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[1]). 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: $\mathsf{VideoApplication}(\mathsf{app1}), \mathsf{NotWatching}(\mathsf{user1})$. Background knowledge is represented in the *ontology* (or *TBox*) using so-called *general concept inclusions* (GCIs) like

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

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

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

to obtain all applications that were out of user focus during the three *previous* ($\bigcirc^-$) moments of observation, were prioritized by the operating system at some point in time, and the priority has *not* ($\neg$) changed *since* (S) then.[2]

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

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

[2] Although our formalism does not support it yet, priority values can be represented if the underlying DL allows for so-called concrete domains.

rigid symbols	TCQ entailment (data complexity)	TCQ entailment (combined complexity)	$\mathcal{EL}$-LTL satisfiability	$\mathcal{EL}$-LTL satisfiability (with global GCIs)
none	P LB: [C+06], UB: Thm. 5	PSPACE LB: [SC85]	PSPACE LB: [SC85], UB: Thm. 11	PSPACE LB: [SC85]
only concepts	CO-NP LB: Thm. 8	PSPACE UB: Thm. 6	NEXPTIME LB: Thm. 9	PSPACE
concepts and roles	CO-NP UB: Thm. 5	CO-NEXPTIME LB: Thm. 7, UB: Thm. 5	NEXPTIME UB: Thm. 11	PSPACE UB: Thm. 12

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

time), contains a *sequence* of ABoxes $\mathcal{A}_0, \mathcal{A}_1, \dots, \mathcal{A}_n$, 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 $\mathcal{ALC}$, 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 $\mathcal{ALC}$ 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 [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 *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 EXPTIME-, NEXPTIME-, and 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) *DL-Lite*$_{krom}$-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 EXPTIME, CO-NEXPTIME, and 2-EXPTIME, respectively, w.r.t. combined complexity, and in CO-NP, CO-NP, and EXPTIME, respectively, w.r.t. data complexity [Baader *et al.*, 2015]. The only lower bounds that directly apply to the problems considered here are PSPACE-hardness of LTL [Sistla and Clarke, 1985] and P-hardness of 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 PSPACE upper bounds. Full proofs of all results can be found in the technical reports [Borgwardt and Thost, 2015a; 2015b].

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 $\mathsf{N_C}$, $\mathsf{N_R}$, $\mathsf{N_I}$ be sets of *concept-*, *role-*, and *individual names*, respectively. *Concepts* are built from concept names using the constructors *conjunction* ($C \sqcap D$), *existential restriction* ($\exists r.C$ for $r \in \mathsf{N_R}$), and *top concept* ($\top$). An *axiom* is either an *assertion* of the form $A(a)$ or $r(a,b)$, where $A \in \mathsf{N_C}$, $r \in \mathsf{N_R}$, and $a, b \in \mathsf{N_I}$, 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 $\langle \mathcal{T}, \mathcal{A} \rangle$ consisting of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$.

An *interpretation* $\mathcal{I}$ has a non-empty *domain* $\Delta^\mathcal{I}$ and an *interpretation function* $\cdot^\mathcal{I}$ that assigns to every $A \in \mathsf{N_C}$ a set $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$, to every $r \in \mathsf{N_R}$ a relation $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$, and to every $a \in \mathsf{N_I}$ an element $a^\mathcal{I} \in \Delta^\mathcal{I}$. This function is extended to concepts as follows: $(C \sqcap D)^\mathcal{I} := C^\mathcal{I} \cap D^\mathcal{I}$; $(\exists r.C)^\mathcal{I} := \{x \mid \exists y\colon (x,y) \in r^\mathcal{I} \wedge y \in C^\mathcal{I}\}$; $\top^\mathcal{I} := \Delta^\mathcal{I}$. An interpretation $\mathcal{I}$ *satisfies* (or is a *model* of) $A(a)$ if $a^\mathcal{I} \in A^\mathcal{I}$; $r(a,b)$ if $(a^\mathcal{I}, b^\mathcal{I}) \in r^\mathcal{I}$; $C \sqsubseteq D$ if $C^\mathcal{I} \subseteq D^\mathcal{I}$; a set of axioms or KB if it satisfies all its axioms. A KB $\mathcal{K}$ *entails* an axiom α

(written $\mathcal{K} \models \alpha$) if every model of $\mathcal{K}$ is also a model of α.

The temporal component of our formalisms is based on propositional LTL [Pnueli, 1977]. *LTL-formulae* are built from a set of *propositional variables* P using *conjunction* ($\phi \wedge \psi$), *negation* ($\neg\psi$), *next* ($\bigcirc\phi$), *previous* ($\bigcirc^-\phi$), *until* ($\phi \mathsf{U} \psi$), and *since* ($\phi \mathsf{S} \psi$). An *LTL-structure* $\mathfrak{I} = (w_i)_{i \geq 0}$ is an infinite sequence of *worlds* $w_i \subseteq P$. *Validity* of an LTL-formula ϕ in $\mathfrak{I}$ at time point $i \geq 0$ (written $\mathfrak{I}, i \models \phi$) is defined inductively:

$\mathfrak{I}, i \models p$	iff	$p \in w_i$ (for $p \in P$)
$\mathfrak{I}, i \models \phi \wedge \psi$	iff	$\mathfrak{I}, i \models \phi$ and $\mathfrak{I}, i \models \psi$
$\mathfrak{I}, i \models \neg\phi$	iff	not $\mathfrak{I}, i \models \phi$
$\mathfrak{I}, i \models \bigcirc\phi$	iff	$\mathfrak{I}, i+1 \models \phi$
$\mathfrak{I}, i \models \bigcirc^-\phi$	iff	$i > 0$ and $\mathfrak{I}, i-1 \models \phi$
$\mathfrak{I}, i \models \phi \mathsf{U} \psi$	iff	there is $k \geq i$ such that $\mathfrak{I}, k \models \psi$ and $\mathfrak{I}, j \models \phi$ for all j with $i \leq j < k$
$\mathfrak{I}, i \models \phi \mathsf{S} \psi$	iff	there is $0 \leq k \leq i$, such that $\mathfrak{I}, k \models \psi$ and $\mathfrak{I}, j \models \phi$, for all j with $k < j \leq i$

An LTL-formula ϕ is *satisfiable* if there is an LTL-structure $\mathfrak{I}$ with $\mathfrak{I}, 0 \models \phi$. Note that this logic is usually called *Past-LTL* due to the operators $\bigcirc^-$ and S. The presence of past operators does not affect the complexity of the satisfiability problem [Lichtenstein *et al.*, 1985], but allows to write some formulae more succinctly [Laroussinie *et al.*, 2002]. As usual, one can express other temporal operators such as *eventually* ($\Diamond\phi$) and *always* ($\Box\phi$) in this logic.

3 Temporal Query Entailment in $\mathcal{EL}$

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 $\mathsf{N_{RC}} \subseteq \mathsf{N_C}$ of *rigid concept names* and a set $\mathsf{N_{RR}} \subseteq \mathsf{N_R}$ of *rigid role names*.

Temporal conjunctive queries (TCQs) [Baader *et al.*, 2015] are constructed exactly as LTL-formulae, except that conjunctive queries (CQs) [Abiteboul *et al.*, 1995] 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 ψ is a conjunction of *atoms* of the form $A(t)$ or $r(t, t')$, where $A \in \mathsf{N_C}$, $r \in \mathsf{N_R}$, 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 $\mathfrak{I} = (\mathcal{I}_i)_{i \geq 0}$ of interpretations. Following [Baader *et al.*, 2015], we make the *constant domain assumption* (i.e., the interpretations all have the same domain Δ). Furthermore, we have to ensure that the rigid names are respected; that is, we require that $s^{\mathcal{I}_i} = s^{\mathcal{I}_j}$ holds for all symbols $s \in \mathsf{N_I} \cup \mathsf{N_{RC}} \cup \mathsf{N_{RR}}$ and $i, j \geq 0$. *Validity* of a TCQ ϕ in $\mathfrak{I}$ at time point $i \geq 0$ (again denoted by $\mathfrak{I}, i \models \phi$) 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 [Chandra and Merlin, 1977]. More precisely, the fact that $\mathfrak{I}, i \models \psi$ for a CQ ψ is equivalent to ψ being satisfied by $\mathcal{I}_i$ (written $\mathcal{I}_i \models \psi$), which is the case if there is a *homomorphism* π mapping the variables and individual names of ψ into Δ such that: $\pi(a) = a^{\mathcal{I}_i}$ for all $a \in \mathsf{N_I}$; $\pi(t) \in A^{\mathcal{I}_i}$ for all concept atoms $A(t)$ in ψ; and $(\pi(t), \pi(t')) \in r^{\mathcal{I}_i}$ for all role atoms $r(t, t')$ in ψ.

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

For a (non-Boolean) TCQ ϕ, a mapping $\mathfrak{a}$ of the free variables in ϕ to the individual names of $\mathcal{K}$ is a *certain answer* to ϕ w.r.t. $\mathcal{K}$ if $\mathcal{K} \models \mathfrak{a}(\phi)$, where $\mathfrak{a}(\phi)$ is obtained from ϕ by replacing the free variables according to $\mathfrak{a}$. 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 $\mathcal{EL}$. We do this indirectly, via the satisfiability problem, which has the same complexity as non-entailment.

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

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 ψ is satisfiable w.r.t. a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ can be reduced to several* P*-tests, whose number is polynomial in the number of conjuncts of ψ and exponential in the size of the largest negated conjunct in ψ.*

Proof Sketch. We can reduce this problem to the UCQ non-entailment problem $\langle \mathcal{T}, \mathcal{A} \cup \mathcal{A}' \rangle \not\models \sigma_1 \vee \ldots \vee \sigma_m$, where $\mathcal{A}'$ is obtained from instantiating the variables of the CQs occurring positively in ψ with fresh individual names, and σ_i are the CQs occurring negatively in ψ. Using the algorithm in [Rosati, 2007], this can be solved by a series of polynomial non-entailment tests, one for every possible CQ resulting from some σ_i by unifying some of the terms of σ_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 [Baader *et al.*, 2012; 2015] to solve the satisfiability problem. In the following, let $\mathcal{K} = \langle \mathcal{T}, (\mathcal{A}_i)_{0 \leq i \leq n} \rangle$ be a TKB and

ϕ be a Boolean TCQ. For ease of presentation, we assume that all concept and role names occurring in $(\mathcal{A}_i)_{0\leq i\leq n}$ or ϕ also occur in $\mathcal{T}$, and that all individual names occurring in ϕ also occur in $(\mathcal{A}_i)_{0\leq i\leq n}$. 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 $\mathcal{EL}$—that together imply satisfiability of ϕ w.r.t. $\mathcal{K}$. The LTL part analyzes the *propositional abstraction* ϕ^{p} of ϕ, which contains the propositional variables $p_1,\dots,p_m$ in place of the CQs $\alpha_1,\dots,\alpha_m$ from ϕ (where each α_i was replaced by p_i). Furthermore, let $\mathcal{S}\subseteq 2^{\{p_1,\dots,p_m\}}$ be a set that specifies the worlds that are allowed to occur in an LTL-structure satisfying ϕ^{p}. This condition is formalized by the following LTL-formula:

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

where $\overline{X}:=\{p_1,\dots,p_m\}\setminus X$ is the complement of $X\in\mathcal{S}$.

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

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

- they share the same domain and respect rigid names;[3]
- the interpretations are models of $\mathcal{T}$;
- each $\mathcal{J}_i$ 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
- each $\mathcal{I}_i$ is a model of $\mathcal{A}_i$ and $\chi_{\iota(i)}$.

The existence of $\mathcal{J}_i$ ensures that the conjunction χ_i of CQ-literals induced by X_i is satisfiable; a set $\mathcal{S}$ containing an X_i for which this does not hold cannot be induced by a model of $\mathcal{K}$. The interpretations $\mathcal{I}_i$ represent the first $n+1$ elements of such a model, which must additionally satisfy the ABoxes $\mathcal{A}_i$. The mapping ι chooses a world for each ABox.

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

Lemma 3. *ϕ is satisfiable w.r.t. $\mathcal{K}$ iff there are $\mathcal{S}$ and ι as above such that $\mathcal{S}$ is r-satisfiable w.r.t. ι and $\mathcal{K}$ and ϕ^{p} is t-satisfiable w.r.t. $\mathcal{S}$ and ι.*

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 $\phi^{\mathsf{p}}_{\mathcal{S}}$ w.r.t. $\mathcal{S}$ and ι is*

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

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

[3]This is defined as for sequences of interpretations.

Theorem 5. *TCQ entailment in $\mathcal{EL}$ is*

- *in* CO-NP *w.r.t. data complexity and in* CO-NEXPTIME *w.r.t. combined complexity even if $\mathsf{N_{RR}}\neq\emptyset$,*
- *in* P *w.r.t. data complexity if $\mathsf{N_{RC}}=\mathsf{N_{RR}}=\emptyset$.*

Proof Sketch. Recall that we show the complementary results by regarding TCQ satisfiability. Let $\mathcal{K}=\langle\mathcal{T},(\mathcal{A}_i)_{0\leq i\leq n}\rangle$ be a TKB and ϕ be a Boolean TCQ. For the first two results, we can simply guess $\mathcal{S}$ and ι as required for Lemma 3; note that $\mathcal{S}$ 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 $\chi_{\mathcal{S},\iota}$ of CQ-literals and TBox $\mathcal{T}_{\mathcal{S},\iota}$ such that it remains to check satisfiability of $\chi_{\mathcal{S},\iota}$ w.r.t. $\mathcal{T}_{\mathcal{S},\iota}$. Since the CQ-literals in $\chi_{\mathcal{S},\iota}$ are essentially of the same size as the CQs in ϕ, 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 $\chi_{\mathcal{S},\iota}$ and the size of $\mathcal{T}_{\mathcal{S},\iota}$ are 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 $\mathcal{S}$ as the set of *all* sets X_j for which χ_j is satisfiable w.r.t. $\mathcal{T}$. Likewise, we consider, for each ABox $\mathcal{A}_i$, the set $\iota'(i)$ of *all* indices j of worlds X_j for which χ_j is satisfiable w.r.t. $\langle\mathcal{T},\mathcal{A}_i\rangle$, and employ a modified t-satisfiability test w.r.t. these sets. This results in a *deterministic* polynomial-time procedure. $\square$

It remains to consider the case where $\mathsf{N_{RR}}=\emptyset$, but possibly $\mathsf{N_{RC}}\neq\emptyset$, 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 $\mathcal{A}_{\mathsf{R}}$ and $Q^{\neg}_{\mathsf{R}}$ below) that allow us to separate these tests. We then combine these with the PSPACE-procedure for LTL-satisfiability of [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 $\mathcal{K}=\langle\mathcal{T},\emptyset\rangle$ and a Boolean TCQ ϕ.

Before stating the main result, we first give some auxiliary definitions. Let ψ 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 $\mathsf{Con}(\psi)$ abbreviates the concept $\mathsf{Con}(\psi,x)$, where

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

A subset $\mathcal{B}$ of the rigid concept names occurring in $\mathcal{T}$ is a *witness* of ψ w.r.t. $\mathcal{T}$ if there are $r_1,\dots,r_\ell$, $\ell\geq 0$, such that $\mathcal{T}\models(\bigsqcap\mathcal{B})\sqsubseteq\exists r_1.\dots\exists r_\ell.\mathsf{Con}(\psi)$. Intuitively, if a model of $\mathcal{T}$ contains an element satisfying $\bigsqcap\mathcal{B}$, then ψ 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* $\mathcal{A}_{\mathsf{R}}$ is a set of such assertions such that $A(a) \in \mathcal{A}_\mathsf{R}$ iff $\neg A(a) \notin \mathcal{A}_\mathsf{R}$. Given $\mathcal{S} = \{X_1, \ldots, X_k\} \subseteq 2^{\{p_1,\ldots,p_m\}}$, we define the KBs $\mathcal{K}^i_\mathsf{R} := \langle \mathcal{T}, \mathcal{A}_\mathsf{R} \cup \mathcal{A}_{Q_i}\rangle$, $1 \leq i \leq k$, where the ABox $\mathcal{A}_{Q_i}$ contains the CQs occurring positively in χ_i with the variables replaced by fresh individual names. A tuple $(\mathcal{A}_\mathsf{R}, Q^-_\mathsf{R})$, where $\mathcal{A}_\mathsf{R}$ is an ABox type and Q^-_R is a subset of $\{\alpha_1, \ldots, \alpha_m\}$, is *r-complete* (w.r.t. $\mathcal{S}$) if the following hold:

(R1) For all $i \in \{1, \ldots, k\}$, $\mathcal{K}^i_\mathsf{R}$ has a model.

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

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

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

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 $\mathcal{S}$ via the guessed data $(\mathcal{A}_\mathsf{R}, Q^-_\mathsf{R})$. (R1) and (R2) ensure that exactly the queries specified by X_i, together with the assertions from $\mathcal{A}_\mathsf{R}$, can be satisfied w.r.t. $\mathcal{T}$. (R3) ensures that there is a model of $\mathcal{K}^i_\mathsf{R}$ that does not satisfy any of the witnesses of the tree-shaped queries in Q^-_R (the *canonical model* [Lutz *et al.*, 2009]). Finally, (R4) makes sure that only the queries from Q^-_R can occur negatively in any $X \in \mathcal{S}$.

We can show that r-satisfiability of $\mathcal{S}$ is characterized by the existence of such an r-complete tuple. To actually obtain a PSPACE decision procedure from this result, we adapt the 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 ϕ^p. We use a modified version of this Turing machine that first guesses a tuple $(\mathcal{A}_\mathsf{R}, Q^-_\mathsf{R})$ as described above, and then proceeds as before, but, for each guessed world X_i, additionally checks whether the KB $\mathcal{K}^i_\mathsf{R}$ satisfies Conditions (R1)–(R4). For (R1), note that the negated assertions in $\mathcal{A}_\mathsf{R}$ 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 $\mathcal{A}_\mathsf{R}$. Finally, to check whether a given set $\mathcal{B}$ is actually a witness of a tree-shaped CQ $\alpha \in Q^-_\mathsf{R}$, it suffices to do a reachability test in the completion graph of $\mathcal{T}$ [Baader *et al.*, 2005].

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

3.2 What Makes It Hard

If $\mathsf{N_{RR}} \neq \emptyset$, we can show CO-NEXPTIME-hardness w.r.t. combined complexity by adapting the proof of 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 ($\sqcup$). 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.

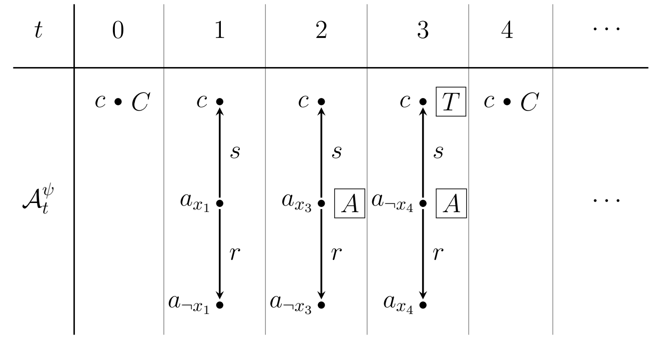

Figure 1: The ABoxes for $(x_1 \vee x_3 \vee \neg x_4) \wedge \ldots$; framed names describe a possible extension to a model of ϕ w.r.t. $\mathcal{K}_\psi$.

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 $\Box\neg(\top \sqsubseteq \neg N)$, we use the TCQ $\Box(\exists x.r(x, a) \wedge N(x))$ 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 $(\top \sqsubseteq A) \vee (\top \sqsubseteq \neg A)$ 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 $\exists r.A \sqsubseteq A$.

Theorem 7. *If $\mathsf{N_{RR}} \neq \emptyset$, then TCQ entailment in $\mathcal{EL}$ is* 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 $\psi = \bigwedge_{0 \leq i < \ell} l_{i,1} \vee l_{i,2} \vee l_{i,3}$. We construct a TCQ ϕ and a TKB $\mathcal{K}_\psi = \langle \mathcal{T}, (\mathcal{A}^\psi_t)_{0 \leq t < 4\ell}\rangle$ such that ψ is satisfiable iff ϕ is satisfiable w.r.t. $\mathcal{K}_\psi$. 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):

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

Then, we enforce through

$$\phi := \Box\Big(\big(C(c) \to \big(\bigcirc T(c) \vee \bigcirc\bigcirc T(c) \vee \bigcirc\bigcirc\bigcirc T(c)\big)\big) \wedge \neg\exists x, y.r(x, y) \wedge A(x) \wedge A(y)\Big)$$

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 $a_{\neg l}$ cannot both be true at the same time. Finally, we use the TBox $\mathcal{T} := \{\exists s.T \sqsubseteq A\}$ to connect the satisfaction of the literal of a clause ($T(c)$) with the truth of the corresponding literal ($A(a_l)$).

Note that both ϕ and $\mathcal{T}$ are of constant size, and the size of $(\mathcal{A}_t^{\psi})_{0\leq t<4\ell}$ is linear in the size of ψ.

Theorem 8. *If $\mathsf{N_{RC}} \neq \emptyset$, 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 ϕ in $\mathfrak{I} = (\mathcal{I}_i)_{i\geq 0}$ at time point $i \geq 0$ (written $\mathfrak{I}, i \models \phi$) is defined as in Section 2, with the exception of axioms α, where we define $\mathfrak{I}, i \models \alpha$ iff $\mathcal{I}_i$ satisfies α. As in [Baader *et al.*, 2012], we investigate the *satisfiability* of $\mathcal{EL}$-LTL-formulae, i.e., deciding whether there is a sequence $\mathfrak{I}$ such that $\mathfrak{I}, 0 \models \phi$. A corresponding entailment problem would be the question whether $\mathfrak{I}, 0 \models \psi$ always implies that $\mathfrak{I}, 0 \models \phi$, but this can easily be reduced to the unsatisfiability of $\psi\wedge\neg\phi$.

$\mathcal{ALC}$-LTL-formulae [Baader *et al.*, 2012] can be reformulated as TCQs over $\mathcal{ALC}$-TKBs [Baader *et al.*, 2015]. However, this is not the case for $\mathcal{EL}$: GCIs of the form $\top \sqsubseteq A$ cannot directly be simulated by TCQs; and conversely, cyclic CQs like $\exists x, y.r(x, y) \wedge r(y, x)$ 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 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 $\mathsf{N_{RC}} \neq \emptyset$, then satisfiability in $\mathcal{EL}$-LTL is* 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 χ_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* NExpTime *even if* $\mathsf{N_{RR}} \neq \emptyset$,
- *in* PSpace *if* $\mathsf{N_{RC}} = \mathsf{N_{RR}} = \emptyset$.

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 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 $\left(\Box \bigwedge \mathcal{T}\right) \wedge \psi$, where $\mathcal{T}$ is a TBox and ψ is an $\mathcal{EL}$-LTL-formula using only assertions. This is also a special case of a Boolean TCQ ψ over the TBox $\mathcal{T}$, where the CQs in ψ do not contain any variables.

By an adaptation of the approach used in the proof of Theorem 6, we can extend the complexity of 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 $\mathcal{S}$, and we obtain the following result by an adaptation of the PSpace-Turing machine from [Sistla and Clarke, 1985].

Theorem 12. *Even if $\mathsf{N_{RR}} \neq \emptyset$, then satisfiability in $\mathcal{EL}$-LTL with global GCIs is in* 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 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 $\mathsf{VideoApplication}$) is available before the start of our context-aware system, then we do not need to guess the ABox type $\mathcal{A}_{\mathsf{R}}$. 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-ExpTime to NExpTime.

In future work, we want to investigate what happens if we replace $\mathcal{EL}$ by *DL-Lite*. While satisfiability in *DL-Lite*-LTL is PSpace-complete in all cases, the complexity of TCQ entailment over *DL-Lite*-TKBs 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

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

[Artale *et al.*, 2007] Alessandro Artale, Roman Kontchakov, Carsten Lutz, Frank Wolter, and Michael Zakharyaschev. Temporalising tractable description logics. In *Proc. of the 14th Int. Symp. on Temporal Representation and Reasoning (TIME'07)*, pages 11–22. IEEE Press, 2007.

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

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

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

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

[Baader *et al.*, 2013] Franz Baader, Stefan Borgwardt, and Marcel Lippmann. Temporalizing ontology-based data access. In *Proc. of the 24th Int. Conf. on Automated Deduction (CADE'13)*, pages 330–344. Springer-Verlag, 2013.

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

[Borgwardt and Thost, 2015a] Stefan Borgwardt and Veronika Thost. LTL over $\mathcal{EL}$ axioms. LTCS-Report 15-07, TU Dresden, Germany, 2015. See http://lat.inf.tu-dresden.de/research/reports.html.

[Borgwardt and Thost, 2015b] Stefan Borgwardt and Veronika Thost. Temporal query answering in $\mathcal{EL}$. LTCS-Report 15-08, TU Dresden, Germany, 2015. See http://lat.inf.tu-dresden.de/research/reports.html.

[Borgwardt *et al.*, 2015] Stefan Borgwardt, Marcel Lippmann, and Veronika Thost. Temporalizing rewritable query languages over knowledge bases. *Journal of Web Semantics*, 2015. In press.

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

[Calvanese *et al.*, 2006] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Data complexity of query answering in description logics. In *Proc. of the 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'06)*, pages 260–270. AAAI Press, 2006.

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

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

[Gastin and Oddoux, 2001] Paul Gastin and Denis Oddoux. Fast LTL to Büchi automata translation. In *Proc. of the 13th Int. Conf. on Computer Aided Verification (CAV'01)*, pages 53–65. Springer-Verlag, 2001.

[Gutiérrez-Basulto *et al.*, 2014] Víctor Gutiérrez-Basulto, Jean Christoph Jung, and Thomas Schneider. Lightweight description logics and branching time: A troublesome marriage. In *Proc. of the 14th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'14)*. AAAI Press, 2014.

[Karp, 1972] Richard Karp. Reducibility among combinatorial problems. In *Proc. of a Symp. on the Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[Laroussinie *et al.*, 2002] François Laroussinie, Nicolas Markey, and Philippe Schnoebelen. Temporal logic with forgettable past. In *Proc. of the 17th Annual IEEE Symp. on Logic in Computer Science (LICS'02)*, pages 383–392. IEEE Press, 2002.

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

[Lichtenstein *et al.*, 1985] Orna Lichtenstein, Amir Pnueli, and Lenore Zuck. The glory of the past. In *Proc. of the Workshop on Logics of Programs*, pages 196–218. Springer-Verlag, 1985.

[Lutz *et al.*, 2008] Carsten Lutz, Frank Wolter, and Michael Zakharyaschev. Temporal description logics: A survey. In *Proc. of the 15th Int. Symp. on Temporal Representation and Reasoning (TIME'08)*, pages 3–14. IEEE Press, 2008.

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

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

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

[Rosati, 2007] Riccardo Rosati. On conjunctive query answering in $\mathcal{EL}$. In *Proc. of the 2007 Int. Workshop on Description Logics (DL'07)*, pages 451–458, 2007.

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