lut08a] since our definition is more restrictive than the one used there.

Boolean $\mathcal{ALC}^\cap$-knowledge base $\mathcal{B} \wedge \top(a)$ has a model that respects $\mathcal{D}$, where a is a fresh individual name not occurring in $\mathcal{B}$. We thus assume w.l.o.g. from now on that $\mathcal{B}$ contains at least one individual name.

For classical $\mathcal{ALC}^\cap$-knowledge bases, the consistency problem (without $\mathcal{D}$) is ExpTime-complete [Tob01]. We now show that the complexity does not increase for checking the existence of a model of a Boolean $\mathcal{ALC}^\cap$-knowledge base that respects $\mathcal{D}$.

Lemma 5.6. *Let $\mathcal{B}$ be a Boolean $\mathcal{ALC}^\cap$-knowledge base of size n, $A_1, \dots, A_k$ be concept names occurring in $\mathcal{B}$, and $\mathcal{D} \subseteq 2^{\{A_1,\dots,A_k\}}$. Then the existence of a model of $\mathcal{B}$ that respects $\mathcal{D}$ can be decided in time exponential in n.*

Proof. The proof 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 [GKWZ03], which shows that consistency of Boolean $\mathcal{ALC}$-knowledge bases can be decided in exponential time. We use some of the notation introduced in [BGL12].

As in the original proof, we assume w.l.o.g. that all $\mathcal{ALC}^\cap$-concept descriptions in $\mathcal{B}$ are built using only $\sqcap$, $\neg$, and $\exists$; that all GCIs in $\mathcal{B}$ are of the form $\top \sqsubseteq C$; and that GCIs and assertions in $\mathcal{B}$ are combined using only $\wedge$ and $\neg$.

For the subsequent construction, we extend the notion of a *quasimodel* in [BGL12] to deal with role conjunctions. For that we introduce additional concept names that function as so-called *pebbles* that mark elements that have specific role predecessors, an idea borrowed from [Dan84, DM00, Mas01].

Let $\mathsf{Sub}(\mathcal{B})$ denote the closure under negation of the set of all subformulae of $\mathcal{B}$ (where $r(a,b)$ is considered a subformula of $(r \cap s)(a,b)$), and define

$$\begin{aligned}\mathsf{Peb}(\mathcal{B}) := \{&A_{*,r_1,E}, \dots, A_{*,r_\ell,E} \mid \\ &E = \exists(r_1 \cap \dots \cap r_\ell).C \text{ occurs in } \mathcal{B} \text{ and } * \in \mathsf{Ind}(\mathcal{B}) \cup \{\circ\}\}.\end{aligned}$$

Moreover, we define

$$\begin{aligned}\mathsf{PebCond}(\mathcal{B}) := \{&\top \sqsubseteq \neg(A_{*,r_1,E} \sqcap \dots \sqcap A_{*,r_\ell,E} \sqcap C) \mid \\ &E = \exists(r_1 \cap \dots \cap r_\ell).C \text{ occurs in } \mathcal{B} \text{ and } * \in \mathsf{Ind}(\mathcal{B}) \cup \{\circ\}\}.\end{aligned}$$

Intuitively, the pebbles $\mathsf{Peb}(\mathcal{B})$ serve as markers and are used to deal with concepts of the form $E = \forall(r_1 \cap \dots \cap r_\ell).\neg C$. If a named (or unnamed) individual satisfies such a concept, then a pebble is propagated to each r_1-, …, r_ℓ-successor. The pebble remembers the name of the individual (or $\circ$ if unnamed), E, and the respective role name. If an individual satisfies all such pebbles, i.e., it is a r_1-, …, and r_ℓ-successor, then $\mathsf{PebCond}(\mathcal{B})$ ensures that this individual also satisfies $\neg C$, as required by E.

Let furthermore $\mathsf{Con}(\mathcal{B})$ be defined as the closure under negation of the set $\mathsf{Con}'(\mathcal{B})$, where $\mathsf{Con}'(\mathcal{B})$ is defined as follows:

$$\begin{aligned}\mathsf{Con}'(\mathcal{B}) := \{C \mid C \text{ occurs in } \mathcal{B}\} &\cup \mathsf{Peb}(\mathcal{B}) \\ &\cup \{\exists r_i.A_{*,r_i,E} \mid A_{*,r_i,E} \in \mathsf{Peb}(\mathcal{B})\} \cup \{C \mid \top \sqsubseteq C \in \mathsf{PebCond}(\mathcal{B})\}.\end{aligned}$$

We identify $\neg\neg\psi$ with ψ for all concept descriptions and Boolean $\mathcal{ALC}^{\cap}$-knowledge bases ψ. Thus, all the above introduced sets are linear in the size of $\mathcal{B}$.

A *concept type for* $\mathcal{B}$ is a set $\mathbf{c} \subseteq \mathsf{Con}(\mathcal{B}) \cup \mathsf{Ind}(\mathcal{B})$ with the following properties:

- $C \sqcap D \in \mathbf{c}$ iff $C, D \in \mathbf{c}$ for all $C \sqcap D \in \mathsf{Con}(\mathcal{B})$;
- $\neg C \in \mathbf{c}$ iff $C \notin \mathbf{c}$ for all $C \in \mathsf{Con}(\mathcal{B})$;
- $a \in \mathbf{c}$ for $a \in \mathsf{Ind}(\mathcal{B})$ implies $b \notin \mathbf{c}$ for all $b \in \mathsf{Ind}(\mathcal{B})$ with $b \neq a$; and
- $\exists(r_1 \cap \cdots \cap r_\ell).C \in \mathbf{c}$ implies $\{\exists r_1.C, \ldots, \exists r_\ell.C\} \subseteq \mathbf{c}$.

A *formula type for* $\mathcal{B}$ is a set $\mathbf{f} \subseteq \mathsf{Sub}(\mathcal{B}) \cup \mathsf{PebCond}(\mathcal{B})$ with the following properties:

- $\mathsf{PebCond}(\mathcal{B}) \subseteq \mathbf{f}$;
- $(r_1 \cap \cdots \cap r_\ell)(a, b) \in \mathbf{f}$ iff $\{r_1(a, b), \ldots, r_\ell(a, b)\} \subseteq \mathbf{f}$ for all $(r_1 \cap \cdots \cap r_\ell)(a, b)$ occurring in $\mathcal{B}$;
- $\psi_1 \wedge \psi_2 \in \mathbf{f}$ iff $\psi_1, \psi_2 \in \mathbf{f}$ for all $\psi_1 \wedge \psi_2 \in \mathsf{Sub}(\mathcal{B})$; and
- $\neg\psi \in \mathbf{f}$ iff $\psi \notin \mathbf{f}$ for all $\psi \in \mathsf{Sub}(\mathcal{B})$.

Obviously, the number of concept and formula types is exponential in n.

A *model candidate for* $\mathcal{B}$ is a triple $(\mathcal{S}, \iota, \mathbf{f})$ such that $\mathcal{S}$ is a set of concept types for $\mathcal{B}$ such that for any $\mathbf{c}, \mathbf{c}' \in \mathcal{S}$ with $\mathbf{c} \neq \mathbf{c}'$, we have $\mathbf{c} \cap \mathbf{c}' \cap \mathsf{Ind}(\mathcal{B}) = \emptyset$, $\iota\colon \mathsf{Ind}(\mathcal{B}) \to \mathcal{S}$ is a function with $a \in \iota(a)$ for all $a \in \mathsf{Ind}(\mathcal{B})$, and $\mathbf{f}$ is a formula type for $\mathcal{B}$ with the following properties:

(a) $\mathcal{B} \in \mathbf{f}$;

(b) $C(a) \in \mathbf{f}$ iff $C \in \iota(a)$; and

(c) $(r_1 \cap \cdots \cap r_\ell)(a, b) \in \mathbf{f}$ implies $\{\neg C \mid \neg(\exists(r_1 \cap \cdots \cap r_\ell).C) \in \iota(a)\} \subseteq \iota(b)$.

The model candidate $(\mathcal{S}, \iota, \mathbf{f})$ for $\mathcal{B}$ is called a *quasimodel for* $\mathcal{B}$ if it additionally satisfies the following properties:

(d) $\mathcal{S}$ is not empty;

(e) for every $\mathbf{c} \in \mathcal{S}$ and every $\exists(r_1 \cap \cdots \cap r_\ell).C \in \mathbf{c}$, there is a $\mathbf{d} \in \mathcal{S}$ with $\mathbf{d} \cap \mathsf{Ind}(\mathcal{B}) = \emptyset$ such that $\{\neg D \mid \neg(\exists(r_1 \cap \cdots \cap r_\ell).D) \in \mathbf{c}\} \cup \{C\} \subseteq \mathbf{d}$;

(f) for every $E = \exists(r_1 \cap \cdots \cap r_\ell).C$, we have that $\{a, \neg E\} \subseteq \mathbf{c}$ implies $\{\neg(\exists r_1.\neg A_{a,r_1,E}), \ldots, \neg(\exists r_\ell.\neg A_{a,r_\ell,E})\} \subseteq \mathbf{c}$;

(g) for every $E = \exists(r_1 \cap \cdots \cap r_\ell).C$, we have that $\neg E \in \mathbf{c}$ and $\mathbf{c} \cap \mathsf{Ind}(\mathcal{B}) = \emptyset$ implies $\{\neg(\exists r_1.\neg A_{\circ,r_1,E}), \ldots, \neg(\exists r_\ell.\neg A_{\circ,r_\ell,E})\} \subseteq \mathbf{c}$.

(h) for every $\mathbf{c} \in \mathcal{S}$ and every $\top \sqsubseteq C \in \mathsf{Sub}(\mathcal{B}) \cup \mathsf{PebCond}(\mathcal{B})$, if $\top \sqsubseteq C \in \mathbf{f}$, then $C \in \mathbf{c}$;

(k) for every $\top \sqsubseteq C \in \mathsf{Sub}(\mathcal{B}) \cup \mathsf{PebCond}(\mathcal{B})$, if $\neg(\top \sqsubseteq C) \in \mathbf{f}$, then there is a $\mathbf{c} \in \mathcal{S}$ such that $C \notin \mathbf{c}$; and

The quasimodel $(\mathcal{S}, \iota, \mathbf{f})$ for $\mathcal{B}$ *respects* $\mathcal{D}$ if it additionally satisfies

(l) for every $\mathbf{c} \in \mathcal{S}$, there is a set $Y \in \mathcal{D}$ such that $Y = \mathbf{c} \cap \{A_1, \ldots, A_k\}$; and

(m) for every $Y \in \mathcal{D}$, there is a $\mathbf{c} \in \mathcal{S}$ such that $Y = \mathbf{c} \cap \{A_1, \ldots, A_k\}$.

Claim. The Boolean $\mathcal{ALC}^\cap$-knowledge base $\mathcal{B}$ has a model that respects $\mathcal{D}$ iff there is a quasimodel for $\mathcal{B}$ that respects $\mathcal{D}$.

For the "if" direction, suppose that $(\mathcal{S}, \iota, \mathbf{f})$ is a quasimodel for $\mathcal{B}$ that respects $\mathcal{D}$. We define a model $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ as follows:

- $\Delta^\mathcal{I} := \mathcal{S}$;
- $a^\mathcal{I} := \iota(a)$ for all $a \in \mathsf{Ind}(\mathcal{B})$;
- $A^\mathcal{I} := \{\mathbf{c} \in \mathcal{S} \mid A \in \mathbf{c}\}$ for all concept names A of $\mathcal{B}$; and
- for all role names r of $\mathcal{B}$,

$$\begin{aligned} r^\mathcal{I} := \{(\mathbf{c}, \mathbf{c}') \mid{}& \mathbf{c}, \mathbf{c}' \in \mathcal{S},\ \{\neg C \mid \neg(\exists r.C) \in \mathbf{c}\} \subseteq \mathbf{c}', \text{ and} \\ & a \in \mathbf{c} \cap \mathsf{Ind}(\mathcal{B}), b \in \mathbf{c}' \cap \mathsf{Ind}(\mathcal{B}) \text{ implies } r(a,b) \in \mathbf{f}\}. \end{aligned}$$

We prove now by structural induction that for all concept descriptions $C \in \mathsf{Con}(\mathcal{B})$, we have:

$$C^\mathcal{I} = \{\mathbf{c} \in \mathcal{S} \mid C \in \mathbf{c}\}$$

For the base case, C being a concept name, the definition of $\mathcal{I}$ immediately implies the claim. For the case that C is of the form $\neg D$, we have by the semantics of

$\mathcal{ALC}^{\cap}$, the induction hypothesis, the definition of $\mathcal{I}$, and the definition of concept types the following:

$$(\neg D)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus D^{\mathcal{I}} = \mathcal{S} \setminus \{\mathbf{c} \in \mathcal{S} \mid D \in \mathbf{c}\} = \{\mathbf{c} \in \mathcal{S} \mid \neg D \in \mathbf{c}\}.$$

For the case that C is of the form $D \sqcap E$, we have by the semantics of $\mathcal{ALC}^{\cap}$, the induction hypothesis, the definition of $\mathcal{I}$, and the definition of concept types the following:

$$(D \sqcap E)^{\mathcal{I}} = D^{\mathcal{I}} \cap E^{\mathcal{I}} = \{\mathbf{c} \in \mathcal{S} \mid D \in \mathbf{c}\} \cap \{\mathbf{c} \in \mathcal{S} \mid E \in \mathbf{c}\} = \{\mathbf{c} \in \mathcal{S} \mid D \sqcap E \in \mathbf{c}\}.$$

For the case that C is of the form $\exists(r_1 \cap \cdots \cap r_\ell).D$, we have that the semantics of $\mathcal{ALC}^{\cap}$, the induction hypothesis, the definition of $\mathcal{I}$, and the properties of quasimodels imply the following:

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

The starred equality $\overset{*}{=}$ holds due to the following arguments. Assume, for the direction ($\supseteq$), that $\mathbf{c} \in \mathcal{S}$ and $\exists(r_1 \cap \cdots \cap r_\ell).D \in \mathbf{c}$. It follows by (e) of the definition of a quasimodel that there is a $\mathbf{c}' \in \mathcal{S}$ with $\mathbf{c}' \cap \mathsf{Ind}(\mathcal{B}) = \emptyset$ such that $\{\neg E \mid \neg(\exists(r_1 \cap \cdots \cap r_\ell).E) \in \mathbf{c}\} \cup \{D\} \subseteq \mathbf{c}'$. Thus, $D \in \mathbf{c}'$, and we have also

$$\bigcup_{1 \leq i \leq \ell} \{\neg E \mid \neg(\exists r_i.E) \in \mathbf{c}\} \ \subseteq \ \{\neg E \mid \neg(\exists(r_1 \cap \cdots \cap r_\ell).E) \in \mathbf{c}\} \ \subseteq \ \mathbf{c}'.$$

The latter holds due to $\mathbf{c}$ being a concept type. In fact, assume to the contrary that there is some i, $1 \leq i \leq \ell$, such that $\{\neg E \mid \neg(\exists r_i.E) \in \mathbf{c}\} \not\subseteq \{\neg E \mid \neg(\exists(r_1 \cap \cdots \cap r_\ell).E) \in \mathbf{c}\}$. This implies that there is some E with $\neg(\exists r_i.E) \in \mathbf{c}$ and $\neg(\exists(r_1 \cap \cdots \cap r_\ell).E) \notin \mathbf{c}$, and hence $\exists r_i.E \notin \mathbf{c}$ and $\exists(r_1 \cap \cdots \cap r_\ell).E \in \mathbf{c}$, which is a contradiction since $\mathbf{c}$ is a concept type. By the definition of $r_i^{\mathcal{I}}$, for $1 \leq i \leq \ell$, we have $(\mathbf{c},\mathbf{c}') \in r_i^{\mathcal{I}}$. The semantics of $\mathcal{ALC}^{\cap}$ yields $(\mathbf{c},\mathbf{c}') \in r_1^{\mathcal{I}} \cap \cdots \cap r_\ell^{\mathcal{I}}$.

For the other direction ($\subseteq$), assume that $\mathbf{c} \in \mathcal{S}$, and that there is a $\mathbf{c}' \in \mathcal{S}$ with $(\mathbf{c},\mathbf{c}') \in r_1^{\mathcal{I}} \cap \cdots \cap r_\ell^{\mathcal{I}}$ and $D \in \mathbf{c}'$. We show now that $E = \exists(r_1 \cap \cdots \cap r_\ell).D \in \mathbf{c}$. Assume to the contrary that $E \notin \mathbf{c}$. Then, $\neg E \in \mathbf{c}$. By the conditions (f) and (g) of a quasimodel, this implies that $\{\neg(\exists r_1.\neg A_{*,r_1,E}), \ldots, \neg(\exists r_\ell.\neg A_{*,r_\ell,E})\} \subseteq \mathbf{c}$ for some $* \in \mathsf{Ind}(\mathcal{B}) \cup \{\circ\}$. By the definition of $r_1^{\mathcal{I}}, \ldots, r_\ell^{\mathcal{I}}$, we have that $\{A_{*,r_1,E}, \ldots, A_{*,r_\ell,E}\} \subseteq \mathbf{c}'$. We have also by the definition of $\mathsf{PebCond}(\mathcal{B})$ and of formula type that

$$\top \sqsubseteq \neg(A_{*,r_1,E} \sqcap \cdots \sqcap A_{*,r_\ell,E} \sqcap D) \ \in \ \mathsf{PebCond}(\mathcal{B}) \ \subseteq \ \mathbf{f}.$$

By condition (h) of a quasimodel, we have $\neg(A_{*,r_1,E} \sqcap \cdots \sqcap A_{*,r_\ell,E} \sqcap D) \in \mathbf{c}'$, and thus $A_{*,r_1,E} \sqcap \cdots \sqcap A_{*,r_\ell,E} \sqcap D \notin \mathbf{c}'$, which implies $\{A_{*,r_1,E}, \ldots, A_{*,r_\ell,E}, D\} \nsubseteq \mathbf{c}'$ since $\mathbf{c}'$ is a concept type. Since $\{A_{*,r_1,E}, \ldots, A_{*,r_\ell,E}\} \subseteq \mathbf{c}'$ as argued above, it follows that $D \notin \mathbf{c}'$, which contradicts our assumption.

This finishes the proof by structural induction.

It is left to be shown that $\mathcal{I}$ is a model of all $\mathcal{B}' \in \mathbf{f}$, and that $\mathcal{I}$ respects $\mathcal{D}$. This indeed finishes the proof of this direction, because it follows that $\mathcal{I}$ is a model of $\mathcal{B}$, since by (a) of the definition of a model candidate, we have that $\mathcal{B} \in \mathbf{f}$.

We prove the claim that $\mathcal{I}$ is a model of all $\mathcal{B}' \in \mathbf{f}$ again by structural induction. We show that for all $\mathcal{B}' \in \mathsf{Sub}(\mathcal{B}) \cup \mathsf{PebCond}(\mathcal{B})$, we have $\mathcal{B}' \in \mathbf{f}$ iff $\mathcal{I} \models \mathcal{B}'$.

For the first base case, assume $\mathcal{B}'$ is of the form $\top \sqsubseteq C$. If $\top \sqsubseteq C \in \mathbf{f}$, then for every $\mathbf{c} \in \mathcal{S}$, we have that $C \in \mathbf{c}$ by (h) of the definition of a quasimodel. Thus, $C^{\mathcal{I}} = \mathcal{S} = \Delta^{\mathcal{I}}$. For the converse direction, if $\top \sqsubseteq C \notin \mathbf{f}$, then by the definition of a formula type, $\neg(\top \sqsubseteq C) \in \mathbf{f}$. Then, by (k) of the definition of a quasimodel, there is a $\mathbf{c} \in \mathcal{S}$ such that $C \notin \mathbf{c}$, which implies $\mathbf{c} \notin C^{\mathcal{I}}$. Hence, $C^{\mathcal{I}} \neq \mathcal{S} = \Delta^{\mathcal{I}}$.

For the second base case, assume $\mathcal{B}'$ is of the form $C(a)$. We have $C(a) \in \mathbf{f}$ iff $C \in \iota(a)$ by (b) of the definition of a model candidate. Thus, $C(a) \in \mathbf{f}$ iff $\iota(a) = a^{\mathcal{I}} \in C^{\mathcal{I}}$.

For the third base case, assume $\mathcal{B}'$ is of the form $(r_1 \cap \cdots \cap r_\ell)(a,b)$. If we have that $(r_1 \cap \cdots \cap r_\ell)(a,b) \in \mathbf{f}$, then we have also that

$$\bigcup_{1 \le i \le \ell} \{\neg D \mid \neg(\exists r_i.D) \in \iota(a)\} \;\subseteq\; \{\neg D \mid \neg(\exists (r_1 \cap \cdots \cap r_\ell).D) \in \iota(a)\} \;\subseteq\; \iota(b).$$

The first $\subseteq$ holds because of the fact that $\iota(a)$ is a concept type as argued above. The second $\subseteq$ holds because of condition (c) of a model candidate. Since $\mathbf{f}$ is a formula type, we have also $\{r_1(a,b), \ldots, r_\ell(a,b)\} \subseteq \mathbf{f}$. By the definition of $r_i^{\mathcal{I}}$ for i, $1 \le i \le \ell$, we have $(\iota(a), \iota(b)) \in r_i^{\mathcal{I}}$ for i, $1 \le i \le \ell$, and thus also $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r_1^{\mathcal{I}} \cap \cdots \cap r_\ell^{\mathcal{I}}$. For the converse direction, suppose $(r_1 \cap \cdots \cap r_\ell)(a,b) \notin \mathbf{f}$. Since $\mathbf{f}$ is a formula type, we have $\{r_1(a,b), \ldots, r_\ell(a,b)\} \nsubseteq \mathbf{f}$, and thus there is some i, $1 \le i \le \ell$, with $r_i(a,b) \notin \mathbf{f}$. By the definition of $r_i^{\mathcal{I}}$ this implies that $(\iota(a), \iota(b)) \notin r_i^{\mathcal{I}}$. In particular, this means that $(a^{\mathcal{I}}, b^{\mathcal{I}}) = (\iota(a), \iota(b)) \notin r_1^{\mathcal{I}} \cap \cdots \cap r_\ell^{\mathcal{I}}$.

For the induction step, assume first that $\mathcal{B}'$ is of the form $\neg\widehat{\mathcal{B}}$. We have $\mathcal{B}' \in \mathbf{f}$ iff $\widehat{\mathcal{B}} \notin \mathbf{f}$ (by the definition of a formula type) iff $\mathcal{I} \not\models \widehat{\mathcal{B}}$ (by the induction hypothesis) iff $\mathcal{I} \models \neg\widehat{\mathcal{B}}$.

Assume now that $\mathcal{B}'$ is of the form $\widehat{\mathcal{B}}_1 \wedge \widehat{\mathcal{B}}_2$. We have $\mathcal{B}' \in \mathbf{f}$ iff $\{\widehat{\mathcal{B}}_1, \widehat{\mathcal{B}}_2\} \subseteq \mathbf{f}$ (by the definition of a formula type) iff $\mathcal{I} \models \widehat{\mathcal{B}}_1$ and $\mathcal{I} \models \widehat{\mathcal{B}}_2$ (by the induction hypothesis) iff $\mathcal{I} \models \widehat{\mathcal{B}}_1 \wedge \widehat{\mathcal{B}}_2$.

It is left to be shown that $\mathcal{I}$ indeed respects $\mathcal{D}$. By condition (l), the definition of $\mathcal{I}$, and the arguments above, we have that for every $d \in \Delta^{\mathcal{I}}$, there is a set $Y \in \mathcal{D}$ such that $d \in (C_Y)^{\mathcal{I}}$. By condition (m), the definition of $\mathcal{I}$, and the arguments above, we have also that for every $Y \in \mathcal{D}$, there is a $d \in \Delta^{\mathcal{I}}$ such that $d \in (C_Y)^{\mathcal{I}}$. Hence, we have that

$$\mathcal{D} = \{Y \subseteq \{A_1, \ldots, A_k\} \mid \text{there is a } d \in \Delta^{\mathcal{I}} \text{ such that } d \in (C_Y)^{\mathcal{I}}\},$$

and thus that $\mathcal{I}$ respects $\mathcal{D}$.

This finishes the proof of the "if" direction of the claim. For the "only if" direction, assume that there is a model $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ of $\mathcal{B}$ that respects $\mathcal{D}$. Due to Lemma 5.4, we can assume w.l.o.g. that $\mathcal{J}$ is a forest model. We extend $\mathcal{J}$ to $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ by also interpreting the elements of $\mathsf{Peb}(\mathcal{B})$. More precisely, $\mathcal{I}$ is defined as $\mathcal{J}$ plus:

- $A^{\mathcal{I}}_{a,r,E} := \left\{d \mid (a^{\mathcal{J}}, d) \in r^{\mathcal{J}} \text{ and } a^{\mathcal{J}} \in (\neg E)^{\mathcal{J}}\right\}$ and
- $A^{\mathcal{I}}_{\circ,r,E} := \Big\{d \mid \text{there is a } d' \in (\neg E)^{\mathcal{J}} \text{ such that } (d', d) \in r^{\mathcal{J}} \text{ and } d' \neq a^{\mathcal{J}}$ for all $a \in \mathsf{Ind}(\mathcal{B})\Big\}$

for all $E = \exists(r_1 \cap \cdots \cap r \cap \ldots r_\ell).C \in \mathsf{Sub}(\mathcal{B})$. Since the elements of $\mathsf{Peb}(\mathcal{B})$ do not occur in $\mathcal{B}$ or $\mathcal{D}$, $\mathcal{I}$ is a model of $\mathcal{B}$ that respects $\mathcal{D}$.

We now construct a quasimodel for $\mathcal{B}$. Let $\tau(e) := \{C \in \mathsf{Con}(\mathcal{B}) \mid e \in C^{\mathcal{I}}\}$ for $e \in \Delta^{\mathcal{I}}$. We define $(\mathcal{S}, \iota, \mathbf{f})$ as follows:

- $\mathcal{S} := \{\tau(d) \mid d \in \Delta^{\mathcal{I}}, d \neq a^{\mathcal{I}} \text{ for all } a \in \mathsf{Ind}(\mathcal{B})\} \cup \{\tau(a^{\mathcal{I}}) \cup \{a\} \mid a \in \mathsf{Ind}(\mathcal{B})\}$;
- $\iota(a) := \tau(a^{\mathcal{I}}) \cup \{a\}$ for $a \in \mathsf{Ind}(\mathcal{B})$; and
- $\mathbf{f} := \{\mathcal{B}' \in \mathsf{Sub}(\mathcal{B}) \mid \mathcal{I} \models \mathcal{B}'\} \cup \mathsf{PebCond}(\mathcal{B})$.

Obviously, $\mathcal{S}$ is a set of concept types for $\mathcal{B}$, and $\mathbf{f}$ is a formula type for $\mathcal{B}$. We show now that $(\mathcal{S}, \iota, \mathbf{f})$ is a quasimodel for $\mathcal{B}$.

For condition (a), it is easy to see that $\mathcal{B} \in \mathbf{f}$, because $\mathcal{I} \models \mathcal{B}$.

For condition (b), we have $C(a) \in \mathbf{f}$ iff $\mathcal{I} \models C(a)$ iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$ iff $C \in \tau(a^{\mathcal{I}}) \subseteq \iota(a)$.

For condition (c), assume that $(r_1 \cap \cdots \cap r_\ell)(a, b) \in \mathbf{f}$. Then, $\mathcal{I} \models (r_1 \cap \cdots \cap r_\ell)(a, b)$, and thus $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r_1^{\mathcal{I}} \cap \cdots \cap r_\ell^{\mathcal{I}}$. Obviously, $\{\neg C \mid \mathcal{I} \models \neg(\exists(r_1 \cap \cdots \cap r_\ell).C)(a)\} \subseteq \tau(b^{\mathcal{I}})$. Hence, $\{\neg C \mid a^{\mathcal{I}} \in (\neg(\exists(r_1 \cap \cdots \cap r_\ell).C))^{\mathcal{I}}\} \subseteq \iota(b)$, and thus $\{\neg C \mid \neg(\exists(r_1 \cap \cdots \cap r_\ell).C) \in \tau(a^{\mathcal{I}})\} \subseteq \iota(b)$, which implies $\{\neg C \mid \neg(\exists(r_1 \cap \cdots \cap r_\ell).C) \in \iota(a)\} \subseteq \iota(b)$.

Condition (d) is easily verified, because $\Delta^{\mathcal{I}} \neq \emptyset$ by definition.

For condition (e), take $\mathbf{c} \in \mathcal{S}$ and $\exists(r_1 \cap \cdots \cap r_\ell).C \in \mathbf{c}$. Then, we have that there is a $d \in \Delta^{\mathcal{I}}$ with $d \in (\exists(r_1 \cap \cdots \cap r_\ell).C)^{\mathcal{I}}$ and $\tau(d) = \mathbf{c} \setminus \mathsf{Ind}(\mathcal{B})$. By the semantics of $\mathcal{ALC}^{\cap}$, there is an $e \in \Delta^{\mathcal{I}}$ such that $(d, e) \in r_1^{\mathcal{I}} \cap \cdots \cap r_\ell^{\mathcal{I}}$, $e \in C^{\mathcal{I}}$, and for each D with $d \in (\neg(\exists(r_1 \cap \cdots \cap r_\ell).D))^\mathcal{I}$, we have that $e \in (\neg D)^\mathcal{I}$. Since $\mathcal{I}$ is a forest model, we have $\tau(e) \in \mathcal{S}$. Indeed, if there is an $a \in N_\mathrm{I}$ with $a^\mathcal{I} = e$, then there is an $e' \in \Delta^\mathcal{I}$ with $e' \neq b^\mathcal{I}$ for any $b \in N_\mathrm{I}$ such that $\tau(e) = \tau(e')$. Hence, $\{\neg D \mid d \in (\neg(\exists(r_1 \cap \cdots \cap r_\ell).D))^\mathcal{I}\} \cup \{C\} \subseteq \tau(e)$. Thus, there is a $\mathbf{d} \in \mathcal{S}$, namely $\mathbf{d} := \tau(e)$, with $\mathbf{d} \cap \mathsf{Ind}(\mathcal{B}) = \emptyset$ and $\{\neg D \mid \neg(\exists(r_1 \cap \cdots \cap r_\ell).D) \in \mathbf{c}\} \cup \{C\} \subseteq \mathbf{d}$.

For condition (f), take $\mathbf{c} \in \mathcal{S}$ and $E = \exists(r_1 \cap \cdots \cap r_\ell).C$ with $\{a, \neg E\} \subseteq \mathbf{c}$. By the definition of $\mathcal{S}$, we have that $\mathbf{c} = \tau(a^\mathcal{I}) \cup \{a\}$. Thus, $a^\mathcal{I} \in (\neg E)^\mathcal{I}$. It is enough to show that $\{\neg(\exists r_1.\neg A_{a,r_1,E}), \ldots, \neg(\exists r_\ell.\neg A_{a,r_\ell,E})\} \subseteq \tau(a^\mathcal{I})$. Assume to the contrary that there is an i, $1 \leq i \leq \ell$, such that $\neg(\exists r_i.\neg A_{a,r_i,E}) \notin \tau(a^\mathcal{I})$. Hence, $a^\mathcal{I} \notin (\neg(\exists r_i.\neg A_{a,r_i,E}))^\mathcal{I}$, and thus $a^\mathcal{I} \in (\exists r_i.\neg A_{a,r_i,E})^\mathcal{I}$. Then, there is a $d \in \Delta^\mathcal{I}$ such that $(a^\mathcal{I}, d) \in r_i^\mathcal{I}$ and $d \in (\neg A_{a,r_i,E})^\mathcal{I}$, and thus $d \notin A^\mathcal{I}_{a,r_i,E}$. By the definition of $A^\mathcal{I}_{a,r_i,E}$, we have $a^\mathcal{I} \notin (\neg E)^\mathcal{I}$, which is a contradiction.

For condition (g), take $\mathbf{c} \in \mathcal{S}$ and $E = (\exists(r_1 \cap \cdots \cap r_\ell).C)$ with $\neg E \in \mathbf{c}$ and $\mathbf{c} \cap \mathsf{Ind}(\mathcal{B}) = \emptyset$. Thus, there is a $d' \in \Delta^\mathcal{I}$ with $\tau(d') = \mathbf{c}$ and $d' \neq a^\mathcal{I}$ for all $a \in \mathsf{Ind}(\mathcal{B})$. This yields $(d')^\mathcal{I} \in (\neg E)^\mathcal{I}$. It is enough to show that $\{\neg(\exists r_1.\neg A_{\circ,r_1,E}), \ldots, \neg(\exists r_\ell.\neg A_{\circ,r_\ell,E})\} \subseteq \tau(d')$. Assume to the contrary that there is an i, $1 \leq i \leq \ell$, such that $\neg(\exists r_i.\neg A_{\circ,r_i,E}) \notin \tau(d')$. Hence, $(d')^\mathcal{I} \notin (\neg(\exists r_i.\neg A_{\circ,r_i,E}))^\mathcal{I}$, and thus $(d')^\mathcal{I} \in (\exists r_i.\neg A_{\circ,r_i,E})^\mathcal{I}$. Then, there is a $d \in \Delta^\mathcal{I}$ such that $(d', d) \in r_i^\mathcal{I}$ and $d \in (\neg A_{\circ,r_i,E})^\mathcal{I}$, and thus $d \notin A^\mathcal{I}_{\circ,r_i,E}$. By the definition of $A^\mathcal{I}_{\circ,r_i,E}$, we have $(d')^\mathcal{I} \notin (\neg E)^\mathcal{I}$, which is a contradiction.

For condition (h), take $\mathbf{c} \in \mathcal{S}$ and $\top \sqsubseteq C \in \mathsf{Sub}(\mathcal{B}) \cup \mathsf{PebCond}(\mathcal{B})$ with $\top \sqsubseteq C \in \mathbf{f}$. We prove $C \in \mathbf{c}$ with a case distinction. Either $\top \sqsubseteq C \in \mathsf{Sub}(\mathcal{B})$ or $\top \sqsubseteq C \in \mathsf{PebCond}(\mathcal{B})$. For the first case, we have $\mathcal{I} \models \top \sqsubseteq C$, 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 $\mathcal{S}$ that $C \in \mathbf{c}$. For the second case, C must be of the form $\neg(A_{*,r_1,E} \sqcap \cdots \sqcap A_{*,r_\ell,E} \sqcap D)$ for some $E = \exists(r_1 \cap \cdots \cap r_\ell).D \in \mathsf{Sub}(\mathcal{B})$ and $* \in \mathsf{Ind}(\mathcal{B}) \cup \{\circ\}$. It is enough to show that $\{A_{*,r_1,E}, \ldots, A_{*,r_\ell,E}, D\} \not\subseteq \mathbf{c}$, which implies $C \in \mathbf{c}$ since $\mathbf{c}$ is a concept type. We need another case distinction: either $a \in \mathbf{c}$ for some $a \in \mathsf{Ind}(\mathcal{B})$ or $\mathbf{c} \cap \mathsf{Ind}(\mathcal{B}) = \emptyset$.

In the first case, we have $\mathbf{c} = \tau(a^\mathcal{I}) \cup \{a\}$. Assume to the contrary that we have $\{A_{*,r_1,E}, \ldots, A_{*,r_\ell,E}, D\} \subseteq \tau(a^\mathcal{I}) \subseteq \mathbf{c}$. We have $* \neq \circ$ since otherwise by definition of $A^\mathcal{I}_{\circ,r_i,E}$ there must be at least one $d \in \Delta^\mathcal{I}$ with $(d, a^\mathcal{I}) \in r_i^\mathcal{I}$ for some i, $1 \leq i \leq \ell$. This contradicts the fact that $\mathcal{I}$ is a forest model. Thus, $* = b$ for $b \in N_\mathrm{I}$. By the definition of $\tau(a^\mathcal{I})$, we have $a^\mathcal{I} \in A^\mathcal{I}_{b,r_1,E}$, $\ldots$, $a^\mathcal{I} \in A^\mathcal{I}_{b,r_\ell,E}$, and $a^\mathcal{I} \in D^\mathcal{I}$. The definition of $A_{b,r_i,E}$, $1 \leq i \leq \ell$, yields that $(b^\mathcal{I}, a^\mathcal{I}) \in r_1^\mathcal{I} \cap \cdots \cap r_\ell^\mathcal{I}$ and $b^\mathcal{I} \in (\neg E)^\mathcal{I}$. The semantics of $\mathcal{ALC}^\cap$ yields $a^\mathcal{I} \notin D^\mathcal{I}$, which is a contradiction.

In the second case, we have $\mathbf{c} \cap \mathsf{Ind}(\mathcal{B}) = \emptyset$. By the definition of $\mathcal{S}$, we have $\mathbf{c} = \tau(d)$ for some $d \in \Delta^\mathcal{I}$ with $d \neq a^\mathcal{I}$ for any $a \in N_\mathrm{I}$. Assume to the contrary that we have $\{A_{*,r_1,E}, \ldots, A_{*,r_\ell,E}, D\} \subseteq \tau(d) = \mathbf{c}$. By the definition of $\tau(d)$, we have $d \in A^\mathcal{I}_{*,r_1,E}$, $\ldots$, $d \in A^\mathcal{I}_{*,r_\ell,E}$, and $d \in D^\mathcal{I}$. By the definition of $A^\mathcal{I}_{*,r_i,E}$, $1 \leq i \leq \ell$, we have that there are $d_1, \ldots, d_\ell \in \Delta^\mathcal{I}$ such that $(d_i, d) \in r_i^\mathcal{I}$, and $d_i \in (\neg E)^\mathcal{I}$. Since $\mathcal{I}$ is a forest model, and $d \neq a^\mathcal{I}$ for any $a \in N_\mathrm{I}$, we have that d is of the form

$(b, d'_1 \dots d'_m)$ for $m > 0$, and $d_1 = \cdots = d_\ell = (b, d'_1 \dots d'_{m-1})$ for some $b \in N_\mathrm{I}$. This implies that $(d_1, d) \in r_1^\mathcal{I} \cap \cdots \cap r_\ell^\mathcal{I}$. The semantics of $\mathcal{ALC}^\cap$ yields that $d \notin D^\mathcal{I}$, which is a contradiction.

For condition (k), take $\neg(\top \sqsubseteq C) \in \mathbf{f}$. By the definition of $\mathbf{f}$, $\top \sqsubseteq C \notin \mathsf{PebCond}(\mathcal{B})$, and thus $\top \sqsubseteq C \in \mathsf{Sub}(\mathcal{B})$. Again by the definition of $\mathbf{f}$, this implies $\mathcal{I} \not\models \top \sqsubseteq C$. By the semantics of $\mathcal{ALC}^\cap$, there is a $d \in \Delta^\mathcal{I}$ with $d \notin C^\mathcal{I}$. Thus, for $\tau(d) \in \mathcal{S}$, we have $C \notin \tau(d)$.

For condition (l), let $\mathbf{c} \in \mathcal{S}$. Then there must be a $d \in \Delta^\mathcal{I}$ with $\tau(d) \subseteq \mathbf{c}$. Since $\mathcal{I}$ respects $\mathcal{D}$, there must be a set $Y \in \mathcal{D}$ such that $d \in A^\mathcal{I}$ for all $A \in Y$, and $d \notin B^\mathcal{I}$ for all $B \notin Y$. Hence, by definition of $\tau(d)$, we have $Y = \mathbf{c} \cap \{A_1, \dots, A_k\}$.

For condition (m), let $Y \in \mathcal{D}$. Since $\mathcal{I}$ respects $\mathcal{D}$, there must be a $d \in \Delta^\mathcal{I}$ such that $d \in A^\mathcal{I}$ for all $A \in Y$, and $d \notin B^\mathcal{I}$ for all $B \notin Y$. Hence, by definition of $\tau(d)$, we have $Y = \tau(d) \cap \{A_1, \dots, A_k\}$.

This finishes the proof of the claim that $\mathcal{B}$ has a model that respects $\mathcal{D}$ iff there is a quasimodel for $\mathcal{B}$ that respects $\mathcal{D}$.

It remains to show that one can check the existence of a quasimodel for $\mathcal{B}$ that respects $\mathcal{D}$ in time exponential in n. This can be achieved by a simple adaptation of the proof of Lemma 6.4 in [BGL12], which shows the claim for $\mathcal{ALC}$-knowledge bases.

The algorithm works as follows. Given $\mathcal{B}$ and $\mathcal{D}$, it enumerates all model candidates $(\mathcal{S} \cup \mathcal{S}_\iota, \iota, \mathbf{f})$ for $\mathcal{B}$ where

- $\mathcal{S}$ is the set of *all* concept types for $\mathcal{B}$ that are subsets of $\mathsf{Con}(\mathcal{B})$ and
- $\mathcal{S}_\iota := \{\iota(a) \mid a \in \mathsf{Ind}(\mathcal{B}), \iota(a) \setminus \{a\} \in \mathcal{S}\}$

We denote these candidates by $\mathfrak{C}_1, \dots, \mathfrak{C}_N$. Note that each of these candidates is of size exponential in n. It should be clear that

$$N \;\leq\; 2^{|\mathsf{Con}(\mathcal{B})|\cdot|\mathsf{Ind}(\mathcal{B})|} \cdot 2^{|\mathsf{Sub}(\mathcal{B}) \cup \mathsf{PebCond}(\mathcal{B})|},$$

and thus the enumeration of all $\mathfrak{C}_1, \dots, \mathfrak{C}_N$ can be done in time exponential in n since $\mathsf{PebCond}(\mathcal{B})$ and $\mathsf{Con}(\mathcal{B})$ are of size polynomial in n.

Now, set $i = 1$ and consider $\mathfrak{C}_i = (\mathcal{S}, \iota, \mathbf{f})$.

Step 1. Check each concept type in $\mathcal{S}$. We call a concept type $\mathbf{c} \in \mathcal{S}$ *defective* if one of the following conditions holds:

- (e) is violated for some $\exists(r_1 \cap \cdots \cap r_\ell).C \in \mathbf{c}$;
- (f) is violated for some $\neg(\exists(r_1 \cap \cdots \cap r_\ell).C) \in \mathbf{c}$ if $\mathbf{c} \cap \mathsf{Ind}(\mathcal{B}) = \{a\}$;

- (g) is violated for some $\neg(\exists(r_1 \cap \cdots \cap r_\ell).C) \in \mathbf{c}$ if $\mathbf{c} \cap \mathsf{Ind}(\mathcal{B}) = \emptyset$;
- (h) is violated for some $\top \sqsubseteq C \in \mathbf{f}$; or
- (l) is violated.

If we have found a defective concept type $\mathbf{c}$ with $\mathbf{c} \cap \mathsf{Ind}(\mathcal{B}) = \emptyset$, then set $\mathcal{S} := \mathcal{S} \setminus \{\mathbf{c}\}$ and continue with Step 1. If we have found a defective concept type $\mathbf{c}$ with $\mathbf{c} \cap \mathsf{Ind}(\mathcal{B}) \neq \emptyset$, then stop considering $\mathfrak{C}_i$ and go to Step 3. If we have found no defective concept types in $\mathcal{S}$, continue with Step 2.

Step 2. Check whether the model candidate $(\mathcal{S}', \iota, \mathbf{f})$ obtained from Step 1 satisfies (d), (k), and (m). If it does, stop with output "quasimodel that respects $\mathcal{D}$ found." Otherwise, continue with Step 3.

Step 3. Set $i := i + 1$. If $i \leq N$, continue with Step 1. Otherwise, stop with output "no quasimodel that respects $\mathcal{D}$ exists."

It is easy to see that the algorithm is sound and complete. It is also not hard to see that it runs in time exponential in n. □

We get the following consequence as corollary; see [BGL12].

Corollary 5.7. *The satisfiability problem of the temporal DL $\mathcal{ALC}^{\cap}$-LTL w.r.t. rigid concepts is* NExpTime*-complete.*

Combining the reductions of this section, we get the desired complexity result.

Theorem 5.8. *If $N_{RC} \neq \emptyset$ and $N_{RR} = \emptyset$, then the entailment problem is in co-*NExpTime *w.r.t. combined complexity.*

6 Conclusions

We have introduced a new temporal query language that extends the temporal DL $\mathcal{ALC}$-LTL to using conjunctive queries as atoms. Our complexity results on the entailment problem for such queries w.r.t. temporal knowledge bases are summarized in Table 3.3. Without any rigid names, we observed that entailment of TCQs is as hard as entailment of CQs w.r.t. atemporal $\mathcal{ALC}$-knowledge bases, i.e., in this case adding temporal operators to the query language does not increase the complexity. However, if we allow for rigid concept names (but no rigid role names), the picture changes. While the data complexity remains the same as in the atemporal case, the combined complexity of query entailment increases to co-NExpTime, i.e., the non-entailment problem is as hard as satisfiability in $\mathcal{ALC}$-LTL. If we further add rigid role names, the combined complexity (of non-entailment) again increases in accordance with the complexity of satisfiability in $\mathcal{ALC}$-LTL. For data complexity, it is still unclear whether adding rigid role names results in an increase. 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 consider temporal queries based on inexpressive DLs such as DL-Lite [CDL+09], and check under what conditions query answering can be realized using classical (temporal or atemporal) database techniques.

References

[AHV95] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases.* Addison-Wesley, 1995.

[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 Martin Giese and Arild Waaler, editors, *Proceedings of the 18th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (Tableaux 2009)*, volume 5607 of *Lecture Notes in Computer Science*, pages 77–92. Springer-Verlag, 2009.

[BCM+03] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press, 2003.

[BGL08] Franz Baader, Silvio Ghilardi, and Carsten Lutz. LTL over description logic axioms. In Gerhard Brewka and Jérôme Lang, editors, *Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning (KR 2008)*, pages 684–694, Los Altos, 2008. Morgan Kaufmann.

[BGL12] Franz Baader, Silvio Ghilardi, and Carsten Lutz. LTL over description logic axioms. *ACM Transactions on Computational Logic*, 13(3), 2012. Extended version of [BGL08].

[BK08] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking.* The MIT Press, Cambridge, Massachusetts, USA, 2008.

[CDL98] Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. On the decidability of query containment under constraints. In Alberto O.

Mendelzon and Jan Paredaens, editors, *Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'98)*, pages 149—158, Seattle, Washington, USA, 1998. ACM Press.

[CDL+06] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Data complexity of query answering in description logics. In Patrick Doherty, John Mylopoulos, and Christopher A. Welty, editors, *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR 2006)*, pages 260–270, Lake District of the United Kingdom, June 2006. AAAI Press.

[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 Sergio Tessaris, Enrico Franconi, Thomas Eiter, Claudio Gutierrez, Siegfried Handschuh, Marie-Christine Rousset, and Renate A. Schmidt, editors, *Reasoning Web, 5th Int. Summer School 2009, Tutorial Lectures*, volume 5689 of *Lecture Notes in Computer Science*, pages 255–356. Springer-Verlag, Brixen-Bressanone, Italy, August 2009.

[CM77] Ashok K. Chandra and Philip M. Merlin. Optimal implementation of conjunctive queries in relational data bases. In John E. Hopcroft, Emily P. Friedman, and Michael A. Harrison, editors, *Proceedings of the 9th Annual ACM Symposium on Theory of Computing (STOC'77)*, pages 77–90, Boulder, Colorado, USA, May 1977. ACM Press.

[Dan84] Ryszard Danecki. Nondeterministic propositional dynamic logic with intersection is decidable. In Andrzej Skowron, editor, *Proceedings of the 5th Symposium on Computation Theory*, volume 208 of *Lecture Notes in Computer Science*, pages 34–53, Zaborów, Poland, December 1984. Springer-Verlag.

[DEFS99] Stefan Decker, Michael Erdmann, Dieter Fensel, and Rudi Studer. Ontobroker: Ontology based access to distributed and semi-structured information. In Robert Meersman, Zahir Tari, and Scott M. Stevens, editors, *Proceedings of the 8th Working Conference on Database Semantics (DS-8)*, volume 138 of *IFIP Conference Proceedings*, pages 351–369, Rotorua, New Zealand, January 1999. Kluwer.

[DM00] Giuseppe De Giacomo and Fabio Massacci. Combining deduction and model checking into tableaux and algorithms for Converse-PDL. *Information and Computation*, 162(1–2):117–137, 2000.

[End95] Mica R. Endsley. Toward a theory of situation awareness in dynamic systems. *Human Factors*, 37(1):32–64, 1995.

[GHLS08] Birte Glimm, Ian Horrocks, Carsten Lutz, and Ulrike Sattler. Answering conjunctive queries in the $\mathcal{SHIQ}$ description logic. *Journal of Artificial Intelligence Research*, 31:150–197, 2008.

[GKWZ03] Dov M. Gabbay, Ágnes Kurucz, Frank Wolter, and Michael Zakharyaschev. *Many-Dimensional Modal Logics: Theory and Applications*. Elsevier Science, 2003.

[GPVW96] Rob Gerth, Doron A. Peled, Moshe Y. Vardi, and Pierre Wolper. Simple on-the-fly automatic verification of linear temporal logic. In Piotr Dembinski and Marek Sredniawa, editors, *Proceedings of the 15th IFIP WG6.1 International Symposium on Protocol Specification, Testing and Verification XV*, pages 3–18, London, UK, 1996. Chapman & Hall, Ltd.

[Lut08a] Carsten Lutz. The complexity of conjunctive query answering in expressive description logics. In Alessandro Armando, Peter Baumgartner, and Gilles Dowek, editors, *Proceedings of the 4th International Joint Conference on Automated Reasoning (IJCAR 2008)*, volume 5195 of *Lecture Notes in Computer Science*, pages 179–193, Sydney, Australia, August 2008. Springer-Verlag.

[Lut08b] Carsten Lutz. Two upper bounds for conjunctive query answering in $\mathcal{SHIQ}$. In Franz Baader, Carsten Lutz, and Boris Motik, editors, *Proceedings of the 21st International Workshop on Description Logics (DL 2008)*, volume 353 of *CEUR Workshop Proceedings*, 2008.

[Mas01] Fabio Massacci. Decision procedures for expressive description logics with intersection, composition, converse of roles and role identity. In Bernhard Nebel, editor, *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI 2001)*, pages 193–198, Seattle, Washington, USA, August 2001. Morgan Kaufmann.

[OCE06] Magdalena Ortiz, Diego Calvanese, and Thomas Eiter. Characterizing data complexity for conjunctive query answering in expressive description logics. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI 2006) and the 18th Innovative Applications of Artificial Intelligence Conference*, pages 275–280, Boston, Massachusetts, USA, July 2006. AAAI Press.

[PCDG+08] Antonella Poggi, Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Linking data to ontologies. *Journal on Data Semantics*, X:133–173, 2008.

[Pnu77] Amir Pnueli. The temporal logic of programs. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science*

(FOCS 1977), pages 46–57, Providence, Rhode Island, USA, 1977. IEEE Computer Society Press.

[RG10] Sebastian Rudolph and Birte Glimm. Nominals, inverses, counting, and conjunctive queries or: Why infinity is your friend! *Journal of Artificial Intelligence Research*, 39(1):429–481, 2010.

[Sch91] Klaus Schild. A correspondence theory for terminological logics: Preliminary report. In John Mylopoulos and Raymond Reiter, editors, *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI'91)*, pages 466–471. Morgan Kaufmann, 1991.

[SS91] Manfred Schmidt-Schauß and Gert Smolka. Attributive concept descriptions with complements. *Artificial Intelligence Journal*, 48(1):1–26, 1991.

[Tes01] Sergio Tessaris. *Questions and Answers: Reasoning and Querying in Description Logic*. PhD thesis, University of Manchester, 2001.

[Tob01] Stephan Tobies. *Complexity Results and Practical Algorithms for Logics in Knowledge Representation*. PhD thesis, Fakultät für Mathematik, Informatik und Naturwissenschaften, RWTH Aachen, 2001.

[VW94] Moshe Y. Vardi and Pierre Wolper. Reasoning about infinite computations. *Information and Computation*, 155(1):1–37, 1994.

[WVS83] Pierre Wolper, Moshe Y. Vardi, and A. Prasad Sistla. Reasoning about infinite computation paths. In *Proceedings of the 24th Annual Symposium on Foundations of Computer Science (FOCS 1983)*, pages 185–194. IEEE Computer Society Press, 1983.