Temporal Query Answering in *DL-Lite* with Negation*

Stefan Borgwardt and Veronika Thost

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

Abstract

Ontology-based query answering augments classical query answering in databases by adopting the open-world assumption and by including domain knowledge provided by an ontology. We investigate temporal query answering w.r.t. ontologies formulated in *DL-Lite*, a family of description logics that captures the conceptual features of relational databases and was tailored for efficient query answering. We consider a recently proposed temporal query language that combines conjunctive queries with the operators of propositional linear temporal logic (LTL). In particular, we consider negation in the ontology and query language, and study both data and combined complexity of query entailment.

1 Introduction

Ontologies play a central role in various applications: by linking data from heterogeneous sources to the concepts and relations described in an ontology, the integration and automated processing of the data can be considerably enhanced. In particular, queries formulated in the abstract vocabulary of the ontology can then be answered over all the linked datasets. Well-known medical domain ontologies like GALEN[1] may, for example, capture the facts that the varicella zoster virus (VZV) is a virus, that chickenpox is a VZV infection, and that a negative allergy test implies that no allergies are present, by so-called *concept inclusions*: $\mathsf{VZV} \sqsubseteq \mathsf{Virus}, \mathsf{Chickenpox} \sqsubseteq \mathsf{VZVInfection}, \mathsf{NegAllergyTest} \sqsubseteq \neg\exists\mathsf{AllergyTo}$. Here, Virus is a *concept name* that represents the set of all viruses, and $\mathsf{AllergyTo}$ is a *role name*, i.e., a binary relation, which connects patients to allergies; $\exists\mathsf{AllergyTo}$ refers to the domain of this relation. A possible data source storing patient data (e.g., allergy test results and findings) could look as follows:

PID	Name
1	Ann
2	Bob
3	Chris

PID	AllergyTest	Date
1	neg	16.01.2011
2	pos	06.01.1970
3	neg	01.06.2015

PID	Finding	Date
1	Chickenpox	13.08.2007
2	VZV-Infection	22.01.2010
3	VZV-Infection	01.11.2011

The data is then connected to the ontology by mappings [35], which in our example may link the tuple $(1, \text{Chickenpox}, 16.01.2011)$ to the facts $\mathsf{HasFinding}(1, \mathtt{x})$ and $\mathsf{Chickenpox}(\mathtt{x})$.

Ontology-based query answering (OBQA) over the above knowledge can then, for example, assist in finding appropriate participants for a clinical study, by formulating the eligibility criteria as queries over the—usually linked and heterogeneous—patient data. The following are examples of in- and exclusion conditions for a recently proposed clinical trial:[2]

- The patient should have been previously infected with VZV or previously vaccinated with VZV vaccine.
- The patient should not be allergic to VZV vaccine.

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

[1] http://www.co-ode.org/ontologies/galen

[2] https://clinicaltrials.gov/ct2/show/NCT01953900

Considering the first condition, OBQA would augment standard query answering (e.g., in SQL) w.r.t. the above ontology and data in that not only Bob and Chris but also Ann would be considered as an appropriate candidate. However, in standard OBQA, the queries neither allow for negation nor can refer to several points in time, both of which would be needed to faithfully represent the data and the stated example criteria. For this reason, we study *temporal* OBQA and allow negation in our query language.

In particular, we focus on *temporal conjunctive queries* (TCQs), which were originally proposed by [9,11]. TCQs allow to combine conjunctive queries (CQs) via the Boolean operators and the temporal operators of propositional linear temporal logic LTL [34]. For example, the above criteria can be specified with the following TCQ $\phi(x)$, to obtain all eligible patients x:

$$\big(\Diamond^- \big(\exists y.\mathsf{HasFinding}(x,y) \wedge \mathsf{VZVInfection}(y)\big) \vee \Diamond^- \big(\exists y.\mathsf{VaccinatedWith}(x,y) \wedge \mathsf{VZVVaccine}(y)\big)\big)$$
$$\wedge \neg\big(\exists y.\mathsf{AllergyTo}(x,y) \wedge \mathsf{VZVVaccine}(y)\big)$$

We here use the temporal operator 'some time in the past' ($\Diamond^-$) and consider the symbols $\mathsf{AllergyTo}$ and $\mathsf{VZVVaccine}$ to be *rigid*, which means that their interpretation does not change over time; e.g., we thus assume someone having an allergy to VZV vaccine to have this allergy for his whole life. TCQs are interpreted under the standard certain answer semantics, as opposed to the epistemic semantics for embedding CQs into a temporal language considered in [29,30], for example.

The semantics of TCQs is based on *temporal knowledge bases* (TKBs), which, in addition to the domain ontology (which is assumed to hold *globally*, i.e., at every point in time), contain finite *sequences* of fact bases. These fact bases represent the data associated to specific points in time—from the past until the *current time point* n ('now'). The problem we focus on is the evaluation of a TCQ w.r.t. such a temporal knowledge base, at the current time point.

In our setting, the information within the ontology and the fact bases does not explicitly refer to the temporal dimension, but is written in a *classical* (atemporal) description logic (DL); only the query is temporalized. In contrast, so-called *temporal DLs* [3,5,7,8,27,28,32] extend classical DLs by temporal operators, which then occur within the ontology. However, as it is shown in [7,8,27,32], most of these logics yield high reasoning complexities, even if the underlying atemporal DL allows for tractable reasoning. For that reason, lower complexities are only obtained by either considerably restricting the set of temporal operators or the DL.

A less expressive variant of TCQs called $\mathcal{ALC}$-LTL, which combines $\mathcal{ALC}$ axioms via LTL operators, has been introduced in [12]. In [11], the problem of answering TCQs over ontologies in the rather expressive DLs $\mathcal{ALC}$ and $\mathcal{SHQ}$ has been investigated (albeit without allowing transitive roles in the queries). However, reasoning in these DLs is not tractable anymore, and applications often need to process large quantities of data fast. Several lightweight logics, including *DL-Lite*, have been considered in [14], but without negation in the TCQs; in contrast, we allow negation to occur in the queries as well as in the ontology language ($\textit{DL-Lite}_{krom}/\textit{DL-Lite}_{bool}$). [7] also consider temporal variants of *DL-Lite*, but use less expressive formulas, similar to those of $\mathcal{ALC}$-LTL. In [16], TCQs are studied in the context of the lightweight DL $\mathcal{EL}$, but it is shown that reasoning is quite hard if rigid symbols are considered. This motivates our study of TCQs in DLs of the *DL-Lite* family, which was tailored for (atemporal) query answering and allows for very efficient reasoning [20,26]. Of particular interest in this setting is the question if temporal queries can be rewritten into first-order queries over a database, which can be expressed (e.g., as SQL queries) and executed using standard database systems; as it is possible in the atemporal case.

In this paper, we investigate the complexity of the TCQ entailment problem over temporal knowledge bases in several members of the extended *DL-Lite* family. In order of expressiv-

Table 1: Our results on the complexity of TCQ entailment compared to related work. All complexities except those marked with $\leq$ are tight.

	Data Complexity			Combined Complexity		
	(i)	(ii)	(iii)	(i)	(ii)	(iii)
$\textit{DL-Lite}^{[\ \mid\mathcal{H}]}_{[core\mid horn]}$	ALogTime LB: 3.6	ALogTime	ALogTime UB: 3.8	PSpace LB: [37]	PSpace	PSpace UB: 3.5
$\mathcal{EL}$ [16]	P	co-NP	co-NP	PSpace	PSpace	co-NExpTime
$\mathcal{ALC}$-$\mathcal{SHQ}$ [11]	co-NP	co-NP	$\leq$ExpTime	ExpTime	co-NExpTime	2-ExpTime
$\textit{DL-Lite}_{[krom\mid bool]}$	co-NP LB: [21]	co-NP	$\leq$ExpTime	ExpTime LB: 4.3, UB: 4.6	co-NExpTime LB: 4.4, UB: 4.6	2-ExpTime LB: 4.4
$\textit{DL-Lite}^{\mathcal{H}}_{[krom\mid bool]}$	co-NP	co-NP UB: 4.5	$\leq$ExpTime UB: 4.5	2-ExpTime LB: 4.3	2-ExpTime	2-ExpTime UB: [10]

ity, we look at $\textit{DL-Lite}_{core}/\textit{DL-Lite}_{horn}$, their variants allowing role inclusions ($^{\mathcal{H}}$), and their counterparts $\textit{DL-Lite}_{krom}/\textit{DL-Lite}_{bool}$ featuring disjunctions on the right-hand side of concept inclusions [4, 20]. In the latter, one can define negated concepts $\neg A$ via the axioms $A \sqcap \neg A \sqsubseteq \bot$ and $\top \sqsubseteq A \sqcup \neg A$ (see also Lemma 4.1). We regard both combined and data complexity and, as usual, distinguish three different settings regarding the rigid symbols:[3] (i) no symbols are allowed to be rigid, (ii) only rigid concept names are allowed, and (iii) both concepts and roles can be rigid.

Table 1 summarizes our results and shows that they are ambivalent. On the one hand, for expressive members of the extended *DL-Lite* family, we obtain at least the same complexities as for $\mathcal{SHQ}$. For logics below $\textit{DL-Lite}^{\mathcal{H}}_{horn}$, however, we have results that are considerably better than those for $\mathcal{EL}$; above all, rigid roles can often be added without affecting the complexity. Unfortunately, our ALogTime lower bound for the data complexity of TCQ entailment in $\textit{DL-Lite}_{core}$ shows that it is not possible to find a (pure) first-order rewriting of TCQs, in this setting; to see this, note that the graph of the parity function is in ALogTime and parity is not first-order definable [2]. However, within the class ALogTime, it may still be possible to find a practical *Datalog rewriting* [23], or apply the so-called *combined approach* [31]. The PSpace and co-NP lower bounds directly follow from the complexity of satisfiability in propositional LTL [37] and CQ entailment in $\textit{DL-Lite}_{krom}$ [21], respectively. Full proofs of all results can be found in the accompanying technical report [15].

2 Preliminaries

We first introduce the ontology languages and queries we consider.

2.1 *DL-Lite* Description Logics

The various description logics of the extended *DL-Lite* family augment the base formalism $\textit{DL-Lite}_{core}$ by allowing for different kinds of axioms. We focus on several of the logics presented in [4], but do not consider number restrictions.

Definition 2.1 (Syntax of *DL-Lite* Logics)**.** *Let* $\mathsf{N_C}$*,* $\mathsf{N_R}$*, and* $\mathsf{N_I}$*, be non-empty, pairwise disjoint sets of* concept names*,* role names*, and* individual names*, respectively. The set* $\mathsf{N_R^-}$ *of all* roles

[3]Note that rigid concepts can be simulated by rigid roles [12], even in $\textit{DL-Lite}_{core}$.

extends $\mathsf{N_R}$ *by* inverse roles *of the form* P^- *with* $P \in \mathsf{N_R}$. *A* basic concept *is either a concept name or an* existential restriction *of the form* $\exists R$, *where* R *is a role. An* axiom *is a* concept inclusion (CI) *of the form* $B_1 \sqcap \ldots \sqcap B_m \sqsubseteq B_{m+1} \sqcup \ldots \sqcup B_{m+n}$ $(*)$, *where* $B_1, \ldots, B_{m+n}$ *are basic concepts; a* role inclusion (RI) *of the form* $R_1 \sqsubseteq R_2$, *where* $R_1, R_2 \in \mathsf{N_R^-}$; *or an* assertion *of the form* $B(a)$ *or* $P(a,b)$, *where* B *is a basic concept,* $P \in \mathsf{N_R}$, *and* $a, b \in \mathsf{N_I}$.

For $c \in \{core, horn, krom, bool\}$, *we denote by DL-Lite*$_c$ *the logic that allows assertions and concept inclusions* $(*)$ *satisfying the following conditions: if* $c = bool$, *then* m, n *are arbitrary; if* $c = krom$, *then* $m + n \leq 2$; *if* $c = horn$, *then* $n \leq 1$; *and if* $c = core$, *then* $m + n \leq 2$ *and* $n \leq 1$. *If role inclusions are allowed in addition, this is indicated by a superscript* $\mathcal{H}$, *and we obtain the four DLs denoted by DL-Lite*$_c^{\mathcal{H}}$. *Regarding a specific DL* $\mathcal{L}$, *an* ($\mathcal{L}$-)ontology *is a finite set of concept and (if allowed in* $\mathcal{L}$*) role inclusions; and an* ABox *is a finite set of assertions. Together, an (*$\mathcal{L}$*-)ontology* $\mathcal{O}$ *and an ABox* $\mathcal{A}$ *form an* ($\mathcal{L}$-)knowledge base (KB) $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$.

In our constructions, we sometimes also consider *negated* assertions of the form $\neg B(a)$ or $\neg P(a,b)$. As usual, the empty conjunction ($\sqcap$) is denoted by $\top$ and the empty disjunction ($\sqcup$) by $\bot$. We further use the abbreviations $P^-(a,b) := P(b,a)$ and $(P^-)^- := P$, for $P \in \mathsf{N_R}$ and $a, b \in \mathsf{N_I}$. We denote by $\mathsf{N_C}(\mathcal{O})$ ($\mathsf{N_R^-}(\mathcal{O})$) the set of concept names (roles) that occur in the ontology $\mathcal{O}$, and use the notation $\mathsf{BC}(\mathcal{O})$ for the set of all basic concepts that can be built from $\mathsf{N_C}(\mathcal{O})$ and $\mathsf{N_R^-}(\mathcal{O})$. The set $\mathsf{BC}^\neg(\mathcal{O})$ contains all elements B of $\mathsf{BC}(\mathcal{O})$ and their negations $\neg B$.

Definition 2.2 (Semantics of *DL-Lite* Logics)**.** *An* interpretation $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ *consists of a non-empty set* $\Delta^\mathcal{I}$ *(called* domain*), 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* $P \in \mathsf{N_R}$ *a binary relation* $P^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$, *and to every* $a \in \mathsf{N_I}$ *an element* $a^\mathcal{I} \in \Delta^\mathcal{I}$, *such that* $a^\mathcal{I} \neq b^\mathcal{I}$ *for all* $a, b \in \mathsf{N_I}$ *with* $a \neq b$ (unique name assumption (UNA)). *We further define* $(P^-)^\mathcal{I} := \{(y,x) \mid (x,y) \in P^\mathcal{I}\}$ *and* $(\exists R)^\mathcal{I} := \{x \mid \exists\, y \in \Delta^\mathcal{I} \colon (x,y) \in R^\mathcal{I}\}$. *The interpretation* $\mathcal{I}$ satisfies *(or is a* model *of)*

- *a CI* $B_1 \sqcap \ldots \sqcap B_m \sqsubseteq B_{m+1} \sqcup \ldots \sqcup B_{m+n}$ *if* $B_1^\mathcal{I} \cap \ldots \cap B_m^\mathcal{I} \subseteq B_{m+1}^\mathcal{I} \cup \ldots \cup B_{m+n}^\mathcal{I}$;
- *an RI* $R_1 \sqsubseteq R_2$ *if* $R_1^\mathcal{I} \subseteq R_2^\mathcal{I}$;
- *a (negated) assertion* $(\neg)B(a)$ *if* $a^\mathcal{I} \in B^\mathcal{I}$ $(a^\mathcal{I} \notin B^\mathcal{I})$;
- *a (negated) assertion* $(\neg)P(a,b)$ *if* $(a^\mathcal{I}, b^\mathcal{I}) \in P^\mathcal{I}$ $((a^\mathcal{I}, b^\mathcal{I}) \notin P^\mathcal{I})$;
- *a knowledge base if it satisfies all axioms contained in it.*

We denote this by $\mathcal{I} \models \alpha$, *where* α *is either an axiom or a KB. A KB* $\mathcal{K}$ *is* consistent *if it has a model, and* $\mathcal{K}$ entails *an axiom* α *(written* $\mathcal{K} \models \alpha$*) if all models of* $\mathcal{K}$ *satisfy* α.

2.2 Temporal Conjunctive Queries

The temporal query language we focus on has originally been proposed in [9] for querying knowledge bases in $\mathcal{ALC}$. The queries are propositional LTL formulas in which the propositions have been replaced by CQs. They are answered over *temporal* KBs, according to a semantics that is suitably lifted from propositional worlds to interpretations.

As it is common [11,12], we assume that a subset of the concept and role names is designated as being *rigid*, which means that their interpretation is not allowed to change over time. The individual names are implicitly assumed to be rigid. We denote by $\mathsf{N_{RC}} \subseteq \mathsf{N_C}$ the rigid concept names, and by $\mathsf{N_{RR}} \subseteq \mathsf{N_R}$ the rigid role names. Names that are not rigid are called *flexible*. We denote by $\mathsf{BC_R}(\mathcal{O})$ the restriction of $\mathsf{BC}(\mathcal{O})$ to basic concepts involving only rigid names, and similarly use $\mathsf{BC_R^-}(\mathcal{O})$.

Definition 2.3 (Temporal Knowledge Base)**.** *An ontology $\mathcal{O}$ and a finite sequence of ABoxes $\mathcal{A}_i$, $0 \leq i \leq n$, form a* temporal knowledge base *(TKB) $\mathcal{K} = \langle \mathcal{O}, (\mathcal{A}_i)_{0 \leq i \leq n} \rangle$. Let $\mathfrak{I} = (\mathcal{I}_i)_{i \geq 0}$ be an infinite sequence of interpretations $\mathcal{I}_i = (\Delta, \cdot^{\mathcal{I}_i})$ over a fixed domain Δ (constant domain assumption). $\mathfrak{I}$ is a model of $\mathcal{K}$ (written $\mathfrak{I} \models \mathcal{K}$) if*

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

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

Definition 2.4 (Syntax of TCQs)**.** *Let $\mathsf{N_V}$ be a set of variables. A* conjunctive query *(CQ) is of the form $\phi = \exists x_1, \ldots, x_m.\psi$, where $x_1, \ldots, x_m \in \mathsf{N_V}$ and ψ is a finite conjunction of atoms of the form $A(t)$ (*concept atom*) or $P(t, t')$ (*role atom*), for $A \in \mathsf{N_C}$, $P \in \mathsf{N_R}$ and $t, t' \in \mathsf{N_I} \cup \mathsf{N_V}$. The empty conjunction is denoted by* true*, and we write $\alpha \in \phi$ if the atom α occurs in ϕ.* Temporal conjunctive queries *(TCQs) are defined inductively, given a CQ ψ and TCQs ϕ_1 and ϕ_2:*

$$\phi := \psi \mid \neg\phi_1 \ \textit{(negation)} \mid \phi_1 \wedge \phi_2 \ \textit{(conjunction)} \mid$$
$$\bigcirc \phi_1 \ \textit{(next)} \mid \bigcirc^- \phi_1 \ \textit{(previous)} \mid \phi_1 \,\mathsf{U}\, \phi_2 \ \textit{(until)} \mid \phi_1 \,\mathsf{S}\, \phi_2 \ \textit{(since)}.$$

We denote the set of individual names occurring in a TCQ ϕ by $\mathsf{N_I}(\phi)$, the set of variables occurring in ϕ by $\mathsf{N_V}(\phi)$, and the set of free variables of ϕ by $\mathsf{N_{FV}}(\phi)$. A TCQ ϕ with $\mathsf{N_{FV}}(\phi) = \emptyset$ is a *Boolean* TCQ. As usual, we use the following abbreviations: false for $\neg\mathsf{true}$, $\phi_1 \vee \phi_2$ (disjunction) for $\neg(\neg\phi_1 \wedge \phi_2)$, $\Diamond\phi_1$ (eventually) for $\mathsf{true}\,\mathsf{U}\,\phi_1$, $\Box\phi_1$ (always) for $\neg\Diamond\neg\phi_1$, and analogously for the past: $\Diamond^-\phi_1$ for $\mathsf{true}\,\mathsf{S}\,\phi_1$, and $\Box^-\phi_1$ for $\neg\Diamond^-\neg\phi_1$. A *CQ-literal* is either a CQ or a negated CQ, and a *union of CQs* (UCQ) is a disjunction of CQs.

The semantics of Boolean CQs and TCQs is the standard one [1, 11].

Definition 2.5 (Semantics of TCQs)**.** *Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be an interpretation and ψ be a Boolean CQ. A mapping $\pi \colon \mathsf{N_V}(\psi) \cup \mathsf{N_I}(\psi) \to \Delta^{\mathcal{I}}$ is a* homomorphism *of ψ into $\mathcal{I}$ if it satisfies $\pi(a) = a^{\mathcal{I}}$, for all $a \in \mathsf{N_I}(\psi)$; $\pi(t) \in A^{\mathcal{I}}$, for all $A(t) \in \psi$; and $(\pi(t), \pi(t')) \in P^{\mathcal{I}}$, for all $P(t, t') \in \psi$. We say that $\mathcal{I}$ is a* model *of ψ (written $\mathcal{I} \models \psi$) if there is such a homomorphism.*

Let now ϕ be a Boolean TCQ and $\mathfrak{I} = (\mathcal{I}_i)_{i \geq 0}$ be an infinite sequence of interpretations. We define the satisfaction relation $\mathfrak{I}, i \models \phi$, where $i \geq 0$, by induction on the structure of ϕ:

$\mathfrak{I}, i \models \psi$	*iff*	$\mathcal{I}_i \models \psi$	*(if ψ is a CQ)*
$\mathfrak{I}, i \models \neg\phi_1$	*iff*	$\mathfrak{I}, i \not\models \phi_1$	
$\mathfrak{I}, i \models \phi_1 \wedge \phi_2$	*iff*	$\mathfrak{I}, i \models \phi_1$ *and* $\mathfrak{I}, i \models \phi_2$	
$\mathfrak{I}, i \models \bigcirc\phi_1$	*iff*	$\mathfrak{I}, i+1 \models \phi_1$	
$\mathfrak{I}, i \models \bigcirc^-\phi_1$	*iff*	$i > 0$ *and* $\mathfrak{I}, i-1 \models \phi_1$	
$\mathfrak{I}, i \models \phi_1 \,\mathsf{U}\, \phi_2$	*iff*	*there is $k \geq i$ with $\mathfrak{I}, k \models \phi_2$ and $\mathfrak{I}, j \models \phi_1$, for all j, $i \leq j < k$*	
$\mathfrak{I}, i \models \phi_1 \,\mathsf{S}\, \phi_2$	*iff*	*there is k, $0 \leq k \leq i$, with $\mathfrak{I}, k \models \phi_2$ and $\mathfrak{I}, j \models \phi_1$, for all j, $k < j \leq i$*	

Given a TKB $\mathcal{K} = \langle \mathcal{O}, (\mathcal{A}_i)_{0 \leq i \leq n} \rangle$, $\mathfrak{I}$ is a model *of ϕ w.r.t. $\mathcal{K}$ if $\mathfrak{I} \models \mathcal{K}$ and $\mathfrak{I}, n \models \phi$; and ϕ is* satisfiable *w.r.t. $\mathcal{K}$ if it has a model w.r.t. $\mathcal{K}$. Furthermore, ϕ is* entailed *by $\mathcal{K}$ (written $\mathcal{K} \models \phi$) if every model of $\mathcal{K}$ is also a model of ϕ.*

Especially note that models of TCQs satisfy them at the current time point n. We often consider conjunctions of CQ-literals ϕ that contain no temporal operators. Then, the satisfaction of ϕ by an infinite sequence of interpretations $\mathfrak{I} = (\mathcal{I}_i)_{i \geq 0}$ at time point i only depends on $\mathcal{I}_i$. For simplicity, we then may write $\mathcal{I}_i \models \phi$ instead of $\mathfrak{I}, i \models \phi$, and use this notation also for UCQs. In this context, it is sufficient to deal with classical knowledge bases $\mathcal{K} = \langle \mathcal{O}, \mathcal{A} \rangle$, which can be seen as TKBs with only one ABox. We now define the semantics of non-Boolean TCQs.

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

As usual, the question of finding all certain answers to a TCQ can be reduced to the entailment problem for Boolean TCQs. The latter, being a decision problem, can be studied from the point of view of computational complexity theory. We determine the complexity of entailment via the satisfiability problem, which has the same complexity as the complement of the entailment problem [11]. We consider two kinds of complexity measures: combined and data complexity. For combined complexity, all parts of the input, meaning the TCQ and the entire TKB, are taken into account. In contrast, for data complexity, the TCQ and the global ontology are assumed to be constant, such that the complexity is measured only w.r.t. the data, the sequence of ABoxes. 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 ($\mathsf{N_{RR}} \neq \emptyset$).

2.3 On Upper Bounds

In this section, we recall a general approach to solve the TCQ satisfiability problem [11, 12], which we apply in this paper. In the following, let $\mathcal{K} = \langle \mathcal{O}, (\mathcal{A}_i)_{0 \leq i \leq n} \rangle$ be a TKB and ϕ be a Boolean TCQ. For ease of presentation, we assume w.l.o.g. that all concept and role names occurring in $(\mathcal{A}_i)_{0 \leq i \leq n}$ or ϕ also occur in $\mathcal{O}$, and that the individual names 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 *DL-Lite*—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 $p_1, \ldots, p_m$ in place of the CQs $\alpha_1, \ldots, \alpha_m$ from ϕ (where each α_i was replaced by p_i). The formula ϕ^{p} is a propositional LTL-formula [34]. The semantics of propositional LTL is defined analogously to Definition 2.5, based on *LTL-structures* $\mathfrak{J} = (w_i)_{i \geq 0}$, which consist of *worlds* $w_i \subseteq \{p_1, \ldots, p_m\}$ that describe which propositional variables are true at a given time point $i \geq 0$.

We additionally consider a set $\mathcal{S} = \{X_1, \ldots, X_k\} \subseteq 2^{\{p_1, \ldots, p_m\}}$, which restricts the worlds that are allowed to occur in an LTL-structure satisfying ϕ^{p} at time point n, and a mapping $\iota\colon \{0, \ldots, n\} \to \{1, \ldots, k\}$, which assigns a world $X_{\iota(i)}$ to each of the input ABoxes $\mathcal{A}_i$.

Definition 2.7 (t-satisfiability)**.** *The LTL-formula ϕ^{p} is* t-satisfiable *w.r.t. $\mathcal{S}$ and ι if there is an LTL-structure $\mathfrak{J} = (w_i)_{i \geq 0}$ such that $\mathfrak{J}, n \models \phi^{\mathsf{p}}$, $w_i \in \mathcal{S}$ for all $i \geq 0$, and $w_i = X_{\iota(i)}$ for all i, $0 \leq i \leq n$.*

However, finding $\mathcal{S}$ and ι and then testing t-satisfiability is not sufficient. We must also ensure that $\mathcal{S}$ can indeed be induced by a model of $\mathcal{K}$, in the following sense.

Definition 2.8 (r-satisfiability)**.** *The set $\mathcal{S}$ is* r-satisfiable *w.r.t. ι and $\mathcal{K}$ if there are interpretations $\mathcal{J}_1, \ldots, \mathcal{J}_k, \mathcal{I}_0, \ldots, \mathcal{I}_n$, which share the same domain, respect rigid names,*[4] *and are models of $\mathcal{O}$; additionally, each $\mathcal{J}_i$, $1 \leq i \leq k$, is a model of $\chi_i := \bigwedge_{p_j \in X_i} \alpha_j \wedge \bigwedge_{p_j \in \overline{X_i}} \neg \alpha_j$ (where $\overline{X_i} := \{p_1, \ldots, p_m\} \setminus X_i$), and each $\mathcal{I}_i$, $0 \leq i \leq n$, is a model of $\mathcal{A}_i$ and $\chi_{\iota(i)}$.*

The following was shown in [11] for $\mathcal{SHQ}$, and the proof remains valid in our setting.

Lemma 2.9. *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.*

[4] This is defined as for infinite sequences of interpretations (see Definition 2.3).

3 Temporal Query Entailment in $\boldsymbol{DL\text{-}Lite}^{\mathcal{H}}_{horn}$

For DLs below $DL\text{-}Lite^{\mathcal{H}}_{horn}$, we first show that we can separate the r-satisfiability condition into independent tests (i.e., tests for each time point), similar to a construction in [16]. For simplicity, we restrict the presentation here to the case that all queries are *r-simple*, i.e., they do not contain role atoms $R(t, t')$ for which there are a flexible role S and a rigid role S' (or their inverses) such that $\mathcal{O} \models S \sqsubseteq S' \sqsubseteq R$. In the technical report, we describe an intricate construction that also covers the case when queries may not be r-simple [15]. Note that without role inclusions or rigid roles all queries are r-simple.

3.1 Characterizing r-satisfiability

We first introduce the auxiliary notions of *consequences* and *witness queries* w.r.t. an ontology $\mathcal{O}$.

Definition 3.1 (Consequences)**.** *For a CQ α, let α' denote the CQ obtained by instantiating all variables x in α with fresh individual names a_x. The set of* consequences *of α is defined as*

$$\mathcal{C}_{\mathcal{O}}(\alpha) := \{C(a) \mid C \in \mathsf{BC}^{\neg}_{\mathsf{R}}(\mathcal{O}),\ a \in \mathsf{N}_{\mathsf{I}}(\alpha'),\ \mathcal{O} \models \bigsqcap \mathsf{BC}^{-}(a, \alpha') \sqsubseteq C\} \cup$$
$$\{R(a, b) \mid R \in \mathsf{N}^{-}_{\mathsf{RR}}(\mathcal{O}),\ S(a, b) \in \alpha',\ \mathcal{O} \models S \sqsubseteq R\},$$

where $\mathsf{BC}^{-}(a, \alpha') := \{A \in \mathsf{N}_{\mathsf{C}} \mid A(a) \in \alpha'\} \cup \{\exists R \mid R \in \mathsf{N}^{-}_{\mathsf{R}},\ R(a, b) \in \alpha'\}$.

We collect all the new individual names a_x in the set $\mathsf{N}^{\mathsf{aux}}_{\mathsf{I}}$. Intuitively, the consequences of α describe those structures that, if α is satisfied at one time point, have to exist at all other time points, because of the rigid names.

Second, we consider so-called *witness queries* for a CQ α (w.r.t. $\mathcal{O}$), which are CQs using only rigid names and whose satisfaction implies the satisfaction of α. Hence, if such a witness query is satisfied at some time point, α must be satisfied at every time point. Due to space constraints, we cannot include the full definition of witness queries here, but only state an important lemma (see [15] for details).

Lemma 3.2. *Let ψ be a witness query for a CQ α and $\mathcal{I} \models \mathcal{O}$. Then, $\mathcal{I} \models \psi$ implies $\mathcal{I} \models \alpha$.*

Let now ϕ be an r-simple Boolean TCQ and $\mathcal{K} = \langle \mathcal{O}, (\mathcal{A}_i)_{0 \leq i \leq n} \rangle$ be a $DL\text{-}Lite^{\mathcal{H}}_{horn}$-TKB. We further assume that $\mathcal{S} = \{X_1, \ldots, X_k\} \subseteq 2^{\{p_1, \ldots, p_m\}}$ and $\iota \colon \{0, \ldots, n\} \to \{1, \ldots, k\}$ are given. We later describe how to actually obtain $\mathcal{S}$ and ι to show the claimed upper bounds. We denote by $\mathcal{Q}_\phi$ the set of CQs occurring in ϕ, and assume w.l.o.g. that they use disjoint variables. We denote by Q_i the set $\{\alpha_j \mid p_j \in X_i\}$, and by $\mathcal{A}_{Q_i}$ the ABox obtained from Q_i by instantiating all variables x in CQs $\alpha \in Q_i$ with the corresponding individual names a_x from $\mathsf{N}^{\mathsf{aux}}_{\mathsf{I}}$. For ease of presentation, for all i, $1 \leq i \leq k$, we define the set $\mathcal{A}_{n+i} = \emptyset$ and extend ι such that $\iota(n+i) := i$.

We now formalize the additional information we guess, in order to be able to split the r-satisfiability test.

Definition 3.3 (r-complete)**.** *An* ABox type *is a set $\mathcal{A}_{\mathsf{R}}$ of rigid (negated) assertions $(\neg)\alpha$ over $\mathsf{N}_{\mathsf{I}}(\mathcal{K})$, $B \in \mathsf{BC}_{\mathsf{R}}(\mathcal{O})$, and $R \in \mathsf{N}^{-}_{\mathsf{RR}}(\mathcal{O})$, such that $\alpha \in \mathcal{A}_{\mathsf{R}}$ iff $\neg\alpha \notin \mathcal{A}_{\mathsf{R}}$. For a triple $(\mathcal{A}_{\mathsf{R}}, Q_{\mathsf{R}}, Q^{\neg}_{\mathsf{R}})$, where $\mathcal{A}_{\mathsf{R}}$ is an ABox type, and $Q_{\mathsf{R}}, Q^{\neg}_{\mathsf{R}} \subseteq \mathcal{Q}_\phi$, we define $\mathcal{K}^i_{\mathsf{R}} := \langle \mathcal{O}, \mathcal{A}_{\mathsf{R}} \cup \mathcal{A}_{Q_{\mathsf{R}}} \cup \mathcal{A}_{Q_{\iota(i)}} \cup \mathcal{A}_i \rangle$, where $\mathcal{A}_{Q_{\mathsf{R}}} := \bigcup_{\alpha \in Q_{\mathsf{R}}} \mathcal{C}_{\mathcal{O}}(\alpha)$. We call this triple* r-complete *(w.r.t. $\mathcal{S}$ and ι) if the following hold:*

(R1) *For all $i \in \{0, \ldots, n+k\}$, $\mathcal{K}^i_{\mathsf{R}}$ is consistent.*

(R2) *For all $i \in \{0, \ldots, n+k\}$ and $p_j \in \overline{X_{\iota(i)}}$, we have $\mathcal{K}^i_{\mathsf{R}} \not\models \alpha_j$.*

(R3) *If there is an* $X \in \mathcal{S}$ *such that* $p_j \in X$*, then* $\alpha_j \in Q_{\mathsf{R}}$*.*

(R4) *If there is an* $X \in \mathcal{S}$ *such that* $p_j \in \overline{X}$*, then* $\alpha_j \in Q_{\mathsf{R}}^{\neg}$*.*

(R5) *For all* $i \in \{0, \ldots, n+k\}$*, all CQs* $\alpha \in Q_{\mathsf{R}}^{\neg}$*, and all witnesses queries* ψ *of* α *w.r.t.* $\mathcal{O}$*, we have* $\mathcal{K}_{\mathsf{R}}^i \not\models \psi$*.*

The idea thus is to fix the interpretation of the rigid names on all named individuals ($\mathcal{A}_{\mathsf{R}}$) and to specify the CQs that have to be satisfied at least once (Q_{R}), and those that are allowed to occur negatively ($Q_{\mathsf{R}}^{\neg}$). Condition (R1) ensures that each $\mathcal{K}_{\mathsf{R}}^i$ has a model, and, in particular, the *canonical model* [17]. Because of the ABox $\mathcal{A}_{Q_{\iota(i)}}$, this implies that the CQs α_j with $p_j \in X_{\iota(i)}$ are satisfied. Condition (R2) implies the dual fact, that the CQs α_j with $p_j \in \overline{X_{\iota(i)}}$ are not satisfied by the canonical model of $\mathcal{K}_{\mathsf{R}}^i$; hence, this model satisfies all conjuncts of $\chi_{\iota(i)}$ (see Definition 2.8). With (R3) and the ABox $\mathcal{A}_{Q_{\mathsf{R}}}$, we ensure that the rigid structures $\mathcal{C}_{\mathcal{O}}(\alpha_j)$ implied by the satisfaction of some α_j are present at every time point. Conversely, by (R4) and (R5), for each α_j occurring negatively at some time point, there cannot be a witness query ψ of α_j that is satisfied at any other time point, since this would yield a contradiction, by Lemma 3.2. We can show that the r-satisfiability of $\mathcal{S}$ is characterized by the existence of such an r-complete tuple.

Lemma 3.4. *$\mathcal{S}$ is r-satisfiable w.r.t. ι and $\mathcal{O}$ iff there is an r-complete tuple w.r.t. $\mathcal{S}$ and ι.*

3.2 Combined Complexity

From the above characterization, we obtain a PSPACE decision procedure by adapting the Turing machine (TM) for LTL satisfiability from [37]. This TM successively guesses propositional worlds and checks whether they can be used to construct an LTL-structure that satisfies ϕ^{P}. The key insight of the previous section is that this TM does not need to store the exponentially large set $\mathcal{S}$ in order to check the conditions of Definition 3.3. Our adapted TM simply guesses a tuple $(\mathcal{A}_{\mathsf{R}}, Q_{\mathsf{R}}, Q_{\mathsf{R}}^{\neg})$ as described above and then proceeds as before, but, for each guessed world X, it additionally checks whether $\mathcal{K}_{\mathsf{R}} := \langle \mathcal{O}, \mathcal{A}_{\mathsf{R}} \cup \mathcal{A}_{Q_{\mathsf{R}}} \cup \mathcal{A}_{Q_X} \cup \mathcal{A}_i \rangle$ satisfies the conditions (R1)–(R5). Here, $\mathcal{A}_i$ is only relevant for the first $n+1$ time points, after which it is empty; and $\mathcal{A}_{Q_X}$ is the ABox obtained by instantiating all CQs α_j with $p_j \in X$. The KB $\mathcal{K}_{\mathsf{R}}$, particularly $\mathcal{A}_{Q_{\mathsf{R}}}$, is of polynomial size and can be constructed with the help of polynomially many P-tests for the entailment of certain assertions [4] (see Definition 3.1). The consistency test for (R1) can also be done in polynomial time [4]. Moreover, the non-entailment tests in (R2) and (R5) can be done in CO-NP (and thus also in PSPACE) using the non-deterministic version of the algorithm in [17]. Although there may be exponentially many witness queries, we can enumerate all of them within PSPACE. Finally, the conditions (R3) and (R4) can be verified easily.

The set $\mathcal{S}$ required for Lemma 2.9 can then be defined as the set of all worlds X encountered during a run of this Turing machine, while ι is obtained by collecting the worlds guessed for the first $n+1$ time points. Given these definitions of $\mathcal{S}$ and ι, it is easy to see that the above checks are actually equivalent to (R1)–(R5) from Definition 3.3. By Lemmas 2.9 and 3.4, the described Turing machine accepts the input $\mathcal{K}$ and ϕ iff ϕ has a model w.r.t. $\mathcal{K}$. Since we do not have to store $\mathcal{S}$ explicitly and all checks can be done with a nondeterministic TM using only polynomial space, according to [36], TCQ entailment can be decided in PSPACE. We show this result in more detail in [15], even for TCQs that are not r-simple. The corresponding lower bound follows from the satisfiability problem for propositional LTL [37].

Theorem 3.5. *TCQ entailment in DL-Lite$^{\mathcal{H}}_{horn}$ is in* PSPACE *w.r.t. combined complexity.*

3.3 Data Complexity

Regarding data complexity, we first prove that TCQ entailment is not FO-rewritable, not even for $\mathit{DL\text{-}Lite}_{core}$, by adapting a proof from [5, 6].

Theorem 3.6. *TCQ entailment in* $\mathit{DL\text{-}Lite}_{core}$ *is* ALogTime*-hard w.r.t. data complexity, even if* $\mathsf{N_{RC}} = \emptyset$ *and* $\mathsf{N_{RR}} = \emptyset$*.*

Proof. There are regular languages that are hard for DLogTime-uniform NC^1 (under constant-depth reductions) [13, Theorem 7], which is equal to ALogTime [33, Lemma 7.2]. Furthermore, for any regular language, there is an NFA recognizing it. We hence can establish ALogTime-hardness by considering an arbitrary NFA $\mathfrak{A}$ and reducing its word problem to TCQ entailment. We consider concept names A_a and Q_q for characters a of the input alphabet and states q, respectively, and define the TCQ

$$\phi := \Box^-\Big(\bigwedge_{q \to_a q'} \big(Q_q(a) \wedge A_a(a)\big) \to \bigcirc Q_{q'}(a)\Big) \to Q_{q_1}(a),$$

where q_1 is the accepting state of $\mathfrak{A}$. Given an input word $w = a_0 \dots a_{n-1}$, we define the sequence of ABoxes $\mathcal{A}_w = (\mathcal{A}_i)_{0 \leq i < n}$ such that $\mathcal{A}_0 := \{Q_{q_0}(a)\}$ and $\mathcal{A}_i := \{A_{a_i}(a)\}$, for all $0 \leq i < n$, with q_0 being the initial state of $\mathfrak{A}$. Thus, $\mathfrak{A}$ accepts w iff all models of $\langle \emptyset, \mathcal{A}_w \rangle$ that satisfy the antecedent of ϕ (which means that they simulate all runs of $\mathfrak{A}$ on w), also satisfy the consequent $Q_{q_1}(a)$, which is equivalent to the entailment $\langle \emptyset, \mathcal{A}_w \rangle \models \phi$. □

We now apply the approach of Lemma 3.4 to show a matching ALogTime upper bound. All details of the construction, which again works also for TCQs that are not r-simple, can be found in [15]. Note that we cannot guess the whole tuple $(\mathcal{A}_{\mathsf{R}}, Q_{\mathsf{R}}, Q_{\mathsf{R}}^{\neg})$ in ALogTime. Instead, we show that we only have to consider a single tuple that is uniquely determined by the choice of $\mathcal{S}$. Since $\mathcal{S}$ is constant under data complexity assumptions (it depends only on ϕ), we can enumerate all possible sets $\mathcal{S}$ and all its elements X. We then provide a first-order rewriting $\mathsf{rsat}_{\mathcal{S},X}(i)$ of the conditions in Definition 3.3 as follows.

Lemma 3.7. *There is an r-complete tuple w.r.t.* $\mathcal{S}$ *and* ι *iff* $\mathsf{DB} \models \mathsf{rsat}_{\mathcal{S},X}(-1)$*, and, for all* i*,* $0 \leq i \leq n$*, we have* $\mathsf{DB} \models \mathsf{rsat}_{\mathcal{S},X_{\iota(i)}}(i)$*.*

Here, DB is the interpretation obtained from viewing the input ABoxes $(\mathcal{A}_i)_{0 \leq i \leq n}$ under the closed-world assumption, i.e., it makes exactly the assertions in $\mathcal{A}_i$ true and all others false. The value -1 refers to a special ABox "$\mathcal{A}_{-1}$" (which is empty) to model the satisfiability tests for χ_i without any ABox (see Definition 2.8).

In order to find ι and check t-satisfiability, we then define an alternating TM for the TCQ satisfiability problem. An important step is to decouple the satisfaction of the future and past operators in ϕ^{P} according to the separation theorem of [24]. Everything that concerns only the future does not depend on the ABoxes, and hence can be checked in constant time. It then only remains to check the satisfaction of those parts of ϕ^{P} that refer to the past, over the input ABoxes $\mathcal{A}_0, \dots, \mathcal{A}_n$. For this purpose, we proceed as for the PSpace TM above, but, instead of guessing the worlds X one after the other and in a linear fashion, we follow the approach sketched in Figure 1 and build a computation tree of depth $\log(n+1)$, where each branching represents a universal state. Each copy of the TM is responsible for checking the existence of an LTL-structure on the subsequence of $0, \dots, n$ represented by the current subtree; e.g., after the first branching there are two copies of the TM, responsible for the time points $0, \dots, \frac{n+1}{2}$, and $\frac{n+1}{2} + 1, \dots, n$, respectively. Before a branching, the current copy of the TM guesses two worlds

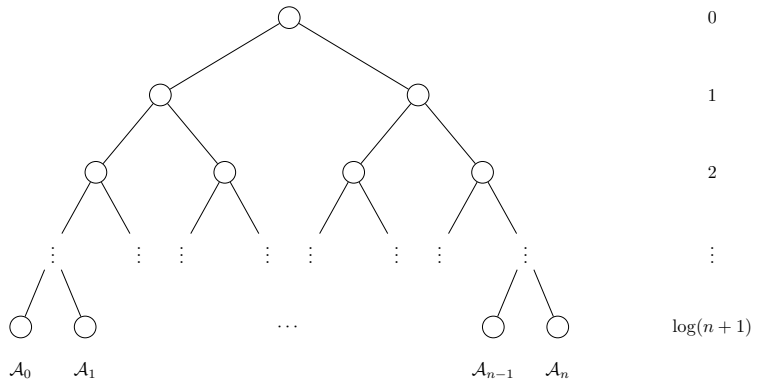

Figure 1: The computation tree of the ALogTime Turing machine for Theorem 3.8

Table 2: The rules of our transformation

CI	TCQ
$A_1 \sqsubseteq \forall R.A_2$	$\neg\exists x, y.A_1(x) \wedge R(x,y) \wedge \overline{A}_2(y)$
$A_1 \sqcap \cdots \sqcap A_m \sqsubseteq A_{m+1} \sqcup \cdots \sqcup A_{m+n}$	$\neg\exists x.A_1(x) \wedge \cdots \wedge A_m(x) \wedge \overline{A}_{m+1}(x) \wedge \cdots \wedge \overline{A}_{m+n}(x)$

$X_{\iota(i)}$ and $X_{\iota(i+1)}$, for the adjacent time points i and $i+1$ between which the next split will occur; it then checks a local condition ensuring that one can build an LTL-structure satisfying ϕ^{P} from them. Subsequently, $X_{\iota(i)}$ is given to the "left" copy of the TM, and the "right" copy receives $X_{\iota(i+1)}$. In this way, each copy knows the two worlds belonging to the left-most and right-most time point it is responsible for. In the end, the copy for time point i knows $X_{\iota(i)}$, and checks the condition $\mathsf{DB} \models \mathsf{rsat}_{S,X_{\iota(i)}}(i)$, for Lemma 3.7. This can be done in DLogTime-uniform AC^0 [1], which is a subclass of ALogTime. The remaining conditions $\mathsf{DB} \models \mathsf{rsat}_{\mathcal{S},X}(-1)$ can be checked in constant time since they do not depend on the ABoxes. In this way, each of the parallel computations of our TM only uses time logarithmic in the size of the input ABoxes. The result now follows from the fact that ALogTime is closed under complement [22].

Theorem 3.8. *TCQ entailment in DL-Lite$^{\mathcal{H}}_{horn}$ is in* ALogTime *w.r.t. data complexity.*

4 Beyond *DL-Lite*$^{\mathcal{H}}_{horn}$

For the *krom* and *bool* variants of *DL-Lite*, we first observe that we do not need to distinguish between them since all CIs of *DL-Lite*$_{bool}$ can be simulated by appropriate (negated) CQs. We can even simulate *qualified* existential restrictions on the left-hand side of CIs (or, equivalently, value restrictions on the right-hand side). For this, we use fresh concept names $\overline{A_i}$ to denote the complements of $A_i \in \mathsf{N_C}$, which can be expressed in *DL-Lite*$_{krom}$ as follows.

Lemma 4.1. *Let $(C \sqsubseteq D, \phi)$ be one of the pairs of a CI and a TCQ given in Table 2, and let $\mathcal{I}$ be a model of $\top \sqsubseteq A_i \sqcup \overline{A}_i$ and $A_i \sqcap \overline{A}_i \sqsubseteq \bot$, for all concept names A_i occurring in D. Then, we have $\mathcal{I} \models C \sqsubseteq D$ iff $\mathcal{I} \models \phi$.*

We thus have $\langle \mathcal{O}, (\mathcal{A}_i)_{0\leq i\leq n}\rangle \models \phi$ iff $\langle \mathcal{O}', (\mathcal{A}_i)_{0\leq i\leq n}\rangle \models ((\Box\Box^{-}\psi) \rightarrow \phi)$; where $\mathcal{O}'$ is obtained by removing all CIs of the forms listed in Table 2 from $\mathcal{O}$ and adding the necessary CIs to express

the complements $\overline{A}_i$; and ψ is the conjunction of the negated CQs simulating the removed CIs. We will use this result to prove some of the lower bounds later on. An immediate consequence is the following reduction.

Corollary 4.2. *TCQ entailment in DL-Lite$_{bool}$ can be polynomially reduced to TCQ entailment in DL-Lite$_{krom}$.*

Hence, we can prove our lower bounds for *DL-Lite$_{bool}$*, and the upper bounds for *DL-Lite$_{krom}$*.

4.1 What Makes it Hard

We directly obtain two rather strong lower bounds, even without considering rigid symbols, from ExpTime-hardness of UCQ entailment in *DL-Lite$_{bool}$* [19, Corollary 2] and 2-ExpTime-hardness of UCQ entailment in *DL-Lite$^{\mathcal{H}}_{bool}$* [18, Theorem 12], by Corollary 4.2.

Theorem 4.3. *Under combined complexity, TCQ entailment is* ExpTime*-hard in DL-Lite$_{krom}$, and* 2-ExpTime*-hard in DL-Lite$^{\mathcal{H}}_{krom}$, even if* $\mathsf{N_{RC}} = \emptyset$ *and* $\mathsf{N_{RR}} = \emptyset$.

For *DL-Lite$_{krom}$*—without role inclusions—, we show two other lower bounds, depending on whether rigid role names are allowed next to rigid concept names. We show that these bounds are the same as in $\mathcal{ALC}$ [11, 12]. We obtain them by modifying the hardness proofs for the satisfiability problem in $\mathcal{EL}_\bot$-LTL with rigid concept names [16] (which, in turn, is an adaptation of a proof in [12]) and in $\mathcal{ALC}$-LTL with rigid role names [12]. The latter adaptation involves many changes due to the fact that we cannot express qualified existential restrictions on the right-hand side of *DL-Lite$_{krom}$*-CIs. The needed value restrictions on the right-hand side of CIs can be simulated using Lemma 4.1.

Theorem 4.4. *Under combined complexity assumptions, TCQ entailment in DL-Lite$_{krom}$ is* co-NExpTime*-hard if* $\mathsf{N_{RR}} = \emptyset$*, and* 2-ExpTime*-hard in general.*

For data complexity, the lower bound of co-NP follows from co-NP-hardness of CQ entailment in *DL-Lite$_{krom}$* [20, Theorem 48].

4.2 Upper Bounds

The upper bounds regarding data complexity are obtained by the following reduction to TCQ entailment in $\mathcal{ALCH}$. Let ϕ be a TCQ and $\mathcal{K}$ a *DL-Lite$^{\mathcal{H}}_{bool}$*-TKB. The idea is to view all inverse roles R^- as role names, and introduce the CI $\exists R.(\neg\exists R^-.\top) \sqsubseteq \bot$, for each $R \in \mathsf{N}^-_{\mathsf{R}}(\mathcal{O})$; and the RI $R^- \sqsubseteq S^-$ for each $R \sqsubseteq S \in \mathcal{O}$. We call the resulting $\mathcal{ALCH}$-TKB $\mathcal{K}'$. The TCQ ϕ' is obtained from ϕ by replacing each CQ $\psi \in \phi$ by the disjunction of all its *variants*, in which some role atoms $R(t, t')$ may be replaced by $R^-(t', t)$. We then have $\mathcal{K}' \models \phi'$ iff $\mathcal{K} \models \phi$. Since the ABoxes remain essentially unchanged, the following bounds follow from [11].

Theorem 4.5. *Under data complexity assumptions, TCQ entailment in DL-Lite$^{\mathcal{H}}_{bool}$ is in* co-NP *if* $\mathsf{N_{RR}} = \emptyset$*, and in* ExpTime *in general.*

Note that, for $\mathcal{ALCH}$, a tight upper bound for the case $\mathsf{N_{RR}} \neq \emptyset$ w.r.t. data complexity is still open. Hence, we also have the same gap between co-NP and ExpTime here.

For combined complexity, the 2-ExpTime upper bound for TCQ entailment in $\mathcal{SHIQ}$ [10], which includes *DL-Lite$^{\mathcal{H}}_{bool}$*, is inherited. The two remaining upper bounds, without role inclusions, can be shown as in [11], taking into account Corollary 4.2 and the fact that UCQ entailment for frontier-one disjunctive inclusion dependencies, and hence in *DL-Lite$_{krom}$*, can be decided in ExpTime [18, Theorem 8].

Theorem 4.6. *Under combined complexity assumptions, TCQ entailment in DL-Lite$_{bool}$ is in* ExpTime *if* $\mathsf{N_{RC}} = \emptyset$ *and* $\mathsf{N_{RR}} = \emptyset$*, and in* co-NExpTime *if* $\mathsf{N_{RR}} = \emptyset$*.*

Proof Sketch. Recall that we show the complementary results by regarding TCQ satisfiability. Let $\mathcal{K} = \langle \mathcal{O}, (\mathcal{A}_i)_{0 \leq i \leq n} \rangle$ be a TKB and ϕ be a Boolean TCQ. Observe first that in the absence of rigid names the satisfiability tests of Definition 2.8 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. $\mathcal{O}$. As in [11], these exponentially many satisfiability tests can be reduced to entailment tests for UCQs of polynomial size, and hence we only have to consider exponentially many ExpTime-tests [18] to construct $\mathcal{S}$. We can further enumerate all possible mappings ι in exponential time and, for each, check the consistency of the conjunction $\chi_{\iota(i)} \wedge \bigwedge_{\alpha \in \mathcal{A}_i} \alpha$ in ExpTime, by the same arguments. For each ι that passes these tests, the t-satisfiability can be checked in ExpTime, by [11, Lemma 4.12]. According to Lemma 2.9, we can thus decide the satisfiability problem in ExpTime, which implies the same complexity for the entailment problem.

For the case where we have rigid concept names, we can guess $\mathcal{S}$ and ι as required for Lemma 2.9 in NExpTime, and the t-satisfiability can again be checked in exponential time. We can further reduce the r-satisfiability problem as in [11, Lemma 6.2] to exponentially many UCQ non-entailment problems (of polynomial size) with an additional side condition: Given a set $\mathcal{D} \subseteq 2^{\mathsf{N_{RC}}(\mathcal{O})}$, which may be of exponential size, we need the interpretations witnessing these non-entailments to *respect* $\mathcal{D}$; in the sense that every domain element satisfies exactly the rigid concept names given by an element of $\mathcal{D}$, and that, conversely, every element of $\mathcal{D}$ is represented in this way by at least one domain element. This condition ensures that we can later join the independent interpretations into a temporal model of the TCQ that respects the rigid names. These non-entailment problems w.r.t. $\mathcal{D}$ can also be decided in ExpTime, by an adaptation of the procedure in [18, Theorem 8], which yields a NExpTime upper bound for TCQ satisfiability. □

5 Summary and Outlook

We have analyzed the computational complexity of TCQ entailment in several members of the extended *DL-Lite* family of Description Logics. As it can be seen in Table 1, many of these fragments turned out to be very complex. Nevertheless, for several others, we obtained encouraging results, which are even better than those for $\mathcal{EL}$. Especially the data complexity of ALogTime implies that it might be possible to solve the entailment problem by applying the combined approach of [31] or by rewriting the TCQ into a Datalog query to be evaluated over a database [23].

We further showed that the combined complexity of PSpace inherited from LTL does not increase—even if rigid role names are considered. If we make the reasonable assumption that all relevant information about these names (e.g., which patients have no allergy and thus do not belong to the rigid concept $\exists\mathsf{AllergyTo}$) is available before query answering, then we do not need to guess the ABox type $\mathcal{A}_{\mathsf{R}}$. It remains to be seen whether existing PSpace-algorithms for LTL [25] can be efficiently combined with reasoning procedures for *DL-Lite* [31].

In future work, it would be worth to study other variants of *DL-Lite* [4] since it might be possible to go beyond *DL-Lite*$^{\mathcal{H}}_{horn}$ while keeping its complexity. We could also combine our approach with other temporal query formalisms based on *DL-Lite* [5–7], and investigate how to transfer and combine existing constructions and results. On the practical side, it would be interesting to see how TCQs perform in applications; some prototype implementations have already been described [38].

References

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

[2] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

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

[4] Alessandro Artale, Diego Calvanese, Roman Kontchakov, and Michael Zakharyaschev. The *DL-Lite* family and relations. *Journal of Artificial Intelligence Research*, 36:1–69, 2009.

[5] Alessandro Artale, Roman Kontchakov, Alisa Kovtunova, Vladislav Ryzhikov, Frank Wolter, and Michael Zakharyaschev. Temporal OBDA with LTL and *DL-Lite*. In Meghyn Bienvenu, Magdalena Ortiz, Riccardo Rosati, and Mantas Šimkus, editors, *Proc. of the 27th Int. Workshop on Description Logics (DL'14)*, volume 1193 of *CEUR Workshop Proceedings*, pages 21–32, 2014.

[6] Alessandro Artale, Roman Kontchakov, Alisa Kovtunova, Vladislav Ryzhikov, Frank Wolter, and Michael Zakharyaschev. First-order rewritability of ontology-mediated temporal queries. In Qiang Yang, editor, *Proc. of the 24th Int. Joint Conf. on Artificial Intelligence (IJCAI'15)*, pages 2706–2712. AAAI Press, 2015.

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

[8] 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.

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

[10] Franz Baader, Stefan Borgwardt, and Marcel Lippmann. Temporal conjunctive queries in expressive DLs with non-simple roles. LTCS-Report 15-17, Chair for Automata Theory, TU Dresden, Germany, 2015. See `https://ddll.inf.tu-dresden.de/web/Ver%C3%B6ffentlichungen/2015/en#Technical_Reports`.

[11] Franz Baader, Stefan Borgwardt, and Marcel Lippmann. Temporal query entailment in the description logic $\mathcal{SHQ}$. *Journal of Web Semantics*, 33:71–93, 2015.

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

[13] David A. Mix Barrington, Kevin Compton, Howard Straubing, and Denis Thérien. Regular languages in NC^1. *Journal of Computer and System Sciences*, 44(3):478–499, 1992.

[14] Stefan Borgwardt, Marcel Lippmann, and Veronika Thost. Temporalizing rewritable query languages over knowledge bases. *Journal of Web Semantics*, 33:50–70, 2015.

[15] Stefan Borgwardt and Veronika Thost. Temporal query answering in *DL-Lite* with negation. LTCS-Report 15-16, Chair for Automata Theory, TU Dresden, Germany, 2015. See `https://ddll.inf.tu-dresden.de/web/Ver%C3%B6ffentlichungen/2015/en#Technical_Reports`.

[16] Stefan Borgwardt and Veronika Thost. Temporal query answering in the description logic $\mathcal{EL}$. In Qiang Yang, editor, *Proc. of the 24th Int. Joint Conf. on Artificial Intelligence (IJCAI'15)*, pages 2819–2825. AAAI Press, 2015.

[17] Elena Botoeva, Alessandro Artale, and Diego Calvanese. Query rewriting in $\textit{DL-Lite}_{horn}^{(\mathcal{HN})}$. In Volker Haarslev, David Toman, and Grant Weddell, editors, *Proc. of the 23rd Int. Workshop on*

Description Logics (DL'10), volume 573 of *CEUR Workshop Proceedings*, pages 267–278, 2010.

[18] Pierre Bourhis, Michael Morak, and Andreas Pieris. The impact of disjunction on query answering under guarded-based existential rules. In Francesca Rossi, editor, *Proc. of the 23rd Int. Joint Conf. on Artificial Intelligence (IJCAI'13)*, pages 796–802. AAAI Press, 2013.

[19] Pierre Bourhis, Michael Morak, and Andreas Pieris. Acyclic query answering under guarded disjunctive existential rules and consequences to DLs. In Meghyn Bienvenu, Magdalena Ortiz, Riccardo Rosati, and Mantas Šimkus, editors, *Proc. of the 27th Int. Workshop on Description Logics (DL'14)*, volume 1193 of *CEUR Workshop Proceedings*, pages 100–111, 2014.

[20] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. *Journal of Automated Reasoning*, 39(3):385–429, 2007.

[21] Diego Calvanese, Giuseppe De Giacomo, Domenico Lemho, Maurizio Lenzerini, and Riccardo Rosati. *DL-Lite*: Tractable description logics for ontologies. In Manuela M. Veloso and Subbaro Kambhampati, editors, *Proc. of the 20th Nat. Conf. on Artificial Intelligence (AAAI'05)*, pages 602–607. AAAI Press, 2005.

[22] Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. Alternation. *Journal of the ACM*, 28(1):114–133, 1981.

[23] Evgeny Dantsin, Thomas Eiter, Georg Gottlob, and Andrei Voronkov. Complexity and expressive power of logic programming. In *Proc. of the 12th Annual IEEE Conf. on Computation Complexity (CCC'97)*, pages 82–101. IEEE Press, 1997.

[24] Dov M. Gabbay. The declarative past and imperative future: Executable temporal logic for interactive systems. In *Temporal Logic in Specification*, pages 409–448. Springer-Verlag, 1987.

[25] Paul Gastin and Denis Oddoux. Fast LTL to Büchi automata translation. In Gérard Berry, Hubert Comon, and Alain Finkel, editors, *Proc. of the 13th Int. Conf. on Computer Aided Verification (CAV'01)*, volume 2102 of *Lecture Notes in Computer Science*, pages 53–65. Springer-Verlag, 2001.

[26] Martin Giese, Peter Haase, Ernesto Jiménez-Ruiz, Davide Lanti, Özgür Özçep, Martin Rezk, Riccardo Rosati, Ahmet Soylu, Guillermo Vega-Gorgojo, Arild Waaler, and Guohui Xiao. Optique: Zooming in on Big Data. *IEEE Computer*, 48(3):60–67, 2015.

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

[28] Víctor Gutiérrez-Basulto, Jean Christoph Jung, and Thomas Schneider. Lightweight temporal description logics with rigid roles and restricted TBoxes. In Qiang Yang, editor, *Proc. of the 24th Int. Joint Conf. on Artificial Intelligence (IJCAI'15)*, pages 3015–3021. AAAI Press, 2015.

[29] Víctor Gutiérrez-Basulto and Szymon Klarman. Towards a unifying approach to representing and querying temporal data in description logics. In Markus Krötzsch and Umberto Straccia, editors, *Proc. of the 6th Int. Conf. on Web Reasoning and Rule Systems (RR'12)*, volume 7497 of *Lecture Notes in Computer Science*, pages 90–105. Springer-Verlag, 2012.

[30] Szymon Klarman and Thomas Meyer. Querying temporal databases via OWL 2 QL. In Roman Kontchakov and Marie-Laure Mugnier, editors, *Proc. of the 8th Int. Conf. on Web Reasoning and Rule Systems (RR'14)*, volume 8741 of *Lecture Notes in Computer Science*, pages 92–107. Springer-Verlag, 2014.

[31] Roman Kontchakov, Carsten Lutz, David Toman, Frank Wolter, and Michael Zakharyaschev. The combined approach to query answering in *DL-Lite*. In Fangzhen Lin, Ulrike Sattler, and Miroslaw Truszczynski, editors, *Proc. of the 12th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'10)*, pages 247–257. AAAI Press, 2010.

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

[33] David A. Mix Barrington, Neil Immerman, and Howard Straubing. On uniformity within NC^1. *Journal of Computer and System Sciences*, 41(3):274–306, 1990.

[34] 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.

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

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

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

[38] Veronika Thost, Jan Holste, and Özgür Özçep. On implementing temporal query answering in *DL-Lite* (extended abstract). In Diego Calvanese and Boris Konev, editors, *Proc. of the 28th Int. Workshop on Description Logics (DL'15)*, volume 1350 of *CEUR Workshop Proceedings*, pages 552–555, 2015.