# Temporalizing Ontology-Based Data Access*

Franz Baader, Stefan Borgwardt, and Marcel Lippmann

TU Dresden, Germany,
{baader,stefborg,lippmann}@tcs.inf.tu-dresden.de

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

## 1 Introduction

Situation awareness tools [2,12] try to help the user to detect certain situations within a running system. Here "system" is seen in a broad sense: it may be a computer system, air traffic observed by radar, or a patient in an intensive care unit. From an abstract point of view, the system is observed by certain "sensors" (e.g., heart-rate and blood pressure monitors for a patient), and the results of sensing are stored in a fact base. Based on the information available in the fact base, the situation awareness tool is supposed to detect certain predefined situations (e.g., heart-rate very high and blood pressure low), which require a reaction (e.g., fetch a doctor or give medication).

In a simple setting, one could realize such a tool by using standard database techniques: the information obtained from the sensors is stored in a relational database, and the situations to be recognized are specified by queries in an appropriate query language (e.g., conjunctive queries [1]). However, in general we cannot assume that the sensors provide us with a complete description of the current state of the system, and thus the closed world assumption (CWA) employed by database systems (where facts not occurring in the database are assumed to be false) is not appropriate (since there may be facts for which it is not known whether they are true or false). In addition, though one usually does not have a complete specification of the working of the system (e.g., a

* Partially supported by DFG SFB 912 (HAEC) and GRK 1763 (QuantLA).

complete biological model of a human patient), one has some knowledge about how the system works. This knowledge can be used to formulate constraints on the interpretation of the predicates used in the queries, which may cause more answers to be found.

Ontology-based data access [11,17] addresses these requirements. The fact base is viewed to be a Description Logic ABox (which is not interpreted with the CWA), and an ontology, also formulated in an appropriate DL, constrains the interpretations of unary and binary predicates, called concepts and roles in the DL community. As an example, assume that the ABox $\mathcal{A}$ contains the following assertions about the patient Bob:

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

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

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

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

The complexity of OBDA, i.e., the complexity of checking whether a given tuple of individuals is a certain answer of a conjunctive query in an ABox w.r.t. an ontology, has been investigated in detail for cases where the ontology is expressed in an appropriate DL and the query is a conjunctive query. One can either consider the *combined complexity*, which is measured in the size of the whole input (consisting of the query, the ontology, and the ABox), or the *data complexity*, which is measured in the size of the ABox only (i.e., the query and the ontology are assumed to be of constant size). The underlying assumption is that query and ontology are usually relatively small, whereas the size of the data may be huge. In the database setting (where there is no ontology and CWA is used), answering conjunctive queries is NP-complete w.r.t. combined complexity and in $\mathrm{AC}^0$ w.r.t. data complexity [8,1]. For expressive DLs, the complexity of checking certain answers is considerably higher. For instance, for the well-known DL $\mathcal{ALC}$, OBDA is ExpTime-complete w.r.t. combined complexity and co-NP-complete w.r.t. data complexity [7,13,6]. For this reason, more light-weight DLs have been developed, for which the data complexity of OBDA is still in $\mathrm{AC}^0$ and for which computing certain answers can be reduced to answering conjunctive queries in the database setting [5].

Unfortunately, OBDA as described until now is not sufficient to achieve situation awareness. The reason is that the situations we want to recognize may depend on states of the system at different time points. For example, assume that we want to find male patients that have a history of hypertension, i.e., patients that are male and at some previous time point had hypertension.[1] In order to express this kind of temporal queries, we propose to extend the well-known propositional temporal logic LTL [16] by allowing the use of conjunctive queries in place of propositional variables. For example, male patients with a history of hypertension can then be described by the query

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

where $\bigcirc^{-}$ stands for "previous" and $\Diamond^{-}$ stands for "sometime in the past." The query language obtained this way extends the temporal description logic $\mathcal{ALC}$-LTL introduced and investigated in [4]. In $\mathcal{ALC}$-LTL, only concept and role assertions (i.e., very restricted conjunctive queries without variables and existential quantification) can be used in place of propositional variables. As in [4], we also consider rigid concepts and roles, i.e., concepts and roles whose interpretation does not change over time. For example, we may want to assume that the concept *Male* is rigid, and thus a patient that is male now also has been male in the past and will stay male in the future.

Our overall setting for recognizing situations will thus be the following. In addition to a global ontology $\mathcal{T}$ (which describes properties of the system that hold at every time point, using the DL $\mathcal{ALC}$), we have a sequence of ABoxes $\mathcal{A}_0, \mathcal{A}_1, \ldots \mathcal{A}_n$, which (incompletely) describe the states of the system at the previous time points $0, 1, \ldots, n-1$ and the current time point $n$. The situation to be recognized is expressed by a temporal conjunctive query, as introduced above, which is evaluated w.r.t. the current time point $n$. We will investigate both the combined and the data complexity of this temporal extension of OBDA in three different settings: (i) both concepts and roles may be rigid; (ii) only concepts may be rigid; and (iii) neither concepts nor roles are allowed to be rigid. For the combined complexity, the obtained complexity results are identical to the ones for $\mathcal{ALC}$-LTL, though the upper bounds are considerably harder to show. For the data complexity, the results for the settings (ii) and (iii) coincides with the one for atemporal OBDA (co-NP-complete). For the setting (i), we can show that the data complexity is in ExpTime (in contrast to 2-ExpTime-completeness for the combined complexity), but we do not have a matching lower bound.

The details of the proofs can be found in the accompanying technical report [3].

## 2 Preliminaries

While in principle our temporal query language can be parameterized with any DL, in this paper we focus on $\mathcal{ALC}$ [20] as a prototypical expressive DL.

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

**Definition 2.1 (syntax of $\mathcal{ALC}$).** *Let $N_C$, $N_R$, and $N_I$, respectively, be non-empty, pairwise disjoint sets of* concept names, role names, *and* individual names. *The set of* concept descriptions *(or* concepts*) is the smallest set such that all concept names $A \in N_C$ are concepts, and if $C, D$ are concepts and $r \in N_R$, then $\neg C$ (negation), $C \sqcap D$ (conjunction), and $\exists r.C$ (existential restriction) are also concepts.*

*A* general concept inclusion (GCI) *is of the form $C \sqsubseteq D$, where $C, D$ are concepts, and an* assertion *is of the form $C(a)$ or $r(a, b)$, where $C$ is a concept, $r \in N_R$, and $a, b \in N_I$. We call both GCIs and assertions* axioms. *A* TBox *(or* ontology*) is a finite set of GCIs and an* ABox *is a finite set of assertions.*

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

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

*This function is extended to concept descriptions as follows: $(\neg C)^\mathcal{I} := \Delta^\mathcal{I} \setminus C^\mathcal{I}$; $(C \sqcap D)^\mathcal{I} := C^\mathcal{I} \cap D^\mathcal{I}$; and $(\exists r.C)^\mathcal{I} := \{d \in \Delta^\mathcal{I} \mid \exists e \in \Delta^\mathcal{I} : (d, e) \in r^\mathcal{I} \wedge e \in C^\mathcal{I}\}$. The interpretation $\mathcal{I}$ is a* model *of the GCI $C \sqsubseteq D$ if $C^\mathcal{I} \subseteq D^\mathcal{I}$, of the assertion $C(a)$ if $a^\mathcal{I} \in C^\mathcal{I}$, and of $r(a, b)$ if $(a^\mathcal{I}, b^\mathcal{I}) \in r^\mathcal{I}$. We write $\mathcal{I} \models \alpha$ if $\mathcal{I}$ is a model of the axiom $\alpha$, $\mathcal{I} \models \mathcal{T}$ if $\mathcal{I}$ is a model of all GCIs in the TBox $\mathcal{T}$, and $\mathcal{I} \models \mathcal{A}$ if $\mathcal{I}$ is a model of all assertions in the ABox $\mathcal{A}$.*

We assume that all interpretations $\mathcal{I}$ satisfy the *unique name assumption (UNA)*, i.e., for all $a, b \in N_\mathrm{I}$ with $a \neq b$ we have $a^\mathcal{I} \neq b^\mathcal{I}$. We now introduce a temporal query language that generalizes a subset of first-order queries called conjunctive queries [1,8] and the temporal DL $\mathcal{ALC}$-LTL [4]. In the following, we assume (as in [4]) that a subset of the concept and role names is designated as being rigid. The intuition is that the interpretation of the rigid names is not allowed to change over time. Let $N_\mathrm{RC}$ denote the *rigid concept names*, and $N_\mathrm{RR}$ the *rigid role names* with $N_\mathrm{RC} \subseteq N_\mathrm{C}$ and $N_\mathrm{RR} \subseteq N_\mathrm{R}$. We sometimes call the names in $N_\mathrm{C} \setminus N_\mathrm{RC}$ and $N_\mathrm{R} \setminus N_\mathrm{RR}$ *flexible.* As usual, all individual names are implicitly assumed to be rigid.

**Definition 2.3.** *A* temporal knowledge base *(TKB) $\mathcal{K} = \langle (\mathcal{A}_i)_{0 \leq i \leq n}, \mathcal{T} \rangle$ consists of a finite sequence of ABoxes $\mathcal{A}_i$ and a global TBox $\mathcal{T}$.*

*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 fixed non-empty domain $\Delta$ (*constant domain assumption*). Then $\mathfrak{I}$ is a* model *of $\mathcal{K}$ (written $\mathfrak{I} \models \mathcal{K}$) if (i) $\mathcal{I}_i \models \mathcal{A}_i$ for all $i, 0 \leq i \leq n$, (ii) $\mathcal{I}_i \models \mathcal{T}$ for all $i \geq 0$, and (iii) $\mathfrak{I}$ respects rigid names, i.e., $x^{\mathcal{I}_i} = x^{\mathcal{I}_j}$ for all $x \in N_I \cup N_{RC} \cup N_{RR}$ and all $i, j \geq 0$.*

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

**Definition 2.4.** *Let $N_V$ be a set of variables. A* conjunctive query (CQ) *is of the form $\phi = \exists y_1, \ldots, y_m.\psi$, where $y_1, \ldots, y_m \in N_V$ and $\psi$ is a finite conjunction* *of* atoms *of the form* $A(z)$ *for* $A \in N_C$ *and* $z \in N_V \cup N_I$ (concept atom)*; or* $r(z_1, z_2)$ *for* $r \in N_R$ *and* $z_1, z_2 \in N_V \cup N_I$ (role atom)*. The empty conjunction is denoted by* $\mathsf{true}$. Temporal conjunctive queries (TCQs) *are built from CQs using the constructors* $\neg\phi_1$ *(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 $\phi$ by $\mathsf{Ind}(\phi)$, the set of variables occurring in $\phi$ by $\mathsf{Var}(\phi)$, and the set of free variables occurring in $\phi$ by $\mathsf{FVar}(\phi)$. We call a TCQ $\phi$ with $\mathsf{FVar}(\phi) = \emptyset$ a *Boolean TCQ.* As usual, we use the following abbreviations: $\phi_1 \vee \phi_2$ (disjunction) for $\neg(\neg\phi_1 \wedge \neg\phi_2)$, $\Diamond\phi$ (eventually) for $\mathsf{true}\mathsf{U}\phi$, $\Box\phi$ (always) for $\neg\Diamond\neg\phi$, and analogously for the past: $\Diamond^-\phi$ for $\mathsf{true}\mathsf{S}\phi$, and $\Box^-\phi$ for $\neg\Diamond^-\neg\phi$. A *union of CQs* is a disjunction of CQs.

For our purposes, it is sufficient to define the semantics of CQs and TCQs only for Boolean queries. As usual, it is given using the notion of homomorphisms [8].

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

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

*We say that* $\mathcal{I}$ *is a* model *of* $\psi$ *(written* $\mathcal{I} \models \psi$*) if there is such a homomorphism. Let now* $\phi$ *be a Boolean TCQ. For an infinite sequence of interpretations* $\mathfrak{I} = (\mathcal{I}_i)_{i \geq 0}$ *and* $i \geq 0$*, we define* $\mathfrak{I}, i \models \phi$ *by induction on the structure of* $\phi$*:*

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

*Given a TKB* $\mathcal{K} = \langle (\mathcal{A}_i)_{0 \leq i \leq n}, \mathcal{T} \rangle$*, we say that* $\mathfrak{I}$ *is a* model *of* $\phi$ *w.r.t.* $\mathcal{K}$ *if* $\mathfrak{I} \models \mathcal{K}$ *and* $\mathfrak{I}, n \models \phi$*. We call* $\phi$ *satisfiable w.r.t.* $\mathcal{K}$ *if it has a model w.r.t.* $\mathcal{K}$*.*

It should be noted that Boolean TCQs generalize $\mathcal{ALC}$-LTL formulae as introduced in [4]. More precisely, every TCQ that contains only assertions instead of general CQs and contains no past operators ($\bigcirc^-$ or $\mathsf{S}$) is an $\mathcal{ALC}$-LTL formula. $\mathcal{ALC}$-LTL formulae may additionally contain local GCIs $C \sqsubseteq D$. Such a GCI can, however, be expressed by the TCQ $\neg\exists x.A(x)$ if we add the (global) GCIs $A \sqsubseteq C \sqcap \neg D$, $C \sqcap \neg D \sqsubseteq A$ to the TBox. Thus, TCQs together with a global TBox can express all $\mathcal{ALC}$-LTL formulae. TCQs are more expressive than $\mathcal{ALC}$-LTL formulae since CQs like $\exists y.r(y, y)$, which says that there is a loop in the model without naming the individual which has the loop, can clearly not be expressed in $\mathcal{ALC}$.

Before defining the main inference problem for TCQs to be investigated in this paper, we introduce some notation that will be used later on. The *propositional abstraction* $\widehat{\phi}$ of a TCQ $\phi$ is built by replacing each CQ occurring in $\phi$ by a propositional variable such that there is a 1–1 relationship between the CQs $\alpha_1, \dots, \alpha_m$ occurring in $\phi$ and the propositional variables $p_1, \dots, p_m$ occurring in $\widehat{\phi}$. The formula $\widehat{\phi}$ obtained this way is a propositional LTL-formula [16]. Recall that the semantics of propositional LTL is defined using the notion of an *LTL-structure*, which 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 time point $i \geq 0$ (written $\mathfrak{J}, i \models p_j$) iff $p_j \in w_i$. The satisfaction of a complex propositional LTL-formula by an LTL-structure is defined as in Definition 2.5.

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

## 3 The Entailment Problem

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

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

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

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

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

Therefore, it is sufficient to analyze the complexity of the *entailment problem*. We consider two kinds of complexity measures: combined complexity and data complexity. For the *combined complexity*, all parts of the input, i.e., the TCQ $\phi$ and the temporal knowledge base $\mathcal{K}$, are taken into account. For the *data complexity*, the TCQ $\phi$ and the TBox $\mathcal{T}$ are assumed to be constant, and the complexity is measured only w.r.t. the data, i.e., the sequence of ABoxes. As usual when investigating the data complexity of OBDA [5], we assume that the ABoxes occurring in a temporal knowledge base and the query contain only concept and role names that also occur in the global TBox.

It turns out that it is actually easier to analyze the complexity of the complement of this problem, i.e., *non-entailment* $\mathcal{K} \not\models \phi$. This problem has the same complexity as the *satisfiability problem*. In fact, $\mathcal{K} \not\models \phi$ iff $\neg\phi$ has a model w.r.t. $\mathcal{K}$, and conversely $\phi$ has a model w.r.t. $\mathcal{K}$ iff $\mathcal{K} \not\models \neg\phi$.

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

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

*Proof (Sketch).* The lower bounds easily follow from the known lower bounds for concept satisfiability in $\mathcal{ALC}$ w.r.t. TBoxes [19] and for the data complexity of query answering of Boolean CQs in $\mathcal{ALC}$ [6]. To check whether there is an interpretation $\mathcal{I}$ with $\mathcal{I} \models \mathcal{K}$ and $\mathcal{I} \models \phi$, we reduce this problem to a query non-entailment problem of known complexity. First, we instantiate the non-negated CQs in $\phi$ by omitting the existential quantifiers and replacing the variables by fresh individual names. The set $\mathcal{A}'$ of the resulting atoms can thus be viewed as an additional ABox that restricts the interpretation $\mathcal{I}$. The above problem is thus equivalent to finding an interpretation $\mathcal{I}$ with $\mathcal{I} \models \langle \mathcal{A} \cup \mathcal{A}', \mathcal{T} \rangle$ and $\mathcal{I} \not\models \rho$, where $\rho$ is the union of Boolean CQs that results from negating the conjunction of all negated CQs in $\phi$. This is the same as asking whether the knowledge base $\langle \mathcal{A} \cup \mathcal{A}', \mathcal{T} \rangle$ does not entail the union of conjunctive queries $\rho$. The complexity of this kind of entailment problems is known: it is ExpTime-complete w.r.t. combined complexity [7,13] and co-NP-complete w.r.t. data complexity [15]. □

We now describe an approach to solving the satisfiability problem (and thus the non-entailment problem) in general. The basic idea is to reduce the problem to two separate satisfiability problems, similar to what was done for $\mathcal{ALC}$-LTL in Lemma 4.3 of [4]. Let $\mathcal{K} = \langle (\mathcal{A}_i)_{0 \leq i \leq n}, \mathcal{T} \rangle$ be a TKB and $\phi$ be a Boolean TCQ, for which we want to decide whether $\phi$ has a model w.r.t. $\mathcal{K}$. Recall that the propositional abstraction $\widehat{\phi}$ of $\phi$ contains the propositional variables $p_1, \ldots, p_m$ in place of the CQs $\alpha_1, \ldots, \alpha_m$ occurring in $\phi$. We assume in the following that $\alpha_i$ was replaced by $p_i$ for all $i$, $1 \leq i \leq m$. We now consider a set $\mathcal{S} \subseteq 2^{\{p_1,\ldots,p_m\}}$, which intuitively specifies the worlds that are allowed to occur in an LTL-structure satisfying $\widehat{\phi}$. To express this restriction, we define the propositional LTL-formula

$$\widehat{\phi}_{\mathcal{S}} := \widehat{\phi} \wedge \Box^{-}\Box \left( \bigvee_{X \in \mathcal{S}} \left( \bigwedge_{p \in X} p \wedge \bigwedge_{p \notin X} \neg p \right) \right).^{2}$$

If $\phi$ has a model w.r.t. $\mathcal{K}$, i.e., there is a sequence of interpretations $\mathfrak{I} = (\mathcal{I}_i)_{i \geq 0}$ that respects rigid names, is a model of $\mathcal{K}$, and satisfies $\mathfrak{I}, n \models \phi$, then there exist a set $\mathcal{S} \subseteq 2^{\{p_1,\ldots,p_m\}}$ and a propositional LTL-structure that satisfies $\widehat{\phi}_{\mathcal{S}}$ at time point $n$. In fact, for each interpretation $\mathcal{I}_i$ of $\mathfrak{I}$, we set $X_i := \{p_j \mid 1 \leq j \leq m \text{ and } \mathcal{I}_i \text{ satisfies } \alpha_j\}$, and then take $\mathcal{S} := \{X_i \mid i \geq 0\}$. We say that $\mathcal{S}$ is *induced* by $\mathfrak{I}$. The fact that $\mathfrak{I}$ satisfies $\phi$ at time point $n$ implies that its propositional abstraction satisfies $\widehat{\phi}_{\mathcal{S}}$ at time point $n$, where the *propositional abstraction* $\widehat{\mathfrak{I}} = (w_i)_{i \geq 0}$ of $\mathfrak{I}$ is defined by $w_i := X_i$ for all $i \geq 0$. However, guessing a set $\mathcal{S}$ and then testing whether the induced LTL-formula $\widehat{\phi}_{\mathcal{S}}$ is satisfiable at time point $n$ is not sufficient for checking whether $\phi$ has a model w.r.t. $\mathcal{K}$. We must also check whether the guessed set $\mathcal{S}$ can indeed be induced by some sequence of interpretations that is a model of $\mathcal{K}$. The following definition introduces a condition that needs to be satisfied for this to hold.

**Definition 3.3.** *Given a set $\mathcal{S} = \{X_1, \ldots, X_k\} \subseteq 2^{\{p_1,\ldots,p_m\}}$ and a mapping $\iota\colon \{0, \ldots, n\} \to \{1, \ldots, k\}$, we say that $\mathcal{S}$ is* r-consistent *w.r.t. $\iota$ and $\mathcal{K}$ if there exist interpretations $\mathcal{J}_1, \ldots, \mathcal{J}_k$, $\mathcal{I}_0, \ldots, \mathcal{I}_n$ such that*

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

The intuition underlying this definition is the following. The existence of the interpretation $\mathcal{J}_i$ ($1 \leq i \leq k$) ensures that the conjunction $\chi_i$ of the CQ-literals specified by $X_i$ is consistent. In fact, a set $\mathcal{S}$ containing a set $X_i$ for which this does not hold cannot be induced by a sequence of interpretations. The interpretations $\mathcal{I}_i$ ($0 \leq i \leq n$) are supposed to constitute the first $n+1$ interpretations in such a sequence. In addition to inducing a set $X_{\iota(i)} \in \mathcal{S}$ and thus satisfying the corresponding conjunction $\chi_{\iota(i)}$, the interpretation $\mathcal{I}_i$ must thus also satisfy the ABox $\mathcal{A}_i$. The first and the second condition ensure that a sequence of interpretations built from $\mathcal{J}_1, \ldots, \mathcal{J}_k$, $\mathcal{I}_0, \ldots, \mathcal{I}_n$ respects rigid names and satisfies the global TBox $\mathcal{T}$. Note that we can use Theorem 3.2 to check whether interpretations satisfying the last three conditions of Definition 3.3 exist.

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

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

As we will see below, the difficulty lies in ensuring that they also satisfy the first condition.

Satisfaction of the temporal structure of $\phi$ by a sequence of interpretations built this way is ensured by testing $\widehat{\phi}_{\mathcal{S}}$ for satisfiability, which can basically be done using algorithms for testing satisfiability in propositional LTL [23].

**Lemma 3.4.** *The TCQ $\phi$ has a model w.r.t. the TKB $\mathcal{K}$ iff there is 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*

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

The proof of this lemma is similar to, but more involved than the proof of a similar characterization for satisfiability in $\mathcal{ALC}$-LTL [4].

As shown later, the overall complexity of the satisfiability problem depends on which symbols are allowed to be rigid. To achieve these complexity results, we obtain the set $\mathcal{S}$ and the function $\iota$ either by enumeration, guessing, or direct construction, depending on the case under consideration. Given $\mathcal{S}$ and $\iota$, it remains to check the two conditions of the lemma. To check the second condition, we construct a Büchi automaton similar to the standard construction for satisfiability of LTL-formulae [23]. Emptiness of this automaton is equivalent to satisfiability of $\widehat{\phi}_{\mathcal{S}}$. The details can be found in [3].

The main difference to the standard construction is the additional condition $w_i = X_{\iota(i)}$ for $i$, $0 \leq i \leq n$. We check this by attaching a counter taking values from $\{0, \dots, n+1\}$ to the states of the automaton. Transitions where the counter is $i < n+1$ check if the current world corresponds to $X_{\iota(i)}$ and increase the counter by 1. At $i = n$, we ensure that $\widehat{\phi}_{\mathcal{S}}$ is satisfied. Similar to what is done in [4], we do not construct the automaton directly for $\widehat{\phi}_{\mathcal{S}}$, which would yield an automaton of double-exponential size in the size of $\phi$, but rather for $\widehat{\phi}$. The additional restrictions of $\widehat{\phi}_{\mathcal{S}}$ are enforced by restricting this automaton to states that satisfy a world from $\mathcal{S}$. The size of the constructed automaton only depends linearly on the number $n$ of input ABoxes, which is important for the results about data complexity, and exponentially on the size of $\phi$. Furthermore, emptiness of Büchi automata can be checked in polynomial time in the size of the automaton [23].

**Lemma 3.5.** *Given a set $\mathcal{S} = \{X_1, \dots, X_k\} \subseteq 2^{\{p_1,\dots,p_m\}}$ and a mapping $\iota\colon \{0, \dots, n\} \to \{1, \dots, k\}$, the problem of deciding the existence of an LTL-structure $\mathfrak{J} = (w_i)_{i \geq 0}$ such that $\mathfrak{J}, n \models \widehat{\phi}_{\mathcal{S}}$ and $w_i = X_{\iota(i)}$ for all $i$, $0 \leq i \leq n$, is in* ExpTime *w.r.t. combined complexity and in* P *w.r.t. data complexity.*

For the r-consistency test, we need to use different constructions depending on which symbols are allowed to be rigid. Using these constructions, we obtain the complexity results for the entailment problem shown in Table 1. Note that rigid concept names can be simulated by rigid role names [4], which is why there are only three cases to consider. The lower bounds can be obtained by

**Table 1.** The complexity of the entailment problem for TCQs.

| | *Data complexity* | *Combined complexity* |
|---|---|---|
| $N_{\mathrm{RC}} = N_{\mathrm{RR}} = \emptyset$ | co-NP-complete | ExpTime-complete |
| $N_{\mathrm{RC}} \neq \emptyset$, $N_{\mathrm{RR}} = \emptyset$ | co-NP-complete | co-NExpTime-complete |
| $N_{\mathrm{RR}} \neq \emptyset$ | co-NP-hard/in ExpTime | 2-ExpTime-complete |

simple reductions from the atemporal entailment problem [6] and the satisfiability problem of $\mathcal{ALC}$-LTL [4]. In the following sections, we only present the ideas for the upper bounds in the most interesting case (no rigid role names, but rigid concept names). For the other two cases, the proofs are quite similar to the ones for $\mathcal{ALC}$-LTL [4]. For rigid concepts, the proofs still follow the lines of the proofs in [4], but need considerably more effort to deal with CQs instead of assertions (see [3] for more details).

## 4 Data Complexity for the Case of Rigid Concepts

To obtain an upper bound for the data complexity of the non-entailment problem in the case where $N_{\mathrm{RC}} \neq \emptyset$ and $N_{\mathrm{RR}} = \emptyset$, we consider the conditions of Lemma 3.4 in more detail. First, note that, since $\mathcal{S} \subseteq 2^{\{p_1,\dots,p_m\}}$ is of constant size w.r.t. the input ABoxes and $\iota\colon \{0,\dots,n\} \to \{1,\dots,k\}$ is of size linear in $n$ (the number of ABoxes), guessing $\mathcal{S}$ and $\iota$ can be done in NP. Additionally, according to Lemma 3.5, LTL-satisfiability can be tested in P.

We now show that the r-consistency of $\mathcal{S}$ w.r.t. $\iota$ and $\mathcal{K}$ can be checked in NP, which yields the desired data complexity of co-NP for the entailment problem. We use a renaming technique similar to the one employed in [4]. For every $i$, $1 \leq i \leq k$, and every *flexible* concept name $A$ (every role name $r$) occurring in $\phi$ or in $\mathcal{T}$, we introduce a copy $A^{(i)}$ ($r^{(i)}$), which is a fresh concept (role) name. We call $A^{(i)}$ ($r^{(i)}$) the $i$-th copy of $A$ ($r$). The CQ $\alpha^{(i)}$ (the GCI $\beta^{(i)}$) is obtained from a CQ $\alpha$ (a GCI $\beta$) by replacing every occurrence of a flexible name by its $i$-th copy. Similarly, for $1 \leq \ell \leq k$, the conjunction $\chi_\ell^{(i)}$ is obtained from $\chi_\ell$ (see Definition 3.3) by replacing each CQ $\alpha_j$ by $\alpha_j^{(i)}$.

The basic idea is to decide the existence of models of the conjunctions of CQ-literals $\gamma_i \wedge \chi_{\mathcal{S}}$ w.r.t. the TBox $\mathcal{T}_{\mathcal{S}}$, where

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

One can see from the proof of Theorem 3.2 that this problem can be decided in NP in the size of the input ABoxes. The main reason is that the negated CQs do not depend on the input ABoxes. In fact, negated CQs only occur in $\chi_{\mathcal{S}}$, which only depends on the query $\phi$. Thus, the union of CQs $\rho$ constructed in the proof of Theorem 3.2 does not depend on the input ABoxes and the same is true for the TBox $\mathcal{T}_{\mathcal{S}}$.

However, for r-consistency we have to make sure that rigid consequences of the form $A(a)$ for a rigid concept name $A$ and an individual name $a$ are shared between these conjunctions of CQ-literals. Let $\mathsf{RCon}(\mathcal{T})$ denote the rigid concept names occurring in $\mathcal{T}$. Similar to what was done in Lemma 6.3 of [4], we now guess a set $\mathcal{D} \subseteq 2^{\mathsf{RCon}(\mathcal{T})}$ and a mapping $\tau\colon \mathsf{Ind}(\phi) \cup \mathsf{Ind}(\mathcal{K}) \to \mathcal{D}$. The idea is that $\mathcal{D}$ fixes the combinations of rigid concept names that occur in the models of $\gamma_i \wedge \chi_{\mathcal{S}}$ and $\tau$ assigns to each individual name one such combination. Note that $\mathcal{D}$ only depends on $\mathcal{T}$ and $\tau$ is of size linear in the size of the input ABoxes, which is why we can guess $\mathcal{D}$ and $\tau$ in NP w.r.t. data complexity. We now define

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

where $A'$ is a rigid concept name that is equivalent to $\neg A$ in $\mathcal{T}$.[4] Note that $\chi_\tau$ is of polynomial size w.r.t. the size of the input ABoxes.

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

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

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

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

*Proof (Sketch).* For the "if" direction, assume that $\mathcal{I}_i$ are the required models for $\gamma_i \wedge \chi_{\mathcal{S}} \wedge \chi_\tau$. Similar to the proof of Lemma 6.3 in [4], we can assume w.l.o.g. that their domains $\Delta_i$ are countably infinite and for each $Y \in \mathcal{D}$ there are countably infinitely many elements $d \in (C_Y)^{\mathcal{I}_i}$. This is a consequence of the Löwenheim-Skolem theorem and the fact that the countably infinite disjoint union of $\mathcal{I}_i$ with itself is again a model of $\gamma_i \wedge \chi_{\mathcal{S}} \wedge \chi_\tau$. The latter follows from the observation that for any CQ there is a homomorphism into $\mathcal{I}_i$ iff there is a homomorphism into the disjoint union of $\mathcal{I}_i$ with itself.

Consequently, we can partition the domains $\Delta_i$ into the countably infinite sets $\Delta_i(Y) := \{d \in \Delta_i \mid d \in (C_Y)^{\mathcal{I}_i}\}$ for $Y \in \mathcal{D}$. It is now easy to see that the domains $\Delta_i$ are essentially the same up to isomorphisms between $\Delta_i$ and $\Delta_j$ for $0 \leq i, j \leq n$ that relate the elements of $\Delta_i(Y)$ to those of $\Delta_j(Y)$, and respect the individual names, i.e., map each $a^{\mathcal{I}_i}$ to $a^{\mathcal{I}_j}$. We can now construct the models required by Definition 3.3 from the models $\mathcal{I}_i$ by appropriately relating the flexible names and their copies. For example, interpreting the rigid concept names as in $\mathcal{I}_i$ and the flexible names as their $\iota(i)$-th copies in $\mathcal{I}_i$ yields a model

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

of $\chi_{\iota(i)}$ w.r.t. $\langle \mathcal{A}_i, \mathcal{T} \rangle$, and similarly for the models of $\chi_j$ and $\mathcal{T}$ for $1 \leq j \leq k$. These models share the same domain and respect rigid names. Note that the interpretation of the names in $N_{\mathrm{RC}} \setminus \mathsf{RCon}(\mathcal{T})$ and $N_{\mathrm{I}} \setminus (\mathsf{Ind}(\phi) \cup \mathsf{Ind}(\mathcal{K}))$ is irrelevant and can be fixed arbitrarily.

For the "only if" direction, it is easy to see that one can combine the interpretations $\mathcal{I}_i$, $\mathcal{J}_1$, ..., $\mathcal{J}_k$ from Definition 3.3 to a model $\mathcal{I}_i'$ of $\gamma_i \wedge \chi_{\mathcal{S}}$ w.r.t. $\mathcal{T}_{\mathcal{S}}$ by interpreting the $j$-th copy of a flexible name as the original name in $\mathcal{J}_j$. For $a \in \mathsf{Ind}(\phi) \cup \mathsf{Ind}(\mathcal{K})$, we define $\tau(a) := Y \subseteq \mathsf{RCon}(\mathcal{T})$ iff $a \in (C_Y)^{\mathcal{I}_0}$. Furthermore, we let $\mathcal{D}$ contain all those sets $Y \subseteq \mathsf{RCon}(\mathcal{T})$ such that there is a $d \in (C_Y)^{\mathcal{I}_i'}$ for some $0 \leq i \leq n$. To obtain models of $\gamma_i \wedge \chi_{\mathcal{S}} \wedge \chi_{\tau}$ w.r.t. $\mathcal{T}_{\mathcal{S}}$ that respect $\mathcal{D}$, we still need to ensure that all $Y \in \mathcal{D}$ are represented in each of the models $\mathcal{I}_i'$. To do this, we construct the disjoint union $\mathcal{I}_i''$ of $\mathcal{I}_i'$ with all other $\mathcal{I}_j'$ for $0 \leq j \leq n$. It remains to show that this interpretation is still a model of $\mathcal{T}_{\mathcal{S}}$ and the conjunction $\gamma_i \wedge \chi_{\mathcal{S}} \wedge \chi_{\tau}$. This can be seen as follows. For the non-negated CQs in this conjunction, clearly there is a homomorphism into $\mathcal{I}_i''$ if there is one into $\mathcal{I}_i'$. For the negated CQs in $\chi_{\mathcal{S}}$, we need the additional assumption that each of them is *connected* in the sense that the variables and individual names are related by roles (see [18] or [3] for an exact definition). It follows from a result in [21] that this is without loss of generality (see [3]). Given this assumption, the non-existence of a homomorphism into any of the components of $\mathcal{I}_i''$ clearly implies the non-existence of a homomorphism into their disjoint union $\mathcal{I}_i''$. $\square$

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

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

where $A_Y$ and $A_{\mathcal{D}}$ are fresh concept names that are restricted by adding the GCIs $A_Y \sqsubseteq C_Y$, $C_Y \sqsubseteq A_Y$ for each $Y \in \mathcal{D}$, and $A_{\mathcal{D}} \sqsubseteq \bigsqcap_{Y \in \mathcal{D}} \neg A_Y$, $\bigsqcap_{Y \in \mathcal{D}} \neg A_Y \sqsubseteq A_{\mathcal{D}}$ to $\mathcal{T}_{\mathcal{S}}$. We call the resulting TBox $\mathcal{T}_{\mathcal{S}}'$. Since $\chi_{\mathcal{D}}$ and $\mathcal{T}_{\mathcal{S}}'$ do not depend on the input ABoxes, by Theorem 3.2 we can check the consistency of $\gamma_i \wedge \chi_{\mathcal{S}} \wedge \chi_{\tau} \wedge \chi_{\mathcal{D}}$ w.r.t. $\mathcal{T}_{\mathcal{S}}'$ in NP w.r.t. data complexity.

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

## 5 Combined Complexity for the Case of Rigid Concepts

Unfortunately, the approach used in the previous section does not yield a *combined complexity* of co-NExpTime. The reason is that the conjunctions $\chi_{\mathcal{S}}$ and $\chi_{\mathcal{D}}$ are of exponential size in the size of $\phi$, and thus Theorem 3.2 only yields an upper bound of 2-ExpTime. In this section, we describe a different approach with a combined complexity of co-NExpTime.

As a first step, we rewrite the Boolean TCQ $\phi$ into a Boolean TCQ $\psi$ of linear size in the size of $\phi$ and $\mathcal{K}$ such that answering $\phi$ at time point $n$ is equivalent to answering $\psi$ at time point 0 w.r.t. a trivial sequence of ABoxes. This is done by compiling the ABoxes into the query and postponing the query $\phi$ using the $\bigcirc$-operator (see [3] for details). We can thus focus on deciding whether a Boolean TCQ $\phi$ has a model w.r.t. a TKB $\mathcal{K} = \langle \emptyset, \mathcal{T} \rangle$ that has only one empty ABox in the sequence. Note that this compilation approach does not allow us to obtain a low *data complexity* for the entailment problem since after encoding the ABoxes into $\phi$ the size of $\chi_{\mathcal{S}}$ is exponential in the size of the ABoxes.

We now again analyze how to check the two conditions in Lemma 3.4. First, observe that guessing $\mathcal{S} = \{X_1, \ldots, X_k\} \subseteq 2^{\{p_1,\ldots,p_m\}}$ can be done in non-deterministic exponential time in the size of $\phi$. Furthermore, by Lemma 3.5, the LTL-satisfiability test required by the second condition can be realized in ExpTime. It remains to determine the complexity of testing r-consistency of $\mathcal{S}$ w.r.t. $\mathcal{K} = \langle \emptyset, \mathcal{T} \rangle$. Similarly to the approach used in the previous section and to the proof of Lemma 6.3 in [4], we start by guessing a set $\mathcal{D} \subseteq 2^{\mathsf{RCon}(\mathcal{T})}$ and a mapping $\tau \colon \mathsf{Ind}(\phi) \to \mathcal{D}$. Since $\mathcal{D}$ is of size exponential in $\mathcal{T}$ and $\tau$ is of size polynomial in the size of $\phi$ and $\mathcal{T}$, guessing $\mathcal{D}$ and $\tau$ can also be done in NExpTime. By Lemma 4.1, it suffices to test whether $\chi_{\mathcal{S}} \wedge \chi_\tau$ has a model w.r.t. $\mathcal{T}_{\mathcal{S}}$ that respects $\mathcal{D}$. Instead of applying Theorem 3.2 directly to this problem, which would yield a complexity of 2-ExpTime, we split the problem into separate sub-problems for each component $\chi_i$ of $\chi_{\mathcal{S}}$. The correctness of this approach is stated in the next lemma. For the special case of $\mathcal{ALC}$-LTL, this was shown in Lemma 6.3 in [4]. The proof for the general case is similar to the proof of Lemma 4.1 above.

**Lemma 5.1.** *If $N_{RC} \neq \emptyset$ and $N_{RR} = \emptyset$, then $\mathcal{S}$ is r-consistent w.r.t. $\mathcal{K} = \langle \emptyset, \mathcal{T} \rangle$ iff there exist $\mathcal{D} \subseteq 2^{\mathsf{RCon}(\mathcal{T})}$ and $\tau \colon \mathsf{Ind}(\phi) \to \mathcal{D}$ such that each of the conjunctions $\widehat{\chi}_i := \chi_i \wedge \chi_\tau$, $1 \leq i \leq k$, has a model w.r.t. $\mathcal{K}$ that respects $\mathcal{D}$.*

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

It thus suffices to show that we can decide query non-entailment $\langle \mathcal{A}, \mathcal{T} \rangle \not\models \rho$ w.r.t. $\mathcal{D}$ in time exponential in the size of $\mathcal{A}$, $\mathcal{T}$, and $\rho$. To this purpose, we further reduce this problem following an idea from [13]. There, the notion of a *spoiler* is introduced. A spoiler is an $\mathcal{ALC}^\cap$-knowledge base that states properties that must be satisfied such that a query is not entailed by a knowledge base.[5] It is shown that $\langle \mathcal{A}, \mathcal{T} \rangle \not\models \rho$ iff there is a spoiler $\langle \mathcal{A}', \mathcal{T}' \rangle$ for $\langle \mathcal{A}, \mathcal{T} \rangle$ such that $\langle \mathcal{A} \cup \mathcal{A}', \mathcal{T} \cup \mathcal{T}' \rangle$ is consistent. Additionally, all spoilers can be computed in time exponential in the size of $\langle \mathcal{A}, \mathcal{T} \rangle$ and $\rho$, and each spoiler is of polynomial size.

[5] $\mathcal{ALC}^\cap$ extends $\mathcal{ALC}$ by role conjunctions of the form $r_1 \cap \cdots \cap r_n$ for $r_1, \ldots, r_n \in N_{\mathrm{R}}$.

We show in [3] that the above reduction is still correct in the presence of $\mathcal{D}$, i.e., we have $\langle\mathcal{A},\mathcal{T}\rangle \not\models \rho$ w.r.t. $\mathcal{D}$ iff there is a spoiler $\langle\mathcal{A}',\mathcal{T}'\rangle$ for $\langle\mathcal{A},\mathcal{T}\rangle$ such that there is a model of $\langle\mathcal{A}\cup\mathcal{A}',\mathcal{T}\cup\mathcal{T}'\rangle$ that respects $\mathcal{D}$. It now remains to show that the existence of such a model can be checked in exponential time in the size of $\langle\mathcal{A}\cup\mathcal{A}',\mathcal{T}\cup\mathcal{T}'\rangle$, and therefore in exponential time in the size of $\phi$ and $\mathcal{T}$.

For classical $\mathcal{ALC}^{\cap}$-knowledge bases, the consistency problem (without $\mathcal{D}$) is ExpTime-complete [22]. The complexity does not increase for checking the existence of a model of a Boolean $\mathcal{ALC}^{\cap}$-knowledge base that respects $\mathcal{D}$.[6] We show this in [3] using a notion of *quasimodels* similar to the one in [4], but extended to deal with role conjunctions. The main difference is that we must introduce additional concept names that function as so-called *pebbles*, which mark elements that have specific role predecessors, an idea borrowed from [9,10,14].

**Lemma 5.2.** *Let $\mathcal{B}$ be a Boolean $\mathcal{ALC}^{\cap}$-knowledge base of size $n$, $A_1,\dots,A_k$ be concept names occurring in $\mathcal{B}$, and $\mathcal{D} \subseteq 2^{\{A_1,\dots,A_k\}}$. Then the existence of a model of $\mathcal{B}$ that respects $\mathcal{D}$ can be decided in time exponential in $n$.*

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

**Theorem 5.3.** *If $N_{RC} \neq \emptyset$ and $N_{RR} = \emptyset$, then the entailment problem is in co-*NExpTime *w.r.t. combined complexity.*

## 6 Conclusions

We have introduced a new temporal query language that extends the temporal DL $\mathcal{ALC}$-LTL to using conjunctive queries as atoms. Our complexity results on the entailment problem for such queries w.r.t. temporal knowledge bases are summarized in Table 1. Without any rigid names, we observed that entailment of TCQs is as hard as entailment of CQs w.r.t. atemporal $\mathcal{ALC}$-knowledge bases, i.e., in this case adding temporal operators to the query language does not increase the complexity. However, if we allow for rigid concept names (but no rigid role names), the picture changes. While the data complexity remains the same as in the atemporal case, the combined complexity of query entailment increases to co-NExpTime, i.e., the non-entailment problem is as hard as satisfiability in $\mathcal{ALC}$-LTL. If we further add rigid role names, the combined complexity (of non-entailment) again increases in accordance with the complexity of satisfiability in $\mathcal{ALC}$-LTL. For data complexity, it is still unclear whether adding rigid role names results in an increase. We have shown an upper bound of ExpTime (which is one exponential better than the combined complexity), but the only lower bound we have is the trivial one of co-NP.

Further work will include trying to close this gap. Moreover, it would be interesting to consider temporal queries based on inexpressive DLs such as DL-Lite [5], and check under what conditions query answering can be realized using classical (temporal or atemporal) database techniques.

[6] Boolean knowledge bases generalize ABoxes and TBoxes by allowing arbitrary Boolean combinations of axioms instead of only conjunctions.

## References

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley (1995)
2. Baader, F., Bauer, A., Baumgartner, P., Cregan, A., Gabaldon, A., Ji, K., Lee, K., Rajaratnam, D., Schwitter, R.: A novel architecture for situation awareness systems. In: Proc. TABLEAUX'09. LNCS, vol. 5607 (2009)
3. Baader, F., Borgwardt, S., Lippmann, M.: On the complexity of temporal query answering. LTCS-Report 13-01, Technische Universität Dresden, Germany (2012), see `http://lat.inf.tu-dresden.de/research/reports.html`.
4. Baader, F., Ghilardi, S., Lutz, C.: LTL over description logic axioms. ACM Trans. Comput. Log. 13(3) (2012)
5. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., Rodriguez-Muro, M., Rosati, R.: Ontologies and databases: The DL-Lite approach. In: RW'09, LNCS, vol. 5689 (2009)
6. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Data complexity of query answering in description logics. In: Proc. KR'06 (2006)
7. Calvanese, D., De Giacomo, G., Lenzerini, M.: On the decidability of query containment under constraints. In: Proc. PODS'98 (1998)
8. Chandra, A.K., Merlin, P.M.: Optimal implementation of conjunctive queries in relational data bases. In: Proc. STOC'77 (1977)
9. Danecki, R.: Nondeterministic propositional dynamic logic with intersection is decidable. In: Proc. SCT'84. LNCS, vol. 208 (1984)
10. De Giacomo, G., Massacci, F.: Combining deduction and model checking into tableaux and algorithms for Converse-PDL. Inform. Comput. 162(1–2) (2000)
11. Decker, S., Erdmann, M., Fensel, D., Studer, R.: Ontobroker: Ontology based access to distributed and semi-structured information. In: Proc. DS'99 (1999)
12. Endsley, M.R.: Toward a theory of situation awareness in dynamic systems. Human Factors 37(1) (1995)
13. Lutz, C.: The complexity of conjunctive query answering in expressive description logics. In: Proc. IJCAR'08. LNCS, vol. 5195 (2008)
14. Massacci, F.: Decision procedures for expressive description logics with intersection, composition, converse of roles and role identity. In: Proc. IJCAI 2001 (2001)
15. Ortiz, M., Calvanese, D., Eiter, T.: Characterizing data complexity for conjunctive query answering in expressive description logics. In: Proc. AAAI'06 (2006)
16. Pnueli, A.: The temporal logic of programs. In: Proc. FOCS'77 (1977)
17. Poggi, A., Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Linking data to ontologies. J. Data Sem. X (2008)
18. Rudolph, S., Glimm, B.: Nominals, inverses, counting, and conjunctive queries or: Why infinity is your friend! J. Artif. Intell. Res. 39(1) (2010)
19. Schild, K.: A correspondence theory for terminological logics: Preliminary report. In: Proc. IJCAI'91 (1991)
20. Schmidt-Schauß, M., Smolka, G.: Attributive concept descriptions with complements. Artif. Intell. 48(1) (1991)
21. Tessaris, S.: Questions and Answers: Reasoning and Querying in Description Logic. Ph.D. thesis, University of Manchester (2001)
22. Tobies, S.: Complexity Results and Practical Algorithms for Logics in Knowledge Representation. Ph.D. thesis, RWTH Aachen (2001)
23. Vardi, M.Y., Wolper, P.: Reasoning about infinite computations. Inform. Comput. 155(1) (1994)