Temporal Query Answering in the Description Logic $\mathcal{EL}$

Stefan Borgwardt and Veronika Thost
Technische Universität Dresden, Germany
firstname.lastname@tu-dresden.de

Abstract
Context-aware systems use data collected at runtime to recognize certain predefined situations and trigger adaptations. This can be implemented using ontology-based data access (OBDA), which augments classical query answering in databases by adopting the open-world assumption and including domain knowledge provided by an ontology. We investigate temporalized OBDA w.r.t. ontologies formulated in $\mathcal{EL}$, a description logic that allows for efficient reasoning and is successfully used in practice. We consider a recently proposed temporalized query language that combines conjunctive queries with the operators of propositional linear temporal logic (LTL), and study both data and combined complexity of query entailment in this setting. We also analyze the satisfiability problem in the similar formalism $\mathcal{EL}$-LTL.

1 Introduction
Context-aware systems use data collected at runtime to recognize certain predefined situations and trigger adaptations. For example, an operating system might be able to recognize that a video application is out of user focus (e.g., by corresponding sensors) and then adapt application parameters to optimize the energy consumption of the system. A straightforward approach is to encode the situations into queries over a database containing the sensor data. However, in general, sensors do not completely describe the environment (e.g., to date, sensors cannot capture the intentions of users), and usually additional knowledge about the behavior of the environment is available. For example, if the user has not been watching the video for a longer period of time because he is using another application on a second screen, then the video does not need to be displayed in the highest resolution.

Ontology-based data access [Poggi et al., 2008; Decker et al., 1998] remedies this situation by adopting the open-world assumption, where facts not present in the data are assumed to be unknown rather than false, and by employing an ontology to encode background knowledge. This is done using axioms of an appropriate ontology language, for example a description logic (DL) [Baader et al., 2003]. In this paper, we focus on ontologies in the lightweight DL $\mathcal{EL}$, which allows for efficient reasoning [Baader et al., 2005; Lutz et al., 2009] and is successfully applied in practice (e.g., in large biomedical ontologies like SNOMED CT). In this setting, the data is collected into a fact base (or ABox) containing assertions about individuals using unary and binary predicates, called concepts and roles, respectively. Thus, we can represent both static knowledge about active applications as well as dynamic knowledge about the current context: VideoApplication(app1), NotWatching(user1). Background knowledge is represented in the ontology (or TBox) using so-called general concept inclusions (GCIs) like

\[
\text{VideoApplication} \sqcap \exists_{\text{hasUser}} \text{NotWatching} \sqsubseteq \exists_{\text{hasState}} \text{OutOfFocus},
\]
saying that a video application whose user is currently not watching the video is out of user focus. ABox and TBox together are called knowledge base. We can use a conjunctive query (CQ) like $\psi(x) := \exists y. \text{hasState}(x, y) \wedge \text{OutOfFocus}(y)$ over this knowledge base to identify applications $x$ that can potentially be assigned a lower priority. However, complex situations typically depend also on the behavior of the environment in the past. For example, the operating system should not switch between configurations every time the user is not watching for one second, but only after this has been the case for a longer period of time.

For that reason, we investigate temporal conjunctive queries (TCQs), originally proposed in [Baader et al., 2013; 2015]. They allow to combine conjunctive queries via the operators of the propositional linear temporal logic LTL [Pnueli, 1977; Lichtenstein et al., 1985]. Hence, we can use the TCQ

\[
(\neg (\exists y. \text{GotPriority}(y) \wedge \neg \text{Equal}(x, y)) \wedge \text{SGotPriority}(x))
\]

to obtain all applications that were out of user focus during the three previous ($\neg$) moments of observation, were prioritized by the operating system at some point in time, and the priority has not ($\neg$) changed since ($\neg$) then.\footnote{http://www.ihtsdo.org/snomed-ct/}

The semantics of TCQs is based on temporal knowledge bases (TKBs), which, in addition to the background knowledge (which is assumed to hold globally, i.e., at every point in

\footnote{Although our formalism does not support it yet, priority values can be represented if the underlying DL allows for so-called concrete domains.}
time), contains a sequence of ABoxes $A_0, A_1, \ldots, A_n$, representing the data collected at specific points in time. We designate with $n$ the most recent time of observation (the current time point), at which the situation recognition is performed.

We also investigate the related temporalized formalism $\mathcal{EL}$-LTL, in which axioms, i.e., assertions or GCIs, are combined using LTL-operators. This approach was first suggested in [Baader et al., 2012]. In our setting, the axioms in the TKB do not explicitly refer to temporal information, but are written in a classical (atemporal) DL; only the query is temporalized. In contrast, [Lutz et al., 2008; Artale et al., 2007; 2014; Gutiérrez-Basulto et al., 2014] extend classical DLs by temporal operators, which then occur within the knowledge base. However, most of these logics yield high reasoning complexities, even if the underlying atemporal DL has tractable reasoning problems. Lower complexities are only obtained by either considerably restricting the set of temporal operators or the underlying DL.

Regarding temporal properties formulated over atemporal DLs, $\mathcal{ALC}$-LTL, a variant of $\mathcal{EL}$-LTL over the more expressive DL $\mathcal{ACC}$, was first considered in [Baader et al., 2012]. This was the basis for introducing TCQs over $\mathcal{ALC}$-TKBs in [Baader et al., 2013], which was extended to $\mathcal{SHQ}$ in [Baader et al., 2015]. However, reasoning in $\mathcal{ACC}$ is not tractable, and context-aware systems often need to deal with large quantities of data and adapt fast. TCQs over several lightweight logics have been regarded in [Borgwardt et al., 2015], but only over a fragment of LTL without negation. In [Artale et al., 2007], the complexity of LTL over axioms of several members of the $\mathcal{DL}$-Lite family of DLs has been investigated. However, nothing is known about TCQs over these logics.

In this paper, we want to answer TCQs over TKBs formulated in $\mathcal{EL}$ and in particular investigate both the combined and the data complexity of the temporal query entailment problem. Moreover, we determine the complexity of satisfiability of $\mathcal{EL}$-LTL-formulae, and additionally consider the special case where only global GCIs are allowed [Baader et al., 2012]. As usual, we consider rigid concepts and roles, whose interpretation does not change over time. In this regard, we distinguish three different settings, depending on whether concepts or roles (or both) are allowed to be rigid. Since rigid concepts can be simulated by rigid roles [Baader et al., 2012], only three cases need to be considered.

Our results are summarized in Table 1. The complexity of $\mathcal{EL}$-LTL is often lower than that of $\mathcal{ALC}$-LTL, for which satisfiability is $\text{ExpTime}$-, $\text{NExpTime}$-, and $\text{2-ExpTime}$-complete, respectively, in the three settings we consider [Baader et al., 2012]. This partially confirms and refutes the conjecture from [Baader et al., 2012] that $\mathcal{EL}$-LTL is as hard as $\mathcal{ALC}$-LTL. Using only global GCIs, the complexity matches that of (unrestricted) $\mathcal{DL}$-Lite$_{ex}$-LTL [Artale et al., 2007]. Regarding TCQs, the complexity is even more reduced compared to $\mathcal{ALC}$ (and $\mathcal{SHQ}$), where TCQ entailment is in $\text{ExpTime}$, $\text{co-NExpTime}$, and $\text{2-ExpTime}$, respectively, w.r.t. combined complexity, and in co-NP, NP, and $\text{ExpTime}$, respectively, w.r.t. data complexity [Baader et al., 2015]. The only lower bounds that directly apply to the problems considered here are $\text{PSPACE}$-hardness of LTL [Sistla and Clarke, 1985] and $\text{P}$-hardness of $\text{CQ}$ entailment in $\mathcal{EL}$ w.r.t. data complexity [Calvanese et al., 2006].

Our results are based on known techniques for $\mathcal{ALC}$-LTL and TCQs over $\mathcal{ALC}$-TKBs [Baader et al., 2012; 2015], but we had to significantly adapt them and to combine them with new approaches, in particular for some of the hardness proofs and for the $\text{PSPACE}$-upper bounds. Full proofs of all results can be found in the technical reports [Borgwardt and Thost, 2015a; 2015b].

<table>
<thead>
<tr>
<th>rigid symbols</th>
<th>$\text{TCQ entailment}$ (data complexity)</th>
<th>$\text{TCQ entailment}$ (combined complexity)</th>
<th>$\mathcal{EL}$-LTL satisfiability</th>
<th>$\mathcal{EL}$-LTL satisfiability (with global GCIs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>P</td>
<td>$\text{PSPACE}$</td>
<td>$\text{PSpace}$</td>
<td>$\text{PSpace}$</td>
</tr>
<tr>
<td></td>
<td>LB: [C⁺06], UB: Thm. 5</td>
<td>LB: [SC85]</td>
<td>LB: [SC85], UB: Thm. 11</td>
<td>LB: [SC85]</td>
</tr>
<tr>
<td>only concepts</td>
<td>co-NP</td>
<td>$\text{PSPACE}$</td>
<td>$\text{NExpTime}$</td>
<td>$\text{PSPACE}$</td>
</tr>
<tr>
<td></td>
<td>LB: Thm. 8</td>
<td>UB: Thm. 6</td>
<td>LB: Thm. 9</td>
<td></td>
</tr>
<tr>
<td>concepts and roles</td>
<td>co-NP</td>
<td>$\text{co-NExpTime}$</td>
<td>$\text{NExpTime}$</td>
<td>$\text{PSPACE}$</td>
</tr>
<tr>
<td></td>
<td>UB: Thm. 5</td>
<td>LB: Thm. 7, UB: Thm. 5</td>
<td>LB: Thm. 11</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Summary of the complexity results; [C⁺06] stands for [Calvanese et al., 2006], [SC85] for [Sistla and Clarke, 1985].

2 $\mathcal{EL}$ and LTL

We introduce the formalisms underlying the temporal languages we consider in this paper.

The DL part focuses on the logic $\mathcal{EL}$. Let $\mathcal{N}_C, \mathcal{N}_R, \mathcal{N}_I$ be sets of concept-, role-, and individual names, respectively. Concepts are built from concept names using the constructors conjunction ($C \cap D$), existential restriction ($\exists r.C$ for $r \in \mathcal{R}_P$), and top concept ($\top$). An axiom is either an assertion of the form $\langle \top \rangle a$ or $\langle r(a, b) \rangle$, where $A \in \mathcal{N}_C$, $r \in \mathcal{R}_R$, and $a, b \in \mathcal{N}_I$ or a general concept inclusion (GCI) of the form $C \sqsubseteq D$ for concepts $C, D$. An ABox is a finite set of assertions, a TBox is a finite set of GCIs, and a knowledge base (KB) is a pair $(T, A)$ consisting of a TBox $T$ and an ABox $A$.

An interpretation $\mathcal{I}$ has a non-empty domain $\Delta^\mathcal{I}$ and an interpretation function $\mathcal{I}$ that assigns to every $A \in \mathcal{N}_C$ a set $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$, to every $r \in \mathcal{R}_R$ a relation $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$, and to every $a \in \mathcal{N}_I$ an element $a^\mathcal{I} \in \Delta^\mathcal{I}$. This function is extended to concepts as follows: $(C \cap D)^\mathcal{I} := C^\mathcal{I} \cap D^\mathcal{I}$; $(\exists r.C)^\mathcal{I} := \{ x \mid \exists y : (x, y) \in r^\mathcal{I} \land y \in C^\mathcal{I} \}$; $\top^\mathcal{I} := \Delta^\mathcal{I}$. An interpretation $\mathcal{I}$ satisfies (or is a model of) $\langle A(a) \rangle$ if $a^\mathcal{I} \in A^\mathcal{I}$; $\langle r(a, b) \rangle$ if $(a^\mathcal{I}, b^\mathcal{I}) \in r^\mathcal{I}$; $a^\mathcal{I} \in C^\mathcal{I}$ if $C^\mathcal{I} \subseteq D^\mathcal{I}$; a set of axioms or KB if it satisfies all its axioms. A KB $K$ entails an axiom $\alpha$
(written $K \models \alpha$) if every model of $K$ is also a model of $\alpha$.

The temporal component of our formalism is based on propositional LTL \cite{Pnueli1977}. LTL-formulae are built from a set of propositional variables $P$ using conjunction ($\phi \land \psi$), negation ($\neg \psi$), next ($\psi^t$), previous ($\psi^s$), until ($\phi U \psi$), and since ($\phi S \psi$). An LTL-structure $\mathcal{J} = (w_i)_{i \geq 0}$ is an infinite sequence of worlds $w_i \subseteq P$. Validity of an LTL-formula $\phi$ in $\mathcal{J}$ at time point $i \geq 0$ (written $\mathcal{J}, i \models \phi$) is defined inductively:

\[
\begin{align*}
\mathcal{J}, i \models p & \quad \text{iff } p \in w_i \quad (p \in P) \\
\mathcal{J}, i \models \phi \land \psi & \quad \text{iff } \mathcal{J}, i \models \phi \land \mathcal{J}, i \models \psi \\
\mathcal{J}, i \models \neg \phi & \quad \text{iff } \neg \mathcal{J}, i \models \phi \\
\mathcal{J}, i \models \phi \lor \psi & \quad \text{iff } \mathcal{J}, i \models \phi \lor \mathcal{J}, i \models \psi \\
\mathcal{J}, i \models \psi^t & \quad \text{iff } \mathcal{J}, i + 1 \models \psi \\
\mathcal{J}, i \models \psi^s & \quad \text{iff } \mathcal{J}, i \models \psi \\
\end{align*}
\]

An LTL-formula $\phi$ is satisfiable if there is an LTL-structure $\mathcal{J}$ with $\mathcal{J}, 0 \models \phi$. Note that this logic is usually called Past-LTL due to the operators $\psi^s$ and $\psi^t$. The past of past operators does not affect the complexity of the satisfiability problem \cite{Lichtenstein1985}, but allows to write some formulae more succinctly \cite{Laroussinie2002}. As usual, one can express other temporal operators such as eventually ($\psi^s \phi$) and always ($\psi^t \phi$) in this logic.

### 3 Temporal Query Entailment in $\mathcal{EC}$

As described in the introduction, in our temporal formalism we can designate certain concept and role names as being rigid, which means that their interpretation is not allowed to change over time. For this purpose, we fix a set $N_{RC} \subseteq N_c$ of rigid concept names and a set $N_{RR} \subseteq N_r$ of rigid role names.

Temporally conjunctive queries (TCQs) \cite{Baader2015} are constructed exactly as LTL-formulae, except that conjunctive queries (CQs) \cite{Abiteboul1995} take the place of the propositional variables. A conjunctive query is of the form $\exists x_1, \ldots, x_m. \psi$, where $x_1, \ldots, x_m$ are variables and $\psi$ is a conjunction of atoms of the form $A(t)$ or $r(t, t')$, where $A \in N_c$, $r \in N_r$, and $t, t'$ are individual names or variables. A Boolean TCQ does not contain free variables. A CQ-literal is either a CQ or a negated CQ; and a union of conjunctive queries (UCQ) is a disjunction of CQs.

The semantics of TCQs is also very similar to that of LTL-formulae. However, instead of LTL-structures one has to consider infinite sequences $\mathcal{I} = (I_i)_{i \geq 0}$ of interpretations. Following \cite{Baader2015}, we make the constant domain assumption (i.e., the interpretations all have the same domain $\Delta$). Furthermore, we have to ensure that the rigid names are respected; that is, we require that $s^{\Delta} = s^{\Delta}$ holds for all symbols $s \in N_c \cup N_{RC} \cup N_{RR}$ and $i, j \geq 0$. Validity of a TCQ $\phi$ in $\mathcal{I}$ at time point $i \geq 0$ (again denoted by $\mathcal{I}, i \models \phi$) is now defined exactly as for LTL in Section 2, with the obvious exception of CQs. For these, we adopt the classical semantics based on homomorphisms \cite{Chandra1977}. More precisely, the fact that $\mathcal{I}, i \models \psi$ for a CQ $\psi$ is equivalent to $\psi$ being satisfied by $I_i$ (written $I_i \models \psi$), which is the case if there is a homomorphism $\pi$ mapping the variables and individual names of $\psi$ into $\Delta$ such that: $\pi(a) = o^{\pi}$ for all $a \in N_i$; $\pi(t) \in A^{\pi}$ for all concept atoms $A(t)$ in $\psi$; and $(\pi(t), \pi(t')) \in r^{\pi}$ for all role atoms $r(t, t')$ in $\psi$.

We now consider temporal knowledge bases (TKBs) of the form $K = (T, (A_i)_{0 \leq i \leq n})$, where $T$ is a TBox and the $A_i$ are ABoxes. As described in the introduction, $T$ represents the global knowledge about the application domain, whereas the $A_i$ contain data about different time points. A sequence $\mathcal{I} = (I_i)_{i \geq 0}$ of interpretations as above satisfies (or is a model of) $K$ (written $\mathcal{I} \models K$) if we have $I_i \models T$ for all $i \geq 0$, and $I_i \models A_i$ for all $i, 0 \leq i \leq n$. A Boolean TCQ $\phi$ is satisfiable w.r.t. $K$ if there is a model $\mathcal{I}$ of $K$ such that $\mathcal{I}, n \models \phi$, and it is entailed by $K$ (written $K \models \phi$) if for all models $\mathcal{I}$ of $K$ it holds that $\mathcal{I}, n \models \phi$. Recall that we are interested in the current time point $n$, for which the most recent data ($A_n$) is available.

For a (non-Boolean) TCQ $\phi$, a mapping $\alpha$ of the free variables in $\phi$ to the individual names of $K$ is a certain answer to $\phi$ w.r.t. $K$ if $K \models \alpha(\phi)$, where $\alpha(\phi)$ is obtained from $\phi$ by replacing the free variables according to $\alpha$. As usual, the problem of computing all certain answers can be reduced to exponentially many entailment tests. Therefore, we investigate in the following the complexity of the TCQ entailment problem in $\mathcal{EC}$. We do this indirectly, via the satisfiability problem, which has the same complexity as non-entailment.

We consider both data complexity, where the TBox $T$ and the TCQ $\phi$ are assumed to be fixed and the complexity is measured only w.r.t. the size of the input ABoxes $(A_i)_{0 \leq i \leq n}$; and combined complexity, where also the influence of $T$ and $\phi$ is taken into account. As described in the introduction, we further distinguish the three cases where (i) no rigid names are available ($N_{RC} = N_{RR} = \emptyset$); (ii) only rigid concept names are allowed ($N_{RR} = \emptyset$, but $N_{RC} \neq \emptyset$); and (iii) also rigid role names can be used ($N_{RR} \neq \emptyset$).

We next state an auxiliary result about satisfiability of (atemporal) conjunctions of CQ-literals. Note that, for this case, it suffices to consider an ordinary KB instead of a TKB.

**Lemma 1.** W.r.t. combined complexity, deciding whether a Boolean conjunction of CQ-literals $\psi$ is satisfiable w.r.t. a KB $K = (T, A)$ can be reduced to several $P$-tests, whose number is polynomial in the number of conjuncts of $\psi$ and exponential in the size of the largest negated conjunct in $\psi$.

**Proof Sketch.** We can reduce this problem to the UCQ non-entailment problem $\langle T, A \cup A' \rangle \not\models \sigma_1 \lor \ldots \lor \sigma_m$, where $A'$ is obtained from instantiating the variables of the CQs occurring positively in $\psi$ with fresh individual names, and $\sigma_i$ are the CQs occurring negatively in $\psi$. Using the algorithm in \cite{Rosati2007}, this can be solved by a series of polynomial non-entailment tests, one for every possible CQ resulting from some $\sigma_i$, by unifying some of the terms of $\sigma_i$. The claim follows from the fact that there are at most exponentially many such unifiers.

### 3.1 The Upper Bounds

In this section, we describe the general approach used in \cite{Baader2012, Baader2015} to solve the satisfiability problem. In the following, let $K = \langle T, (A_i)_{0 \leq i \leq n} \rangle$ be a TKB and
\( \phi \) be a Boolean TCQ. For ease of presentation, we assume that all concept and role names occurring in \((A_i)_{0 \leq i \leq n}\) or \( \phi \) also occur in \( T \), and that all individual names occurring in \( \phi \) also occur in \((A_i)_{0 \leq i \leq n}\). These assumptions do not affect the complexity results.

The main idea is to consider two separate satisfiability problems—one in LTL and the other in \( \mathcal{EC} \)—that together imply satisfiability of \( \phi \) w.r.t. \( K \). The LTL part analyzes the propositional abstraction \( \phi^p \) of \( \phi \), which contains the propositional variables \( p_1, \ldots, p_m \) in place of the CQs \( \alpha_1, \ldots, \alpha_m \) from \( \phi \) (where each \( \alpha_i \) was replaced by \( p_i \)). Furthermore, let \( S \subseteq \{p_1, \ldots, p_m\} \) be a set that specifies the worlds that are allowed to occur in an LTL-structure satisfying \( \phi^p \). This condition is formalized by the following LTL-formula:

\[
\phi^p_S = \phi^p \land \Box \left( \bigvee_{X \in S} \left( \bigwedge_{p \in X} p \land \bigwedge_{p \in \overline{X}} \neg p \right) \right),
\]

where \( \overline{X} := \{p_1, \ldots, p_m\} \setminus X \) is the complement of \( X \in S \).

However, for checking satisfiability of \( \phi \) w.r.t. \( K \), it is not sufficient to find such a set \( S \) and then test whether \( \phi^p_S \) is satisfiable (at time point \( n \)). We must also ensure that \( S \) can indeed be induced by a model of \( K \) of the following sense.

**Definition 2.** Let \( S = \{X_1, \ldots, X_k\} \subseteq 2^{\{p_1, \ldots, p_m\}} \) and \( \iota: \{0, \ldots, n\} \to \{1, \ldots, k\} \). Then \( S \) is r-satisfiable (w.r.t. \( \iota \) and \( K \)) if there are interpretations \( J_1, \ldots, J_k, I_0, \ldots, I_n \) such that:

1. they share the same domain and respect rigid names;
2. the interpretations are models of \( T \);
3. each \( J_i \) is a model of \( \chi_i := \bigwedge_{p_j \in X_i} \alpha_j \land \bigwedge_{p_j \in \overline{X_i}} \neg \alpha_j \); and
4. each \( I_i \) is a model of \( A_i \) and \( \chi_i(\iota(i)) \).

The existence of \( J_i \) ensures that the conjunction \( \chi_i \) of CQ-literals induced by \( X_i \) is satisfiable, a set \( S \) containing an \( X_i \) for which this does not hold cannot be induced by a model of \( K \). The interpretations \( I_i \) represent the first \( n+1 \) elements of such a model, which must additionally satisfy the ABoxes \( A_i \). The mapping \( \iota \) chooses a world for each ABox.

We now call \( \phi^p \) t-satisfiable (w.r.t. \( S \) and \( \iota \) as above) if there is an LTL-structure \( \mathcal{J} = (w_i)_{i \geq 0} \) with \( \mathcal{J}, n \models \phi^p_S \) and \( w_i = X_{\iota(i)} \) for all \( i, 0 \leq i \leq n \). Intuitively, \( \mathcal{J} \) is the propositional abstraction of the model of \( \phi^p \) \( K \) we are looking for. The following was shown in [Baader et al., 2015] for \( \mathcal{SHQ} \), and remains valid in our setting.

**Lemma 3.** \( \phi \) is satisfiable w.r.t. \( K \) iff there are \( S \) and \( \iota \) as above such that \( S \) is r-satisfiable w.r.t. \( \iota \) and \( K \) and \( \phi^p \) is t-satisfiable w.r.t. \( S \) and \( \iota \).

Since t-satisfiability is independent of the DL part, we can also reuse the following result from [Baader et al., 2015].

**Lemma 4.** Checking t-satisfiability of \( \phi^p_S \) w.r.t. \( S \) and \( \iota \) is:

1. in ExPTIME w.r.t. combined complexity, and
2. in P w.r.t. data complexity.

Given this, we already obtain some of the upper bounds.

---

This is defined as for sequences of interpretations.

**Theorem 5.** TCQ entailment in \( \mathcal{EC} \) is

- in co-NP w.r.t. data complexity and in co-NExpTime w.r.t. combined complexity even if \( N_{RR} \neq \emptyset \).
- in P w.r.t. data complexity if \( N_{RC} = N_{RR} = \emptyset \).

**Proof Sketch.** Recall that we show the complementary results by regarding TCQ satisfiability. Let \( K = (\mathcal{T}, (A_i)_{0 \leq i \leq n}) \) be a TKB and \( \phi \) be a Boolean TCQ. For the first two results, we can simply guess \( S \) and \( \iota \) as required for Lemma 3; note that \( S \) is of constant size in the size of the input ABoxes. By Lemma 4, the required t-satisfiability test can be done within the claimed time bounds. For the r-satisfiability test, we use a technique from [Baader et al., 2015] that constructs an exponentially large conjunction \( \chi_{S, r} \) of CQ-literals and TBox \( T_S \), such that it remains to check satisfiability of \( \chi_{S, r} \) w.r.t. \( T_S \). Since the CQ-literals in \( \chi_{S, r} \) are essentially of the same size as the CQs in \( \phi \), we can apply Lemma 1 to decide this problem via exponentially many ExpTime-tests w.r.t. combined complexity. Moreover, the number of conjuncts of \( \chi_{S, r} \) and the size of \( T_S \) are in linear in the size of the input ABoxes, and thus we obtain an upper bound of \( P \) w.r.t. data complexity.

For the last result, observe first that in the absence of rigid names the satisfiability tests of Definition 2 are largely independent of each other. Hence, it suffices to define \( S \) as the set of all sets \( X_j \) for which \( \chi_j \) is satisfiable w.r.t. \( T \). Likewise, we consider, for each ABox \( A_i \), the set \( \iota'(i) \) of all indices \( j \) of worlds \( X_j \) for which \( \chi_j \) is satisfiable w.r.t. \( (T, A_i) \), and employ a modified t-satisfiability test w.r.t. these sets. This results in a deterministic polynomial-time procedure.

It remains to consider the case where \( N_{RR} = \emptyset \), but possibly \( N_{RC} \neq \emptyset \), under combined complexity (see Table 1). Note that the satisfiability tests of Definition 2 are not independent in this case. Nevertheless, we can guess polynomially many additional data (see \( A_R \) and \( Q_{\emptyset}^d \)) below that allow us to separate these tests. We then combine these with the PSPACE-procedure for \( \mathcal{LT} \)-satisfiability [Sistla and Clarke, 1985] in order to obtain the claimed upper bound.

We assume here that the sequence of input ABoxes consists only of one empty ABox; this is without loss of generality since the ABoxes can be encoded into the TCQ without affecting the (combined) complexity [Baader et al., 2015]. We thus consider a TKB \( K = (\mathcal{T}, \emptyset) \) and a Boolean TCQ \( \phi \).

Before stating the main result, we first give some auxiliary definitions. Let \( \psi \) be a CQ that does not contain any individual names and is tree-shaped (i.e., the directed graph described by its atoms is a tree), and let \( x \) be the root of this tree. Then \( \text{Con} (\psi, x) \) abbreviates the concept \( \text{Con} (\psi, x) \), where

\[
\text{Con} (\psi, y) := \bigcap_{A(y) \in \psi} A \cap \bigcap_{r(y,z) \in \psi} \exists r. \text{Con} (\psi, z).
\]

A subset \( B \) of the rigid concept names occurring in \( T \) is a witness of \( \psi \) w.r.t. \( T \) if there are \( r_1, \ldots, r_{\ell}, \ell \geq 0 \), such that \( T \models (\bigcap B) \subseteq \exists r_1, \ldots, \exists r_{\ell}. \text{Con} (\psi) \). Intuitively, if a model of \( T \) contains an element satisfying \( \bigcap B \), then \( \psi \) is satisfied.

We now consider all possible assertions over the individual names and the rigid concept names occurring in the input, together with their negations. An ABox type \( A_R \) is a set
of such assertions such that \( A(a) \in A_R \) iff \( \neg A(a) \notin A_R \). Given \( S = \{X_1, \ldots, X_k\} \subseteq \{p_1, \ldots, p_m\} \), we define the KBs \( K_R^i := (T_R \cup A_Q^i), 1 \leq i \leq k, \) where the ABox \( A_Q^i \) contains the CQs occurring positively in \( \psi \), with the variables replaced by fresh individual names. A tuple \( \langle A_R, Q_R^i \rangle \), where \( A_R \) is an ABox type and \( Q_R^i \) is a subset of \( \{\alpha_1, \ldots, \alpha_m\} \), is \( r \)-complete (w.r.t. \( S \)) if the following hold:

(R1) For all \( i \in \{1, \ldots, k\}, K_R^i \) has a model.

(R2) For all \( i \in \{1, \ldots, k\} \) and \( p_j \in X_i \), we have \( K_R^i \not\models \alpha_j \).

(R3) For all \( i \in \{1, \ldots, k\} \), all tree-shaped \( \alpha \in Q_R^i \), and all witnesses \( \beta \) of \( \alpha \) w.r.t. \( T \), we have \( K_R^i \not\models \exists x.\beta(x) \).

(R4) For all \( \alpha_j \in Q_R \setminus Q_R^i \), we have \( p_j \in \bigcap S \).

The idea is to fix the interpretation of the rigid names on all named individuals and specify the CQs that are allowed to occur negatively in \( S \) via the guessed data \( \langle A_R, Q_R^i \rangle \). (R1) and (R2) ensure that exactly the queries specified by \( X_i \), together with the assertions from \( A_R \), can be satisfied w.r.t. \( T \). (R3) ensures that there is a model of \( K_R^i \) that does not satisfy any of the witnesses of the tree-shaped queries in \( Q_R^i \) (the canonical model [Lutz et al., 2009]). Finally, (R4) makes sure that only the queries from \( Q_R^i \) can occur negatively in any \( X_i \).

We can show that \( r \)-satisfiability of \( S \) is characterized by the existence of such an \( r \)-complete tuple. To actually obtain a \( \text{PSPACE} \) decision procedure from this result, we adapt the \( \text{PSPACE} \)-Turing machine from [Sistla and Clarke, 1985] that successively guesses propositional worlds and checks whether these can be assembled into an LTL-structure satisfying \( \phi^R \). We use a modified version of this Turing machine that first guesses a tuple \( \langle A_R, Q_R^i \rangle \) as described above, and then proceeds as before, but, for each guessed world \( X_i \), additionally checks whether the KB \( K_R^i \) satisfies Conditions (R1)–(R4). For (R1), note that the negated assertions in \( A_R \) do not pose a problem, as they can be simulated using nominals and the bottom constructor [Baader et al., 2005]. Moreover, the non-entailment tests in (R2) and (R3) can be done using only the positive assertions in \( A_R \). Finally, to check whether a given set \( B \) is actually a witness of a tree-shaped CQ \( \alpha \in Q_R^i \), it suffices to do a reachability test in the completion graph of \( T \) [Baader et al., 2005].

**Theorem 6.** If \( N_{RR} = 0 \), but possibly \( N_{RC} \neq 0 \), then \( TCQ \) entailment in \( \mathcal{EL} \) is in \( \text{PSPACE} \) w.r.t. combined complexity.

### 3.2 What Makes It Hard

If \( N_{RR} \neq 0 \), we can show \( \text{co-NEXPTime} \)-hardness w.r.t. combined complexity by adapting the proof of \( \text{NEXPTime} \)-hardness of satisfiability in \( \mathcal{ALC} \)-LTL from [Baader et al., 2012]. The latter reduces the \( 2^{n+1} \)-bounded domino problem [Lewis, 1978; Börger et al., 1997] and the result already holds if only concept names are allowed to be rigid. However, \( \mathcal{ALC} \)-LTL-formulae are built by replacing the propositional variables in LTL-formulae by axioms of the more expressive DL \( \mathcal{ALC} \), which may contain concept negation (\( \neg \)) and disjunction (\( \lor \)). In a nutshell, the original proof represents the positions in the \( 2^{n+1} \times 2^{n+1} \) domino grid in two different ways: for each position, there is a specific time point representing it, as well as a domain element \( x_i \). This dual representation facilitates the encoding of the domino conditions.

<table>
<thead>
<tr>
<th>( t )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>\ldots</th>
</tr>
</thead>
<tbody>
<tr>
<td>( e \cdot C )</td>
<td>( c )</td>
<td>( c )</td>
<td>( c )</td>
<td>( c \cdot \top )</td>
<td>( c \cdot C )</td>
<td></td>
</tr>
<tr>
<td>( A^\psi_i )</td>
<td>( a_{x_1} )</td>
<td>( a_{x_2} )</td>
<td>( a_{x_3} )</td>
<td>( a_{x_4} )</td>
<td>( A )</td>
<td>\ldots</td>
</tr>
<tr>
<td>( a_{-x_1} )</td>
<td>( a_{-x_2} )</td>
<td>( a_{-x_3} )</td>
<td>( a_{-x_4} )</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 1: The ABoxes for \( (x_1 \lor x_3 \lor \neg x_4) \land \ldots \); framed names describe a possible extension to a model of \( \phi \) w.r.t. \( K_\psi \).

We describe some interesting adaptations necessary to apply the proof for the case of conjunctive queries and a global \( \mathcal{EL} \)-TBox; the detailed proof can be found in the technical report.

- Instead of the formula \( \square \neg (\top \subseteq \neg N) \), we use the TCQ \( \square (\exists x. r(x, a) \land N(x)) \) to create the elements \( x_i \). This connects them to a fixed individual \( a \) via the new rigid role \( r \) and allows us to refer back to them later.

- Formulae of the form \( (\top \subseteq A) \lor (\top \subseteq \neg A) \) are used to express that \( A \) is either satisfied by all domain elements or by none. The second axiom can be expressed by the negated CQ \( \neg \exists x. A(x) \). The first axiom cannot easily be expressed by a TCQ; however, for the hardness proof it suffices to ensure that \( A \) is satisfied by all elements \( x_i \), which can be identified via their \( r \)-connection to \( a \). Thus, we can replace the first axiom by the CQ \( A(a) \) and the (global) GCI \( \exists x. A \subseteq A \).

**Theorem 7.** If \( N_{RR} \neq 0 \), then \( TCQ \) entailment in \( \mathcal{EL} \) is \( \text{co-NEXPTime-hard} \) w.r.t. combined complexity.

The last remaining result concerns the data complexity of \( TCQ \) entailment if rigid concept names are allowed. We show NP-hardness of satisfiability by a reduction of the 3-SAT problem [Karp, 1972], considering a propositional 3-CNF formula \( \psi = \bigwedge_{0 \leq i < \ell} l_{i,1} \lor l_{i,2} \lor l_{i,3} \). We construct a TCQ \( \phi \) and a TKB \( K_\psi = \langle T, \{A^\psi_i\}_{0 \leq i < \ell} \rangle \) such that \( \psi \) is satisfiable iff \( \phi \) is satisfiable w.r.t. \( K_\psi \). We use four ABoxes to represent each clause: one to identify the start of a new clause (via \( C(c) \)), and the following three to encode the literals of this clause via the individual names \( a_l \) (see also Figure 1):}

\[ A^\psi_i := \{C(c)\} \]
\[ A^\psi_{i+l} := \{r(a_{l_{i,j}}, a_{-l_{i,j}}), s(a_{l_{i,j}}, c)\} \]

Then, we enforce through

\[ \phi := \square \left( (C(c) \rightarrow (\bigcirc T(c) \lor \bigcirc \bigcirc T(c) \lor \bigcirc \bigcirc \bigcirc T(c))) \right) \]
\[ \land \neg \exists x, y. r(x, y) \land A(x) \land A(y) \]

that one of the clause’s literals is satisfied (indicated by \( T(c) \)). Using the rigid concept \( A \), we express that a literal \( a_l \) and its complement \( a_{-l} \) cannot both be true at the same time. Finally, we use the TBox \( T := \{\exists x. T \subseteq A\} \) to connect the
satisfaction of the literal of a clause \((T(c))\) with the truth of the corresponding literal \((A(a))\).

Note that both \(\phi\) and \(T\) are of constant size, and the size of \((A^\psi_{i,t})\) is linear in the size of \(\psi\).

**Theorem 8.** If \(N_{RC} \neq \emptyset\), then TCQ entailment in \(\mathcal{EL}\) is co-NP-hard w.r.t. data complexity.

### 4 Temporal Subsumption in \(\mathcal{EL}\)

We now consider a related temporal formalism based on \(\mathcal{EL}\), where the atoms of the temporal formulae are not CQs, but axioms [Baader et al., 2012]. More formally, \(\mathcal{EL}\)-LTL-formulae are defined exactly as LTL-formulae, except that instead of propositional variables they contain assertions and GCIs. As in Section 3, the semantics are given by infinite sequences of interpretations. Validity of an \(\mathcal{EL}\)-LTL-formula \(\phi\) in \(J = (I_t)_{t \geq 0}\) at time point \(i \geq 0\) (written \(J,i \models \phi\)) is defined as in Section 2, with the exception of axioms \(\alpha\), where we define \(J,i \models \alpha\) iff \(I_i\) satisfies \(\alpha\). As in [Baader et al., 2012], we investigate the satisfiability of \(\mathcal{EL}\)-LTL-formulae, i.e., deciding whether there is a sequence \(J\) such that \(3,0 \models \phi\). A corresponding entailment problem would be the question whether \(3,0 \models \psi\) always implies that \(3,0 \models \phi\), but this can easily be reduced to the unsatisfiability of \(\psi \land \neg \phi\).

ALC-LTL-formulae [Baader et al., 2012] can be reformulated as TCQs over ALC-TKBs [Baader et al., 2015]. However, this is not the case for \(\mathcal{EL}\): GCIs of the form \(T \subseteq A\) cannot directly be simulated by TCQs; and conversely, cyclic CQs like \(\exists x,y.r(x,y) \land r(y,x)\) cannot be expressed by \(\mathcal{EL}\)-LTL-formulae. Hence, these two satisfiability problems are not directly comparable. Nevertheless, satisfiability of \(\mathcal{EL}\)-LTL-formulae turns out to be always harder than that of TCQs (see Table 1). It does not make sense to consider data complexity here because the assertions are part of the formula.

The proof techniques employed for \(\mathcal{EL}\)-LTL-formulae are similar to those we have presented in Section 3. For instance, we can show \textsc{NExpTime-hardness} using a similar construction as in the proof of Theorem 7, which is even closer to that of [Baader et al., 2012] and does not use rigid role names.

**Theorem 9.** If \(N_{RC} \neq \emptyset\), then satisfiability in \(\mathcal{EL}\)-LTL is \(\text{NExpTime-hard}\).

For the upper bounds, we use the ideas from Theorem 5. The \(r\)-satisfiability condition is simpler since we do not have to consider ABoxes, and the \(\chi_i\) are now conjunctions of \(\mathcal{EL}\)-literals, which are axioms or negated axioms. As in Lemma 1, we first determine the complexity of satisfiability of such conjunctions. The main idea is to instantiate negated GCIs and to simulate negated assertions using nominals and the bottom constructor to construct an \(\mathcal{EL}^{++}\)-KB that has a model iff the original conjunction has a model. The former problem can be decided in polynomial time [Baader et al., 2005].

**Lemma 10.** Satisfiability of conjunctions of \(\mathcal{EL}\)-literals can be decided in \(P\).

This helps us to prove the following upper bounds.

**Theorem 11.** Satisfiability in \(\mathcal{EL}\)-LTL is

- in \(\text{NExpTime}\) even if \(N_{RR} \neq \emptyset\),
- in \(\text{PSPACE}\) if \(N_{RC} = N_{RR} = \emptyset\).

**Proof Sketch.** The first result is obtained exactly as in the proof of Theorem 5, using the renaming technique from [Baader et al., 2012] and Lemma 10.

For the second upper bound, observe once more that the satisfiability tests of Definition 2 are independent in the absence of rigid concept and role names. Thus, we can again use the \(\text{PSPACE}\)-Turing machine from [Sistla and Clarke, 1985], where, in each step, we additionally execute a P-test according to Lemma 10.

Given the rather negative results for \(\mathcal{EL}\)-LTL in the presence of rigid symbols, we now consider \(\mathcal{EL}\)-LTL with global GCIs, as introduced in [Baader et al., 2012]. In this case, \(\mathcal{EL}\)-LTL-formulae are restricted to the form \((\square \land T) \land \psi\), where \(T\) is a TBox and \(\psi\) is an \(\mathcal{EL}\)-LTL-formula using only assertions. This is also a special case of a Boolean TCQ \(\psi\) over the TBox \(T\), where the CQs in \(\psi\) do not contain any variables.

By an adaptation of the approach used in the proof of Theorem 6, we can extend the complexity of \(\text{PSPACE}\) even to the case where rigid roles are allowed. It suffices to guess an ABox type, which must now contain also (negated) role assertions for all rigid role names, together with a set of assertions of the form \(\exists r.A(a)\) for a rigid role name \(r\). Intuitively, they specify the kinds of \(r\)-successors \(a\) must have at every time point. Again, the existence of such an ABox type and assertions characterizes the \(r\)-satisfiability of \(S\), and we obtain the following result by an adaptation of the \(\text{PSPACE}\)-Turing machine from [Sistla and Clarke, 1985].

**Theorem 12.** Even if \(N_{RR} \neq \emptyset\), then satisfiability in \(\mathcal{EL}\)-LTL with global GCIs is in \(\text{PSPACE}\).

### 5 Conclusions

We have characterized the computational complexity of two recently proposed temporal query languages over ontologies in \(\mathcal{EL}\). The data complexity of TCQ entailment implies that it may be possible to apply the approach of [Lutz et al., 2009] if no rigid names are allowed. But this is not a very interesting case since one cannot formulate temporal dependencies.

On the positive side, we show that the combined complexity of \(\text{PSPACE}\) inherited from LTL does not increase if rigid role names are disallowed, and also in the case that the query contains no variables and rigid role names are allowed. Furthermore, if we make the reasonable assumption that all relevant information about the rigid names (e.g., which applications belong to the concept VideoApplication) is available before the start of our context-aware system, then we do not need to guess the ABox type \(A_R\). It remains to be seen whether one can efficiently combine existing algorithms for LTL [Gastin and Oddoux, 2001] and \(\mathcal{EL}\) [Lutz et al., 2009].

Regarding the conjecture about \(\mathcal{EL}\)-LTL from [Baader et al., 2012], we have verified that \(\mathcal{EL}\)-LTL has the same complexity as \(\mathcal{ALC}\)-LTL if only rigid concept names are allowed. However, if rigid role names are considered, then the complexity decreases from \(2\times\text{ExpTime}\) to \(\text{ExpTime}\).

In future work, we want to investigate what happens if we replace \(\mathcal{EL}\) by \(\mathcal{DL}\)-LTL. While satisfiability in \(\mathcal{DL}\)-LTL is \(\text{PSPACE}\)-complete in all cases, the complexity of TCQ entailment over \(\mathcal{DL}\)-LTL remains open. Our hope is
that TCQs can be rewritten into a first-order query over the database resulting from viewing the ABox sequence under the closed world assumption [Calvanese et al., 2006]. If the size of the rewriting is not too large, this may yield efficient algorithms for answering temporal queries. For a practical application, an implementation should also be based on suitable windows of the data rather than the whole history. We are also currently evaluating the utility of temporal query languages for situation recognition in operating systems.

Acknowledgments

This work was partially supported by the DFG in CRC 912. We also want to thank Franz Baader, Marcel Lippmann, and Carsten Lutz for fruitful discussions on the topic of this paper.

References


