A Decidable Temporal DL-Lite Logic with Undecidable First-Order and Datalog-rewritability of Ontology-Mediated Atomic Queries
(Extended Abstract)

Alessandro Artale¹, Anton Gnatenko¹, Vladislav Ryzhikov² and Michael Zakharyaschev²

¹Free University of Bozen-Bolzano, Italy
²Birkbeck, University of London, U.K.

Abstract
We design a logic in the temporal DL-Lite family (with non-Horn role inclusions and restricted temporalised roles), for which answering ontology-mediated atomic queries (OMAQs) can be done in ExpSpace and even in PSpace for ontologies without existential quantification in the rule heads but determining FO-rewritability or (linear) Datalog-rewritability of OMAQs is undecidable. On the other hand, we show (by reduction to monadic disjunctive Datalog) that deciding FO-rewritability of OMAQs in the non-temporal fragment of our logic can be done in 3NExpTime.

Keywords
Temporal description logics, DL-Lite, ontology-mediated query, first-order rewritability.

1. Introduction

Temporal description logics are used to reason about relational data that evolves in time. Along with the standard DL constructs on concepts and roles they admit temporal operators such as \( \circ_F \) (at the next moment), \( \square_F \) (always in the future), \( \Diamond_F \) (sometime later) and their past-time counterparts, which give rise to temporalised concepts and roles. Being non-monodic [1], temporalised roles notoriously lead to high computational complexity even if coupled with a lightweight DL component [2, 3, 4]. Thus, any type of DL-Lite temporalised concept inclusions (CIs) with unguarded non-Horn temporalised role inclusions (RIs) (say, \( P \sqsubseteq R \sqcup \circ_F S \)) result in an undecidable logic, while with Horn, Krom or core temporalised RIs the logic becomes decidable in ExpSpace or PSpace depending on the available temporal operators [3].

The proliferation of ontology-based data access [5] in the past decade has extended the list of traditional reasoning problems in DLs—such as satisfiability checking and answering ontology-mediated queries (OMQs)—with the rewritability problem into a target query language...
\[ \mathcal{L} \{6, 7, 8, 9\} \text{: given an OMQ } \mathcal{Q}, \text{ decide whether there exists an } \mathcal{L} \text{-rewriting of } \mathcal{Q}, \text{ that is, an } \mathcal{L} \text{-query } \varphi \text{ returning the same answers as } \mathcal{Q} \text{ over any data instance. Typical target query languages are the first-order logic (FO, possibly with various built-in predicates) and (linear) Datalog. For temporal OMQs given in the propositional temporal logic LTL, the FO-rewritability problem has recently been studied in [10]. Here, we present our initial observations on the FO- and Datalog-rewritability problems for OMQs with temporal DL-Lite ontologies. We design a logic with a restricted form of temporalised roles and non-Horn RIs whose syntax and potential usefulness are illustrated by the following example.}

**Example 1.** Imagine that we are modelling the European transport network. Some passengers may like to go by plane in one direction but return by train (say, to safely bring back a selection of wines, cheeses and fine teas). To highlight such routes, one could use the following RI:

\[
\text{flight} \sqcap \Diamond \text{F=train}^- \sqsubseteq \text{safeFlightConnection}, \tag{1}
\]

where \( \text{train}^- \) denotes the inverse of the role \( \text{train} \) (connection) from one city to another. (In our modelling, we do not regard the roles \( \text{flight} \) and \( \text{train} \) as ‘global’ because they depend on the season, day of a week, etc.) We may further partition the connections into certain classes, e.g., national and international, and infer properties of connected cities based on this classification. This is achieved by means of non-Horn RIs and CIs:

\[
\text{safeFlightConnection} \sqsubseteq \text{national} \sqcup \text{international},
\]

\[
\exists \text{international} \sqsubseteq \text{InternationalAirport}. \tag{2}
\]

Axiom (1) is an example of a guarded \( \Diamond \)-RI, where the temporalised role \( \Diamond \text{F=train}^- \) is ‘guarded’ by the role name \( \text{flight} \). Thus, \( \text{safeFlightConnection} \) may be inferred at a time instant only when there exists a \( \text{flight} \) from one city to another and, sometime later, there is a return \( \text{train} \). –

The temporal description logic \( \text{TDL-Lite}^{[\Diamond]}_{\text{bool}} \) we introduce in this paper admits arbitrary Boolean CIs with temporalised concepts, and only guarded \( \Diamond \)-RIs with temporalised roles that take the form

\[
R_1 \sqcap \cdots \sqcap R_n \sqcap \Diamond_{n+1} R_{n+1} \sqcap \cdots \sqcap \Diamond_k R_k \sqsubseteq R_{k+1} \sqcup \cdots \sqcup R_m, \tag{3}
\]

where the \( R_i \) are role names or their inverses, the \( \Diamond_j \) are sequences of \( \Diamond_F \) and \( \Diamond_P \), and \( n \geq 1 \). We prove that answering ontology-mediated atomic queries (OMAQs) in \( \text{TDL-Lite}^{[\Diamond]}_{\text{bool}} \) can be done in \( \text{ExpSpace} \) for combined complexity, and in \( \text{PSpace} \) for the flat fragment of \( \text{TDL-Lite}^{[\Diamond]}_{\text{bool}} \), which disallows positive occurrences of \( \exists R \) in ontology axioms (Th. 1). However, determining whether such a query can be rewritten into an FO- or a (linear) Datalog query is algorithmically undecidable (Th. 2). On the other hand, we observe that FO-rewritability of OMAQs in \( \text{DL-Lite}_{\text{bool}}^{[\Diamond]} \), i.e. in \( \text{TDL-Lite}^{[\Diamond]}_{\text{bool}} \) without temporalised concepts and roles, becomes decidable in \( 3\text{NExpTime} \) (Th. 3). For the proofs consult the full version of the paper at https://eprints.bbk.ac.uk/id/eprint/51763.

2. OMAQ Answering and FO-rewritability in \( \text{TDL-Lite}^{[\Diamond]}_{\text{bool}} \)

The logic \( \text{TDL-Lite}^{[\Diamond]}_{\text{bool}} \) is a member of the \( \text{TDL-Lite} \) family [4, 3], which comprises temporal extensions of various languages in the atemporal \( \text{DL-Lite} \) family [11, 12, 13]. The alphabet of
**TDL-Lite$^0_{\text{bool}}$** consists of individual names $a_0, a_1, \ldots$, concept names $A_0, A_1, \ldots$, and role names $P_0, P_1, \ldots$. A basic role $R$ is either a role name $P_i$ or its inverse $P_i^-$. A basic concept $C$ is either a concept name $A_i$ or $\exists R$. Temporalised concepts, $D$, and roles, $S$, are defined by the grammar:

$$
D ::= C \mid \bigcirc_F D \mid \bigcirc_P D \mid \diamond_F D \mid \diamond_P D \mid \square_F D \mid \square_P D,
$$

$$
S ::= R \mid \diamond_F S \mid \diamond_P S.
$$

A concept inclusion (CI) in TDL-Lite$^0_{\text{bool}}$ takes the form $D_1 \sqcap \cdots \sqcap D_k \sqsubseteq D_{k+1} \sqcup \cdots \sqcup D_{k+m}$, where the $D_i$ are temporised concepts. A guarded $\diamond$-role inclusion (guarded $\diamond$-RI) takes the form (3). As usual, the empty $\sqsubseteq$ is $\top$ and the empty $\sqcup$ is $\bot$.

A TBox, $\mathcal{T}$, is a finite set of CIs, and an RBox, $\mathcal{R}$, is a finite set of guarded $\diamond$-RIs. Taken together, they form an ontology, $\mathcal{O}$, in TDL-Lite$^0_{\text{bool}}$. An ABox signature, $\Sigma$, is a set of concept and role names. An ABox, $\mathcal{A}$, over $\Sigma$ is a finite set of facts of the form $A_i(a, \ell)$ and $P_j(a, b, \ell)$, where $A_i, P_j \in \Sigma$ and $\ell \in \mathbb{Z}$ is a timestamp. We denote by $\text{ind}(\mathcal{A})$ the set of individual names in $\mathcal{A}$, by $\text{min}(\mathcal{A})$ and $\text{max}(\mathcal{A})$ the minimal and maximal timestamps in $\mathcal{A}$, respectively, and by $\text{tem}(\mathcal{A})$ the closed interval $[\text{min}(\mathcal{A}), \text{max}(\mathcal{A})]$.

An ontology-mediated atomic query (OMAQ) is a triple $Q(x, t) = (\mathcal{O}, \Sigma, A(x, t))$, where $\mathcal{O}$ is an ontology in TDL-Lite$^0_{\text{bool}}$. $\Sigma$ is an ABox signature, $A$ a concept name, $x$ ranges over $\text{ind}(\mathcal{A})$, and $t$ over $\text{tem}(\mathcal{A})$. (Note that the symbols in $\mathcal{O}$ and $A$ do not have to be in $\Sigma$.) A pair $(a, \ell) \in \text{ind}(\mathcal{A}) \times \text{tem}(\mathcal{A})$ is a certain answer to $Q(x, t)$ over a $\Sigma$-ABox $\mathcal{A}$ if $A(a, \ell)$ is true in every model $\mathcal{M}$ of $(\mathcal{O}, \mathcal{A})$; for a detailed definition of a model the reader is referred to [3]. We denote by $\text{ans}_Q(\mathcal{A})$ the set of all certain answers to $Q$ over $\mathcal{A}$. The query answering problem for $Q$ over $\mathcal{A}$ is the decision problem for $\text{ans}_Q(\mathcal{A})$. We also consider ontology-mediated Boolean atomic queries (OMBAQs) of the form $Q = (\mathcal{O}, \Sigma, A)$ that require a ‘yes/no’ answer: a certain answer to an OMBAQ $Q$ over $\mathcal{A}$ is ‘yes’ if, in every model $\mathcal{M}$ of $(\mathcal{O}, \mathcal{A})$, there exists a pair $(a, \ell) \in \text{ind}(\mathcal{A}) \times \text{tem}(\mathcal{A})$ such that $A(a, \ell)$ is true in $\mathcal{M}$, and ‘no’ otherwise.

With an ABox $\mathcal{A}$ we associate a temporal FO-structure $\mathcal{S}_{\mathcal{A}}$ with domain $\text{ind}(\mathcal{A}) \times \text{tem}(\mathcal{A})$, over which we can evaluate $\text{FO}(\prec)$- and Datalog-queries with atoms of the form $A(x, t)$, $P(x, y, t)$, $(t_1 < t_2)$. Let $\mathcal{L}$ be any relevant query language: $\text{FO}(\prec)$, linear or arbitrary Datalog queries. An OMAQ $Q(x, t) = (\mathcal{O}, \Sigma, A(x, t))$ is $\mathcal{L}$-rewritable if there exists an $\mathcal{L}$-query $\varphi(x, t)$, such that $\text{ans}_Q(\mathcal{A}) = \{ (a, \ell) \in \text{ind}(\mathcal{A}) \times \text{tem}(\mathcal{A}) \mid \mathcal{S}_{\mathcal{A}} \models \varphi(a, \ell) \}$, for every $\Sigma$-ABox $\mathcal{A}$. An OMBAQ $Q$ is $\mathcal{L}$-rewritable if there is an $\mathcal{L}$-query $\varphi$ without answer variables such that $\mathcal{S}_{\mathcal{A}} \models \varphi$ if the certain answer to $Q$ over $\mathcal{A}$ is ‘yes’. It is known that $\text{FO}(\prec)$-, linear Datalog-, and Datalog-rewritability guarantee answering OMAQs/OMBAQs in $\text{AC}^0$ [14], NL [15], and $\text{PTime}$ [16] for data complexity, respectively. For the non-temporal fragment of TDL-Lite$^0_{\text{bool}}$, i.e. DL-Lite$^0_{\text{bool}}$, answering OMAQs/OMBAQs is $\text{ExpTime}$-complete for combined complexity [17]. We establish an $\text{ExpSpace}$ upper bound for answering OMAQs/OMBAQs in full TDL-Lite$^0_{\text{bool}}$.

We also consider the flat TDL-Lite$^0_{\text{bool}}$ that disallows concepts $\exists R$ on the right-hand side of CIs. This restriction reduces the complexity of answering OMAQs/OMBAQs to $\text{PSpace}$.

**Theorem 1.** Answering TDL-Lite$^0_{\text{bool}}$ OMAQs/OMBAQs is in $\text{ExpSpace}$ and $\text{ExpTime}$-hard for combined complexity. Answering flat TDL-Lite$^0_{\text{bool}}$ OMAQs/OMBAQs is $\text{PSpace}$-complete for combined complexity.
However, checking whether a query is rewritable into FO(<) or (linear) Datalog turns out to be undecidable even for flat ontologies.

**Theorem 2.** FO(<)-rewritability, linear Datalog-rewritability (if NL ≠ coNP), and Datalog-rewritability (if PTIME ≠ coNP) are undecidable for flat TDL-Lite\textsuperscript{[0]} OMQAQs/OMBAQs.

The proof of Th. 2 makes use of the interaction between non-Horn RIs and temporal axioms. FO-rewritability of OMAQs/OMBAQs becomes decidable in the case when ontologies do not contain temporal operators. We obtain a positive result for the language DL-Lite\textsuperscript{[0]} that disallows temporalised concepts and temporalised roles in TDL-Lite\textsuperscript{[0]}. The proof is via a translation to monadic disjunctive Datalog, adapting a similar technique for ALCI [7]. However, non-Horn RIs increase the complexity from 2NExpTime to 3NExpTime.

**Theorem 3.** Checking FO-rewritability of DL-Lite\textsuperscript{[0]} OMAQs/OMBAQs is in 3NExpTime.

### 3. Related Work

The problem of deciding if a given OMQ is rewritable into a conventional query language \( L \) has been investigated for several important description logics. Bienvenu et al. [6] considered ALC and its extensions with OMQs through the lenses of CSP and MMSNP. In particular, FO-rewritability of ALCF OMQs is undecidable (ALCF is an extension of ALC with functionality constraints on roles). Feier et al. [7] proved 2NExpTime-completeness of FO-rewritability of OMQs with CQs in ALCI. Lutz and Sabellek [8] established that every conjunctive OMQ in \( \mathcal{EL} \) is either FO-rewritable, or linear Datalog-rewritable, or PTIME-complete, and showed that each of the associated decision problems is ExpTime-complete. Gerasimova et al. [9] showed FO-rewritability to be in 2NExpTime for OMQs with CQs and the non-Horn ontology \( \{ A \sqsubseteq B \sqcup C \} \). Description logics with non-Horn RIs such as DL-Lite\textsuperscript{[0]}, however, have not been considered. Separately, rewritability to first-order languages was studied by Artale et al. [18] and Kurucz et al. [10] for pure temporal logics, using automata- and group-theoretic techniques.

In this paper, we take a first step towards understanding (temporal) DL-Lite with guarded non-Horn role inclusions by establishing two results: a decision procedure for FO-rewritability of OMQs in DL-Lite\textsuperscript{[0]}, and an undecidability result for FO(<)- and Datalog-rewritability of OMQs in TDL-Lite\textsuperscript{[0]}, i.e., DL-Lite\textsuperscript{[0]} with temporalised concepts and (guarded) roles. An open question is if rewritability is decidable for DL-Lite\textsuperscript{[0]} with temporalised concepts only.

FO-rewritability (aka boundedness) has been studied for Datalog itself. Predicate boundedness is undecidable for binary programs [19], and even for linear programs with one binary IDB relation [20], while for linear monadic programs, it is in PSpace [21]. Uniform boundedness is undecidable for ternary programs [19], even if they are linear [22] (consult the latter for an explanation on different forms of Datalog boundedness). Temporalised RIs in TDL-Lite\textsuperscript{[0]} can be viewed from the Datalog perspective as using binary, or even ternary IDBs. However, the undecidability proofs of [19], [20] and [22] make use of chains, i.e., Datalog rules where the right-hand part contains constructs like \( R(X_1, X_2) \land R(X_2, X_3) \), which is inexpressible in DL-Lite. Compared to ALCF, DL-Lite lacks negation and qualified existential restrictions.
References


