Temporal Query Answering in $\mathcal{EL}^*$

Stefan Borgwardt and Veronika Thost

Theoretical Computer Science, TU Dresden, Germany
firstname.lastname@tu-dresden.de

Motivation Context-aware systems use data collected at runtime to recognize predefined situations and trigger adaptations; e.g., an operating system may use sensors to recognize that a video application is out of user focus, and then adapt application parameters to optimize the energy consumption. Using ontology-based data access [12, 19], the situations can be encoded into queries that are answered over an ABox containing the sensor data. In the TBox, we can encode background knowledge about the domain. For example, if the user has been working with another application on a second screen for a longer period, then we may assume that he does not need the video to be displayed in the highest resolution.

In this paper, we focus on the lightweight DL $\mathcal{EL}$. We can state static knowledge about applications ($\text{VideoApplication(app1)}$), dynamic knowledge about the current context ($\text{NotWatchingVideo(user1)}$), as well as background knowledge like

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

saying that a video application whose user is currently not watching the video is out of user focus. Given such a knowledge base, we can use the conjunctive query (CQ) $\psi(x) := \exists y. \text{hasState}(x, y) \land \text{OutOfFocus}(y)$ to identify applications $x$ that can potentially be assigned a lower priority. More complex situations typically depend also on the behavior of the environment in the past—the operating system should not switch configurations every time the user is not watching for one second, but only after this has been the case for a longer period.

For that reason, we investigate temporal conjunctive queries (TCQs), originally proposed in [3, 4]. They combine conjunctive queries via the operators of the propositional linear temporal logic LTL [14, 18]. We can use the TCQ

$$\left( \circ^3 \neg \psi(x) \right) \land \left( \circ^3 \circ^3 \neg \psi(x) \right) \land \left( \circ^3 \circ^3 \circ^3 \neg \psi(x) \right) \land \neg \left( \exists y. \text{GotPriority}(y) \land \text{notEqual}(x, y) \right) \text{S GotPriority}(x)$$

to obtain all applications that were out of user focus during the three previous ($\circ^3$) moments of observation, were prioritized by the operating system at some point in time, and the priority has not ($\neg$) changed since (S) then. The semantics of TCQs is based on temporal knowledge bases (TKBs), which, in addition to the TBox (which is assumed to hold globally, i.e., at every point in time), contains a sequence of ABoxes $A_0, A_1, \ldots, A_n$, representing the data collected at specific

* Partially supported by the DFG in CRC 912 (HAEC).
Related Work

The axioms in a TKB do not explicitly refer to time, but are written in a classical (atemporal) DL; only the query is temporalized. In contrast, \cite{1,2,13,17} extend classical DLs by temporal operators that occur within concepts and axioms. However, most of these logics yield high reasoning complexities, even if the underlying atemporal DL is tractable. Lower complexities are obtained by considerably restricting either the 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 $\mathcal{ALC}$, was first considered in \cite{6}. This was the basis for introducing TCQs over $\mathcal{ALC}$-TKBs in \cite{3}, which was extended to $\mathcal{SHQ}$ in \cite{4}. However, reasoning in $\mathcal{ALC}$ 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 \cite{7}, but only over a fragment of LTL without negation. In \cite{1}, 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.

Results

We investigate the combined and data complexity of the TCQ entailment problem over TKBs formulated in $\mathcal{EL}$. Moreover, we determine the complexity of satisfiability of $\mathcal{EL}$-LTL-formulae, and additionally consider the special case where only global GCIs are allowed \cite{6}. 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 \cite{6}, only three cases need to be considered: (i) no symbols are allowed to be rigid, (ii) only rigid concepts are allowed, and (iii) both concepts and roles can be rigid. Tables 1 and 2 summarize our results and provide a comparison to related work. The only previously known results that directly apply here are P-hardness of CQ entailment in $\mathcal{EL}$ w.r.t. data complexity \cite{11} and PSPACE-hardness of LTL \cite{20}. Hence, we needed to prove three additional complexity lower bounds.

With a single exception, the complexity of TCQ entailment in $\mathcal{EL}$ turns out to be lower than that in $\mathcal{ALC}$ (and $\mathcal{SHQ}$) \cite{4}. Regarding satisfiability in $\mathcal{EL}$-LTL, Table 2 shows that rigid symbols lead to an increase in complexity that does not affect $\mathcal{DL}$-Lite$_{krom}$-LTL \cite{1}, and even matches the complexity of $\mathcal{ALC}$-LTL and $\mathcal{SHQ}$-LTL in case (ii) \cite{6,15}. Thus, we partially confirm and refute the conjecture of \cite{6} that $\mathcal{EL}$-LTL is as hard as $\mathcal{ALC}$-LTL. In the following, we shortly describe some of the ideas behind them. More details can be found in \cite{8–10}.

The upper bounds are obtained by a combination of techniques that were developed for $\mathcal{ALC}$-LTL \cite{6} and refined for TCQs over $\mathcal{SHQ}$-TKBs \cite{4}, methods for checking LTL-satisfiability \cite{4,20,21}, and algorithms for atemporal reasoning
Table 1. The complexity of TCQ entailment. All results except the one for the data complexity of case (iii) from [4] are tight.

<table>
<thead>
<tr>
<th></th>
<th>Data Complexity</th>
<th>Combined Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(i) (ii) (iii)</td>
<td>(i) (ii) (iii)</td>
</tr>
<tr>
<td>$\mathcal{EL}$</td>
<td>P co-NP co-NP</td>
<td>PSpace PSpace co-NExpTime</td>
</tr>
<tr>
<td>$\mathcal{ALC}/S\mathcal{H}Q$ [4]</td>
<td>co-NP co-NP ExpTime ExpTime co-NExpTime</td>
<td>2-ExpTime</td>
</tr>
</tbody>
</table>

Table 2. The complexity of satisfiability in LTL over DL axioms.

<table>
<thead>
<tr>
<th></th>
<th>Global GCIs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(i) (ii) (iii)</td>
</tr>
<tr>
<td>$DL-Lite_{krom}$ [1]</td>
<td>PSpace PSpace PSpace</td>
</tr>
<tr>
<td>$\mathcal{EL}$</td>
<td>PSpace NExpTime NExpTime PSpace PSpace PSpace</td>
</tr>
</tbody>
</table>

in $\mathcal{EL}$ [5,16]. However, considerable work was necessary to obtain tight complexity bounds in all cases we considered. The main approach is to separate the temporal operators from the CQs (or axioms), which leaves us to solve a variant of the satisfiability problem for LTL (in P w.r.t. data complexity and in PSpace w.r.t. combined complexity), as well as the following problem for the DL part.

**Definition 1.** Let $\mathcal{K} = \langle T, (A_i)_{0 \leq i \leq n} \rangle$ be a TKB and $\alpha_1, \ldots, \alpha_m$ be CQs. A set $S = \{X_1, \ldots, X_k\} \subseteq 2^{\{\alpha_1, \ldots, \alpha_m\}}$ is r-satisfiable w.r.t. a mapping $\iota : \{0, \ldots, n\} \to \{1, \ldots, k\}$ and $\mathcal{K}$ if there are interpretations $J_1, \ldots, J_k$ and $I_0, \ldots, I_n$ such that

- they share the same domain and interpret all rigid symbols in the same way;
- each $J_i$ is a model of $T$ and $\chi_i := \bigwedge X_i \land \bigwedge \{\lnot \alpha_j \mid \alpha_j \notin X_i\}$; and
- each $I_i$ is a model of $\langle T, A_i \rangle$ and $\chi_{\iota(i)}$.

Individually, the satisfiability of the conjunctions $\chi_i$ can be tested in P w.r.t. data complexity and in PSpace w.r.t. combined complexity. However, the problem is to ensure the first condition, namely that all rigid names are interpreted in the same way by all relevant interpretations.

In case (i), this restriction is obviously irrelevant. For case (iii), one can answer an exponentially large UCQ over an exponentially large atemporal knowledge base instead to obtain the upper bounds. The most difficult cases were case (ii) for the combined complexity of TCQ entailment, and the case of global GCIs in $\mathcal{EL}$-LTL, where we needed to obtain PSpace upper bounds in the presence of rigid names. For these cases, we proved that it suffices to guess additional data of polynomial size that can be added to the knowledge bases in order to separate the satisfiability tests in Definition 1. These tests can then be integrated into a PSpace-Turing machine for LTL-satisfiability [20] without increasing the complexity.

**Acknowledgements** We want to thank Franz Baader, Marcel Lippmann, and Carsten Lutz for fruitful discussions on the topic of this paper.

---

1 In the case of $\mathcal{EL}$-LTL, these are axioms.
References