TECHNISCHE
UNIVERSITÄT
DRESDEN

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

LTCS–Report

Temporal Query Answering in $\mathcal{EL}$

Stefan Borgwardt Veronika Thost

LTCS-Report 15-08

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

Abstract

Context-aware systems use data about their environment for adaptation at runtime, e.g., for optimization of power consumption or user experience. Ontology-based data access (OBDA) can be used to support the interpretation of the usually large amounts of data. OBDA augments query answering in databases by dropping the closed-world assumption (i.e., the data is not assumed to be complete any more) and by including domain knowledge provided by an ontology. We focus on a recently proposed temporalized query language that allows to combine conjunctive queries with the operators of the well-known propositional temporal logic LTL. In particular, we investigate temporalized OBDA w.r.t. ontologies in the DL $\mathcal{EL}$, which allows for efficient reasoning and has been successfully applied in practice. We study both data and combined complexity of the query entailment problem.

Contents

1 Introduction

Context-aware systems use data about their environment for adaptation at runtime [BBB+09, HSK09], e.g., for optimization of power consumption or user experience. This data is usually collected in a large scale and continuously by different sensors (e.g., the operating system or other, possibly external, sources) and stored in a database. Interpreting the information available in the database, the context-aware system is supposed to recognize certain predefined situations (e.g., that an application is out of user focus), which require an adaptation (e.g., the optimization of application parameters w.r.t. power consumption).

OBDA

In a simple setting, such a context-aware system can be realized by using standard database techniques: the sensor information is stored in a database, and the situations to be recognized are specified as database queries [AHV95]. However, we cannot assume that the sensors provide a complete description of the current state of the environment. Thus, the closed-world assumption employed by database systems (i.e., facts not present in the database are assumed to be false) is not appropriate since there may be facts of which the truth is not known. For example, a sensor for specific information might not be available for some time or not even exist.

In addition, though a complete specification of the environment usually does not exist, some knowledge about its behavior is often available (e.g., that a video application is out of user focus if the user does not watch the video for a while). This background knowledge could be used to support the interpretation of the sensor data to identify predefined, more complex contexts at runtime (e.g., that an application actually is out of user focus); by answering queries based on the predefined contexts, the contexts identified in this way then can be used to dynamically recognize complex situations.

Ontology-based data access (OBDA) [PLC+08, DEFS98] addresses these two points by (i) viewing the data as an *ABox*, which is interpreted under the open-world assumption, and (ii) representing additional background knowledge in a *TBox* (or *ontology*). ABox and TBox together form a *knowledge base*, and are written in an appropriate ontology language; for example, a *Description Logic* (DL) [BCM+03].

For example, assume that we have an ABox containing the following facts about *individuals*, formed using unary and binary predicates, in DL terminology called *concepts* and *roles*, respectively:

$$\begin{array}{ll} \mathsf{User}(\mathsf{bob}), & \mathsf{NotWatchingVideo}(\mathsf{bob}), \\ \mathsf{VideoApplication}(\mathsf{xPlayer}), & \mathsf{hasUser}(\mathsf{xPlayer}, \mathsf{bob}), \\ \mathsf{TextApplication}(\mathsf{openOffice}), & \mathsf{hasUser}(\mathsf{openOffice}, \mathsf{bob}), \\ \mathsf{OperatingSystem}(\mathsf{os}) & \end{array}$$

We can thus describe that the individual Bob is a user that is not watching a video, that there are two applications used by Bob, and that the system is currently optimizing the user experience w.r.t. the video application, e.g., by setting a high resolution.

In addition, a corresponding TBox may contain the following background information:

$$\mathsf{VideoApplication} \sqcap \exists \mathsf{hasUser}.\mathsf{NotWatchingVideo} \sqsubseteq \exists \mathsf{hasState}.\mathsf{OutOfFocus},$$

Hence, a video application is described to have the state 'out of user focus' if its user does not watch the video.

Given that kind of information, we can recognize the situation when the system is optimizing for an application that is out of user focus to potentially adapt and optimize w.r.t. a different application; for example, by answering the following simple *conjunctive query* (CQ) over the example knowledge base, we can identify applications x that can potentially be assigned a lower priority:

$$\psi(x) := \exists y.\mathsf{hasState}(x, y) \wedge \mathsf{OutOfFocus}(y)$$

This method has several drawbacks. For example, a context-aware system usually optimizes the application parameters once and adjusts them in random intervals, but not continuously. Moreover, it is questionable to assume that a user not watching the video at a single moment in time is not focusing on the application any more.

For that reason, we want to investigate *temporal conjunctive queries* (TCQs) [BBL15], where the query may refer to several points in time.

Temporalized OBDA

Originally proposed by [BBL13, BBL15], TCQs allow to combine CQs via Boolean operators and the temporal operators of the well-known propositional temporal logic LTL [Pnu77]. For example, the situation described above could be specified more elaborately as follows:

$$\begin{aligned} &\big(\bigcirc^{-}\, \psi(x)\big) \wedge \big(\bigcirc^{-}\, \bigcirc^{-}\psi(x)\big) \wedge \big(\bigcirc^{-}\, \bigcirc^{-}\, \bigcirc^{-}\, \psi(x)\big) \wedge \\ &\Big(\neg\Big(\exists y.\mathsf{GotPriority}(y) \wedge \mathsf{notEqual}(x, y)\Big)\, \mathsf{S}\, \mathsf{GotPriority}(x)\Big) \end{aligned}$$

to obtain all applications that were out of user focus during the three *previous* ($\bigcirc^-$) moments of observation, were prioritized by the operating system at some point in time, and the priority has *not* ($\neg$) changed *since* (S) then.

To apply context-aware situation recognition by answering TCQs, we extend the overall setting of OBDA as proposed in [BBL15]. Specifically, we consider a *temporal knowledge base*, which, in addition to the TBox for the background knowledge (this knowledge is assumed to hold at all points in time), contains a *sequence* of ABoxes $\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_n$, each containing the sensor data observed —and thus describing the state of the system—at a specific point in time. We designate with n the most recent time point at which we have observed the state of the system, and will call it the *current time point.* Given this data, we want to evaluate a TCQ recognizing a certain situation at the current time point.

In our setting, the information within the TBox and the ABoxes thus does not explicitly refer to the temporal dimension, but is written in a *classical* (atemporal) DL; only the query is temporalized. In contrast, so-called *temporal DLs* [LWZ08, AKL[+]07, AKRZ14, AKK[+]14, GJS14, ABM[+]14] extend classical DLs by temporal operators, which then occur within the knowledge base. However, as it is shown in [LWZ08, AKL[+]07, AKRZ14, GJS14], most of these logics yield high reasoning complexities, even if the underlying atemporal DL allows for tractable reasoning. For that reason, lower complexities are only obtained by either considerably restricting the set of temporal operators or the underlying DL.

A simplified version of TCQs called $\mathcal{ALC}$-LTL, which allows to combine only a very restricted subset of CQs (i.e., $\mathcal{ALC}$ axioms) via LTL operators, has been introduced in [BGL12]. In [BBL13, BBL15], the problem of answering TCQs over knowledge bases in the rather expressive DLs $\mathcal{ALC}$ and $\mathcal{SHQ}$ has been investigated. However, reasoning in these DLs is not tractable anymore, and context-aware systems often need to deal with large quantities of data and adapt fast. Several lightweight logics have been considered in [BLT15], but this article does not consider full TCQs since it does not allow negation in the query language. Similarly, the formulas considered in [AKL[+]07] w.r.t. KBs in tractable DLs are very restricted. This motivates our study focusing on TCQs and the DL $\mathcal{EL}$, which allows for efficient reasoning [BBL05] and has been successfully applied in practice, e.g., in large biomedical ontologies like SNOMED CT.[1]

Contribution

In this report, we consider TCQ answering over temporal knowledge bases in $\mathcal{EL}$ and investigate the complexity of the query entailment problem.

As in [BGL12, BBL15], we also consider *rigid* concepts and roles, whose interpretation does not change over time. This makes sense regarding our application

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

Table 1.1: The complexity of TCQ entailment in $\mathcal{EL}$

allowed rigid symbols	data complexity	combined complexity
none	P	PSPACE
	LB: [CDL+06], UB: 5.2	LB: [SC85]
concept names	CO-NP	PSPACE
	LB: 5.4	UB: 4.14
role names	CO-NP	CO-NEXPTIME
	UB: 5.5	LB: 4.16, UB: 4.17

scenario of a context-aware system, where certain concepts and roles should definitely be interpreted rigidly (e.g., an application will always be an application).

We investigate both the combined and the data complexity of the query entailment problem in three different settings: (i) both concepts and roles may be rigid (Sections 4.2 and 5.2); (ii) only concepts may be rigid (Sections 4.1 and 5.2); and (iii) neither concepts nor roles are allowed to be rigid (Sections 4.1 and 5.1). The case where roles, but not concepts, are allowed to be rigid, is the same as setting (i) since rigid concepts can be simulated using rigid roles [BGL12].

Our results are summarized in Table 1.1. Compared to TCQs in $\mathcal{ALC}$ and $\mathcal{SHQ}$ [BBL15], the combined complexity decreases in all cases (from 2-EXPTIME to CO-NEXPTIME, from CO-NEXPTIME to PSPACE, and from EXPTIME to PSPACE, respectively). For the data complexity, we can show reduced upper bounds for cases (i) and (iii) (CO-NP instead of EXPTIME and P instead of CO-NP, respectively), whereas the data complexity remains in CO-NP for the second case. Apart from the latter case, the only previous results that directly apply to TCQ answering in $\mathcal{EL}$ are the PSPACE lower bound for satisfiability in propositional LTL [SC85] and the P lower bound for the data complexity of CQ answering in atemporal $\mathcal{EL}$ [CDL+06].

2 Preliminaries

We first introduce the description logic $\mathcal{EL}$ and then define TCQs over temporal knowledge bases formulated in $\mathcal{EL}$, as it was done for $\mathcal{ALC}$ in [BBL15].

2.1 The Description Logic $\mathcal{EL}$

The syntax of $\mathcal{EL}$ is defined as follows.

Definition 2.1 (Syntax of $\mathcal{EL}$). *Let* $\mathsf{N_C}$, $\mathsf{N_R}$, *and* $\mathsf{N_I}$, *respectively, be non-empty, pairwise disjoint sets of* concept names, role names, *and* individual names. *In* *the description logic $\mathcal{EL}$, the set of* (complex) concepts *is the smallest set such that*

- *all concept names $A \in \mathsf{N_C}$ are concepts,*
- *if C and D are concepts, and $r \in \mathsf{N_R}$, then $\top$ (top), $C \sqcap D$ (conjunction), and $\exists r.C$ (existential restriction) are concepts.*

A general concept inclusion *(GCI) is of the form $C \sqsubseteq D$, where C and D are concepts, and an* assertion *is of the form $A(a)$ or $r(a, b)$, where $A \in \mathsf{N_C}$, $r \in \mathsf{N_R}$, and $a, b \in \mathsf{N_I}$. An* axiom *is either a GCI or a assertion.*

A TBox *is a finite set of GCIs and an* ABox *is a finite set of assertions. Together, a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$ form a* knowledge base *$\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$.*

We furthermore denote by $\mathsf{N_I}(\mathcal{K})$ the set of individual names that occur in the knowledge base $\mathcal{K}$, by $\mathsf{N_C}(\mathcal{T})$ ($\mathsf{N_{RC}}(\mathcal{T})$) the set of (rigid) concept names that occur in the TBox $\mathcal{T}$, and by $\mathsf{Sub}(\mathcal{T})$ the set of all subconcepts that occur in the TBox $\mathcal{T}$. Sometimes, we use the abbreviation $\exists r_1 \ldots r_\ell.C$ for the concept $\exists r_1. \ldots \exists r_\ell.C$.

We define the semantics of $\mathcal{EL}$ as usual in a model-theoretic way.

Definition 2.2 (Semantics of $\mathcal{EL}$)**.** *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 $A \in \mathsf{N_C}$ a set $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$, to every $r \in \mathsf{N_R}$ a binary relation $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$, and to every $a \in \mathsf{N_I}$ an element $a^\mathcal{I} \in \Delta^\mathcal{I}$.*

This function is extended to complex concepts as follows:

- $\top^\mathcal{I} := \Delta^\mathcal{I}$*;*
- $(C \sqcap D)^\mathcal{I} := C^\mathcal{I} \cap D^\mathcal{I}$*; and*
- $(\exists r.C)^\mathcal{I} := \{d \in \Delta^\mathcal{I} \mid \exists e \in \Delta^\mathcal{I}, (d, e) \in r^\mathcal{I}, e \in C^\mathcal{I}\}$.

The interpretation $\mathcal{I}$ satisfies *(or is a* model *of)*

- *a GCI $C \sqsubseteq D$ if $C^\mathcal{I} \subseteq D^\mathcal{I}$;*
- *an assertion $A(a)$ if $a^\mathcal{I} \in A^\mathcal{I}$;*
- *an assertion $r(a, b)$ if $(a^\mathcal{I}, b^\mathcal{I}) \in r^\mathcal{I}$;*
- *an knowledge base if it satisfies all axioms contained in it.*

We write $\mathcal{I} \models \alpha$ if $\mathcal{I}$ satisfies the axiom α, $\mathcal{I} \models \mathcal{T}$ if $\mathcal{I}$ satisfies all GCIs in the TBox $\mathcal{T}$, $\mathcal{I} \models \mathcal{A}$ if $\mathcal{I}$ satisfies all assertions in the ABox $\mathcal{A}$, and $\mathcal{I} \models \mathcal{K}$ if $\mathcal{I}$ is a model of the knowledge base $\mathcal{K}$. Further, a knowledge base $\mathcal{K}$ is said to be consistent *iff it has model.*

Throughout the report, we assume that all interpretations $\mathcal{I}$ satisfy the unique name assumption (UNA), (i.e., for all $a, b \in \mathsf{N_I}$ with $a \neq b$, we have that $a^\mathcal{I} \neq b^\mathcal{I}$).

We sometimes consider also ABoxes that contain *negated concept assertions* of the form $\neg A(a)$, which are satisfied by an interpretation $\mathcal{I}$ if $a^\mathcal{I} \notin A^\mathcal{I}$. However, they can be simulated in the extension $\mathcal{EL}^{++}$ of $\mathcal{EL}$ by GCIs of the form $\{a\} \sqcap A \sqsubseteq \bot$.[2] Thus, consistency of knowledge bases containing negated assertions can be decided in polynomial time [BBL05].

2.2 Temporal Conjunctive Queries

This report focuses on a temporal query language originally proposed in [BBL13], but we consider here knowledge bases formulated in $\mathcal{EL}$ instead of $\mathcal{ALC}$. The queries are formulas of propositional LTL, where the propositions are replaced by CQs, and are then answered over temporal knowledge bases, according to a semantics that is suitably lifted from propositional worlds to interpretations.

In the following, we assume (as in [BGL12, BBL15]) that a subset of the concept and role names is designated as being *rigid* (as opposed to *flexible*). The intuition is that the interpretation of the rigid names is not allowed to change over time. In particular, the individual names are implicitly assumed to be rigid (i.e., an individual always has the same name). We denote by $\mathsf{N_{RC}} \subseteq \mathsf{N_C}$ the rigid concept names, and by $\mathsf{N_{RR}} \subseteq \mathsf{N_R}$ the rigid role names.

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

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 non-empty domain Δ that is fixed (constant domain assumption). Then $\mathfrak{I}$ is a model of $\mathcal{K}$ (written $\mathfrak{I} \models \mathcal{K}$) if

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

[2] The constructor $\bot$ (*bottom*) is interpreted as the empty set, whereas $\{a\}$ (*nominal*) is interpreted as the singleton set $\{a^\mathcal{I}\}$ [BBL05].

We denote by $\mathsf{N_I}(\mathcal{K})$ the set of all individual names occurring in the TKB $\mathcal{K}$.

As mentioned above, our query language combines conjunctive queries via LTL operators.

Definition 2.4 (Syntax of TCQs)**.** *Let $\mathsf{N_V}$ be a set of variables. A* conjunctive query *(CQ) is of the form $\phi = \exists x_1, \ldots, x_m.\psi$, where $x_1, \ldots, x_m \in \mathsf{N_V}$ and ψ is a (possibly empty) finite conjunction of atoms of the form*

- $A(t)$ *(*concept atom*), for $A \in \mathsf{N_C}$ and $t \in \mathsf{N_I} \cup \mathsf{N_V}$, or*
- $r(t_1, t_2)$ *(*role atom*), for $r \in \mathsf{N_R}$ and $t_1, t_2 \in \mathsf{N_I} \cup \mathsf{N_V}$.*

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 the following are also TCQs:*
  - $\neg\phi_1$ *(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 $\mathsf{N_I}(\phi)$, the set of variables occurring in ϕ by $\mathsf{N_V}(\phi)$, the set of free variables of ϕ by $\mathsf{FVar}(\phi)$, and the set of atoms occurring in ϕ by $\mathsf{At}(\phi)$. A TCQ ϕ with $\mathsf{FVar}(\phi) = \emptyset$ is called a *Boolean* TCQ. A *CQ-literal* is either a CQ or a negated CQ, and a *union of CQs* (UCQ) is a disjunction of CQs.

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

Since we focus on the analysis of entailment of TCQs, we define the semantics of CQs and TCQs only for Boolean queries. As usual, it is given through the notion of homomorphisms [CM77].

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

- $\pi(a) = a^\mathcal{I}$*, for all $a \in \mathsf{N_I}(\psi)$;*
- $\pi(t) \in A^\mathcal{I}$*, for all concept atoms $A(t)$ in ψ; and*
- $(\pi(t_1), \pi(t_2)) \in r^\mathcal{I}$*, for all role atoms $r(t_1, t_2)$ 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 and $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$ be an infinite sequence of interpretations. We define the satisfaction relation $\mathfrak{I}, i \models \phi$, where $i \geq 0$, by induction on the structure of ϕ:

$$
\begin{array}{lll}
\mathfrak{I}, i \models \exists x_1, \ldots, x_m.\psi & \textit{iff} & \mathcal{I}_i \models \exists x_1, \ldots, x_m.\psi \\
\mathfrak{I}, i \models \neg\phi_1 & \textit{iff} & \mathcal{I}_i \not\models \exists\phi_1 \\
\mathfrak{I}, i \models \phi_1 \wedge \phi_2 & \textit{iff} & \mathfrak{I}, i \models \phi_1 \textit{ and } \mathfrak{I}, i \models \phi_2 \\
\mathfrak{I}, i \models \bigcirc\phi_1 & \textit{iff} & \mathfrak{I}, i+1 \models \phi_1 \\
\mathfrak{I}, i \models \bigcirc^{-}\phi_1 & \textit{iff} & i > 0 \textit{ and } \mathfrak{I}, i-1 \models \phi_1 \\
\mathfrak{I}, i \models \phi_1 \,\mathsf{U}\, \phi_2 & \textit{iff} & \textit{there is some } k \geq i \textit{ such that } \mathfrak{I}, k \models \phi_2 \\
 & & \textit{and } \mathfrak{I}, j \models \phi_1, \textit{ for all } j,\ i \leq j < k \\
\mathfrak{I}, i \models \phi_1 \,\mathsf{S}\, \phi_2 & \textit{iff} & \textit{there is some } k,\ 0 \leq k \leq i, \textit{ such that } \mathfrak{I}, k \models \phi_2 \\
 & & \textit{and } \mathfrak{I}, j \models \phi_1, \textit{ for all } j,\ k < j \leq i.
\end{array}
$$

Given a TKB $\mathcal{K} = \langle \mathcal{T}, (\mathcal{A}_i)_{0\leq i\leq n} \rangle$, $\mathfrak{I}$ is called 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. $\mathcal{K}$. Furthermore, ϕ is* entailed *by $\mathcal{K}$ (written $\mathcal{K} \models \phi$) if every model of $\mathcal{K}$ is also a model of ϕ.*

Especially note that, as mentioned in the introduction, models of TCQs consider the current time point n.

We will often deal with conjunctions of CQ-literals ϕ. Since ϕ contains no temporal operators, the satisfaction of ϕ by an infinite sequence of interpretations $\mathcal{I} = (\mathcal{I}_i)_{i\geq 0}$ 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 $\mathcal{I}, i \models \phi$. For the same reason, we use this notation also for unions of CQs. In this context, it is sufficient to deal with classical knowledge bases $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, which can be seen as TKBs with only one ABox.

We now define the semantics of non-Boolean TCQs.

Definition 2.6 (Certain Answer)**.** *Let ϕ be a TCQ and $\mathcal{K} = \langle \mathcal{T}, (\mathcal{A}_i)_{0\leq i\leq n} \rangle$, be a temporal knowledge base. The mapping $\mathfrak{a}\colon \mathsf{FVar}(\phi) \to \mathsf{N_I}(\mathcal{K})$ is a* certain answer *to ϕ w.r.t. $\mathcal{K}$ if, for every $\mathcal{I} \models \mathcal{K}$, we have $\mathcal{I}, n \models \mathfrak{a}(\phi)$, where $\mathfrak{a}(\phi)$ denotes the Boolean TCQ that is obtained from ϕ by replacing the free variables according to $\mathfrak{a}$.*

As usual, the problem of computing all certain answers to a TCQ reduces to exponentially many entailment problems. In this report, we study the complexity of entailment via the satisfiability problem, which has the same complexity as the complement of the entailment problem [BBL15].

We consider two kinds of complexity measures: combined complexity and data complexity. For the combined complexity, all parts of the input, meaning the TCQ ϕ and the entire temporal knowledge base $\mathcal{K}$, are taken into account. In contrast, for the data complexity, the TCQ ϕ and the TBox $\mathcal{T}$ are assumed to be constant, and thus the complexity is measured only w.r.t. the data, i.e., the sequence of ABoxes. Note that the data complexity is actually suited quite well for our use case, where we can assume that both the domain knowledge and the specifications of the situations we want to recognize are given at design time as a TBox and a set of TCQs, respectively.

Recall that we assumed that all concept names in the ABoxes also occur in the TBox. If this was not the case, we could simply add trivial axioms like $A \sqsubseteq \top$ to $\mathcal{T}$ in order to satisfy this requirement. Although formally this increases the size of $\mathcal{T}$, these axioms do not affect the semantics of $\mathcal{T}$, and can thus be ignored in all reasoning problems involving $\mathcal{T}$. All complexity results remain valid without this assumption.

We will also assume that TCQs use only individual names that occur in the ABoxes, and only concept and role names that occur in the TBox; this is clearly without loss of generality.

All our proofs of upper bounds are based on the approach described in [BGL12, BBL15]. We now introduce definitions that are important in this construction.

The *propositional abstraction* ϕ^{p} 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 $\alpha_1, \dots, \alpha_m$ occurring in ϕ and the propositional variables $p_1, \dots, p_m$ occurring in ϕ^{p}. The formula ϕ^{p} obtained in this way is a propositional LTL-formula [Pnu77].

Definition 2.7 (LTL). *Let $\{p_1, \dots, p_m\}$ be a finite set of propositional variables. An* LTL-formula *ϕ is built inductively from these variables using the constructors negation ($\neg\phi_1$), conjunction ($\phi_1 \wedge \phi_2$), next ($\bigcirc\phi_1$), previous ($\bigcirc^-\phi_1$), until ($\phi_1 \mathsf{U} \phi_2$), and since ($\phi_1 \mathsf{S} \phi_2$).*

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

Note that the above definition extends the usual definition of LTL, which only considers the temporal operators $\bigcirc$ and U [Pnu77]. For this reason, this extended logic is often referred to as Past-LTL.

2.3 Atemporal Queries and Canonical Models

We conclude the introductory definitions by considering some properties of atemporal queries.

Definition 2.8 (Tree-shaped). *We call a CQ* tree-shaped *if it does not contain individual names and the directed graph described by its atoms is a tree, i.e., it has a unique* root variable *from which all other variables can be reached by a unique path described by role atoms.*

For a tree-shaped CQ α with root variable x, we set $\mathsf{Con}(\alpha) := \mathsf{Con}(\alpha, x)$, where

$$\mathsf{Con}(\alpha, y) := \bigsqcap_{A(y)\in\alpha} A \sqcap \bigsqcap_{r(y,z)\in\alpha} \exists r.\mathsf{Con}(\alpha, z).$$

This definition of $\mathsf{Con}(\alpha)$ is similar to the notion of "rolled-up" queries used by [Ros07].

For simplicity, we assume that all Boolean CQs we encounter are *connected*, meaning that the variables and individual names are related by roles, as defined in [RG10], for example.

Definition 2.9 (Connected). *A Boolean CQ ψ is called* connected *if, for all $t, t' \in \mathsf{N_I}(\psi) \cup \mathsf{N_V}(\psi)$, there exists a sequence $t_1, \dots, t_\ell \in \mathsf{N_I}(\psi) \cup \mathsf{N_V}(\psi)$ such that $t = t_1$ and $t' = t_\ell$ and for all i,$1 \leq i \leq \ell$, there is a $r \in \mathsf{N_R}$ such that either $r(t_i, t_{i+1}) \in \mathsf{At}(\psi)$ or $r(t_{i+1}, t_i) \in \mathsf{At}(\psi)$. A collection of Boolean CQs $\psi_1, \dots, \psi_m$ is a partition of ψ if $\mathsf{At}(\psi) = \mathsf{At}(\psi_1) \cup \dots \cup \mathsf{At}(\psi_m)$, the sets $\mathsf{N_I}(\psi_i) \cup \mathsf{N_V}(\psi_i)$, $1 \leq i \leq m$, are pairwise disjoint, and each ψ_i is connected.*

It follows from a result in [Tes01] that we can assume Boolean TCQs to only contain connected CQs without loss of generality: if a Boolean TCQ ϕ contains a CQ ψ that is not connected, then we can replace ψ by the conjunction $\psi_1 \wedge \dots \wedge \psi_\ell$, where $\psi_1, \dots, \psi_\ell$ 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 by a collection of homomorphisms of $\psi_1, \dots, \psi_\ell$ into $\mathcal{I}$.

We now recall the well-known construction of so-called *canonical models* for knowledge bases in $\mathcal{EL}$ [KL07, LTW09, Ros07, KRH07]. We consider elements c_ϱ, where ϱ is a *path* of the form $ar_1C_1 \dots r_nC_n$, where a is an individual name, $r_1, \dots, r_n$ are role names, and $C_1, \dots, C_n$ are concepts appearing in the knowledge base. Intuitively, ϱ describes a role path in a model of the knowledge base that starts at the domain element denoted by a and proceeds through role connections via $r_1, \dots, r_n$ to new elements $e_1, \dots, e_n$ such that each e_i satisfies C_i. The canonical model contains only those elements c_ϱ for which the presence of a path corresponding to ϱ is enforced by the knowledge base.

Definition 2.10 (Canonical Model). *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a knowledge base. We first define the set*

$$\Delta_{\mathsf{u}}^{\mathcal{I}_\mathcal{K}} := \bigcup_{j=0}^{\infty} \Delta_{\mathsf{u}}^j,$$

where

$$\Delta_{\mathsf{u}}^{0} := \{c_{arD} \mid a \in \mathsf{N_I}(\mathcal{A}),\ D \in \mathsf{Sub}(\mathcal{T}),\ \mathcal{K} \models \exists r.D(a)\} \text{ and}$$
$$\Delta_{\mathsf{u}}^{j+1} := \{c_{\varrho rDsE} \mid \exists c_{\varrho rD} \in \Delta_{\mathsf{u}}^{j}, \mathcal{T} \models D \sqsubseteq \exists s.E\}.$$

The canonical interpretation $\mathcal{I}_\mathcal{K}$ for $\mathcal{K}$ is defined as follows, for all $a \in \mathsf{N_I}(\mathcal{A})$, $A \in \mathsf{N_C}$, and $r \in \mathsf{N_R}$:

$$\begin{aligned}
\Delta^{\mathcal{I}_\mathcal{K}} &:= \mathsf{N_I}(\mathcal{A}) \cup \Delta_{\mathsf{u}}^{\mathcal{I}_\mathcal{K}},\\
a^{\mathcal{I}_\mathcal{K}} &:= a,\\
A^{\mathcal{I}_\mathcal{K}} &:= \{a \in \mathsf{N_I}(\mathcal{A}) \mid \mathcal{K} \models A(a)\} \cup\\
&\quad\ \{c_{\varrho rD} \in \Delta_{\mathsf{u}}^{\mathcal{I}_\mathcal{K}} \mid \mathcal{T} \models D \sqsubseteq A\}, \text{ and}\\
r^{\mathcal{I}_\mathcal{K}} &:= \{(a,b) \mid r(a,b) \in \mathcal{A}\} \cup\\
&\quad\ \{(a, c_{arD}) \in \mathsf{N_I}(\mathcal{A}) \times \Delta_{\mathsf{u}}^{\mathcal{I}_\mathcal{K}}\} \cup\\
&\quad\ \{(c_\varrho, c_{\varrho rD}) \in \Delta_{\mathsf{u}}^{\mathcal{I}_\mathcal{K}} \times \Delta_{\mathsf{u}}^{\mathcal{I}_\mathcal{K}}\}.
\end{aligned}$$

It is easy to see that this indeed defines a model of the input knowledge base. It is also a prototype for all other models of the KB in the sense that it includes only those domain elements whose presence is enforced by the axioms. Therefore, the canonical interpretation can be embedded into every other model and we have the property that entailment of CQs w.r.t. the KB can simply be answered over the canonical model.

Proposition 2.11 ([LTW09])**.** *$\mathcal{I}_\mathcal{K}$ is a model of $\mathcal{K}$ and, for all CQs ψ, we have $\mathcal{K} \models \psi$ iff $\mathcal{I}_\mathcal{K} \models \psi$.*

The following auxiliary lemma is easy to prove by induction on the structure of concepts (cf. Lemma 4.9).

Lemma 2.12. *For all elements $c_{\varrho rD} \in \Delta_{\mathsf{u}}^{\mathcal{I}_\mathcal{K}}$ and concepts $C \in \mathsf{Sub}(\mathcal{T})$, we have $c_{\varrho rD} \in C^{\mathcal{I}_\mathcal{K}}$ iff $\mathcal{T} \models D \sqsubseteq C$.*

3 On Upper Bounds

In this section, we describe a general approach to solve the satisfiability problem (and thus the entailment problem), which has been proposed in [BBL15, BGL12]. This procedure is then used in later sections to obtain several upper bounds.

In a nutshell, the satisfiability problem of a TCQ w.r.t. a TKB is reduced to two separate satisfiability problems—one in LTL and one in $\mathcal{EL}$. We describe this approach in the following. Let $\mathcal{K} = \langle \mathcal{T}, (\mathcal{A}_i)_{0 \le i \le n} \rangle$ be a TKB and ϕ be a Boolean TCQ. For the LTL part, we consider the propositional abstraction ϕ^{p} of ϕ, which contains the propositional variables $p_1, \dots, p_m$ in place of the CQs $\alpha_1, \dots, \alpha_m$ occurring in ϕ. Let them be such that α_i was replaced by p_i, for $1 \le i \le m$. Furthermore, we define a set $\mathcal{S} \subseteq 2^{\{p_1,\dots,p_m\}}$, which specifies the worlds that are allowed to occur in an LTL-structure satisfying ϕ^{p}. This can be described with the following propositional LTL-formula:

$$\phi^{\mathsf{p}}_{\mathcal{S}} = \phi^{\mathsf{p}} \wedge \Box \left(\bigvee_{X \in \mathcal{S}} \left(\bigwedge_{p \in X} p \wedge \bigwedge_{p \in \overline{X}} \neg p \right) \right),$$

where we denote by $\overline{X} := \{p_1, \dots, p_m\} \setminus X$ the complement of a world $X \in \mathcal{S}$.

Nevertheless, for checking whether ϕ has a model w.r.t. $\mathcal{K}$ it is not sufficient to guess a set $\mathcal{S}$ and to then test whether the induced LTL-formula $\phi^{\mathsf{p}}_{\mathcal{S}}$ is satisfiable at time point n. 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 needs to be satisfied for this to hold. That is, it covers the part of satisfiability regarding $\mathcal{EL}$.

Definition 3.1 (r-satisfiable). *Given a set $\mathcal{S} = \{X_1, \dots, X_k\} \subseteq 2^{\{p_1,\dots,p_m\}}$ and a mapping $\iota \colon \{0, \dots, n\} \to \{1, \dots, k\}$, $\mathcal{S}$ is called* r-satisfiable *w.r.t. ι and $\mathcal{K}$ if there are interpretations $\mathcal{J}_1, \dots, \mathcal{J}_k, \mathcal{I}_0, \dots, \mathcal{I}_n$ such that*

- *the interpretations share the same domain and respect rigid names;*[3]
- *the interpretations are models of $\mathcal{T}$;*
- *for all i, $1 \le i \le k$, $\mathcal{J}_i$ is a model of $\chi_i := \bigwedge_{p_j \in X_i} \alpha_j \wedge \bigwedge_{p_j \in \overline{X_i}} \neg \alpha_j$; and*
- *for all i, $0 \le i \le n$, $\mathcal{I}_i$ is a model of $\mathcal{A}_i$ and $\chi_{\iota(i)}$.*

Note that, through the existence of the interpretations $\mathcal{J}_i$, $1 \le i \le k$, it is ensured that the conjunction χ_i of CQ-literals specified by X_i is consistent. A set $\mathcal{S}$ containing a set X_i for which this does not hold cannot be induced by any model of $\mathcal{K}$. Moreover, the ABoxes are considered through the interpretations $\mathcal{I}_i$, $0 \le i \le n$, which represent the first $n+1$ interpretations in such a model.

This two-fold approach for solving the satisfiability problem, which we sketched above, is formalized in the next lemma.

Lemma 3.2 ([BBL15, Lemma 4.7]). *The TCQ ϕ has a model w.r.t. the TKB $\mathcal{K}$ iff there are a set $\mathcal{S} = \{X_1, \dots, X_k\} \subseteq 2^{\{p_1,\dots,p_m\}}$ and a mapping $\iota \colon \{0, \dots, n\} \to \{1, \dots, k\}$ such that*

- *there is an LTL-structure $\mathfrak{J} = (w_i)_{i \ge 0}$ such that $\mathfrak{J}, n \models \phi^{\mathsf{p}}_{\mathcal{S}}$ and $w_i = X_{\iota(i)}$, for all i, $0 \le i \le n$, and*

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

- *$\mathcal{S}$ is r-satisfiable w.r.t. ι and $\mathcal{K}$.* $\square$

This result still holds in our setting since every TKB formulated in $\mathcal{EL}$ is also a TKB according to [BBL15], which considers the DL $\mathcal{SHQ}$.

Note that the choice of methods to obtain the set $\mathcal{S}$ and the mapping ι strongly depends on which symbols are allowed to be rigid. In particular, we can obtain $\mathcal{S}$ and the ι by enumeration, guessing, or direct construction, depending on the complexity class we are aiming for. Given $\mathcal{S}$ and ι, we then need to check the two conditions of Lemma 3.2, which basically describe two satisfiability problems: one in LTL and one (or rather several) in $\mathcal{EL}$. In the following, we recall results that provide upper bounds for these two tests.

Lemma 3.3 ([BBL15, Lemma 4.12])**.** *Given a set $\mathcal{S} = \{X_1, \dots, X_k\} \subseteq 2^{\{p_1,\dots,p_m\}}$ and a mapping $\iota : \{0, \dots, n\} \to \{1, \dots, k\}$, the problem of deciding the existence of an LTL structure $\mathfrak{J} = (w_i)_{i\geq 0}$ such that $\mathfrak{J}, n \models \phi^{\mathsf{p}}_{\mathcal{S}}$ 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.* $\square$

The $\mathcal{EL}$ part consists of testing of the r-satisfiability of $\mathcal{S}$. It is especially important whether rigid names are considered or not. In the latter case, the satisfiability of each of the conjunctions χ_i, $1 \leq i \leq k$, and $\chi_{\iota(i)} \wedge \bigwedge_{\alpha \in \mathcal{A}_i} \alpha$, $0 \leq i \leq n$, from Definition 3.1 can be checked separately. Otherwise, each such conjunction has to be regarded in context of the other conjunctions.

To this end, we apply the renaming technique from [BGL12], which introduces copies of the flexible symbols and then regards the conjunction of all relevant conjunctions as an atemporal query. More formally, for $1 \leq i \leq k$ and every flexible concept name $A \in \mathsf{N_C} \setminus \mathsf{N_{RC}}$ (flexible role name $r \in \mathsf{N_R} \setminus \mathsf{N_{RR}}$) that occurs in $\mathcal{T}$ or ϕ, the symbol $A^{(i)}$ ($r^{(i)}$) is introduced and called the i-th copy of A (r). The conjunctive query $\alpha^{(i)}$ (the GCI $\beta^{(i)}$) is then obtained from a CQ α (a GCI β) by replacing every occurrence of a flexible name by its i-th copy. Similarly, for $1 \leq i \leq k$, the conjunction of CQ-literals $\chi_i^{(i)}$ is obtained from χ_i (cf. Definition 3.1) by replacing each CQ α occurring in χ_i by $\alpha^{(i)}$. Finally, we define

$$\begin{aligned}
\chi_{\mathcal{S},\iota} &:= \bigwedge_{1\leq i\leq k} \chi^{(i)} \wedge \bigwedge_{0\leq i\leq n} \left(\chi_{\iota(i)}^{(k+i+1)} \bigwedge_{\alpha\in\mathcal{A}_i} \alpha^{(k+i+1)} \right) \text{ and} \\
\mathcal{T}_{\mathcal{S},\iota} &:= \{\beta^{(i)} \mid \beta \in \mathcal{T},\ 1 \leq i \leq k+n+1\}.
\end{aligned}$$

Note that, for this approach it is essential that the ABoxes do not contain complex concepts since otherwise we could not view the assertions as CQs. We now again refer to a result from [BBL15].

Lemma 3.4 ([BBL15, Lemma 4.14]). *The set $\mathcal{S}$ is r-satisfiable w.r.t. ι and $\mathcal{K}$ iff the conjunction of CQ-literals $\chi_{\mathcal{S},\iota}$ has a model w.r.t. $\mathcal{T}_{\mathcal{S},\iota}$.* □

The next lemma specifies upper bounds for deciding satisfiability of such a conjunction of CQ literals, i.e., for the atemporal case.

Lemma 3.5. *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a knowledge base and ψ be a Boolean conjunction of CQ-literals. Then, the decision whether ψ has a model w.r.t. $\mathcal{K}$ can be reduced to several deterministic polynomial tests w.r.t. combined complexity, the number of which is polynomial in the number of conjuncts of ψ and exponential in the size of the largest negated conjunct in ψ.*

Proof. We first proceed as in [BBL15] and reduce the problem of deciding whether ψ has a model w.r.t. $\mathcal{K}$ to a UCQ non-entailment problem. Let

$$\psi = \rho_1 \wedge \ldots \wedge \rho_\ell \wedge \neg\sigma_1 \wedge \ldots \wedge \neg\sigma_m,$$

where $\rho_1, \ldots, \rho_\ell, \sigma_1, \ldots, \sigma_m$ are Boolean CQs. Now, the positive CQs $\rho_1, \ldots, \rho_\ell$ are instantiated by omitting the existential quantifiers and replacing the variables by fresh individual names. The set $\mathcal{A}'$ of all resulting assertions is then regarded as an additional ABox restricting possible models of ψ. It can be easily seen that ψ is satisfiable w.r.t. $\mathcal{K}$ iff there is an interpretation $\mathcal{I}'$ such that $\mathcal{I}' \models \langle \mathcal{T}, \mathcal{A} \cup \mathcal{A}' \rangle$ and $\mathcal{I}' \models \neg\sigma_1 \wedge \ldots \wedge \neg\sigma_m$.

This is the complement of the entailment problem $\langle \mathcal{T}, \mathcal{A} \cup \mathcal{A}' \rangle \models \sigma_1 \vee \ldots \vee \sigma_m$. In [Ros07], it is proven that this problem is NP-complete w.r.t. combined complexity. The proof is based on the algorithm computeQueryEntailment for deciding UCQ entailment. In particular, it is stated in [Ros07] that the nondeterminism is caused only by the first step of the algorithm; all other steps run in deterministic polynomial time w.r.t. their inputs. This first step (*Unify*) nondeterministically chooses one CQ σ_i, $1 \leq i \leq m$, and one substitution unifying some terms of σ_i. But this means that we can instead consider all (exponentially many) possible unifiers, for each σ_i, $1 \leq i \leq m$, and execute the remaining deterministic steps of the algorithm computeQueryEntailment for each of them in polynomial time. The entailment holds iff one of these runs succeeds. Thus, also the complement problem, satisfiability, can be decided deterministically by applying exponentially many (in the size of the largest negated conjunct in ψ) polynomial tests. □

In particular, this implies that the satisfiability problem for conjunctions of CQ-literals is P-complete w.r.t. data complexity, as it is P-hard already for a single CQ [CDL$^+$06]. We will show in Section 5 that this also holds for TCQs if no rigid names are allowed; however, the complexity jumps to co-NP as soon as rigid concept names are allowed.

4 Regarding Combined Complexity

In this section, we investigate the combined complexity and show that the entailment problem, even w.r.t. rigid concept names, can be solved in PSPACE, which matches the lower bound given by propositional LTL. In a nutshell, this can be done by guessing the rigid concept names satisfied by the named individuals, a certain set of CQs characterizing the set $\mathcal{S}$, and—in a step-wise fashion—$\mathcal{S}$ itself and the mapping ι. Nevertheless, if rigid role names are considered, similar guessing leads to a complexity of in NEXPTIME, and we indeed prove NEXPTIME-completeness for this case.

4.1 The Case With(out) Rigid Concept Names

We first show that in the case that $\mathsf{N_{RR}}$ is empty, the complexity of PSPACE carries over from propositional LTL.

Theorem 4.1. *If $\mathsf{N_{RC}} \neq \emptyset$ but $\mathsf{N_{RR}} = \emptyset$, then TCQ entailment in $\mathcal{EL}$ is* PSPACE-*complete w.r.t. combined complexity.*

PSPACE-hardness follows from the fact that the satisfiability problem of propositional LTL is PSPACE-complete [Pnu77]. The remainder of this section is dedicated to the proof of the matching upper bound.

For ease of presentation, we encode the ABoxes into the query, as proposed in [BBL15]. This is done by rewriting the Boolean TCQ ϕ into a Boolean TCQ ϕ' of polynomial size in the size of ϕ and the TKB $\mathcal{K}$ such that answering ϕ at time point n is equivalent to answering ϕ' at time point 0 w.r.t. the trivial sequence of ABoxes. However, this obviously does not work for data complexity, as the resulting TCQ is no longer independent of the data.

Proposition 4.2 ([BBL15, Lemma 6.1])**.** *Let $\mathcal{K} = \langle \mathcal{T}, (\mathcal{A}_i)_{0\leq i\leq n}\rangle$ be a TKB and ϕ be a Boolean TCQ. Then, there is a Boolean TCQ ψ of size polynomial in the size of ϕ and $\mathcal{K}$ such that $\mathcal{K} \models \phi$ iff $\langle \mathcal{T}, \emptyset\rangle \models \psi$.*

Note that, according to Definition 3.1, we have to ensure that there is a world $X_{\iota(0)}$ that is consistent w.r.t. the knowledge base $\langle \mathcal{T}, \emptyset\rangle$. However, this is true as soon as $\mathcal{S}$ contains any world that is consistent w.r.t. $\mathcal{T}$. Moreover, we always require that $|\mathcal{S}| \geq 1$, and thus this holds whenever $\mathcal{S}$ satisfies the first three requirements of Definition 3.1. This means that we do not have to guess a mapping $\iota\colon \{0\} \to \{1, \dots, k\}$ in the following.

Let now ϕ be a TCQ and $\mathcal{K} = \langle \mathcal{T}, \emptyset\rangle$ be a TKB. Note that in this section we have to drop the assumption that all individual names in the query ϕ also occur in the ABoxes; in fact, ϕ is now the only place where individual names may occur.

We assume without loss of generality that the CQs occurring in ϕ use disjoint variables[4] and denote by $\mathcal{Q}_\phi$ the set of exactly those CQs. We further assume that all concepts of the form $\mathsf{Con}(\alpha)$, for all tree-shaped CQs $\alpha \in \mathcal{Q}_\phi$, also occur in $\mathcal{T}$.

For now, we assume that a set $\mathcal{S} = \{X_1, \dots, X_k\} \subseteq 2^{\{p_1,\dots,p_m\}}$ is given; in the proof of Lemma 4.14, we describe how to actually obtain $\mathcal{S}$ within PSPACE. For all i, $1 \leq i \leq k$, we denote by Q_i the set $\{\alpha_j \mid p_j \in X_i\}$, and by $\mathcal{A}_{Q_i}$ the ABox obtained from Q_i by instantiating all variables x with fresh individual names a_x. We collect all these new individual names in the set $\mathsf{N}_\mathsf{I}^{\mathsf{aux}}$.

To check the conditions of Lemma 3.2, we guess polynomially many additional assertions and queries that allow us to separate the r-satisfiability test for $\mathcal{S}$ into independent consistency tests for the individual time points. In the following, we use sets $\mathcal{B} = \{B_1, \dots, B_\ell\} \subseteq \mathsf{N}_{\mathsf{RC}}(\mathcal{T})$ as witnesses for the satisfaction of tree-shaped CQs. In an abuse of notation, we denote by $\mathcal{B}$ also the associated concept $B_1 \sqcap \dots \sqcap B_\ell$, and write $\mathcal{B}(x)$ for the conjunction $B_1(x) \wedge \dots \wedge B_\ell(x)$.

Definition 4.3. *A set $\mathcal{B} \subseteq \mathsf{N}_{\mathsf{RC}}(\mathcal{T})$ is a* witness *of a concept C w.r.t. $\mathcal{T}$ if there are $r_1, \dots, r_\ell \in \mathsf{N}_\mathsf{R}$, $\ell \geq 0$, such that $\mathcal{T} \models \mathcal{B} \sqsubseteq \exists r_1 \dots r_\ell . C$. Furthermore, $\mathcal{B}$ is a* witness *of a tree-shaped CQ α w.r.t. $\mathcal{T}$ if it is a witness of $\mathsf{Con}(\alpha)$ w.r.t. $\mathcal{T}$.*

It should be clear from these definitions that, if a model of $\mathcal{T}$ contains an element that satisfies a witness for α, then this model satisfies α.

Lemma 4.4. *Let $\mathcal{I}$ be a model of $\mathcal{T}$ and $\mathcal{B}$ be a witness of a tree-shaped CQ α w.r.t. $\mathcal{T}$. Then, $\mathcal{I} \models \exists x . \mathcal{B}(x)$ implies that $\mathcal{I} \models \alpha$.*

We will use witnesses to fully characterize the satisfaction of the CQs in $\mathcal{Q}_\phi$ in the anonymous part of an interpretation. We now describe a property that has to be fulfilled by the polynomially many additional assertions and queries which we guess.

Definition 4.5. *An* ABox type *for $\mathcal{K}$ is a set*

$$\mathcal{A}_\mathsf{R} \subseteq \{A(a), \neg A(a) \mid a \in \mathsf{N}_\mathsf{I}(\phi),\ A \in \mathsf{N}_{\mathsf{RC}}(\mathcal{T})\}$$

with the property that $A(a) \in \mathcal{A}_\mathsf{R}$ iff $\neg A(a) \notin \mathcal{A}_\mathsf{R}$. Given an ABox type $\mathcal{A}_\mathsf{R}$, for all i, $1 \leq i \leq k$, we define $\mathcal{K}_\mathsf{R}^i := \langle \mathcal{T}, \mathcal{A}_\mathsf{R} \cup \mathcal{A}_{Q_i} \rangle$.

A tuple $(\mathcal{A}_\mathsf{R}, Q_\mathsf{R}^\neg)$ consisting of an ABox type $\mathcal{A}_\mathsf{R}$ for $\mathcal{K}$ and a set $Q_\mathsf{R}^\neg \subseteq \mathcal{Q}_\phi$ is called r-complete *(w.r.t. $\mathcal{S}$) if the following hold:*

(R1) *For all $i \in \{1, \dots, k\}$, $\mathcal{K}_\mathsf{R}^i$ is consistent.*

(R2) *For all $i \in \{1, \dots, k\}$ and $p_j \in \overline{X_i}$, we have $\mathcal{K}_\mathsf{R}^i \not\models \alpha_j$.*

[4] If this was not the case, we could simply rename them.

(R3) *For all* $i \in \{1, \dots, k\}$*, all tree-shaped CQs* $\alpha \in Q_{\mathsf{R}}^{\neg}$*, and all witnesses* $\mathcal{B}$ *of* α *w.r.t.* $\mathcal{T}$*, we have* $\mathcal{K}_{\mathsf{R}}^i \not\models \exists x.\mathcal{B}(x)$*.*

(R4) *For all* $\alpha_j \in \mathcal{Q}_\phi \setminus Q_{\mathsf{R}}^{\neg}$*, we have* $p_j \in \bigcap \mathcal{S}$*.*

The idea is to fix the interpretation of the rigid names on all named individuals ($\mathcal{A}_{\mathsf{R}}$) and specify a set of CQs that are allowed to occur negatively in $\mathcal{S}$ ($Q_{\mathsf{R}}^{\neg}$). The first two conditions ensure that, for all considered worlds X_i, $1 \leq i \leq k$, exactly the queries specified by X_i can be satisfied w.r.t. $\mathcal{T}$, together with the assertions from $\mathcal{A}_{\mathsf{R}}$. The third condition ensures that the canonical model of $\mathcal{K}_{\mathsf{R}}^i$ does not satisfy any of the witnesses of the tree-shaped queries in $Q_{\mathsf{R}}^{\neg}$ (cf. Proposition 2.11). Finally, the last condition checks that only the queries from $Q_{\mathsf{R}}^{\neg}$ can occur negatively in any $X \in \mathcal{S}$.

In the main part of this section we show that the existence of an r-complete tuple w.r.t. $\mathcal{S}$ fully characterizes the r-satisfiability of $\mathcal{S}$.

Lemma 4.6. $\mathcal{S}$ *is r-satisfiable iff there is an r-complete tuple w.r.t.* $\mathcal{S}$*.*

The proof of this lemma is split over the following two subsections. The last subsection then describes how this lemma can be used to decide the entailment problem using only polynomial space.

4.1.1 If $\mathcal{S}$ is r-satisfiable, then there is an r-complete tuple w.r.t. $\mathcal{S}$.

Let $\mathcal{J}_1, \dots, \mathcal{J}_k$ be the interpretations over the domain Δ that exist according to the r-satisfiability of $\mathcal{S}$ (cf. Definition 3.1). We define the tuple $(\mathcal{A}_{\mathsf{R}}, Q_{\mathsf{R}}^{\neg})$ as follows:

$$
\begin{aligned}
\mathcal{A}_{\mathsf{R}} := {} & \{A(a) \mid a \in \mathsf{N_I}(\phi), A \in \mathsf{N_{RC}}(\mathcal{T}), a^{\mathcal{J}_1} \in A^{\mathcal{J}_1}\} \cup \\
& \{\neg A(a) \mid a \in \mathsf{N_I}(\phi), A \in \mathsf{N_{RC}}(\mathcal{T}), a^{\mathcal{J}_1} \notin A^{\mathcal{J}_1}\}; \\
Q_{\mathsf{R}}^{\neg} := {} & \{\alpha_j \in \mathcal{Q}_\phi \mid p_j \notin \bigcap \mathcal{S}\}.
\end{aligned}
$$

Obviously, $\mathcal{A}_{\mathsf{R}}$ is an ABox type for $\mathcal{K}$, and $Q_{\mathsf{R}}^{\neg}$ satisfies Condition (R4). Furthermore, it is easy to verify that each $\mathcal{J}_i$, $1 \leq i \leq k$, can be extended to a model $\mathcal{J}_i'$ of $\mathcal{K}_{\mathsf{R}}^i$ by appropriately defining the interpretations of the new individual names a_x that are introduced by $\mathcal{A}_{Q_i}$. Thus, Condition (R1) is also satisfied.

Regarding Condition (R2), assume that there are i, $1 \leq i \leq k$, and $p_j \in \overline{X_i}$ such that $\mathcal{K}_{\mathsf{R}}^i \models \alpha_j$, and thus $\mathcal{J}_i' \models \alpha_j$. This means that also $\mathcal{J}_i \models \alpha_j$ since α_j does not contain any of the new individual names. But this contradicts the assumption that $\mathcal{J}_i \models \chi_i$.

The proof of Condition (R3) is also by contradiction. We assume that there are i, $1 \leq i \leq k$, a tree-shaped CQ $\alpha_j \in Q_{\mathsf{R}}^{\neg}$, and a witness $\mathcal{B}$ of α_j such that

$\mathcal{K}^i_{\mathsf{R}} \models \exists x.\mathcal{B}(x)$, and thus $\mathcal{J}_i \models \exists x.\mathcal{B}(x)$ as above. However, by the definition of Q^{-}_{R}, there must be an i', $1 \leq i' \leq k$, such that $p_j \notin X_{i'}$, and thus $\mathcal{J}_{i'} \not\models \alpha_j$. Lemma 4.4 then yields that $\mathcal{J}_{i'} \not\models \exists x.\mathcal{B}(x)$, which contradicts the facts that $\mathcal{B} \subseteq \mathsf{N_{RC}}(\mathcal{T})$ and $\mathcal{J}_i$ and $\mathcal{J}_{i'}$ respect the rigid names. □

4.1.2 If there is an r-complete tuple w.r.t. $\mathcal{S}$, then $\mathcal{S}$ is r-satisfiable.

The proof of the converse direction is more involved. For each i, $1 \leq i \leq k$, we consider the canonical interpretation $\mathcal{I}_i := \mathcal{I}_{[\mathcal{K}^i_{\mathsf{R}}]^+}$, where $[\mathcal{K}^i_{\mathsf{R}}]^+$ is equal to $\mathcal{K}^i_{\mathsf{R}}$ without the negated assertions in $\mathcal{A}_{\mathsf{R}}$. Since $\mathcal{K}^i_{\mathsf{R}}$ is consistent by Condition (R1), we know that $[\mathcal{K}^i_{\mathsf{R}}]^+ \not\models A(a)$ for any negated assertion $\neg A(a) \in \mathcal{A}_{\mathsf{R}}$. By Proposition 2.11, it follows that $\mathcal{I}_i \models \neg A(a)$, and hence $\mathcal{I}_i$ is a model of $\mathcal{K}^i_{\mathsf{R}}$.

To distinguish the elements contained in $\mathsf{N_I^{aux}}$, we define $\Delta^{\mathcal{I}_i}_{\mathsf{a}} := \mathsf{N_I^{aux}} \cap \Delta^{\mathcal{I}_i}$, and write a^i_x instead of a_x for the elements of this set. We further write $\Delta^{\mathcal{I}_i}_{\mathsf{u}}$ for the set containing the unnamed domain elements unique to the canonical interpretation $\mathcal{I}_i$, and similarly write $c^i_{\varrho rD}$ for every element $c_{\varrho rD} \in \Delta^{\mathcal{I}_i}_{\mathsf{u}}$. Thus, the domain of each $\mathcal{I}_i$ is composed of the pairwise disjoint components $\mathsf{N_I}(\phi)$, $\Delta^{\mathcal{I}_i}_{\mathsf{a}}$, and $\Delta^{\mathcal{I}_i}_{\mathsf{u}}$. We next state that as fact for future reference.

Fact 4.7. *For all $i, j \in \{1, \ldots, k\}$, the sets $\mathsf{N_I}(\phi)$, $\Delta^{\mathcal{I}_j}_{\mathsf{a}}$, and $\Delta^{\mathcal{I}_i}_{\mathsf{u}}$ are pairwise disjoint.*

In our construction, we make use of the subset $\Delta^{\mathcal{I}_i}_{\mathsf{uR}} := \bigcup_{j=0}^{\infty} \Delta^{i,j}_{\mathsf{uR}}$ of $\Delta^{\mathcal{I}_i}_{\mathsf{u}}$, which is inductively defined as follows:

$$\begin{aligned}
\Delta^{i,0}_{\mathsf{uR}} &:= \{c^i_{\varrho rD} \mid \mathcal{B} \subseteq \mathsf{N_{RC}}(\mathcal{T}), c^i_{\varrho} \in \mathcal{B}^{\mathcal{I}_i} \cap \Delta^{\mathcal{I}_i}_{\mathsf{u}}, D \in \mathsf{Sub}(\mathcal{T}), \mathcal{T} \models \mathcal{B} \sqsubseteq \exists r.D\} \cup \\
&\qquad \{c^i_{a^i_x rD} \mid \mathcal{B} \subseteq \mathsf{N_{RC}}(\mathcal{T}), a^i_x \in \mathcal{B}^{\mathcal{I}_i} \cap \Delta^{\mathcal{I}_i}_{\mathsf{a}}, D \in \mathsf{Sub}(\mathcal{T}), \mathcal{T} \models \mathcal{B} \sqsubseteq \exists r.D\} \\
\Delta^{i,j+1}_{\mathsf{uR}} &:= \{c^i_{\varrho rDsE} \mid c^i_{\varrho rD} \in \Delta^{i,j}_{\mathsf{uR}}, E \in \mathsf{Sub}(\mathcal{T}), \mathcal{T} \models D \sqsubseteq \exists s.E\}.
\end{aligned}$$

This definition is similar to that of $\Delta^{\mathcal{I}_i}_{\mathsf{u}}$ (cf. Definition 2.10), the only difference being that we here only consider those elements whose existence is enforced by some combination of rigid concept names at an already unnamed domain element. Thus, there are no direct role connections between elements of $\mathsf{N_I}(\phi)$ and $\Delta^{\mathcal{I}_i}_{\mathsf{uR}}$.

Fact 4.8. *For all i, $1 \leq i \leq k$, we have $\Delta^{\mathcal{I}_i}_{\mathsf{uR}} \subseteq \Delta^{\mathcal{I}_i}_{\mathsf{u}}$.*

We now construct the interpretations $\mathcal{J}_1, \ldots, \mathcal{J}_k$ as required for the r-satisfiability of $\mathcal{S}$, that is, they share the same domain and respect rigid names, and each $\mathcal{J}_i$ is a model of $\mathcal{T}$ and $\chi_i = \bigwedge_{p_j \in X_i} \alpha_j \wedge \bigwedge_{p_j \in \overline{X_i}} \neg\alpha_j$. Recall that we then do not need to specifically define an interpretation for time point 0, since any $\mathcal{J}_{\iota(0)}$ will be a model of $\mathcal{A}_0 = \emptyset$ and $\chi_{\iota(0)}$. To obtain interpretations $\mathcal{J}_1, \ldots, \mathcal{J}_k$ as required, we join the domains of the interpretations $\mathcal{I}_i$ and ensure that they interpret all rigid concept names in the same way. We first construct the common domain

$$\Delta := \mathsf{N_I}(\phi) \cup \bigcup_{i=1}^{k} (\Delta_{\mathsf{a}}^{\mathcal{I}_i} \cup \Delta_{\mathsf{u}}^{\mathcal{I}_i})$$

and then define the interpretations $\mathcal{J}_i$, $1 \leq i \leq k$, as follows:

- For all $a \in \mathsf{N_I}(\phi)$, we set $a^{\mathcal{J}_i} := a$.
- For all rigid concept names A, we define $A^{\mathcal{J}_i} := \bigcup_{j=1}^{k} A^{\mathcal{I}_j}$.
- For all flexible concept names A, we define

$$A^{\mathcal{J}_i} := A^{\mathcal{I}_i} \cup \bigcup_{j=1}^{k} \bigcup_{\substack{\mathcal{B} \subseteq \mathsf{N_{RC}}(\mathcal{T}), \\ \mathcal{T} \models \mathcal{B} \sqsubseteq A}} \mathcal{B}^{\mathcal{I}_j} \cup \bigcup_{j=1}^{k} \{c_{\varrho r D}^{j} \in \Delta_{\mathsf{uR}}^{\mathcal{I}_j} \mid \mathcal{T} \models D \sqsubseteq A\}.$$

- For all (flexible) role names r, we define

$$r^{\mathcal{J}_i} := r^{\mathcal{I}_i} \cup \bigcup_{j=1}^{k} \Big(\{(c_{\varrho}^{j}, c_{\varrho r D}^{j}) \in \Delta_{\mathsf{u}}^{\mathcal{I}_j} \times \Delta_{\mathsf{uR}}^{\mathcal{I}_j}\} \cup \{(a_x^j, c_{a_x^j r D}^{j}) \in \Delta_{\mathsf{a}}^{\mathcal{I}_j} \times \Delta_{\mathsf{uR}}^{\mathcal{I}_j}\} \Big).$$

In this way, we have constructed interpretations $\mathcal{J}_1, \dots, \mathcal{J}_k$ that have the same domain and respect the rigid concept names since, for all $A \in \mathsf{N_{RC}}$, the definition of $A^{\mathcal{J}_i}$ is independent of i. It remains to show that they satisfy the other requirements for the r-satisfiability of $\mathcal{S}$ as described above.

We start by showing some facts about the elements in the sets $\Delta_{\mathsf{uR}}^{\mathcal{I}_j}$.

Lemma 4.9. *For all $i, j \in \{1, \dots, k\}$ and $c_{\varrho r D}^{j} \in \Delta_{\mathsf{uR}}^{\mathcal{I}_j}$, the following hold:*

a) *For all concepts $C \in \mathsf{Sub}(\mathcal{T})$, we have $c_{\varrho r D}^{j} \in C^{\mathcal{J}_i}$ iff $\mathcal{T} \models D \sqsubseteq C$.*

b) *There is a witness $\mathcal{B}$ of $\exists r.D$ w.r.t. $\mathcal{T}$ such that $\mathcal{B}^{\mathcal{I}_j}$ is non-empty.*

Proof. We begin with the proof of a), for which we use induction on the shape of C. For the induction start, let $c_{\varrho r D}^{j} \in \Delta_{\mathsf{uR}}^{\mathcal{I}_j}$ and $C = A \in \mathsf{N_C}(\mathcal{T})$. We consider $(\Rightarrow)$ and the definition of $\mathcal{J}_i$. If $c_{\varrho r D}^{j} \in A^{\mathcal{I}_j}$, then we immediately have $\mathcal{T} \models D \sqsubseteq A$ by Definition 2.10. If $c_{\varrho r D}^{j} \in \mathcal{B}^{\mathcal{I}_j}$ for some $\mathcal{B} \subseteq \mathsf{N_{RC}}(\mathcal{T})$ with $\mathcal{T} \models \mathcal{B} \sqsubseteq A$, then we also get $c_{\varrho r D}^{j} \in A^{\mathcal{I}_j}$ since $\mathcal{I}_j \models \mathcal{T}$, and thus $\mathcal{T} \models D \sqsubseteq A$ as above. The other direction, $(\Leftarrow)$, immediately follows from Definition 2.10 and the definition of $\mathcal{J}_i$.

The claim for $C = \top$ holds because of the interpretation of $\top$.

For $C = C_1 \sqcap C_2$, we have $c_{\varrho r D}^{j} \in C_1^{\mathcal{J}_i} \cap C_2^{\mathcal{J}_i}$ iff $\mathcal{T} \models D \sqsubseteq C_1$ and $\mathcal{T} \models D \sqsubseteq C_2$ by the induction hypothesis. This is equivalent to $\mathcal{T} \models D \sqsubseteq C_1 \sqcap C_2$.

Let now $C = \exists s.C_1$. If $c^j_{\varrho r D} \in (\exists s.C_1)^{\mathcal{J}_i}$, then there exists $c^j_{\varrho r D s E} \in \Delta^{\mathcal{I}_j}_{\mathsf{uR}}$ with $(c^j_{\varrho r D}, c^j_{\varrho r D s E}) \in s^{\mathcal{J}_i}$ and $c^j_{\varrho r D s E} \in C_1^{\mathcal{J}_i}$. By the induction hypothesis, we get $\mathcal{T} \models E \sqsubseteq C_1$. Moreover, we have $\mathcal{T} \models D \sqsubseteq \exists s.E$ by Definition 2.10, which implies that $\mathcal{T} \models D \sqsubseteq \exists s.C_1$. On the other hand, if $\mathcal{T} \models D \sqsubseteq \exists s.C_1$, then we have $c^j_{\varrho r D s C_1} \in \Delta^{\mathcal{I}_j}_{\mathsf{uR}}$. Since $\mathcal{T} \models C_1 \sqsubseteq C_1$, by the induction hypothesis we get $c^j_{\varrho r D s C_1} \in C_1^{\mathcal{J}_i}$. By the definition of $\mathcal{J}_i$, we also have $(c^j_{\varrho r D}, c^j_{\varrho r D s C_1}) \in s^{\mathcal{J}_i}$, and thus $c^j_{\varrho r D} \in (\exists s.C_1)^{\mathcal{J}_i}$.

For the proof of b), we proceed by induction on the construction of $\Delta^{\mathcal{I}_j}_{\mathsf{uR}}$. For elements $c^j_{\varrho r D} \in \Delta^{j,0}_{\mathsf{uR}}$, the definition of $\Delta^{j,0}_{\mathsf{uR}}$ directly yields the claim.

For the induction step, we consider $c^j_{\varrho s E r D} \in \Delta^{j,l+1}_{\mathsf{uR}}$, $l \geq 0$, i.e., we have $c^j_{\varrho s E} \in \Delta^{j,l}_{\mathsf{uR}}$ and $\mathcal{T} \models E \sqsubseteq \exists r.D$. By the induction hypotheses, there are $r_1, \ldots, r_\ell \in \mathsf{N_R}$, $\ell \geq 0$, and a set $\mathcal{B} \subseteq \mathsf{N_{RC}}(\mathcal{T})$ such that $\mathcal{T} \models \mathcal{B} \sqsubseteq \exists r_1 \ldots r_\ell s.E$ and $\mathcal{B}^{\mathcal{I}_j}$ is non-empty. This implies that $\mathcal{T} \models \mathcal{B} \sqsubseteq \exists r_1 \ldots r_\ell s r.D$, which concludes the proof. □

We now state a basic connection between the interpretations $\mathcal{J}_i$ and $\mathcal{I}_i$ concerning the interpretation of role names.

Lemma 4.10. *For all $i \in \{1, \ldots, k\}$, role names $r \in \mathsf{N_R}$, and $d, e \in \Delta^{\mathcal{I}_i}$, we have $(d, e) \in r^{\mathcal{J}_i}$ iff $(d, e) \in r^{\mathcal{I}_i}$.*

Proof. The "if"-direction follows directly from the definition of $r^{\mathcal{J}_i}$. For the "only if"-direction, Facts 4.7 and 4.8 and the definition of $\mathcal{J}_i$ either directly yield $(d, e) \in r^{\mathcal{I}_i}$, or $e \in \Delta^{\mathcal{I}_i}_{\mathsf{uR}}$ is of the form $c^j_{\varrho r D}$ and either $d = c^j_{\varrho}$ or $d = a^j_x = \varrho$. By Definition 2.10, the latter two options also imply that $(d, e) \in r^{\mathcal{I}_i}$. □

There is a similar connection between the interpretations of concepts in $\mathcal{I}_j$ and $\mathcal{J}_i$.

Lemma 4.11. *For all $i, j \in \{1, \ldots, k\}$ and all concepts $C \in \mathsf{Sub}(\mathcal{T})$, the following hold:*

a) *For all $e \in \mathsf{N_I}(\phi)$, we have $e \in C^{\mathcal{J}_i}$ iff $e \in C^{\mathcal{I}_i}$.*

b) *For all $e \in \Delta^{\mathcal{I}_j}_{\mathsf{a}} \cup (\Delta^{\mathcal{I}_j}_{\mathsf{u}} \setminus \Delta^{\mathcal{I}_j}_{\mathsf{uR}})$, we have $e \in C^{\mathcal{J}_i}$ iff*

   - *$i = j$ and $e \in C^{\mathcal{I}_i}$, or*
   - *there is a $\mathcal{B} \subseteq \mathsf{N_{RC}}(\mathcal{T})$ such that $e \in \mathcal{B}^{\mathcal{I}_j}$ and $\mathcal{T} \models \mathcal{B} \sqsubseteq C$.*

c) *For all $e \in \Delta^{\mathcal{I}_j}_{\mathsf{uR}}$, we have $e \in C^{\mathcal{J}_i}$ iff $e \in C^{\mathcal{I}_j}$.*

Proof. Item c) is a direct consequence of Lemmata 2.12, and 4.9a) and Fact 4.8. We now prove the other two items simultaneously by induction on the structure of C.

For the induction start, we begin with a). For rigid concept names, it follows from the fact that each $\mathcal{I}_j$, $1 \leq j \leq k$, is a model of $\mathcal{A}_\mathsf{R}$ and from the definition of $\mathcal{J}_i$. For flexible concept names C, by Fact 4.7 we have $e \in C^{\mathcal{J}_i}$ iff $e \in C^{\mathcal{I}_i}$ or $e \in \mathcal{B}^{\mathcal{I}_j}$ for some j, $1 \leq j \leq k$ and $\mathcal{B} \subseteq \mathsf{N}_{\mathsf{RC}}(\mathcal{T})$ with $\mathcal{T} \models \mathcal{B} \sqsubseteq C$. Since both $\mathcal{I}_i$ and $\mathcal{I}_j$ are models of $\mathcal{A}_\mathsf{R}$, in the latter case we also have $e \in \mathcal{B}^{\mathcal{I}_i}$. Since $\mathcal{I}_i \models \mathcal{T}$, this implies that $e \in C^{\mathcal{I}_i}$.

We consider b). Because of Fact 4.7, for $C \in \mathsf{N}_{\mathsf{RC}}$, the definition of $\mathcal{J}_i$ directly yields $e \in C^{\mathcal{J}_i}$ iff $e \in C^{\mathcal{I}_j}$. For $C \in \mathsf{N_C} \setminus \mathsf{N}_{\mathsf{RC}}$, we obtain the claim from Fact 4.7, the definition of $\mathcal{J}_i$, and the fact that all $\mathcal{I}_j$ are models of $\mathcal{T}$.

The claim for $C = \top$ is again trivial by the interpretation of $\top$.

We consider $C = C_1 \sqcap C_2$ for a) and b) under the assumption that $i = j$. The induction hypothesis directly yields the equivalence between $e \in C_1^{\mathcal{J}_i} \cap C_2^{\mathcal{J}_i}$ and $e \in C_1^{\mathcal{I}_i} \cap C_2^{\mathcal{I}_i}$. Moreover, any $\mathcal{B} \subseteq \mathsf{N}_{\mathsf{RC}}(\mathcal{T})$ with $e \in \mathcal{B}^{\mathcal{I}_i}$ and $\mathcal{T} \models \mathcal{B} \sqsubseteq C_1 \sqcap C_2$ also yields that $e \in (C_1 \sqcap C_2)^{\mathcal{I}_i}$, and thus $e \in (C_1 \sqcap C_2)^{\mathcal{J}_i}$ as above.

For case b) with $i \neq j$, by the induction hypothesis $e \in (C_1 \sqcap C_2)^{\mathcal{J}_i}$ implies that there are $\mathcal{B}_1, \mathcal{B}_2 \subseteq \mathsf{N}_{\mathsf{RC}}(\mathcal{T})$ with $e \in (\mathcal{B}_1 \sqcap \mathcal{B}_2)^{\mathcal{I}_j}$, $\mathcal{T} \models \mathcal{B}_1 \sqsubseteq C_1$, and $\mathcal{T} \models \mathcal{B}_2 \sqsubseteq C_2$. But then it also holds that $\mathcal{T} \models \mathcal{B}_1 \sqcap \mathcal{B}_2 \sqsubseteq C_1 \sqcap C_2$, and thus $e \in (C_1 \sqcap C_2)^{\mathcal{I}_j}$. On the other hand, if $e \in \mathcal{B}^{\mathcal{I}_j}$ and $\mathcal{T} \models \mathcal{B} \sqsubseteq C_1 \sqcap C_2$ for some $\mathcal{B} \subseteq \mathsf{N}_{\mathsf{RC}}(\mathcal{T})$, then also $\mathcal{T} \models \mathcal{B} \sqsubseteq C_1$ and $\mathcal{T} \models \mathcal{B} \sqsubseteq C_2$. Together with the induction hypothesis, this leads to $e \in (C_1 \sqcap C_2)^{\mathcal{J}_i}$.

Finally, we consider the case of an existential restriction $C = \exists r.C_1$. For a) and b) with $i = j$, by the definition of $\mathcal{J}_i$, Lemma 4.10, Item c), and the induction hypothesis the existence of a $d \in (C_1)^{\mathcal{J}_i}$ with $(e, d) \in r^{\mathcal{J}_i}$ is equivalent to the existence of a $d \in (C_1)^{\mathcal{I}_i}$ with $(e, d) \in r^{\mathcal{I}_i}$. As before, the second option of b) is subsumed by the first one, in this case.

For case b) with $i \neq j$, $e \in (\exists r.C_1)^{\mathcal{J}_i}$ implies that there is a $c^j_{\varrho r D} \in (C_1)^{\mathcal{J}_i} \cap \Delta_{\mathsf{uR}}^{\mathcal{I}_j}$ such that either $e = c^j_\varrho$ or $e = a^j_x = \varrho$. Since $e \notin \Delta_{\mathsf{uR}}^{\mathcal{I}_j}$, in both case we must have $c^j_{\varrho r D} \in \Delta_{\mathsf{uR}}^{j,0}$, and thus there exists a $\mathcal{B} \subseteq \mathsf{N}_{\mathsf{RC}}(\mathcal{T})$ such that $e \in \mathcal{B}^{\mathcal{I}_j}$ and $\mathcal{T} \models \mathcal{B} \sqsubseteq \exists r.D$. Furthermore, the fact that $c^j_{\varrho r D} \in (C_1)^{\mathcal{J}_i}$ implies that $\mathcal{T} \models D \sqsubseteq C_1$ by Lemma 4.9, and thus $\mathcal{T} \models \mathcal{B} \sqsubseteq \exists r.C_1$. On the other hand, if there exists a $\mathcal{B} \subseteq \mathsf{N}_{\mathsf{RC}}(\mathcal{T})$ with $e \in \mathcal{B}^{\mathcal{I}_j}$ and $\mathcal{T} \models \mathcal{B} \sqsubseteq \exists r.C_1$, then by the definition of $\Delta_{\mathsf{uR}}^{j,0}$ we have $c^j_{\varrho r C_1} \in \Delta^j_{\mathsf{uR}}$, where again either $e = c^j_\varrho$ or $e = a^j_x = \varrho$. Thus, in particular it holds that $(e, c^j_{\varrho r C_1}) \in r^{\mathcal{J}_i}$. Since $\mathcal{T} \models C_1 \sqsubseteq C_1$, by Lemma 4.9 we have $c^j_{\varrho r C_1} \in (C_1)^{\mathcal{J}_i}$, and thus $e \in (\exists r.C_1)^{\mathcal{J}_i}$, as required. □

We finally show that $\mathcal{J}_i$ is in fact as intended.

Lemma 4.12. *Each $\mathcal{J}_i$, $1 \leq i \leq k$, is a model of $\mathcal{T}$.*

Proof. Consider a GCI $C \sqsubseteq D \in \mathcal{T}$ and an element $d \in C^{\mathcal{J}_i}$. If $d \in \mathsf{N_I}(\phi) \cup \Delta_{\mathsf{uR}}^{\mathcal{I}_j}$ for some j, $1 \leq j \leq k$, we get $d \in C^{\mathcal{I}_j}$ by Lemma 4.11. Since $\mathcal{I}_j \models \mathcal{T}$, we obtain

$d \in D^{\mathcal{I}_j}$, and thus $d \in D^{\mathcal{J}_i}$ again by Lemma 4.11. A similar argument applies in case that $d \in \Delta_{\mathsf{a}}^{\mathcal{I}_i} \cup (\Delta_{\mathsf{u}}^{\mathcal{I}_i} \setminus \Delta_{\mathsf{uR}}^{\mathcal{I}_i})$.

For $d \in \Delta_{\mathsf{a}}^{\mathcal{I}_j} \cup (\Delta_{\mathsf{u}}^{\mathcal{I}_j} \setminus \Delta_{\mathsf{uR}}^{\mathcal{I}_j})$ with $i \neq j$, by Lemma 4.11b) we know that there is a set $\mathcal{B} \subseteq \mathsf{N}_{\mathsf{RC}}(\mathcal{T})$ such that $d \in \mathcal{B}^{\mathcal{I}_j}$ and $\mathcal{T} \models \mathcal{B} \sqsubseteq C$. But then we also have $\mathcal{T} \models \mathcal{B} \sqsubseteq D$, which leads to $d \in D^{\mathcal{J}_i}$ by another application of Lemma 4.11. $\square$

We now provide the final missing piece to show r-satisfiability of $\mathcal{S}$.

Lemma 4.13. *Each $\mathcal{J}_i$, $1 \leq i \leq k$, is a model of χ_i.*

Proof. Consider first any CQ α that occurs positively in the conjunction χ_i. Since $\mathcal{I}_i \models \mathcal{A}_{Q_i}$ and $\mathcal{A}_{Q_i}$ contains an instantiation of α, we know that there is a homomorphism π of α into $\mathcal{I}_i$ that maps all variables to elements in $\Delta_{\mathsf{a}}^{\mathcal{I}_i}$. By Lemmata 4.10 and 4.11b), we know that π is also a homomorphism of α into $\mathcal{J}_i$.

We now consider a CQ α that occurs negatively in χ_i. By Condition (R2), we know that $[\mathcal{K}_{\mathsf{R}}^i]^+ \not\models \alpha$, and thus $\mathcal{I}_i \models \neg\alpha$ by Proposition 2.11. We now assume to the contrary that there is a homomorphism π of α into $\mathcal{J}_i$. Since α is connected and domain elements $d, e \in \Delta$ can only be connected by $r^{\mathcal{J}_i}$ if they belong to the same domain $\Delta^{\mathcal{I}_j}$ (cf. Fact 4.7), we can assume that there is an index j, $1 \leq j \leq k$, such that π maps all terms of α into $\Delta^{\mathcal{I}_j}$.

Assume first that π maps all terms into $\Delta^{\mathcal{I}_i}$, which in particular includes $\mathsf{N}_{\mathsf{I}}(\phi)$. Then by Lemmata 4.10 and 4.11, π is also a homomorphism of α into $\mathcal{I}_i$, which contradicts the fact that $\mathcal{I}_i \models \neg\alpha$.

Otherwise, we have $j \neq i$ and π maps at least one term into $\Delta^{\mathcal{I}_j} \setminus \mathsf{N}_{\mathsf{I}}(\phi)$. By the interpretation of roles in $\mathcal{J}_i$ and since α is connected, this means that no term of α can be mapped into $\mathsf{N}_{\mathsf{I}}(\phi)$, (i.e., α contains no individual names and π maps all variables into $\Delta_{\mathsf{a}}^{\mathcal{I}_j} \cup \Delta_{\mathsf{u}}^{\mathcal{I}_j}$). Furthermore, for all role atoms $r(y, z) \in \mathsf{At}(\alpha)$, we either have (i) $\pi(y) \in \Delta_{\mathsf{a}}^{\mathcal{I}_j} \cup (\Delta_{\mathsf{u}}^{\mathcal{I}_j} \setminus \Delta_{\mathsf{uR}}^{\mathcal{I}_j})$ and $\pi(z) \in \Delta_{\mathsf{uR}}^{\mathcal{I}_j}$, or (ii) $\pi(y), \pi(z) \in \Delta_{\mathsf{uR}}^{\mathcal{I}_j}$. Since α is connected, there is at most one variable in α that is mapped into $\Delta_{\mathsf{a}}^{\mathcal{I}_j} \cup (\Delta_{\mathsf{u}}^{\mathcal{I}_j} \setminus \Delta_{\mathsf{uR}}^{\mathcal{I}_j})$ by π. Thus, in α, there are only role connections starting from this variable and role connections between other variables (mapped into $\Delta_{\mathsf{uR}}^{\mathcal{I}_j}$) via a single role and in one direction. This means that α is tree-shaped.

We now show that there is a witness $\mathcal{B}$ of α w.r.t. $\mathcal{T}$ such that $\mathcal{B}^{\mathcal{I}_j}$ is non-empty. For this, let x be the root variable of α. By our assumption that π is a homomorphism of α into $\mathcal{J}_i$, we know that $\pi(x) \in (\mathsf{Con}(\alpha))^{\mathcal{J}_i}$.

- If $\pi(x)$ is contained in $\Delta_{\mathsf{a}}^{\mathcal{I}_j} \cup (\Delta_{\mathsf{u}}^{\mathcal{I}_j} \setminus \Delta_{\mathsf{uR}}^{\mathcal{I}_j})$, then Lemma 4.11b) yields that there is a witness $\mathcal{B}$ of α w.r.t. $\mathcal{T}$ such that $\pi(x) \in \mathcal{B}^{\mathcal{I}_j}$.[5]

[5] Recall that we assumed that $\mathsf{Con}(\alpha)$ occurs in $\mathcal{T}$.

- If $\pi(x)$ is of the form $c^j_{\varrho r D} \in \Delta^{\mathcal{I}_j}_{\mathsf{uR}}$, then by Lemma 4.9b) there is a witness $\mathcal{B}$ of $\exists r.D$ such that $\mathcal{B}^{\mathcal{I}_j}$ is non-empty. Since $c^j_{\varrho r D} \in (\mathsf{Con}(\alpha))^{\mathcal{J}_i}$, Lemma 4.9a) implies that $\mathcal{B}$ is also a witness of α.

However, by Condition (R4) we know that $\alpha \in Q^{\neg}_{\mathsf{R}}$. Hence, by Condition (R3) we have $[\mathcal{K}^j_{\mathsf{R}}]^+ \not\models \exists x.\mathcal{B}(x)$, and thus $\mathcal{I}_j \models \neg\exists x.\mathcal{B}(x)$ by Proposition 2.11. This contradicts the fact that $\mathcal{B}^{\mathcal{I}_j}$ is non-empty. $\square$

This finishes the proof of Lemma 4.6. We next use this characterization to solve TCQ satisfiability (and entailment) in polynomial space.

4.1.3 The Upper Bound ctd.

The key insight of the previous section is that we do not need to store the exponentially large set $\mathcal{S}$ in order to check the conditions of Definition 4.5. It suffices to guess an ABox type $\mathcal{A}_{\mathsf{R}}$ and a set $Q^{\neg}_{\mathsf{R}}$ in advance, and then check, in each step of an LTL-satisfiability test for ϕ^{p}, if there is a world $X_i \subseteq \{p_1, \dots, p_m\}$ that satisfies the requirements specified in Definition 4.5.

For this purpose, we use the polynomial-space-bounded Turing machines for LTL-satisfiability constructed in [SC85]. Given the propositional LTL-formula ϕ^{p}, the machine $\mathcal{M}_{\phi^{\mathsf{p}}}$ iteratively guesses complete sets of (negated) subformulae of ϕ^{p} specifying which subformulae are satisfied at each point in time. Every such set induces a unique world $X_i \subseteq \{p_1, \dots, p_m\}$ containing the propositional variables that are true.

In [SC85, Theorem 4.7], it is shown that if ϕ^{p} is satisfiable, then there must be a *periodic* model of ϕ^{p} with a period that is exponential in the size of ϕ^{p}. Hence, $\mathcal{M}_{\phi^{\mathsf{p}}}$ first guesses two polynomial-sized indices specifying the beginning and end of the first period. Then it continuously increments a (polynomial-sized) counter and in each step guesses a complete set of (negated) subformulae of ϕ^{p}. It then checks Boolean consistency of this set and consistency with the set of the previous time point according to the temporal operators. For example, if the previous set contains the formula $p_1 \mathsf{U} p_2$, then either it also contains p_2 or it must contain p_1 and the current set must contain $p_1 \mathsf{U} p_2$. In this way, the satisfaction of the U-formula is deferred to the next time point.

In each step, the oldest set is discarded and replaced by the next one. When the counter reaches the beginning of the period, it stores the current set and continues until it reaches the end of the period. At that point, instead of guessing the next set of subformulae, the set stored at the beginning of the period is used and checked for consistency with the previous set as described above. $\mathcal{M}_{\phi^{\mathsf{p}}}$ additionally has to ensure that all U-subformulae are satisfied within the period. Thus, the Turing machine never has to remember more than three sets of polynomial size.

Note that [SC85] do not directly regard past operators, which are considered by us. However, we can certainly adapt the complete sets of subformulae guessed by $\mathcal{M}_{\phi^{\mathsf{p}}}$ to also include the past operators. This does not affect the space requirements of the Turing machines; in particular, the period that has to be guessed is still exponential in the size of ϕ^{p}. We now modify this procedure to prove the desired PSPACE upper bound.

Lemma 4.14. *If* $\mathsf{N_{RC}} \neq \emptyset$ *but* $\mathsf{N_{RR}} = \emptyset$*, then TCQ entailment in* $\mathcal{EL}$ *is in* PSPACE *w.r.t. combined complexity.*

Proof. Let $\mathcal{K} = \langle \mathcal{T}, \emptyset \rangle$ be a TKB and ϕ be a TCQ. We analyze the complexity of the satisfiability problem by showing how an r-satisfiable set $\mathcal{S}$ can be found. By Lemma 4.6, it suffices to find a tuple $(\mathcal{A}_{\mathsf{R}}, Q_{\mathsf{R}}^{\neg})$ satisfying conditions (R1)–(R4). All these conditions are such that it is not necessary to actually construct the whole set $\mathcal{S}$—it is enough to show that each world X_i we encounter when checking ϕ^{p} (*not* $\phi_{\mathcal{S}}^{\mathsf{p}}$) for satisfiability induces a knowledge base $\mathcal{K}_{\mathsf{R}}^{i}$ that satisfies all requirements.

We can thus run a modified version of the Turing machine $\mathcal{M}_{\phi^{\mathsf{p}}}$ that first guesses the sets $\mathcal{A}_{\mathsf{R}}$ and $Q_{\mathsf{R}}^{\neg}$ required by Definition 4.5, which can clearly be done in polynomial space, and then proceeds as before, but additionally executes the following checks for the world X induced by each guessed complete set of propositional subformulae:

(R1) Check the KB $\mathcal{K}_{\mathsf{R}} = \langle \mathcal{T}, \mathcal{A}_{\mathsf{R}} \cup \mathcal{A}_{Q_X} \rangle$ for consistency, where $\mathcal{A}_{Q_X}$ is formed by instantiating all CQs α_j with $p_j \in X$.

This consistency test can be done in polynomial time in the (polynomial) size of $\mathcal{K}_{\mathsf{R}}$ [BBL05] and thus needs only polynomial space.

(R2) Check, for each $p_j \in \overline{X}$, whether $\mathcal{K}_{\mathsf{R}} \not\models \alpha_j$ holds.

Note that we have $\mathcal{K}_{\mathsf{R}} \not\models \alpha_j$ iff $[\mathcal{K}_{\mathsf{R}}]^+ \not\models \alpha_j$. For ($\Leftarrow$), we have that, if $[\mathcal{K}_{\mathsf{R}}]^+ \not\models \alpha_j$, then $\mathcal{I}_{[\mathcal{K}_{\mathsf{R}}]^+} \not\models \alpha_j$ by Proposition 2.11. Furthermore, the canonical model $\mathcal{I}_{[\mathcal{K}_{\mathsf{R}}]^+}$ is also a model of $\mathcal{K}_{\mathsf{R}}$ (cf. the beginning of Section 4.1.2), and thus $[\mathcal{K}_{\mathsf{R}}]^+ \not\models \alpha_j$. For ($\Rightarrow$), we directly have that every model of $\mathcal{K}_{\mathsf{R}}$ that does not satisfy α_j is also a model of $[\mathcal{K}_{\mathsf{R}}]^+$. Hence, it suffices to check the non-entailment $[\mathcal{K}_{\mathsf{R}}]^+ \not\models \alpha_j$, which can be done in (deterministic) exponential time and polynomial space by Lemma 3.5.

(R3) Check, for each $\alpha \in Q_{\mathsf{R}}^{\neg}$ and every witness $\mathcal{B}$ of α w.r.t. $\mathcal{T}$, whether it holds that $\mathcal{K}_{\mathsf{R}} \not\models \exists x.\mathcal{B}(x)$.

Since each $\mathcal{B}$ is of polynomial size, the actual non-entailment test can be done in polynomial space by the same arguments as above. However, while we can easily enumerate all $\alpha \in Q_{\mathsf{R}}^{\neg}$ and $\mathcal{B} \subseteq \mathsf{N_{RC}}(\mathcal{T})$ in polynomial space, we still have to determine whether $\mathcal{B}$ is actually a witness of α w.r.t. $\mathcal{T}$.

In [BBM11, Lemma 12], it is shown that, for two concept names A, B, it can be decided in polynomial time whether there are role names $r_1, \ldots, r_\ell$ such that $\mathcal{T} \models A \sqsubseteq \exists r_1 \ldots r_\ell.B$. Essentially, it suffices to check reachability of B from A in an appropriate graph derived from $\mathcal{T}$. This idea is also implicitly used in the form of the reachability relation $\rightsquigarrow$ in [BBL05, KKS12].

We can use this approach for our problem by introducing two new concept names $A_\mathcal{B}$ and A_α and then checking in polynomial time whether

$$\mathcal{T} \cup \{A_\mathcal{B} \sqsubseteq \mathcal{B}, \mathsf{Con}(\alpha) \sqsubseteq A_\alpha\} \models A_\mathcal{B} \sqsubseteq \exists r_1 \ldots r_\ell.A_\alpha$$

holds for some role names $r_1, \ldots, r_\ell$, which is equivalent to the fact that $\mathcal{B}$ is a witness of α w.r.t. $\mathcal{T}$.

(R4) Check, for each $p_j \in \overline{X}$, whether $\alpha_j \in Q_\mathsf{R}^{\neg}$.

The set $\mathcal{S}$ required for Lemma 3.2 corresponds to the set of all worlds X encountered during a run of this modified Turing machine. Under this definition of $\mathcal{S}$, it is easy to see that the above checks are actually equivalent to (R1)–(R4) from Definition 4.5. By Lemmata 3.2 and 4.6, the described Turing machine accepts the input $\mathcal{K}$ and ϕ iff ϕ has a model w.r.t. $\mathcal{K}$ (recall that we can disregard the mapping ι due to our assumptions). Since we do not have to store $\mathcal{S}$ explicitly and all checks can be done with a nondeterministic Turing machine using only polynomial space, according to [Sav70], TCQ entailment can be decided in PSpace. □

This finishes the proof of Theorem 4.1.

4.2 The Case With Rigid Role Names

If the set $\mathsf{N_{RR}}$ is allowed to be non-empty, the combined complexity of the entailment problem increases significantly—in particular, because a polynomial amount of information does not suffice anymore to test the r-satisfiability of $\mathcal{S}$ by testing the worlds contained in $\mathcal{S}$ individually.

Theorem 4.15. *If $\mathsf{N_{RR}} \neq \emptyset$, then TCQ entailment in $\mathcal{EL}$ is* co-NExpTime-*complete w.r.t. combined complexity.*

In the following, we first prove the lower bound and then describe a procedure to obtain a corresponding co-NExpTime upper bound.

4.2.1 The Lower Bound

Lemma 4.16. *If $\mathsf{N_{RR}} \neq \emptyset$, then TCQ entailment in $\mathcal{EL}$ is* co-NExpTime-*hard w.r.t. combined complexity.*

Proof. The proof is by reduction of the 2^{n+1}-bounded domino problem [Lew78, BGG97], known to be NExpTime-hard [BGG97], to the satisfiability problem of Boolean TCQs w.r.t. a TO with rigid role names. The basic idea of the reduction is the same as for $\mathcal{ALC}$-LTL in [BGL12]. However, the lower expressivity of $\mathcal{EL}$ imposes restrictions that complicate the construction. We specifically describe the differences to the proof in [BGL12] in detail during our construction.

We start introducing the bounded version of the domino problem used in our reduction. A *domino system* is a triple $\mathcal{D} = (D, H, V)$, where D is a finite set of *domino types* and $H, V \subseteq D \times D$ are the horizontal and vertical *matching conditions*. Let $\mathcal{D}$ be a domino system and $I = d_0, \ldots, d_{n-1} \in D^n$ an *initial condition*, which is a sequence of domino types of length $n > 0$. A mapping $\tau\colon \{0, \ldots, 2^{n+1} - 1\} \times \{0, \ldots, 2^{n+1} - 1\} \to D$ is a 2^{n+1}*-bounded solution* of $\mathcal{D}$ respecting the initial condition I iff, for all $x, y < 2^{n+1}$, the following holds:

- If $\tau(x, y) = d$ and $\tau(x \oplus_{2^{n+1}} 1, y) = d'$, then $(d, d') \in H$;
- If $\tau(x, y) = d$ and $\tau(x, y \oplus_{2^{n+1}} 1) = d'$, then $(d, d') \in V$;
- $\tau(i, 0) = d_i$ for $i < n$;

where $\oplus_{2^{n+1}}$ denotes addition modulo 2^{n+1}. It is shown in [BGG97, Theorem 6.1.2] that there is a domino system $\mathcal{D} = (D, H, V)$ such that, given an initial condition $I = d_0, \ldots, d_{n-1} \in D^n$, the problem of deciding if $\mathcal{D}$ has a 2^{n+1}-bounded solution respecting I is NExpTime-hard. In what follows, we show that this problem can be reduced in polynomial time to satisfiability of Boolean TCQs w.r.t. a TO using rigid role names.

Our reduction focuses on a specific named individual a, which serves as successor w.r.t. a rigid role r, to certain other (at least 2^{2n+2} many) individuals. Furthermore, we particularly discern *global*[6] concept names that are flexible and are satisfied either by a and all its r-predecessors or by none of the above; in contrast, *local* concept names are rigid and used to identify specific domain elements. We need the following concept and individual names:

- an individual name a;
- a rigid role name r;
- flexible (global) concept names G_d, G_d^h, G_d^v, and a rigid (local) concept name L_d for all $d \in D$;
- rigid (local) concept names $X_0, \ldots, X_n$ and $Y_0, \ldots, Y_n$ that are used to realize two binary counters modulo 2^{n+1}, where the X-counter describes the horizontal and the Y-counter the vertical position of a domino;

[6]Not to be confused with *rigid* or *always* (in time).

- flexible (global) concept names $Z_0, \ldots, Z_{2n+1}$, $Z_0^h \ldots, Z_{2n+1}^h$, $Z_0^v, \ldots, Z_{2n+1}^v$ that are used to realize three binary counters modulo 2^{2n+2}, whose function is explained below;
- concept names $\overline{X}_0, \ldots, \overline{X}_n$, $\overline{Y}_0, \ldots, \overline{Y}_n$, $\overline{Z}_0, \ldots, \overline{Z}_{2n+1}$, $\overline{Z}_0^h, \ldots, \overline{Z}_{2n+1}^h$, and $\overline{Z}_0^v, \ldots, \overline{Z}_{2n+1}^v$ representing the complements of the above counters;
- auxiliary flexible concept names N, $E_0^h, \ldots, E_{2n+1}^h$, $E_0^v, \ldots, E_{2n+1}^v$.

The first $n+1$ bits of the Z-, Z^h- and Z^v-counters are used to represent 2^{n+1} horizontal components $0 \leq x < 2^{n+1}$, and the second $n+1$ bits of these counters are used to represent 2^{n+1} vertical components $0 \leq y < 2^{n+1}$. By counting with the Z-counter up to 2^{2n+2} in the temporal dimension, we ensure that every position $(x, y) \in \{0, \ldots, 2^{n+1}-1\} \times \{0, \ldots, 2^{n+1}-1\}$ is represented at some time point. To count, we enforce that, for every possible value of the Z-counter, there is a world where a belongs to the concepts from the corresponding subset of $\{Z_0, \ldots, Z_{2n+1}\}$. We will restrict the concept names Z_i to be global, and thus the value of the Z-counter is in every world transferred to all r-predecessors of a. For every position given by the Z-counter, the Z^h- and Z^v-counters represent the top and right neighbor position, respectively.

The rigid concept names $X_0, \ldots, X_n$ and $Y_0, \ldots, Y_n$ are then used to ensure that, in every world, there is one r-predecessor of a whose X- and Y-values match the value of the global Z-counter. Since they are rigid, this enforces that every position $(x, y) \in \{0, \ldots, 2^{n+1}-1\} \times \{0, \ldots, 2^{n+1}-1\}$ is represented by at least one r-predecessor of a in every world. Thus, for every position, we have a world representing it with the help of the global Z-counter, but we also have an individual representing it in every world with the help of the local X- and Y-counters.

Furthermore, appropriate assertions on a and specific GCIs are used to ensure that (i) every global/local position has exactly one domino type (given by G_d/L_d), and two global domino types for two neighbors (G_d^h, G_d^v); (ii) the domino types of G_d and L_d are the same, and G_d^h/G_d^v represent the same types as the value of L_d at the individuals corresponding to the correct neighbors; (iii) the horizontal and vertical matching conditions are respected; and that (iv) the initial condition is satisfied.

One of the main differences to the proof for $\mathcal{ALC}$-LTL [BGL12] lies in the presence of three global domino types. In $\mathcal{ALC}$-LTL, it was enough to have one local and one global type in order to enforce the matching conditions. Here, we enforce the matching conditions globally and then ensure that the local types of certain individuals are the same. Another difference is the presence of the concept names of the form $\overline{X_i}$ representing the complements of the various counters. In $\mathcal{ALC}$, these can be directly expressed as $\neg X_i$.

We now construct the Boolean TCQ $\phi_{\mathcal{D},I}$ as a conjunction of several formulae listed in the following. At the same time, we add GCIs to a global TBox $\mathcal{T}_{\mathcal{D},I}$.

- For every possible value of the Z-counter, there is a world where a belongs to the concepts from the corresponding subset of $\{Z_0, \ldots, Z_{2n+1}\}$. This is expressed using the following conjunct of $\phi_{\mathcal{D},I}$:

$$\Box \bigwedge_{0 \le i \le 2n+1} \Big(\Big(\bigwedge_{0 \le j < i} Z_j(a) \Big) \leftrightarrow \big(Z_i(a) \leftrightarrow \bigcirc \neg Z_i(a) \big) \Big)$$

  This formula expresses that the i-th bit of the Z-counter is flipped from one world to the next iff all preceding bits are true. Thus, the value of the Z-counter at the next world is equal to the value at the current world incremented by one.

- In every world, the counters Z^h and Z^v are synchronized with the Z-counter, meaning that a belongs to the concepts from the subsets of $\{Z_0^h, \ldots, Z_{2n+1}^h\}$ and $\{Z_0^v, \ldots, Z_{2n+1}^v\}$ that, respectively, point to the right and top neighbor position of the position distinguished by the Z-counter. This is enforced using formulae similar to the ones for the Z-counter above. First, the horizontal component of the Z^h-counter is equal to the horizontal component of the Z-counter plus 1:

$$\Box \bigwedge_{0 \le i \le n} \Big(\Big(\bigwedge_{0 \le j < i} Z_j(a) \Big) \leftrightarrow \big(Z_i(a) \leftrightarrow \neg Z_i^h(a) \big) \Big)$$

  The vertical component of the Z^h-counter is equal to that of the Z-counter:

$$\Box \bigwedge_{n+1 \le i \le 2n+1} \big(Z_i(a) \leftrightarrow Z_i^h(a) \big)$$

  And similarly for the Z^v-counter:

$$\Box \bigwedge_{n+1 \le i \le 2n+1} \Big(\Big(\bigwedge_{n+1 \le j < i} Z_j(a) \Big) \leftrightarrow \big(Z_i(a) \leftrightarrow \neg Z_i^v(a) \big) \Big)$$

$$\Box \bigwedge_{0 \le i \le n} \big(Z_i(a) \leftrightarrow Z_i^v(a) \big)$$

- The interpretation of the concept names $\overline{Z}_i$, $\overline{Z}_i^h$, and $\overline{Z}_i^v$, as the complements of Z_i, Z_i^h, and Z_i^v is enforced by the following formula:

$$\Box \bigwedge_{0 \le i \le 2n+1} \Big(\big(\overline{Z}_i(a) \leftrightarrow \neg Z_i(a) \big) \wedge \big(\overline{Z}_i^h(a) \leftrightarrow \neg Z_i^h(a) \big) \wedge \big(\overline{Z}_i^v(a) \leftrightarrow \neg Z_i^v(a) \big) \Big)$$

- The values of the three global counters Z, Z^h, and Z^v (and their complements) are shared by a and all its r-predecessors in each world. This is expressed by the following GCIs in $\mathcal{T}_{\mathcal{D},I}$ for $0 \le i \le 2n+1$:

$$\exists r.Z_i \sqsubseteq Z_i, \; \exists r.Z_i^h \sqsubseteq Z_i^h, \; \exists r.Z_i^v \sqsubseteq Z_i^v,$$
$$\exists r.\overline{Z}_i \sqsubseteq \overline{Z}_i, \; \exists r.\overline{Z}_i^h \sqsubseteq \overline{Z}_i^h, \; \exists r.\overline{Z}_i^v \sqsubseteq \overline{Z}_i^v$$

We also need the following formula to enforce that satisfaction of Z_i prevents satisfaction of $\overline{Z}_i$, and vice versa:

$$\Box \bigwedge_{0 \leq i \leq 2n+1} \neg\exists x. Z_i(x) \wedge \overline{Z}_i(x)$$

- In every world, there is at least one r-predecessor of a for which the combined values of the X- and the Y-counter correspond to the value of the global Z-counter in this world. For this, we use the following formula and GCIs, for $0 \leq i \leq n$ and $n+1 \leq j \leq 2n+1$:

  $$\Box\exists x. r(x, a) \wedge N(x)$$

  $$N \sqcap Z_i \sqsubseteq X_i, \;\; N \sqcap X_i \sqsubseteq Z_i, \;\; N \sqcap Z_j \sqsubseteq Y_{j-(n+1)}, \;\; N \sqcap Y_{j-(n+1)} \sqsubseteq Z_j$$

  Since the concept names X_i, Y_i are rigid, this ensures that, in every world, every possible combination of values of the X- and Y-counters is realized by some r-predecessor of a. For a given such combination, the corresponding individual represents the same value combination in every world.

- In the same way, we enforce the correct interpretation of the complements of the local counters:

  $$N \sqcap \overline{Z}_i \sqsubseteq \overline{X}_i, \;\; N \sqcap \overline{X}_i \sqsubseteq \overline{Z}_i, \;\; N \sqcap \overline{Z}_j \sqsubseteq \overline{Y}_{j-(n+1)}, \;\; N \sqcap \overline{Y}_{j-(n+1)} \sqsubseteq \overline{Z}_j$$

- Every world gets exactly one (global) domino type that belongs to the position given by the global Z-counter:

  $$\Box \bigvee_{d \in D} \Big(G_d(a) \wedge \bigwedge_{d' \in D \setminus \{d\}} \neg\exists x. G_{d'}(x) \Big)$$

  To enforce the global domino types in the r-predecessors of a, we again need the GCI

  $$\exists r. G_d \sqsubseteq G_d$$

  for every $d \in D$. The converse direction, i.e., that $\neg G_d(a)$ implies that all r-predecessors of a do not satisfy G_d, is covered already by the negated CQs $\neg\exists x. G_{d'}(x)$ in the above formula.

  We do the same for the global domino type G^h_d and G^v_d for the right and top neighbor positions, respectively (corresponding to the positions given by Z^h and Z^v):

  $$\Box \bigvee_{d \in D} \Big(G^h_d(a) \wedge \bigwedge_{d' \in D \setminus \{d\}} \neg\exists x. G^h_{d'}(x) \Big) \qquad \exists r. G^h_d \sqsubseteq G^h_d$$

  $$\Box \bigvee_{d \in D} \Big(G^v_d(a) \wedge \bigwedge_{d' \in D \setminus \{d\}} \neg\exists x. G^v_{d'}(x) \Big) \qquad \exists r. G^v_d \sqsubseteq G^v_d$$

- Given the global types of the neighbor positions, the horizontal and vertical matching condition can be enforced easily:

$$\Box\Big(\bigvee_{(d,d')\in H}\Big(G_d(a)\wedge G^h_{d'}(a)\Big)\wedge\bigvee_{(d,d')\in V}\Big(G_d(a)\wedge G^v_{d'}(a)\Big)\Big)$$

- To synchronize the domino types G_d, G^h_d, and G^v_d among the different worlds (otherwise G^h_d would not need to be equal to the value of G_d at the world whose Z-counter is equal to the current Z^h-counter), we use the local (rigid) domino types L_d. First, we ensure that the local type of the individual representing the same position as the current world is the same as the current global type. We use the following GCIs for all $d \in D$:

$$N \sqcap G_d \sqsubseteq L_d,\ \ N \sqcap L_d \sqsubseteq G_d$$

  Since the concept names L_d are rigid, this type is then associated with the individual in every world. And because every world has exactly one global domino type G_d (which is shared by all its individuals), every individual also has exactly one local domino type: the one of the world representing the same position.

  To synchronize the domino types of the neighbors given by G^h_d and G^v_d, we employ the auxiliary concept names E^h_i, E^v_i within the following GCIs, for $0 \le i \le n$ and $n+1 \le j \le 2n+1$:

$$\begin{gathered}Z^h_i \sqcap X_i \sqsubseteq E^h_i,\ \overline{Z}^h_i \sqcap \overline{X}_i \sqsubseteq E^h_i,\ Z^h_j \sqcap Y_{j-(n+1)} \sqsubseteq E^h_j,\ \overline{Z}^h_j \sqcap \overline{Y}_{j-(n+1)} \sqsubseteq E^h_j,\\ Z^v_i \sqcap X_i \sqsubseteq E^v_i,\ \overline{Z}^v_i \sqcap \overline{X}_i \sqsubseteq E^v_i,\ Z^v_j \sqcap Y_{j-(n+1)} \sqsubseteq E^v_j,\ \overline{Z}^v_j \sqcap \overline{Y}_{j-(n+1)} \sqsubseteq E^v_j\end{gathered}$$

  In this way, the interpretation of $E^h_1 \sqcap \cdots \sqcap E^h_{2n+1}$ must include all those domain elements whose X- and Y-counters match the current Z^h-counter. This particularly includes the one individual that was created in the corresponding world using the CQ $\exists x.r(x,a) \wedge N(x)$—at which the local domino type equals the current global domino type. Thus, all that remains to do is to ensure that the global domino type G^h_d matches the local domino type L_d at all domain elements satisfying $E^h_1 \sqcap \cdots \sqcap E^h_{2n+1}$. Of course, similar arguments apply for the vertical direction.

$$\begin{gathered}E^h_0 \sqcap \ldots \sqcap E^h_{2n+1} \sqcap G^h_d \sqsubseteq L_d,\ E^h_0 \sqcap \ldots \sqcap E^h_{2n+1} \sqcap L_d \sqsubseteq G^h_d,\\ E^v_0 \sqcap \ldots \sqcap E^v_{2n+1} \sqcap G^v_d \sqsubseteq L_d,\ E^v_0 \sqcap \ldots \sqcap E^v_{2n+1} \sqcap L_d \sqsubseteq G^v_d\end{gathered}$$

- It remains to represent the initial condition $I = d_0, \ldots, d_{n-1}$. For this, we use the following GCI for all $i = 0, \ldots, n-1$:

$$(C^x_Z = i) \sqcap \overline{Z}_{n+1} \sqcap \cdots \sqcap \overline{Z}_{2n+1} \sqsubseteq G_{d_i},$$

where, for any $b_j \in \{0,1\}$, $0 \leq j \leq n$,

$$\Big(C_Z^x = \sum_{0\leq j\leq n} 2^j * b_j\Big) := \bigsqcap_{\substack{0\leq j\leq n\\ b_j=0}} \overline{Z}_j \sqcap \bigsqcap_{\substack{0\leq j\leq n\\ b_j=1}} Z_j.$$

This conjunction identifies a particular x-position in the Z-counter. If the y-component of the Z-counter is 0, additionally, then the corresponding type of the initial condition is enforced.

This finishes the definition of the Boolean TCQ $\phi_{\mathcal{D},I}$ and the global TBox $\mathcal{T}_{\mathcal{D},I}$, which consist of the conjuncts and GCIs specified above. It is easy to see that the size of $\phi_{\mathcal{D},I}$ and $\mathcal{T}_{\mathcal{D},I}$ is polynomial in n. Moreover, $\phi_{\mathcal{D},I}$ is satisfiable w.r.t. $\langle \mathcal{T}_{\mathcal{D},I}, \emptyset \rangle$ iff $\mathcal{D}$ has a 2^{n+1}-bounded solution respecting I. □

4.2.2 The Upper Bound

Lemma 4.17. *If* $\mathsf{N_{RR}} \neq \emptyset$*, then TCQ entailment in* $\mathcal{EL}$ *is in* co-NExpTime *w.r.t. combined complexity.*

Proof. As before, we analyze the satisfiability problem w.r.t. the conditions of Lemma 3.2 and combined complexity.

- The set $\mathcal{S}$, which is of exponential size, and the mapping ι, which is of linear size, can thus be nondeterministically guessed in NExpTime.
- Further, by Lemma 3.3, the LTL-satisfiability test can be done in ExpTime.
- Lemma 3.4 states that, to decide the r-satisfiability of $\mathcal{S}$, we can decide the satisfiability of the conjunction of CQ literals $\chi_{\mathcal{S},\iota}$, which is of size exponential in ϕ and polynomial in $\mathcal{K}$, w.r.t. the TBox $\mathcal{T}_{\mathcal{S},\iota}$, whose size is linear in that of $\mathcal{T}$ and exponential in that of ϕ. By Lemma 3.5, this test can be done by solving exponentially many (in the size of ϕ) tests in polynomial time w.r.t. the input $\chi_{\mathcal{S},\iota}$ and $\mathcal{T}_{\mathcal{S},\iota}$, and hence in exponential time.

Altogether, this means that we can decide TCQ satisfiability in NExpTime, and hence entailment in co-NExpTime. □

5 Regarding Data Complexity

In this section, we show that, if rigid symbols are not considered at all, the TCQ entailment problem is not harder than the problem of entailment of CQs in $\mathcal{EL}$; particularly, because we can enumerate all possible sets $\mathcal{S}$, which are independent of the data, in constant time and do the check if a mapping ι exists in a step-wise fashion, independently for each time point. Subsequently, we show that assertions of rigid concept names on specific individuals (at arbitrary time points) introduce nondeterminism w.r.t. the concepts satisfied by predecessors (at possibly other time points) of these individuals, and prove co-NP-hardness, for this case. Nevertheless, the matching co-NP upper bound can be achieved also for the case with rigid role names.

5.1 The Case Without Rigid Names

We first regard the case where we do not consider rigid names at all.

Theorem 5.1. *If* $\mathsf{N_{RC}} = \mathsf{N_{RR}} = \emptyset$*, then TCQ entailment in* $\mathcal{EL}$ *is* P*-complete w.r.t. data complexity.*

P-hardness follows from the fact that entailment of CQs in $\mathcal{EL}$ is already P-hard with respect to data complexity [CDL+06]. The corresponding upper bound is provided by the following lemma.

Lemma 5.2. *If* $\mathsf{N_{RC}} = \mathsf{N_{RR}} = \emptyset$*, then TCQ entailment in* $\mathcal{EL}$ *is in* P *w.r.t. data complexity*

Proof. We again follow the basic approach of Lemma 3.2 for the satisfiability problem.

- To check r-satisfiability of a set $\mathcal{S} = \{X_1, \dots, X_k\} \subseteq 2^{\{p_1,\dots,p_m\}}$, it clearly suffices to check satisfiability of the conjunctions χ_i, $1 \leq i \leq k$, and $\chi_{\iota(i)} \wedge \bigwedge_{\alpha \in \mathcal{A}_i} \alpha$, $0 \leq i \leq n$, w.r.t. $\mathcal{T}$ individually.[7] This is because, without rigid names, it is impossible to enforce any dependency between the sets $X \in \mathcal{S}$, and hence it suffices to define $\mathcal{S}$ as the set of *all* sets X_i for which χ_i is satisfiable w.r.t. $\mathcal{T}$. This can be computed in constant time w.r.t. the size of the input ABoxes. However, it remains to show how to obtain ι for the remaining satisfiability tests.
- To stay in P while obtaining the mapping ι, we clearly cannot guess one ι or enumerate the exponential number of possible ι. Instead, we first check, for each $X_j \in \mathcal{S}$ and input ABox $\mathcal{A}_i$, whether χ_j is satisfiable w.r.t. $\langle \mathcal{T}, \mathcal{A}_i \rangle$. We collect all indices j that pass this test into the set $\iota'(i)$. In this way, we obtain all possible worlds for each of the input ABoxes. Each of the conjunctions χ_j is of constant size and the number of conjunctions ($|\mathcal{S}|$) is

[7]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, and that all individual names are interpreted by the same domain elements in all models.

constant. By Lemma 3.5, these tests can thus be done in polynomial time in the size of the input ABoxes. Furthermore, each $\iota'(i)$ is of constant size.

Correspondingly, we have to change the condition "$w_i = X_{\iota(i)}$" in Lemma 3.3 to "$w_i = X_j$ for some $j \in \iota'(i)$". The result remains valid since one can easily modify the automaton used in the proof of that result in [BBL15] to check whether the first $n+1$ encountered worlds fall into the pre-specified sets of (constantly many) worlds $\iota'(i)$, instead of equality with a single pre-specified world $\iota(i)$.

If both of the above polynomial tests succeed, then we can simply choose one ι among the many possible identified by ι' in order to satisfy the conditions of Lemma 3.2. Conversely, the existence of some $\mathcal{S}$ and ι imply that we have $\iota(i) \in \iota'(i)$ for every i, $0 \leq i \leq n$, and thus the above checks succeed. This proves that our deterministic definitions of the maximal possible $\mathcal{S}$ and ι' suffice to satisfy Lemma 3.2, which means that we can decide TCQ satisfiability (and entailment) in P. □

5.2 The Case With Rigid Names

As outlined previously, the use of rigid symbols in the ABoxes presents a source of nondeterminism causing NP-hardness. For our following proof of the lower bound, it is especially important that $\mathcal{EL}$ allows for qualified existential restrictions.

Theorem 5.3. *If $\mathsf{N_{RC}} = \mathsf{N_{RR}} \neq \emptyset$, then TCQ entailment in $\mathcal{EL}$ is* co-NP*-complete w.r.t. data complexity.*

In the following, we first prove the required lower bound and then describe how a corresponding upper bound can be obtained.

5.2.1 The Lower Bound

Lemma 5.4. *If $\mathsf{N_{RC}} \neq \emptyset$, then TCQ entailment in $\mathcal{EL}$ is* co-NP*-hard w.r.t. data complexity.*

Proof. We show NP-hardness of the satisfiability problem. The proof is by reduction of the 3-SAT problem, which is known to be NP-complete [Kar72]. Consider a propositional 3-CNF formula

$$\psi = \bigwedge_{0 \leq i < \ell} l_{i,1} \vee l_{i,2} \vee l_{i,3},$$

with the literals $l_{i,j}$, and let $x_1, \ldots, x_m$ be all the propositional variables occurring in ψ. We denote by Lit the set of all literals over these variables. For a literal l, we denote by $\neg l$ its complement literal.

We now construct a TCQ ϕ and a TKB $\mathcal{K}_\psi = \langle \mathcal{T}, (\mathcal{A}_t^\psi)_{0 \leq t < 4\ell} \rangle$ such that ψ is satisfiable iff ϕ is satisfiable w.r.t. $\mathcal{K}_\psi$. Our definitions of ϕ and $\mathcal{T}$ will not depend on ψ and the combined size of the ABoxes $\mathcal{A}_t^\psi$ will be linear in the size of ψ, and hence we obtain the desired result.

We use four ABoxes to represent each clause: one to identify the start of a new clause, and the following three for the literals. Then, we enforce through ϕ that every model of these ABoxes has to satisfy one of the clause's literals. By using a single rigid concept, we can additionally enforce that every variable has to be interpreted either always true or always false, yielding a model of ψ. More formally, we use the following symbols:

- individual names a_l for all literals $l \in \mathsf{Lit}$;
- an individual name c representing the 'current' clause;
- a rigid concept name A that describes the truth values of all literals;
- a flexible concept name C to mark the current time point t as the start of the encoding of clause $\frac{t}{4}$;
- a flexible concept name T to identify which literal of a clause is satisfied;
- a role name r to link each a_l to $a_{\neg l}$ to ensure that the truth assignment is consistent;
- a role name s to relate a clause with its literals.

The TQ ϕ and TBox $\mathcal{T}$ can now be defined independent of the concrete input problem:

$$\begin{aligned}\phi := \Box\Big(\big(C(c) \rightarrow \big(\bigcirc T(c) \vee \bigcirc\bigcirc T(c) \vee \bigcirc\bigcirc\bigcirc T(c)\big)\big)&\\ \wedge \neg\exists x, y. r(x, y) \wedge A(x) \wedge A(y)\Big),&\text{ and}\\ \mathcal{T} := \{\exists s.T \sqsubseteq A\}.&\end{aligned}$$

Thus, whenever $C(c)$ holds, one of the next three time points, of which each is pointing to one of the three literals of the current clause, must satisfy $T(c)$. The TQ ϕ additionally ensures that individuals linked by r cannot both satisfy the rigid concept name A at the same time. The TBox is used to transfer the information about the choice of literal l to the truth value (represented by A) of the individual name a_l. Note that our reduction requires several features: a

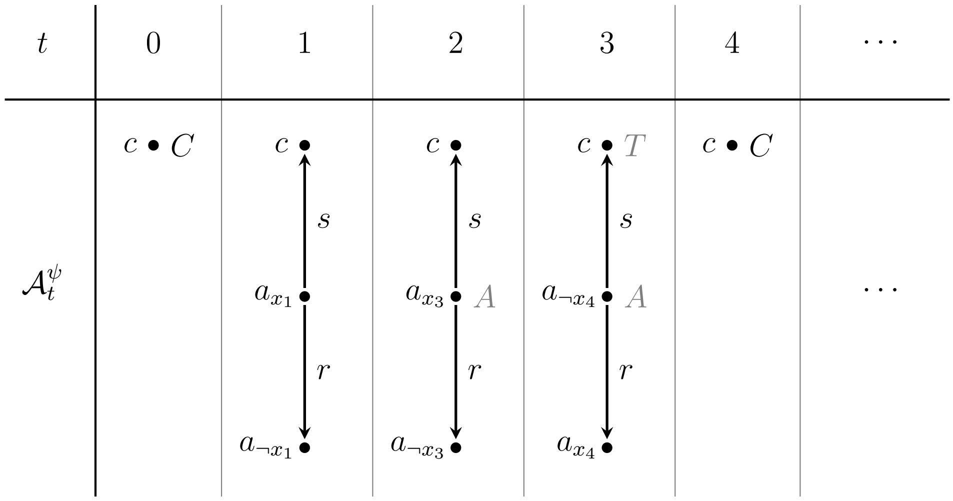

Figure 1: The content of the ABoxes encoding $(x_1 \vee x_3 \vee \neg x_4) \wedge \ldots$; names in gray describe a possible extension to a model of ϕ w.r.t. $\mathcal{K}_\psi$.

quantified existential restriction within a GCI, a rigid concept, and the Boolean negation operator together with the temporal operators in the TCQ.

The clauses of ψ are encoded in the ABoxes $\mathcal{A}_t^\psi$, $0 \leq t < 4\ell$, defined as follows for all $0 \leq i < \ell$ and $1 \leq j \leq 3$ (see also Figure 1):

$$\mathcal{A}_{4i}^\psi := \{C(c)\}$$
$$\mathcal{A}_{4i+j}^\psi := \{r(a_{l_{i,j}}, a_{\neg l_{i,j}}), s(a_{l_{i,j}}, c)\}$$

We now show that there is an assignment $v\colon \{x_1, \ldots, x_m\} \to \{0, 1\}$ that satisfies ψ iff ϕ is satisfiable w.r.t. $\mathcal{K}_\psi$.

($\Rightarrow$) Let v be such an assignment. We define the model $\mathfrak{I} = (\mathcal{I}_t)_{t\geq 0}$ of ϕ w.r.t. $\mathcal{K}_\psi$ with domain $\Delta := \{c, a_{x_1}, \ldots, a_{x_m}, a_{\neg x_1}, \ldots, a_{\neg x_m}\}$, where all individual names occuring in the ABoxes are interpreted as themselves:

$$A^{\mathcal{I}_t} := \{a_l \mid l \in \mathsf{Lit},\ v(l) = 1\};$$
$$T^{\mathcal{I}_t} := \{c \mid 0 \leq i < \ell,\ 1 \leq j \leq 3,\ t = 4i + j,\ v(l_{i,j}) = 1\};$$
$$C^{\mathcal{I}_t} := \{e \mid t < 4\ell,\ C(e) \in \mathcal{A}_t^\psi\};$$
$$r^{\mathcal{I}_t} := \{(e, e') \mid t < 4\ell,\ r(e, e') \in \mathcal{A}_t^\psi\};$$
$$s^{\mathcal{I}_t} := \{(e, e') \mid t < 4\ell,\ s(e, e') \in \mathcal{A}_t^\psi\}.$$

We obviously have $\mathcal{I}_t \models \mathcal{A}_t^\psi$, for all $0 \leq t < 4\ell$. Consider now the GCI $\exists s.T \sqsubseteq A$. By the definition of the ABoxes $\mathcal{A}_t^\psi$, the left-hand side concept can only be satisfied by an individual of the form a_l. If $a_l \in (\exists s.T)^{\mathcal{I}_t}$, then we have $l = l_{i,j}$ for $t = i + 4j$ and $c \in T^{\mathcal{I}_t}$. By the definition of $T^{\mathcal{I}_t}$, this yields $v(l) = 1$. But then we also have $a_l \in A^{\mathcal{I}_t}$, which shows that $\mathfrak{I}$ is a model of $\mathcal{K}_\psi$.

Since v satisfies each clause of ψ, it is clear that $\mathfrak{I}$ satisfies the implication

$$C(c) \rightarrow \Big(\bigcirc T(c) \vee \bigcirc \bigcirc T(c) \vee \bigcirc \bigcirc \bigcirc T(c) \Big)$$

at every time point by its definition, especially w.r.t. T. Moreover, whenever $(d, e) \in r^{\mathcal{I}_t}$, we must have $d = a_l$ and $e = a_{\neg l}$ for some $l \in \mathsf{Lit}$, and thus by the definition of $A^{\mathcal{I}_t}$ we cannot have both $d \in A^{\mathcal{I}_t}$ and $e \in A^{\mathcal{I}_t}$. This shows that $\mathfrak{I}$ also satisfies ϕ.

($\Leftarrow$) Let $\mathfrak{I} = (\mathcal{I}_t)_{t \geq 0}$ be a model of ϕ w.r.t. $\mathcal{K}_\psi$ that interprets all individual names as themselves. We define $v(x_k) := 1$ if $a_{x_k} \in A^{\mathcal{I}_0}$, and $v(x_k) := 0$ otherwise.

Consider now any clause $l_{i,1} \vee l_{i,2} \vee l_{i,3}$ of ψ. We have $C(c) \in A_{4i}$, and thus by the definition of ϕ there must be an index j, $1 \leq j \leq 3$, such that $c \in T^{\mathcal{I}_{4i+j}}$. By the definition of A_{4i+j}, we also have $(a_{l_{i,j}}, c) \in s^{\mathcal{I}_{4i+j}}$, and thus $a_{l_{i,j}} \in A^{\mathcal{I}_{4i+j}} = A^{\mathcal{I}_0}$ because of the GCI $\exists s.T \sqsubseteq A$.

If $l_{i,j}$ is a variable, then by the definition of v we immediately get $v(l_{i,j}) = 1$, which shows that the clause is satisfied by v. Otherwise, we have $l_{i,j} = \neg x_k$ for some k, $1 \leq k \leq m$. By the definition of $\mathcal{A}^\psi_{4i+j}$, we know that $(a_{x_k}, a_{\neg x_k}) \in r^{\mathcal{I}_{4i+j}}$. Since $\mathfrak{I}$ satisfies ϕ and $a_{\neg x_k} \in A^{\mathcal{I}_0}$, it cannot be the case that $a_{x_k} \in A^{\mathcal{I}_0}$. This means that $v(x_k) = 0$, and thus we again have $v(l_{i,j}) = 1$. □

5.2.2 The Upper Bound

Lemma 5.5. *If $\mathsf{N_{RR}} \neq \emptyset$, then TCQ entailment in $\mathcal{EL}$ is in* co-NP *w.r.t. data complexity.*

Proof. We analyze the satisfiability problem w.r.t. the conditions from Lemma 3.2 and data complexity.

- In this case, the set $\mathcal{S}$ is of constant size and the mapping ι is of linear size. They can thus be guessed nondeterministically in polynomial time.
- Further, by Lemma 3.3, the corresponding LTL-satisfiability test of $\phi^{\mathsf{p}}_{\mathcal{S}}$ can be done in P.
- For testing the r-satisfiability of $\mathcal{S}$ w.r.t. data complexity, by Lemma 3.4 we only have to check the satisfiability of $\chi_{\mathcal{S},\iota}$ w.r.t. $\mathcal{T}_{\mathcal{S},\iota}$. The conjuncts of $\chi_{\mathcal{S},\iota}$ induced by the input ABoxes $\mathcal{A}_i$ can be regarded as an ABox that is essentially of the same size as the sequence $(\mathcal{A}_i)_{0 \leq i \leq n}$, and the remaining conjunction is of linear size. However, the individual size of the remaining conjuncts is independent of the input ABoxes. The size of $\mathcal{T}_{\mathcal{S},\iota}$ is also linear in n. By Lemma 3.5, the above satisfiability test can thus be done in polynomial time.

This means that we can decide the satisfiability problem in NP, and thus entailment in co-NP w.r.t. data complexity. □

6 Conclusions

In this report, we focused on temporalized OBDA to support the interpretation of sensor data in a context-aware system by recognizing complex situations. In particular, we investigated the combined and data complexity of TCQ entailment w.r.t. knowledge bases in the DL $\mathcal{EL}$.

Our results are summarized in Table 1.1. It turns out that the data complexity, which is of most interest for our scenario, only stays tractable if rigid symbols are not allowed. In this case, it may be possible to adapt the so-called combined approach of [KLT⁺11], which proposes a procedure for CQ answering w.r.t. an $\mathcal{EL}$-knowledge base where the assertional data can be accessed through a traditional database system. The PSpace result for combined complexity is interesting in that it does not increase the complexity given by the satisfiability problem of propositional LTL—even if rigid concept names are considered. In addition, Table 1.1 shows that this contrasts the complexity of the very similar satisfiability problem in $\mathcal{EL}$-LTL.

In future work, we want to further investigate TCQs w.r.t. knowledge bases formulated in OWL 2 EL,[8] a profile of the current version of the web ontology language OWL 2 that is based on a maximally tractable extension of $\mathcal{EL}$ [BBL08]. The combined complexity of CQ answering increases from NP to PSpace when extending $\mathcal{EL}$ to OWL 2 EL [SMKR14], while the data complexity stays in P [ORŠ11], and thus it is possible that the complexity of TCQ entailment remains the same. Further, the paper [SMKR14] also provides a construction of canonical models for such KBs, which are critical for our PSpace upper bound w.r.t. combined complexity.

Moreover, we plan to consider TCQs in the context of the *DL-Lite* family of lightweight DLs. Since the features provided by $\mathcal{EL}$ are critical for both of our proofs of the lower bounds, it would be interesting to learn more about full TCQs[9] w.r.t. such DLs—in particular, about the data complexity in case rigid symbols are considered.

Last but not least, a practical application of TCQs would give insight into specialized use cases and maybe enable the development of optimized answering procedures.

[8] `http://www.w3.org/TR/owl2-profiles/#OWL_2_EL`

[9] Recall that several subsets of TCQs have already been considered in literature.

References

[ABM$^+$14] Alessandro Artale, Davide Bresolin, Angelo Montari, Guido Sciavicco, and Vladislav Ryzhikov. *DL-Lite* and interval temporal logics: a marriage proposal. In Torsten Schaub, editor, *Proc. of the 21st Eur. Conf. on Artificial Intelligence (ECAI'14)*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 957–958. IOS Press, 2014.

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

[AKK$^+$14] Alessandro Artale, Roman Kontchakov, Alisa Kovtunova, Vladislav Ryzhikov, Frank Wolter, and Michael Zakharyaschev. Temporal OBDA with LTL and dl-lite. In *Informal Proceedings of the 27th International Workshop on Description Logics, Vienna, Austria, July 17-20, 2014.*, pages 21–32, 2014.

[AKL$^+$07] Alessandro Artale, Roman Kontchakov, Carsten Lutz, Frank Wolter, and Michael Zakharyaschev. Temporalising tractable description logics. In Valentin Goranko and X. Sean Wang, editors, *Proceedings of the 14th International Symposium on Temporal Representation and Reasoning (TIME 2007)*, pages 11–22. IEEE Press, 2007.

[AKRZ14] Alessandro Artale, Roman Kontchakov, Vladislav Ryzhikov, and Michael Zakharyaschev. A cookbook for temporal conceptual data modelling with description logics. *ACM Transactions on Computational Logic*, 15(3):25, 2014.

[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.

[BBL05] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI'05)*, pages 364–369. Professional Book Center, 2005.

[BBL08] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope further. In Kendall Clark and Peter F. Patel-Schneider, editors, *Proc. of the 4th Workshop on OWL: Experiences and Directions*, pages 1–10, 2008.

[BBL13] Franz Baader, Stefan Borgwardt, and Marcel Lippmann. Temporalizing ontology-based data access. In Maria Paola Bonacina, editor, *Proc. of the 24th Int. Conf. on Automated Deduction (CADE'13)*, volume 7898 of *Lecture Notes in Computer Science*, pages 330–344. Springer-Verlag, 2013.

[BBL15] Franz Baader, Stefan Borgwardt, and Marcel Lippmann. Temporal query entailment in the description logic $\mathcal{SHQ}$. *Journal of Web Semantics*, 2015. In press.

[BBM11] Franz Baader, Stefan Borgwardt, and Barbara Morawska. Unification in the description logic $\mathcal{EL}$ w.r.t. cycle-restricted tboxes. LTCS-Report 11-05, Chair for Automata Theory, TU Dresden, Germany, 2011. See http://lat.inf.tu-dresden.de/research/reports.html.

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

[BGG97] Egon Börger, Erich Grädel, and Yuri Gurevich. *The Classical Decision Problem.* Perspectives in Mathematical Logic. Springer-Verlag, 1997.

[BGL12] Franz Baader, Silvio Ghilardi, and Carsten Lutz. LTL over description logic axioms. *ACM Transactions on Computational Logic*, 13(3):21:1–21:32, 2012.

[BLT15] Stefan Borgwardt, Marcel Lippmann, and Veronika Thost. Temporalizing rewritable query languages over knowledge bases. *Journal of Web Semantics*, 2015. In 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 Welty, editors, *Proc. of the 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'06)*, pages 260–270. AAAI Press, 2006.

[CM77] Ashok K. Chandra and Philip M. Merlin. Optimal implementation of conjunctive queries in relational data bases. In *Proc. of the 9th Annual ACM Symp. on Theory of Computing (STOC'77)*, pages 77–90. ACM, 1977.

[DEFS98] Stefan Decker, Michael Erdmann, Dieter Fensel, and Rudi Studer. Ontobroker: Ontology based access to distributed and semi-structured information. In *Database Semantics: Semantic Issues in Multimedia Systems*, pages 351–369. Kluwer Academic Publisher, 1998.

[GJS14] Víctor Gutiérrez-Basulto, Jean Christoph Jung, and Thomas Schneider. Lightweight description logics and branching time: A troublesome marriage. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014, Vienna, Austria, July 20-24, 2014*, 2014.

[HSK09] Jong-yi Hong, Eui-ho Suh, and Sung-Jin Kim. Context-aware systems: A literature review and classification. *Expert Systems with Applications*, 36(4):8509 – 8522, 2009.

[Kar72] Richard Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Proc. of a Symp. on the Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[KKS12] Yevgeny Kazakov, Markus Krötzsch, and František Simančík. Practical reasoning with nominals in the $\mathcal{EL}$ family of description logics. In Gerhard Brewka, Thomas Eiter, and Sheila A. McIlraith, editors, *Proc. of the 13th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'12)*, pages 264–274. AAAI Press, 2012.

[KL07] Adila Krisnadhi and Carsten Lutz. Data complexity in the $\mathcal{EL}$ family of description logics. In Nachum Dershowitz and Andrei Voronkov, editors, *Proc. of the 14th Int. Conf. on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'07)*, volume 4790 of *Lecture Notes in Computer Science*, pages 333–347. Springer-Verlag, 2007.

[KLT+11] Roman Kontchakov, Carsten Lutz, David Toman, Frank Wolter, and Michael Zakharyaschev. The combined approach to ontology-based data access. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 2656–2661, 2011.

[KRH07] Markus Krötzsch, Sebastian Rudolph, and Pascal Hitzler. Conjunctive queries for a tractable fragment of OWL 1.1. In Karl Aberer, Key-Sun Choi, Natasha Noy, Dean Allemang, Kyung-Il Lee, Lyndon Nixon, Jennifer Goldbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors, *Proc. of the 6th Int. Semantic Web Conf. (ISWC'07)*, volume 4825 of *Lecture Notes in Computer Science*, pages 310–323. Springer-Verlag, 2007.

[Lew78] Harry R. Lewis. Complexity of solvable cases of the decision problem for the predicate calculus. In *Proc. of the 19th Annual Symp. on Foundations of Computer Science (SFCS'78)*, pages 35–47. IEEE Press, 1978.

[LTW09] Carsten Lutz, David Toman, and Frank Wolter. Conjunctive query answering in the description logic $\mathcal{EL}$ using a relational database system. In Craig Boutilier, editor, *Proc. of the 21st Int. Joint Conf. on Artificial Intelligence (IJCAI'09)*, pages 2070–2075. AAAI Press, 2009.

[LWZ08] Carsten Lutz, Frank Wolter, and Michael Zakharyaschev. Temporal description logics: A survey. In Stéphane Demri and Christian S. Jensen, editors, *Proceedings of the 15th International Symposium on Temporal Representation and Reasoning (TIME 2008)*, pages 3–14. IEEE Press, 2008.

[ORŠ11] Magdalena Ortiz, Sebastian Rudolph, and Mantas Šimkus. Query answering in the horn fragments of the description logics $\mathcal{SHOIQ}$ and $\mathcal{SROIQ}$. In Toby Walsh, editor, *Proc. of the 22nd Int. Joint Conf. on Artificial Intelligence (IJCAI'11)*, pages 1039–1044. AAAI Press, 2011.

[PLC+08] Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. Linking data to ontologies. *Journal of Data Semantics*, 10:133–173, 2008.

[Pnu77] Amir Pnueli. The temporal logic of programs. In *Proc. of the 18th Annual Symp. on Foundations of Computer Science (SFCS'77)*, pages 46–57. IEEE Press, 1977.

[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.

[Ros07] Riccardo Rosati. On conjunctive query answering in $\mathcal{EL}$. In Diego Calvanese, Enrico Franconi, Volker Haarslev, Domenico Lembo, Boris Motik, Anni-Yasmin Turhan, and Sergio Tessaris, editors, *Proc. of the 2007 Int. Workshop on Description Logics (DL'07)*, volume 250 of *CEUR Workshop Proceedings*, pages 451–458, 2007.

[Sav70] Walter J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4(2):177–192, 1970.

[SC85] A. Prasad Sistla and Edmund M. Clarke. The complexity of propositional linear temporal logics. *Journal of the ACM*, 32(3):733–749, 1985.

[SMKR14] Giorgio Stefanoni, Boris Motik, Markus Krötzsch, and Sebastian Rudolph. The complexity of answering conjunctive and navigational queries over OWL 2 EL knowledge bases. *Journal of Artificial Intelligence Research*, 51:645–705, 2014.

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