Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence

Temporal Description Logic for Ontology-Based Data Access

Alessandro Artale[1], Roman Kontchakov[2], Frank Wolter[3] and Michael Zakharyaschev[2]
[1]Faculty of Computer Science
Free University of Bozen-Bolzano, Italy
[3] Department of Computer Science,
University of Liverpool, U.K.
[2]Department of Computer Science and
Information Systems
Birkbeck, University of London, U.K.

Abstract

Our aim is to investigate ontology-based data access over temporal data with validity time and ontologies capable of temporal conceptual modelling. To this end, we design a temporal description logic, *TQL*, that extends the standard ontology language *OWL 2 QL*, provides basic means for temporal conceptual modelling and ensures first-order rewritability of conjunctive queries for suitably defined data instances with validity time.

1 Introduction

One of the most promising and exciting applications of description logics (DLs) is to supply ontology languages and query answering technologies for ontology-based data access (OBDA), a way of querying incomplete data sources that uses ontologies to provide additional conceptual information about the domains of interest and enrich the query vocabulary. The current W3C standard language for OBDA is *OWL 2 QL*, which was built on the *DL-Lite* family of DLs [Calvanese *et al.*, 2006; 2007]. To answer a conjunctive query $\boldsymbol{q}$ over an *OWL 2 QL* ontology $\mathcal{T}$ and instance data $\mathcal{A}$, an OBDA system first 'rewrites' $\boldsymbol{q}$ and $\mathcal{T}$ into a new first-order query $\boldsymbol{q}'$ and then evaluates $\boldsymbol{q}'$ over $\mathcal{A}$ (without using the ontology). The evaluation task is performed by a conventional relational database management system. Finding efficient and practical rewritings has been the subject of extensive research [Pérez-Urbina *et al.*, 2009; Rosati and Almatelli, 2010; Kontchakov *et al.*, 2010; Chortaras *et al.*, 2011; Gottlob *et al.*, 2011; König *et al.*, 2012]. Another fundamental feature of *OWL 2 QL*, supplementing its first-order rewritability, is the ability to capture basic conceptual data modelling constructs [Berardi *et al.*, 2005; Artale *et al.*, 2007].

In applications, instance data is often time-dependent: employment contracts come to an end, parliaments are elected, children are born. Temporal data can be modelled by pairs consisting of facts and their validity time; for example, $\mathit{givesBirth}(\mathit{diana}, \mathit{william}, 1982)$. To query data with validity time, it would be useful to employ an ontology that provides a conceptual model for both static and temporal aspects of the domain of interest. Thus, when querying the fact above, one could use the knowledge that, if x gives birth to y, then x becomes a mother of y from that moment on:

$$\Diamond_P \mathit{givesBirth} \sqsubseteq \mathit{motherOf}, \tag{1}$$

where $\Diamond_P$ reads 'sometime in the past.' *OWL 2 QL* does not support temporal conceptual modelling and, rather surprisingly, no attempt has yet been made to lift the OBDA framework to temporal ontologies and data.

Temporal extensions of DLs have been investigated since 1993; see [Gabbay *et al.*, 2003; Lutz *et al.*, 2008; Artale and Franconi, 2005] for surveys and [Franconi and Toman, 2011; Gutiérrez-Basulto and Klarman, 2012; Baader *et al.*, 2012] for more recent developments. Temporalised *DL-Lite* logics have been constructed for temporal conceptual data modelling [Artale *et al.*, 2010]. But unfortunately, none of the existing temporal DLs supports first-order rewritability.

The aim of this paper is to design a temporal DL that contains *OWL 2 QL*, provides basic means for temporal conceptual modelling and, at the same time, ensures first-order rewritability of conjunctive queries (for suitably defined data instances with validity time).

The temporal extension *TQL* of *OWL 2 QL* we present here is interpreted over sequences $\mathcal{I}(n)$, $n \in \mathbb{Z}$, of standard DL structures reflecting possible evolutions of data. TBox axioms are interpreted globally, that is, are assumed to hold in all of the $\mathcal{I}(n)$, but the concepts and roles they contain can vary in time. ABox assertions (temporal data) are time-stamped unary (for concepts) and binary (for roles) predicates that hold at the specified moments of time. Concept (role) inclusions of *TQL* generalise *OWL 2 QL* inclusions by allowing intersections of basic concepts (roles) in the left-hand side, possibly prefixed with temporal operators $\Diamond_P$ (sometime in the past) or $\Diamond_F$ (sometime in the future). Among other things, one can express in *TQL* that a concept/role name is rigid (or time-independent), persistent in the past/future or instantaneous. For example, $\Diamond_F \Diamond_P \mathit{Person} \sqsubseteq \mathit{Person}$ states that the concept *Person* is rigid, $\Diamond_P \mathit{hasName} \sqsubseteq \mathit{hasName}$ says that the role *hasName* is persistent in the future, while $\mathit{givesBirth} \sqcap \Diamond_P \mathit{givesBirth} \sqsubseteq \bot$ implies that *givesBirth* is instantaneous. Inclusions such as $\Diamond_P \mathit{Start} \sqcap \Diamond_F \mathit{End} \sqsubseteq \mathit{Employed}$ represent convexity (or existential rigidity) of concepts or roles. However, in contrast to most existing temporal DLs, we cannot use temporal operators in the right-hand side of inclusions (e.g., to say that every student will eventually graduate: $\mathit{Student} \sqsubseteq \Diamond_F \mathit{Graduate}$).

In conjunctive queries (CQs) over *TQL* knowledge bases, we allow time-stamped predicates together with atoms of the form $(\tau < \tau')$ or $(\tau = \tau')$, where τ, τ' are temporal constants denoting integers or variables ranging over integers.

Our main result is that, given a *TQL* TBox $\mathcal{T}$ and a CQ $\boldsymbol{q}$, one can construct a union $\boldsymbol{q}'$ of CQs such that the answers to $\boldsymbol{q}$ over $\mathcal{T}$ and any temporal ABox $\mathcal{A}$ can be computed by evaluating $\boldsymbol{q}'$ over $\mathcal{A}$ extended with the temporal precedence relation $<$ between the moments of time in $\mathcal{A}$. For example, the query *motherOf*(x, y, t) over (1) can be rewritten as

$$\textit{motherOf}(x,y,t) \lor \exists t' \big((t' < t) \land \textit{givesBirth}(x,y,t')\big).$$

Note that the addition of the transitive relation $<$ to the ABox is unavoidable: without it, there exists no first-order rewriting even for the simple example above [Libkin, 2004, Cor. 4.13].

From a technical viewpoint, one of the challenges we are facing is that, in contrast to known OBDA languages with CQ rewritability (including fragments of datalog$^\pm$ [Calì *et al.*, 2012]), witnesses for existential quantifiers outside the ABox are not independent from each other but interact via the temporal precedence relation. For this reason, a reduction to known languages appears to be impossible and a novel approach to rewriting has to be found. We also observe that straightforward temporal extensions of *TQL* lose first-order rewritability. For example, query answering over the ontology $\{\textit{Student} \sqsubseteq \Diamond_F \textit{Graduate}\}$ is shown to be non-tractable.

All omitted proofs can be found in [Artale *et al.*, 2013].

2 *TQL*: a Temporal Extension of *OWL 2 QL*

Concepts C and *roles* S of *TQL* are defined by the grammar:

$$\begin{array}{lllllllll} R & ::= & \bot & \mid & P_i & \mid & P_i^-, & & \\ B & ::= & \bot & \mid & A_i & \mid & \exists R, & & \\ C & ::= & B & \mid & C_1 \sqcap C_2 & \mid & \Diamond_P C & \mid & \Diamond_F C, \\ S & ::= & R & \mid & S_1 \sqcap S_2 & \mid & \Diamond_P S & \mid & \Diamond_F S, \end{array}$$

where A_i is a *concept name*, P_i a *role name* $(i \geq 0)$, and $\Diamond_P$ and $\Diamond_F$ are temporal operators 'sometime in the past' and 'sometime in the future,' respectively. We call concepts and roles of the form B and R *basic*. A *TQL TBox*, $\mathcal{T}$, is a finite set of *concept* and *role inclusions* of the form

$$C \sqsubseteq B, \qquad S \sqsubseteq R,$$

which are assumed to hold globally (over the whole timeline). Note that the $\Diamond_{F/P}$-free fragment of *TQL* is an extension of the description logic *DL-Lite*$^{\mathcal{H}}_{horn}$ [Artale *et al.*, 2009] with role inclusions of the form $R_1 \sqcap \dots \sqcap R_n \sqsubseteq R$; it properly contains *OWL 2 QL* (the missing role constraints can be safely added to the language).

A *TQL ABox*, $\mathcal{A}$, is a (finite) set of atoms $P_i(a, b, n)$ and $A_i(a, n)$, where a, b are *individual constants* and $n \in \mathbb{Z}$ a *temporal constant*. The set of individual constants in $\mathcal{A}$ is denoted by $\mathsf{ind}(\mathcal{A})$, and the set of temporal constants by $\mathsf{tem}(\mathcal{A})$. A *TQL knowledge base* (*KB*) is a pair $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, where $\mathcal{T}$ is a TBox and $\mathcal{A}$ an ABox.

A *temporal interpretation*, $\mathcal{I}$, is given by the ordered set $(\mathbb{Z}, <)$ of *time points* and standard (atemporal) interpretations $\mathcal{I}(n) = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}(n)})$, for each $n \in \mathbb{Z}$. Thus, $\Delta^{\mathcal{I}} \neq \emptyset$ is the common domain of all $\mathcal{I}(n)$, $a_i^{\mathcal{I}(n)} \in \Delta^{\mathcal{I}}$, $A_i^{\mathcal{I}(n)} \subseteq \Delta^{\mathcal{I}}$ and $P_i^{\mathcal{I}(n)} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. We assume that $a_i^{\mathcal{I}(n)} = a_i^{\mathcal{I}(0)}$, for all $n \in \mathbb{Z}$. To simplify presentation, we adopt the *unique name assumption*, that is, $a_i^{\mathcal{I}(n)} \neq a_j^{\mathcal{I}(n)}$ for $i \neq j$ (although the obtained results hold without it). The role and concept constructs are interpreted in $\mathcal{I}$ as follows, where $n \in \mathbb{Z}$:

$$\begin{aligned}
&\bot^{\mathcal{I}(n)} = \emptyset \text{ (for both concepts and roles)},\\
&(P_i^-)^{\mathcal{I}(n)} = \{(x,y) \mid (y,x) \in P_i^{\mathcal{I}(n)}\},\\
&(\exists R)^{\mathcal{I}(n)} = \{x \mid (x,y) \in R^{\mathcal{I}(n)}, \text{ for some } y\},\\
&(C_1 \sqcap C_2)^{\mathcal{I}(n)} = C_1^{\mathcal{I}(n)} \cap C_2^{\mathcal{I}(n)},\\
&(\Diamond_P C)^{\mathcal{I}(n)} = \{x \mid x \in C^{\mathcal{I}(m)}, \text{ for some } m < n\},\\
&(\Diamond_F C)^{\mathcal{I}(n)} = \{x \mid x \in C^{\mathcal{I}(m)}, \text{ for some } m > n\},\\
&(S_1 \sqcap S_2)^{\mathcal{I}(n)} = S_1^{\mathcal{I}(n)} \cap S_2^{\mathcal{I}(n)},\\
&(\Diamond_P S)^{\mathcal{I}(n)} = \{(x,y) \mid (x,y) \in S^{\mathcal{I}(m)}, \text{ for some } m < n\},\\
&(\Diamond_F S)^{\mathcal{I}(n)} = \{(x,y) \mid (x,y) \in S^{\mathcal{I}(m)}, \text{ for some } m > n\}.
\end{aligned}$$

The *satisfaction relation* $\models$ is defined by taking

$$\begin{array}{lll}
\mathcal{I} \models A_i(a,n) & \text{iff} & a^{\mathcal{I}(n)} \in A_i^{\mathcal{I}(n)},\\
\mathcal{I} \models P_i(a,b,n) & \text{iff} & (a^{\mathcal{I}(n)}, b^{\mathcal{I}(n)}) \in P_i^{\mathcal{I}(n)},\\
\mathcal{I} \models C \sqsubseteq B & \text{iff} & C^{\mathcal{I}(n)} \subseteq B^{\mathcal{I}(n)}, \text{ for all } n \in \mathbb{Z},\\
\mathcal{I} \models S \sqsubseteq R & \text{iff} & S^{\mathcal{I}(n)} \subseteq R^{\mathcal{I}(n)}, \text{ for all } n \in \mathbb{Z}.
\end{array}$$

If all inclusions in $\mathcal{T}$ and atoms in $\mathcal{A}$ are satisfied in $\mathcal{I}$, we call $\mathcal{I}$ a *model* of $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and write $\mathcal{I} \models \mathcal{K}$.

A *conjunctive query* (*CQ*) is a (two-sorted) first-order formula $\boldsymbol{q}(\vec{x}, \vec{s}) = \exists \vec{y}, \vec{t}\, \varphi(\vec{x}, \vec{y}, \vec{s}, \vec{t})$, where $\varphi(\vec{x}, \vec{y}, \vec{s}, \vec{t})$ is a conjunction of atoms of the form $A_i(\xi, \tau)$, $P_i(\xi, \zeta, \tau)$, $(\tau = \sigma)$ and $(\tau < \sigma)$, with ξ, ζ being *individual terms*—individual constants or variables in $\vec{x}$, $\vec{y}$—and τ, σ *temporal terms*—temporal constants or variables in $\vec{t}$, $\vec{s}$. In a *positive existential query* (PEQ) $\boldsymbol{q}$, the formula φ can also contain $\lor$. A *union of CQs* (UCQ) is a disjunction of CQs (so every PEQ is equivalent to an exponentially larger UCQ).

Given a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and a CQ $\boldsymbol{q}(\vec{x}, \vec{s})$, we call tuples $\vec{a} \subseteq \mathsf{ind}(\mathcal{A})$ and $\vec{n} \subseteq \mathsf{tem}(\mathcal{A})$ a *certain answer* to $\boldsymbol{q}(\vec{x}, \vec{s})$ over $\mathcal{K}$ and write $\mathcal{K} \models \boldsymbol{q}(\vec{a}, \vec{n})$, if $\mathcal{I} \models \boldsymbol{q}(\vec{a}, \vec{n})$ for every model $\mathcal{I}$ of $\mathcal{K}$ (understood as a two-sorted first-order model).

Example 1 Suppose Bob was a lecturer at UCL between times n_1 and n_2, after which he was appointed professor on a permanent contract. To model this situation, we use individual names, e_1 and e_2, to represent the two events of Bob's employment. The ABox will contain $n_1 < n_2$ and the atoms *lect*(bob, e_1, n_1), *lect*(bob, e_1, n_2), *prof*$(bob, e_2, n_2 + 1)$. In the TBox, we make sure that everybody is holding the corresponding post over the duration of the contract, and include other knowledge about the university life:

$$\begin{array}{ll}
\Diamond_P \textit{lect} \sqcap \Diamond_F \textit{lect} \sqsubseteq \textit{lect}, & \Diamond_P \textit{prof} \sqsubseteq \textit{prof},\\
\exists \textit{lect} \sqsubseteq \textit{Lecturer}, & \exists \textit{prof} \sqsubseteq \textit{Professor},\\
\textit{Professor} \sqsubseteq \exists \textit{supervisesPhD}, & \textit{Professor} \sqsubseteq \textit{Staff},\\
\Diamond_P \textit{supervisesPhD} \sqcap \Diamond_F \textit{supervisesPhD} \sqsubseteq \textit{supervisesPhD}, & \\
 & \text{etc.}
\end{array}$$

We can now obtain staff who supervised PhDs between times k_1 and k_2 by posing the following CQ:

$$\exists y, t\,\big((k_1 < t < k_2) \wedge \mathit{Staff}(x,t) \wedge \mathit{supervisesPhD}(x,y,t)\big).$$

The key idea of OBDA is to reduce answering CQs over KBs to evaluating FO-queries over relational databases. To obtain such a reduction for *TQL* KBs, we employ a very basic type of temporal databases. With every *TQL* ABox $\mathcal{A}$, we associate a data instance $[\mathcal{A}]$ which contains all atoms from $\mathcal{A}$ as well as the atoms $(n_1 < n_2)$ such that $n_i \in \mathbb{Z}$ with $\min \mathsf{tem}(\mathcal{A}) \leq n_i \leq \max \mathsf{tem}(\mathcal{A})$ and $n_1 < n_2$. Thus, in addition to $\mathcal{A}$, we explicitly include in $[\mathcal{A}]$ the temporal precedence relation over the *convex closure* of the time points that occur in $\mathcal{A}$. (Note that, in standard temporal databases, the order over timestamps is built-in.) The main result of this paper is the following:

Theorem 2 *Let $\boldsymbol{q}(\vec{x}, \vec{s})$ be a CQ and $\mathcal{T}$ a TQL TBox. Then one can construct a UCQ $\boldsymbol{q}'(\vec{x}, \vec{s})$ such that, for any consistent KB $(\mathcal{T}, \mathcal{A})$ such that $\mathcal{A}$ contains all temporal constants from $\boldsymbol{q}$, any $\vec{a} \subseteq \mathsf{ind}(\mathcal{A})$ and $\vec{n} \subseteq \mathsf{tem}(\mathcal{A})$, we have $(\mathcal{T}, \mathcal{A}) \models \boldsymbol{q}(\vec{a}, \vec{n})$ iff $[\mathcal{A}] \models \boldsymbol{q}'(\vec{a}, \vec{n})$.*

Such a UCQ $\boldsymbol{q}'(\vec{x}, \vec{s})$ is called a *rewriting* for $\boldsymbol{q}$ and $\mathcal{T}$. We begin by showing how to compute rewritings for CQs over KBs with empty TBoxes.

For an ABox $\mathcal{A}$, we denote by $\mathcal{A}^{\mathbb{Z}}$ the *infinite* data instance which contains the atoms in $\mathcal{A}$ as well as all $(n_1 < n_2)$ such that $n_1, n_2 \in \mathbb{Z}$ and $n_1 < n_2$. It will be convenient to regard CQs $\boldsymbol{q}(\vec{x}, \vec{s})$ as *sets* of atoms, so that we can write, e.g., $A(\xi, \tau) \in \boldsymbol{q}$. We say that $\boldsymbol{q}$ is *totally ordered* if, for any temporal terms τ, τ' in $\boldsymbol{q}$, at least one of the constraints $\tau < \tau'$, $\tau = \tau'$ or $\tau' < \tau$ is in $\boldsymbol{q}$ and the set of such constraints is consistent (in the sense that it can be satisfied in $\mathbb{Z}$). Clearly, every CQ is equivalent to a union of totally ordered CQs (note that the empty union is $\bot$).

Lemma 3 *For every UCQ $\boldsymbol{q}(\vec{x}, \vec{s})$, one can compute a UCQ $\boldsymbol{q}'(\vec{x}, \vec{s})$ such that, for any ABox $\mathcal{A}$ containing all temporal constants from $\boldsymbol{q}$ and any $\vec{a} \subseteq \mathsf{ind}(\mathcal{A})$, $\vec{n} \subseteq \mathsf{tem}(\mathcal{A})$, we have*

$$\mathcal{A}^{\mathbb{Z}} \models \boldsymbol{q}(\vec{a}, \vec{n}) \quad \textit{iff} \quad [\mathcal{A}] \models \boldsymbol{q}'(\vec{a}, \vec{n}).$$

Proof. We assume that every CQ $\boldsymbol{q}_0$ in $\boldsymbol{q}$ is totally ordered. In each such $\boldsymbol{q}_0$, we remove a bound temporal variable t together with the atoms containing t if at least one of the following two conditions holds:

– there is no temporal constant or free temporal variable τ with $(\tau < t) \in \boldsymbol{q}_0$, and for no temporal term τ' and atom of the form $A(\xi, \tau')$ or $P(\xi, \zeta, \tau')$ in $\boldsymbol{q}_0$ do we have $(\tau' < t)$ or $(\tau' = t)$ in $\boldsymbol{q}_0$;

– the same as above but with $<$ replaced by $>$.

It is readily checked that the resulting UCQ is as required. ❑

Example 4 Suppose $\mathcal{T} = \{\Diamond_{\scriptscriptstyle F} C \sqsubseteq A,\ \Diamond_{\scriptscriptstyle P} A \sqsubseteq B\}$ and $\boldsymbol{q}(x, s) = B(x, s)$. Then, for any $\mathcal{A}$, $a \in \mathsf{ind}(\mathcal{A})$, $n \in \mathsf{tem}(\mathcal{A})$, we have $(\mathcal{T}, \mathcal{A}) \models \boldsymbol{q}(a, n)$ iff $\mathcal{A}^{\mathbb{Z}} \models \boldsymbol{q}'(a, n)$, where

$$\boldsymbol{q}'(x, s) = B(x, s) \ \vee\ \exists t\,\big((t < s) \wedge A(x, t)\big) \ \vee\ \exists t, r\,\big((t < s) \wedge (t < r) \wedge C(x, r)\big).$$

Note, however, that $\boldsymbol{q}'$ is *not* a rewriting for $\boldsymbol{q}$ and $\mathcal{T}$. Take, for example, $\mathcal{A} = \{C(a, 0)\}$. Then $(\mathcal{T}, \mathcal{A}) \models B(a, 0)$ but $[\mathcal{A}] \not\models \boldsymbol{q}'(a, 0)$. A correct rewriting is obtained by replacing the last disjunct in $\boldsymbol{q}'$ with $\exists r\, C(x, r)$; it can be computed by applying Lemma 3 to $\boldsymbol{q}'$ and slightly simplifying the result.

In view of Lemma 3, from now on we will only focus on rewritings over $\mathcal{A}^{\mathbb{Z}}$.

The problem of finding rewritings for CQs and *TQL* TBoxes can be reduced to the case where the TBoxes only contain inclusions of the form

$$\begin{array}{lll} B_1 \sqcap B_2 \sqsubseteq B, & \Diamond_{\scriptscriptstyle F} B_1 \sqsubseteq B_2, & \Diamond_{\scriptscriptstyle P} B_1 \sqsubseteq B_2, \\ R_1 \sqcap R_2 \sqsubseteq R, & \Diamond_{\scriptscriptstyle F} R_1 \sqsubseteq R_2, & \Diamond_{\scriptscriptstyle P} R_1 \sqsubseteq R_2. \end{array}$$

We say that such TBoxes are in *normal form*.

Theorem 5 *For every TQL TBox $\mathcal{T}$, one can construct in polynomial time a TQL TBox $\mathcal{T}'$ in normal form (possibly containing additional concept and role names) such that $\mathcal{T}' \models \mathcal{T}$ and, for every model $\mathcal{I}$ of $\mathcal{T}$, there exists a model of $\mathcal{T}'$ that coincides with $\mathcal{I}$ on all concept and role names in $\mathcal{T}$.*

Suppose now that we have a UCQ rewriting $\boldsymbol{q}'$ for a CQ $\boldsymbol{q}$ and the TBox $\mathcal{T}'$ in Theorem 5. We obtain a rewriting for $\boldsymbol{q}$ and $\mathcal{T}$ simply by removing from $\boldsymbol{q}'$ those CQs that contain symbols occurring in $\mathcal{T}'$ but not in $\mathcal{T}$. From now on, we assume that *all TQL TBoxes are in normal form*. The set of role names in $\mathcal{T}$ and with their inverses is denoted by $\mathsf{R}_{\mathcal{T}}$, while $|\mathcal{T}|$ is the number of concept and role names in $\mathcal{T}$.

We begin the construction of rewritings by considering the case when all concept inclusions are of the form $C \sqsubseteq A_i$, so existential quantification $\exists R$ does not occur in the right-hand side. *TQL* TBoxes of this form will be called *flat*. Note that RDFS statements can be expressed by means of flat TBoxes.

3 UCQ Rewriting for Flat TBoxes

Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a KB with a flat TBox $\mathcal{T}$ (in normal form). Our first aim is to construct a model $\mathcal{C}_{\mathcal{K}}$ of $\mathcal{K}$, called the *canonical model*, for which the following theorem holds:

Theorem 6 *For any consistent KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ with flat $\mathcal{T}$ and any CQ $\boldsymbol{q}(\vec{x}, \vec{s})$, we have $\mathcal{K} \models \boldsymbol{q}(\vec{a}, \vec{n})$ iff $\mathcal{C}_{\mathcal{K}} \models \boldsymbol{q}(\vec{a}, \vec{n})$, for all tuples $\vec{a} \subseteq \mathsf{ind}(\mathcal{A})$ and $\vec{n} \subseteq \mathbb{Z}$.*

The construction uses a closure operator, cl, which applies the rules **(ex)**, **(c1)–(c3)**, **(r1)–(r3)** below to a set, $\mathcal{S}$, of atoms of the form $R(u, v, n)$, $A(u, n)$, $\exists R(u, n)$ or $(n < n')$; $\mathsf{cl}(\mathcal{S})$ is the result of (non-recursively) applying those rules to $\mathcal{S}$,

$$\mathsf{cl}^0(\mathcal{S}) = \mathcal{S},\ \mathsf{cl}^{i+1}(\mathcal{S}) = \mathsf{cl}(\mathsf{cl}^i(\mathcal{S})),\ \mathsf{cl}^{\infty}(\mathcal{S}) = \bigcup_{i \geq 0} \mathsf{cl}^i(\mathcal{S}).$$

(ex) If $R(u, v, n) \in \mathcal{S}$ then add $\exists R(u, n)$, $\exists R^-(v, n)$ to $\mathcal{S}$;

(c1) if $(B_1 \sqcap B_2 \sqsubseteq B) \in \mathcal{T}$ and $B_1(u, n)$, $B_2(u, n) \in \mathcal{S}$, then add $B(u, n)$ to $\mathcal{S}$;

(c2) if $(\Diamond_{\scriptscriptstyle P} B \sqsubseteq B') \in \mathcal{T}$, $B(u, m) \in \mathcal{S}$ for some $m < n$ and n occurs in $\mathcal{S}$, then add $B'(u, n)$ to $\mathcal{S}$;

(c3) if $(\Diamond_{\scriptscriptstyle F} B \sqsubseteq B') \in \mathcal{T}$, $B(u, m) \in \mathcal{S}$ for some $m > n$ and n occurs in $\mathcal{S}$, then add $B'(u, n)$ to $\mathcal{S}$;

(r1) if $(R_1 \sqcap R_2 \sqsubseteq R) \in \mathcal{T}$ and $R_1(u, v, n)$, $R_2(u, v, n)$ are in $\mathcal{S}$, then add $R(u, v, n)$ to $\mathcal{S}$;

(r2) if $(\Diamond_P R \sqsubseteq R') \in \mathcal{T}$, $R(u, v, m) \in \mathcal{S}$ for some $m < n$ and n occurs in $\mathcal{S}$, then add $R'(u, v, n)$ to $\mathcal{S}$;

(r3) if $(\Diamond_F R \sqsubseteq R') \in \mathcal{T}$, $R(u, v, m) \in \mathcal{S}$ for some $m > n$ and n occurs in $\mathcal{S}$, then add $R'(u, v, n)$ to $\mathcal{S}$.

Note first that $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is inconsistent iff $\bot \in \mathsf{cl}^\infty(\mathcal{A}^\mathbb{Z})$. If $\mathcal{K}$ is consistent, we define the *canonical model* $\mathcal{C}_\mathcal{K}$ of $\mathcal{K}$ by taking $\Delta^{\mathcal{C}_\mathcal{K}} = \mathsf{ind}(\mathcal{A})$, $a \in A^{\mathcal{C}_\mathcal{K}(n)}$ iff $A(a, n) \in \mathsf{cl}^\infty(\mathcal{A}^\mathbb{Z})$, and $(a, b) \in P^{\mathcal{C}_\mathcal{K}(n)}$ iff $P(a, b, n) \in \mathsf{cl}^\infty(\mathcal{A}^\mathbb{Z})$, for $n \in \mathbb{Z}$. (As $\mathcal{T}$ is flat, atoms of the form $\exists R(u, n)$ can only be added by **(ex)**.) This gives us Theorem 6. The following lemma shows that to construct $\mathcal{C}_\mathcal{K}$ we actually need only a bounded number of applications of cl which does not depend on $\mathcal{A}$:

Lemma 7 *Suppose $\mathcal{T}$ is a flat TBox, let $n_\mathcal{T} = (4 \cdot |\mathcal{T}|)^4$. Then $\mathsf{cl}^\infty(\mathcal{A}^\mathbb{Z}) = \mathsf{cl}^{n_\mathcal{T}}(\mathcal{A}^\mathbb{Z})$, for any ABox $\mathcal{A}$.*

Proof. It is not hard to see that $\mathsf{cl}^\infty(\mathcal{S})$ can be obtained by first exhaustively applying **(r1)–(r3)**, then **(ex)**, and after that **(c1)–(c3)**. Since no recursion of **(ex)** is needed, it is sufficient to bound the recursion depth for applications of **(r1)–(r3)** and **(c1)–(c3)** separately. As both behave similarly, we focus on **(r1)–(r3)**. One can show that it is enough to consider ABoxes with two individuals, say a and b, and it is not difficult to find a bound for the recursion depth of the separated rule sets **(r1)**, **(r2)** and, respectively, **(r1)**, **(r3)**; the interesting part of the analysis is how often one has to alternate between applications of **(r1)**, **(r2)** and applications of **(r1)**, **(r3)**. The key observation here is that each alternation introduces a fresh *cross over* (i.e., a pair (R_1, R_2) of roles such that there are $m_1, m_2 \in \mathbb{Z}$ with $m_1 + 1 \geq m_2$, $R_1(a, b, n) \in \mathcal{S}$ for all $n \leq m_1$, and $R_2(a, b, n) \in \mathcal{S}$ for all $n \geq m_2$). The number of such cross overs is bounded by $|\mathcal{T}|^2$, and so the number of required alternations between exhaustively applying **(r1)**, **(r2)** and **(r1)**, **(r3)** is bounded by $|\mathcal{T}|^2$. ❑

We now use Lemma 7 to construct a rewriting for any flat TBox $\mathcal{T}$ and CQ $\boldsymbol{q}(\vec{x}, \vec{s})$. For a concept C and a role S, denote by $C^\sharp$ and $S^\sharp$ their standard FO-translations; for example, $(\Diamond_F A)^\sharp(\xi, \tau) = \exists t\,((\tau < t) \land A(\xi, t))$ and $(\exists R)^\sharp(\xi, \tau) = \exists y\, R(\xi, y, \tau)$. Now, given a PEQ φ, we set $\varphi^{0\downarrow} = \varphi$ and define, inductively, $\varphi^{(n+1)\downarrow}$ as the result of replacing every

– $A(\xi, \tau)$ with $A(\xi, \tau) \lor \bigvee_{(C \sqsubseteq A) \in \mathcal{T}} (C^\sharp(\xi, \tau))^{n\downarrow}$,

– $P(\xi, \zeta, \tau)$ with $P(\xi, \zeta, \tau) \lor \bigvee_{(S \sqsubseteq P) \in \mathcal{T}} (S^\sharp(\xi, \zeta, \tau))^{n\downarrow}$.

Finally, we set

$$\mathsf{ext}_{\boldsymbol{q}}^\mathcal{T}(\vec{x}, \vec{s}) = (\boldsymbol{q}(\vec{x}, \vec{s}))^{n_\mathcal{T}\downarrow}.$$

Clearly, $\mathsf{ext}_{\boldsymbol{q}}^\mathcal{T}(\vec{x}, \vec{s})$ is a PEQ, and so can be equivalently transformed into a UCQ. Denote by $\mathcal{T}^\bot$ the result of replacing $\bot$ with a fresh concept name, say F, in all concept inclusions and with a fresh role name, say Q, in all role inclusions of $\mathcal{T}$. Clearly $(\mathcal{T}^\bot, \mathcal{A})$ is consistent for any ABox $\mathcal{A}$. Let $\boldsymbol{q}^\bot = (\exists x, t\, F(x, t)) \lor (\exists x, y, t\, Q(x, y, t))$. By Theorem 6 and Lemma 7, we obtain:

Theorem 8 *Let $\mathcal{T}$ be a flat TBox and $\boldsymbol{q}(\vec{x}, \vec{s})$ a CQ. Then, for any consistent KB $(\mathcal{T}, \mathcal{A})$, any $\vec{a} \subseteq \mathsf{ind}(\mathcal{A})$ and $\vec{n} \subseteq \mathbb{Z}$,*

$$(\mathcal{T}, \mathcal{A}) \models \boldsymbol{q}(\vec{a}, \vec{n}) \quad \textit{iff} \quad \mathcal{A}^\mathbb{Z} \models \mathsf{ext}_{\boldsymbol{q}}^\mathcal{T}(\vec{a}, \vec{n}).$$

$(\mathcal{T}, \mathcal{A})$ is inconsistent iff $(\mathcal{T}^\bot, \mathcal{A}) \models \boldsymbol{q}^\bot$.

Thus, we obtain a rewriting for $\boldsymbol{q}$ and $\mathcal{T}$ using Lemma 3.

4 Canonical Models for Arbitrary TBoxes

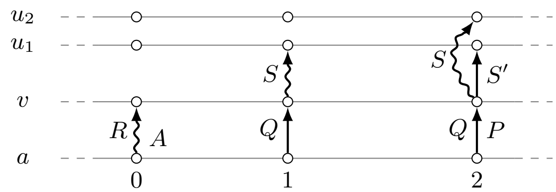
Canonical models for consistent KBs $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ with not necessarily flat TBoxes $\mathcal{T}$ (in normal form) can be constructed starting from $\mathcal{A}^\mathbb{Z}$ and using the rules given in the previous section together with the following one:

($\rightsquigarrow$) if $\exists R(u, n) \in \mathcal{S}$ and $R(u, v, n) \notin \mathcal{S}$ for any v, then add $R(u, v, n)$ to $\mathcal{S}$, for some fresh individual name v; in this case we write $u \rightsquigarrow_R^n v$.

Denote by cl_1 the closure operator under the resulting 8 rules. Again, $\mathcal{K}$ is inconsistent iff $\bot \in \mathsf{cl}_1^\infty(\mathcal{A}^\mathbb{Z})$. If $\mathcal{K}$ is consistent, we define the *canonical model* $\mathcal{C}_\mathcal{K}$ for $\mathcal{K}$ by the set $\mathsf{cl}_1^\infty(\mathcal{A}^\mathbb{Z})$ in the same way as in Section 3 but taking the domain $\Delta^{\mathcal{C}_\mathcal{K}}$ to contain all the individual names in $\mathsf{cl}_1^\infty(\mathcal{A}^\mathbb{Z})$.

Theorem 9 *For every consistent $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and every CQ $\boldsymbol{q}(\vec{x}, \vec{s})$, we have $\mathcal{K} \models \boldsymbol{q}(\vec{a}, \vec{n})$ iff $\mathcal{C}_\mathcal{K} \models \boldsymbol{q}(\vec{a}, \vec{n})$, for any tuples $\vec{a} \subseteq \mathsf{ind}(\mathcal{A})$ and $\vec{n} \subseteq \mathbb{Z}$.*

Example 10 Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ with $\mathcal{A} = \{A(a, 0)\}$ and $\mathcal{T} = \{A \sqsubseteq \exists R,\ \Diamond_P R \sqsubseteq Q,\ \exists Q^- \sqsubseteq \exists S,\ \Diamond_P Q \sqsubseteq P,\ \Diamond_P S \sqsubseteq S'\}$. A fragment of the model $\mathcal{C}_\mathcal{K}$ is shown in the picture below:



We say that the individuals $a \in \mathsf{ind}(\mathcal{A})$ are of *depth* 0 in $\mathcal{C}_\mathcal{K}$; now, if u is of depth d in $\mathcal{C}_\mathcal{K}$ and $u \rightsquigarrow_R^n v$, for some $n \in \mathbb{Z}$ and R, then v is of *depth* $d + 1$ in $\mathcal{C}_\mathcal{K}$. Thus, both u_1 and u_2 in Example 10 are of depth 2 and v is of depth 1. The restriction of $\mathcal{C}_\mathcal{K}$, treated as a set of atoms, to the individual names of depth $\leq d$ is denoted by $\mathcal{C}_\mathcal{K}^d$. Note that this set is not necessarily closed under the rule ($\rightsquigarrow$).

In the remainder of this section, we describe the structure of $\mathcal{C}_\mathcal{K}$, which is required for the rewriting in the next section. We split $\mathcal{C}_\mathcal{K}$ into two parts: one consists of the elements in $\mathsf{ind}(\mathcal{A})$, while the other contains the fresh individuals introduced by ($\rightsquigarrow$). As this rule always uses *fresh* individuals, to understand the structure of the latter part it is enough to consider KBs of the form $\mathcal{K}_{\mathcal{T},R} = (\mathcal{T} \cup \{A \sqsubseteq \exists R\}, \{A(a, 0)\})$ with fresh A. We begin by analysing the behaviour of the atoms $R'(a, u, n)$ entailed by $R(a, u, 0)$, where $a \rightsquigarrow_R^0 u$.

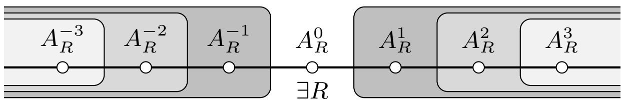
Lemma 11 (monotonicity) *Let $a \rightsquigarrow_R^0 u$ in $\mathcal{C}_{\mathcal{K}_{\mathcal{T},R}}$. If either $m < n < 0$ or $0 < n < m$, then $R'(a, u, n) \in \mathcal{C}_{\mathcal{K}_{\mathcal{T},R}}$ implies $R'(a, u, m) \in \mathcal{C}_{\mathcal{K}_{\mathcal{T},R}}$; moreover, if $n < m = -|\mathsf{R}_\mathcal{T}|$ or $|\mathsf{R}_\mathcal{T}| = m < n$, then $R'(a, u, n) \in \mathcal{C}_{\mathcal{K}_{\mathcal{T},R}}$ iff $R'(a, u, m) \in \mathcal{C}_{\mathcal{K}_{\mathcal{T},R}}$.*

The atoms $R'(a, u, n)$ entailed by $R(a, u, 0)$ in $\mathcal{C}_{\mathcal{K}_{\mathcal{T},R}}$ via **(r1)–(r3)**, also have an impact, via **(ex)**, on the atoms of the form $B(a, n)$ and $B(u, n)$ in $\mathcal{C}_{\mathcal{K}_{\mathcal{T},R}}$. Thus, in Example 10, $R(a, v, 0)$ entails $\exists Q(a, n)$, for $n > 0$. To analyse the behaviour of such atoms, it is helpful to assume that $\mathcal{T}$ is in *concept normal form* (CoNF) in the following sense: for every role $R \in \mathsf{R}_\mathcal{T}$, the TBox $\mathcal{T}$ contains

$$\exists R \sqsubseteq A_R^0, \quad \Diamond_F \exists R \sqsubseteq A_R^{-1}, \quad \Diamond_F A_R^{-m} \sqsubseteq A_R^{-m-1},$$
$$\Diamond_P \exists R \sqsubseteq A_R^1, \quad \Diamond_P A_R^m \sqsubseteq A_R^{m+1},$$

for $0 \leq m \leq |\mathsf{R}_{\mathcal{T}}|$ and some concepts A^i_R, and

$$A^m_R \sqsubseteq \exists R', \text{ for } |m| \leq |\mathsf{R}_{\mathcal{T}}| \text{ and } R'(a, v, m) \in \mathcal{C}_{\mathcal{K}_{\mathcal{T},R}}.$$



(In Example 10, $\mathcal{C}_{\mathcal{K}}$ will contain the atoms $A^1_R(a, n)$ and $A^2_R(a, n + 1)$, for $n \geq 1$.) By Lemma 11, if $\mathcal{T}$ is in CoNF, then we can compute the atoms $B(a, n)$ and $B(u, n)$ in $\mathcal{C}_{\mathcal{K}_{\mathcal{T},R}}$ without using the rules **(r1)–(r3)**. Lemma 11 also implies that we can add the inclusions above (with fresh A^i_R) to $\mathcal{T}$ if required, thereby obtaining a conservative extension of $\mathcal{T}$; so from now on we always assume $\mathcal{T}$ *to be in CoNF*. These observations enable the proof of the following two lemmas. The first one characterises the atoms $B(u, n)$ in $\mathcal{C}_{\mathcal{K}_{\mathcal{T},R}}$:

Lemma 12 (monotonicity) *Let $a \rightsquigarrow^0_R u$ in $\mathcal{C}_{\mathcal{K}_{\mathcal{T},R}}$. If either $m < n < 0$ or $0 < n < m$, then $B(u, n) \in \mathcal{C}_{\mathcal{K}_{\mathcal{T},R}}$ implies $B(u, m) \in \mathcal{C}_{\mathcal{K}_{\mathcal{T},R}}$; moreover, if either $n < m = -|\mathcal{T}|$ or $|\mathcal{T}| = m < n$, then $B(u, n) \in \mathcal{C}_{\mathcal{K}_{\mathcal{T},R}}$ iff $B(u, m) \in \mathcal{C}_{\mathcal{K}_{\mathcal{T},R}}$.*

The second lemma characterises the ABox part of $\mathcal{C}_{\mathcal{K}}$ and is a straightforward generalisation of Lemma 7:

Lemma 13 *For any KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and any atom α of the form $A(a, n)$, $\exists R(a, n)$ or $R(a, b, n)$, where $a, b \in \mathsf{ind}(\mathcal{A})$ and $n \in \mathbb{Z}$, we have $\alpha \in \mathcal{C}_{\mathcal{K}}$ iff $\alpha \in \mathsf{cl}^{n_{\mathcal{T}}}(\mathcal{A}^{\mathbb{Z}})$.*

An obvious extension of the rewriting of Theorem 8 provides, for every CQ $\boldsymbol{q}(\vec{x}, \vec{s})$, a UCQ $\mathsf{ext}^{\mathcal{T}}_{\boldsymbol{q}}(\vec{x}, \vec{s})$ such that for all $\vec{a} \subseteq \mathsf{ind}(\mathcal{A})$ and $\vec{n} \subseteq \mathbb{Z}$ of the appropriate length,

$$\mathcal{C}^0_{\mathcal{K}} \models \boldsymbol{q}(\vec{a}, \vec{n}) \quad \text{iff} \quad \mathcal{A}^{\mathbb{Z}} \models \mathsf{ext}^{\mathcal{T}}_{\boldsymbol{q}}(\vec{a}, \vec{n}). \tag{2}$$

Moreover, for a basic concept $\exists R$, we find a UCQ $\mathsf{ext}^{\mathcal{T}}_{\exists R}(\xi, \tau)$ such that, for any $a \in \mathsf{ind}(\mathcal{A})$ and $n \in \mathbb{Z}$, $\exists R(a, n) \in \mathcal{C}_{\mathcal{K}}$ iff $\mathcal{A}^{\mathbb{Z}} \models \mathsf{ext}^{\mathcal{T}}_{\exists R}(a, n)$.

We now use the obtained results to show that one can find all answers to a CQ $\boldsymbol{q}$ over a *TQL* KB $\mathcal{K}$ by only considering a fragment of $\mathcal{C}_{\mathcal{K}}$ whose size is polynomial in $|\mathcal{T}|$ and $|\boldsymbol{q}|$. This property is called the *polynomial witness property* [Gottlob and Schwentick, 2011]. Denote by $\mathcal{C}^{d,\ell}_{\mathcal{K}}$, for $d, \ell \geq 0$, the restriction of $\mathcal{C}^d_{\mathcal{K}}$ to the moments of time in the interval $[\min \mathsf{tem}(\mathcal{A}) - \ell, \max \mathsf{tem}(\mathcal{A}) + \ell]$.

Let $\boldsymbol{q}(\vec{x}, \vec{s})$ be a CQ. Tuples $\vec{a} \subseteq \mathsf{ind}(\mathcal{A})$ and $\vec{n} \subseteq \mathsf{tem}(\mathcal{A})$ give a certain answer to $\boldsymbol{q}(\vec{x}, \vec{s})$ over $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ iff there is a *homomorphism* h from $\boldsymbol{q}$ to $\mathcal{C}_{\mathcal{K}}$, which maps individual (temporal) terms of $\boldsymbol{q}$ to individual (respectively, temporal) terms of $\mathcal{C}_{\mathcal{K}}$ in such a way that the following conditions hold:

– $h(\vec{x}) = \vec{a}$ and $h(b) = b$, for all $b \in \mathsf{ind}(\mathcal{A})$;

– $h(\vec{s}) = \vec{n}$ and $h(m) = m$, for all $m \in \mathsf{tem}(\mathcal{A})$;

– $h(\boldsymbol{q}) \subseteq \mathcal{C}_{\mathcal{K}}$,

where $h(\boldsymbol{q})$ denotes the set of atoms obtained by replacing every term in $\boldsymbol{q}$ with its h-image, e.g., $P(\xi, \zeta, \tau)$ with $P(h(\xi), h(\zeta), h(\tau))$, $(\tau_1 < \tau_2)$ with $h(\tau_1) < h(\tau_2)$, etc.

Now, using the monotonicity lemmas for the temporal dimension and the fact that atoms of depth $> |\mathsf{R}_{\mathcal{T}}|$ in the canonical models duplicate atoms of smaller depth, we obtain

Theorem 14 *There are polynomials f_1 and f_2 such that, for any consistent TQL KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, any CQ $\boldsymbol{q}(\vec{x}, \vec{s})$ and any $\vec{a} \subseteq \mathsf{ind}(\mathcal{A})$ and $\vec{n} \subseteq \mathsf{tem}(\mathcal{A})$, we have $\mathcal{K} \models \boldsymbol{q}(\vec{a}, \vec{n})$ iff there is a homomorphism $h \colon \boldsymbol{q} \to \mathcal{C}_{\mathcal{K}}$ such that $h(\boldsymbol{q}) \subseteq \mathcal{C}^{d,\ell}_{\mathcal{K}}$, where $d = f_1(|\mathcal{T}|, |\boldsymbol{q}|)$ and $\ell = f_2(|\mathcal{T}|, |\boldsymbol{q}|)$.*

We are now in a position to define a rewriting for any given CQ and *TQL* TBox.

5 UCQ Rewriting

Suppose $\boldsymbol{q}(\vec{x}, \vec{s})$ is a CQ and $\mathcal{T}$ a *TQL* TBox (in CoNF). Without loss of generality we assume $\boldsymbol{q}$ to be totally ordered. By a *sub-query* of $\boldsymbol{q}$ we understand any subset $\boldsymbol{q}' \subseteq \boldsymbol{q}$ containing all temporal constraints $(\tau < \tau')$ and $(\tau = \tau')$ that occur in $\boldsymbol{q}$. In the rewriting for $\boldsymbol{q}$ and $\mathcal{T}$ given below, we consider all possible splittings of $\boldsymbol{q}$ into two sub-queries (sharing the same temporal terms). One is to be mapped to the ABox part of the canonical model $\mathcal{C}_{(\mathcal{T},\mathcal{A})}$, and so we can rewrite it using (2). The other sub-query is to be mapped to the non-ABox part of $\mathcal{C}_{(\mathcal{T},\mathcal{A})}$ and requires a different rewriting.

For every $R \in \mathsf{R}_{\mathcal{T}}$, we construct the set $\mathcal{C}^{d,\ell}_{\mathcal{K}_{\mathcal{T},R}}$, where d and ℓ are provided by Theorem 14. Let h be a map from a sub-query $\boldsymbol{q}_h \subseteq \boldsymbol{q}$ to $\mathcal{C}^{d,\ell}_{\mathcal{K}_{\mathcal{T},R}}$ such that $h(\boldsymbol{q}_h) \subseteq \mathcal{C}^{d,\ell}_{\mathcal{K}_{\mathcal{T},R}}$. Denote by $\mathcal{X}_h$ the set of individual terms ξ in $\boldsymbol{q}_h$ with $h(\xi) = a$, and let $\mathcal{Y}_h$ be the remaining set of individual terms in $\boldsymbol{q}_h$. We call h a *witness for* R if

– $\mathcal{X}_h$ contains at most one individual constant;

– every term in $\mathcal{Y}_h$ is a quantified variable in $\boldsymbol{q}$;

– $\boldsymbol{q}_h$ contains all atoms in $\boldsymbol{q}$ with a variable from $\mathcal{Y}_h$.

Let h be a witness for R. Denote by $\rightsquigarrow$ the union of all $\rightsquigarrow^n_{R'}$ in $\mathcal{C}^{d,\ell}_{\mathcal{K}_{\mathcal{T},R}}$. Clearly, $\rightsquigarrow$ is a tree order on the individuals in $\mathcal{C}^{d,\ell}_{\mathcal{K}_{\mathcal{T},R}}$, with root a. Let T_h be its minimal sub-tree containing a and the h-images of all the individual terms in $\boldsymbol{q}_h$. For each $v \in T_h \setminus \{a\}$, we take the (unique) moment $\mathfrak{g}(v)$ with $u \rightsquigarrow^{\mathfrak{g}(v)}_R v$, for some u and R, and set $\mathfrak{g}(a) = 0$. For $A(y, \tau) \in \boldsymbol{q}_h$, we say that $h(y)$ *realises* $A(y, \tau)$. For any $P(\xi, \xi', \tau) \in \boldsymbol{q}_h$, there are $u, u' \in T_h$ with $u \rightsquigarrow u'$ and $\{u, u'\} = \{h(\xi), h(\xi')\}$; we say that u' *realises* $P(\xi, \xi', \tau)$. Let $\vec{r}$ be a list of fresh temporal variables r_u, for $u \in T_h \setminus \{a\}$. Consider the following formula, whose free variables are r_a and the temporal variables of $\boldsymbol{q}_h$:

$$\mathsf{t}_h = \exists \vec{r} \, \big(\bigwedge_{u \rightsquigarrow v} \delta^{\mathfrak{g}(v) - \mathfrak{g}(u)}(r_u, r_v) \wedge \bigwedge_{u \text{ realises } \alpha(\vec{\xi}, \tau)} \delta^{h(\tau) - \mathfrak{g}(u)}(r_u, \tau) \big),$$

where the formulas $\delta^n(t, s)$ say that t is at least n moments before s: that is, $\delta^0(t, s)$ is $(t = s)$ and $\delta^n(t, s)$ is

$$\exists s_1, \ldots, s_{n-1} (t < s_1 < \cdots < s_{n-1} < s), \qquad \text{if } n > 0,$$
$$\exists s_1, \ldots, s_{|n|-1} (t > s_1 > \cdots > s_{|n|-1} > s), \qquad \text{if } n < 0.$$

Take a fresh variable x_h and associate with h the formula

$$\mathsf{w}_h = \exists r_a \exists x_h \, \big[\mathsf{ext}^{\mathcal{T}}_{\exists R}(x_h, r_a) \ \wedge \bigwedge_{h(\xi) = a} (\xi = x_h) \ \wedge \ \mathsf{t}_h \big].$$

To give the intuition behind w_h, suppose that $\mathcal{C}_{(\mathcal{T},\mathcal{A})} \models^g \mathsf{w}_h$, for some assignment g. Then g maps all terms in $\mathcal{X}_h$ to

$g(x_h) \in \mathsf{ind}(\mathcal{A})$ such that $\exists R(g(x_h), g(r_a)) \in \mathcal{C}_{(\mathcal{T},\mathcal{A})}$, so $(g(x_h), g(r_a))$ is the root of a substructure of $\mathcal{C}_{(\mathcal{T},\mathcal{A})}$ isomorphic to $\mathcal{C}_{\mathcal{K}_{\mathcal{T},R}}$ in which the variables from $\mathcal{Y}_h$ can be mapped according to h. For temporal terms, the formula t_h cannot specify the values prescribed by h: without $\neg$ in UCQs, we can only say that τ is at least (not exactly) n moments before τ'. However, by Lemmas 11 and 12, this is still enough to ensure that g and h give a homomorphism from $\boldsymbol{q}_h$ to $\mathcal{C}_{(\mathcal{T},\mathcal{A})}$.

Example 15 Let $\mathcal{T}$ be the same as in Example 10 and let

$$\boldsymbol{q}(x,t) = \exists y, z, t' \left((t < t') \wedge Q(x,y,t) \wedge S'(y,z,t')\right).$$

The map $h = \{x \mapsto a, y \mapsto v, z \mapsto u_1, t \mapsto 1, t' \mapsto 2\}$ is a witness for R, with $\boldsymbol{q}_h = \boldsymbol{q}$ and w_h is the following formula

$$\exists r_a \exists x_h \left(\mathsf{ext}^{\mathcal{T}}_{\exists R}(x_h, r_a) \wedge (x_h = x) \wedge \right.$$
$$\left.\exists r_v \exists r_{u_1} \left(\delta^0(r_a, r_v) \wedge \delta^1(r_v, r_{u_1}) \wedge \delta^1(r_v, t) \wedge \delta^1(r_{u_1}, t')\right)\right).$$

We can now define a rewriting for $\boldsymbol{q}(\vec{x}, \vec{s})$ and $\mathcal{T}$. Let $\mathfrak{T}$ be the set of all witnesses for $\boldsymbol{q}$ and $\mathcal{T}$. We call a subset $\mathfrak{S} \subseteq \mathfrak{T}$ *consistent* if $(\mathcal{X}_{h_1} \cup \mathcal{Y}_{h_1}) \cap (\mathcal{X}_{h_1} \cup \mathcal{Y}_{h_2}) \subseteq \mathcal{X}_{h_1} \cap \mathcal{X}_{h_2}$, for any distinct $h_1, h_2 \in \mathfrak{S}$. Assuming that $\vec{y}$ contains all the quantified variables in $\boldsymbol{q}$ and $\boldsymbol{q} \setminus \mathfrak{S}$ is the sub-query of $\boldsymbol{q}$ obtained by removing the atoms in $\boldsymbol{q}_h$, $h \in \mathfrak{S}$, other than $(\tau < \tau')$ and $(\tau = \tau')$, we set:

$$\boldsymbol{q}^*(\vec{x}, \vec{s}) \;=\; \exists \vec{y} \bigvee_{\substack{\mathfrak{S} \subseteq \mathfrak{T} \\ \mathfrak{S} \text{ consistent}}} \Big(\bigwedge_{h \in \mathfrak{S}} \mathsf{w}_h \;\wedge\; \mathsf{ext}^{\mathcal{T}}_{\boldsymbol{q} \setminus \mathfrak{S}} \Big).$$

Theorem 16 *Let $\mathcal{T}$ be a TQL TBox in CoNF and $\boldsymbol{q}(\vec{x}, \vec{s})$ a totally ordered CQ. Then, for any consistent KB $(\mathcal{T}, \mathcal{A})$ and any tuples $\vec{a} \subseteq \mathsf{ind}(\mathcal{A})$ and $\vec{n} \subseteq \mathbb{Z}$,*

$$(\mathcal{T}, \mathcal{A}) \models \boldsymbol{q}(\vec{a}, \vec{n}) \quad \textit{iff} \quad \mathcal{A}^{\mathbb{Z}} \models \boldsymbol{q}^*(\vec{a}, \vec{n}).$$

$(\mathcal{T}, \mathcal{A})$ is inconsistent iff $(\mathcal{T}^{\perp}, \mathcal{A}) \models \boldsymbol{q}^{\perp}$.

Theorem 2 now follows by Lemma 3.

6 Non-Rewritability

In this section, we show that the language *TQL* is nearly optimal as far as rewritability of CQs and ontologies is concerned.

We note first, that the syntax of *TQL* allows concept inclusions and role inclusions; 'mixed' axioms such as the datalog rule $A(x,t) \wedge R(x,y,t) \rightarrow B(x,t)$ are not expressible. The reason is that mixed rules often lead to non-rewritability, as is well known from the DL $\mathcal{EL}$. For example, there does not exist an FO-query $\boldsymbol{q}(x,t)$ such that $(\mathcal{T}, \mathcal{A}) \models A(a,n)$ iff $\mathcal{A}^{\mathbb{Z}} \models \boldsymbol{q}(a,n)$ for $\mathcal{T} = \{A(y,t) \wedge R(x,y,t) \rightarrow A(x,t)\}$ since such a query has to express that at time-point t there is an R-path from x to some y with $A(y,t)$.

Second, it would seem to be natural to extend *TQL* with the temporal next/previous-time operators as concept or role constructs. However, again this would lead to non-rewritability: any FO-rewriting for $A(x,t)$ and $\{\bigcirc_P A \sqsubseteq B,\ \bigcirc_P B \sqsubseteq A\}$ has to express that there exists $n \geq 0$ such that $A(x, t-2n)$ or $B(x, t-(2n+1))$, which is impossible [Libkin, 2004].

Another natural extension would be inclusions of the form $A \sqsubseteq \Diamond_F B$. (Note that inclusions of the form $A \sqsubseteq \exists R.B$ are expressible in *OWL 2 QL*.) But again such an extension would ruin rewritability. The reason is that temporal precedence $<$ is a total order, and so one can construct an ABox $\mathcal{A}$ and a UCQ $\boldsymbol{q}(x) = \boldsymbol{q}_1 \vee \boldsymbol{q}_2$ such that $(\mathcal{T}, \mathcal{A}) \models \boldsymbol{q}(a)$ but $(\mathcal{T}, \mathcal{A}) \not\models \boldsymbol{q}_i(a)$, $i = 1, 2$, for $\mathcal{T} = \{A \sqsubseteq \Diamond_F B\}$. Indeed, we take $\mathcal{A} = \{A(a,0), C(a,1)\}$ and

$$\boldsymbol{q}_1(x) = \exists t\, (C(x,t) \wedge B(x,t)),$$
$$\boldsymbol{q}_2(x) = \exists t, t'\, ((t < t') \wedge C(x,t) \wedge B(x,t')).$$

In fact, by reduction of 2+2-SAT [Schaerf, 1993], we prove the following:

Theorem 17 *Answering CQs over the TBox $\{A \sqsubseteq \Diamond_F B\}$ is* coNP*-hard for data complexity.*

7 Related Work

The Semantic Web community has developed a variety of extensions of RDF/S and OWL with validity time [Motik, 2012; Pugliese *et al.*, 2008; Gutierrez *et al.*, 2007]. The focus of this line of research is on representing and querying time stamped RDF triples or OWL axioms. In contrast, in our language only instance data are time stamped , while the ontology formulates time independent constraints that describe how the extensions of concepts and roles can change over time. In the temporal DL literature, a similar distinction has been discussed as the difference between temporalised axioms and temporalised concepts/roles; the expressive power of the respective languages is incomparable [Gabbay *et al.*, 2003; Baader *et al.*, 2012].

In Theorem 8, we show rewritability using boundedness of recursion. This connection between first-order definability and boundedness is well known from the datalog and logic literature where boundedness has been investigated extensively [Gaifman *et al.*, 1987; van der Meyden, 2000; Kreutzer *et al.*, 2007]. Grohe and Schwandtner [2009] investigate boundedness for datalog programs on linear orders; the results are different from ours since the linear order is the only predicate symbol of the datalog programs considered and no further restrictions (comparable to ours) are imposed.

8 Conclusion

In this paper, we have proved UCQ rewritability for conjunctive queries and *TQL* ontologies over data instances with validity time. Our focus was solely on the existence of rewritings, and we did not consider efficiency issues such as finding shortest rewritings, using temporal intervals in the data representation or mappings between temporal databases and ontologies. We only note here that these issues are of practical importance and will be addressed in future work. It would also be of interest to investigate the possibilities to increase the expressive power of both ontology and query language. For example, we believe that the extension of *TQL* with the next/previous time operators, which can only occur in TBox axioms not involved in cycles, will still enjoy rewritability. We can also increase the expressivity of conjunctive queries by allowing the arithmetic operations $+$ and $\times$ over temporal terms, which would make the CQ $A(x,t)$ and the TBox $\{\bigcirc_P A \sqsubseteq B,\ \bigcirc_P B \sqsubseteq A\}$ rewritable in the extended language.

References

[Artale and Franconi, 2005] A. Artale and E. Franconi. Temporal description logics. In *Handbook of Temporal Reasoning in Artificial Intelligence*, Foundations of Artificial Intelligence, pages 375–388. Elsevier, 2005.

[Artale *et al.*, 2007] A. Artale, D. Calvanese, R. Kontchakov, V. Ryzhikov, and M. Zakharyaschev. Reasoning over extended ER models. In *Proc. of ER-2007*, volume 4801 of *LNCS*, pages 277–292. Springer, 2007.

[Artale *et al.*, 2009] A. Artale, D. Calvanese, R. Kontchakov, and M. Zakharyaschev. The DL-Lite family and relations. *J. of Artifical Intelligence Research*, 36:1–69, 2009.

[Artale *et al.*, 2010] A. Artale, R. Kontchakov, V. Ryzhikov, and M. Zakharyaschev. Complexity of reasoning over temporal data models. In *Proc. of ER-2010*, volume 6412 of *LNCS*, pages 174–187. Springer, 2010.

[Artale *et al.*, 2013] A. Artale, R. Kontchakov, F. Wolter, and M. Zakharyaschev. Temporal Description Logic for Ontology-Based Data Access (Extended Version). *CoRR*, abs/1304.5185, 2013.

[Baader *et al.*, 2012] F. Baader, S. Ghilardi, and C. Lutz. LTL over description logic axioms. *ACM Trans. Comput. Log.*, 13(3):21, 2012.

[Berardi *et al.*, 2005] D. Berardi, D. Calvanese, and G. De Giacomo. Reasoning on UML class diagrams. *Artificial Intelligence*, 168(1–2):70–118, 2005.

[Calì *et al.*, 2012] A. Calì, G. Gottlob, and T. Lukasiewicz. A general datalog-based framework for tractable query answering over ontologies. *J. Web Sem.*, 14:57–83, 2012.

[Calvanese *et al.*, 2006] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Data complexity of query answering in description logics. In *Proc. of KR 2006*, pages 260–270, 2006.

[Calvanese *et al.*, 2007] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *J. of Automated Reasoning*, 39(3):385–429, 2007.

[Chortaras *et al.*, 2011] A. Chortaras, D. Trivela, and G. Stamou. Goal-oriented query rewriting for OWL 2 QL. In *Proc. of DL 2011*, volume 745 of *CEUR-WS*, 2011.

[Franconi and Toman, 2011] E. Franconi and D. Toman. Fixpoints in temporal description logics. In *Proc. of IJCAI 2011*, pages 875–880, 2011.

[Gabbay *et al.*, 2003] D. Gabbay, A. Kurucz, F. Wolter, and M. Zakharyaschev. *Many-dimensional modal logics: theory and applications*. Studies in Logic. Elsevier, 2003.

[Gaifman *et al.*, 1987] H. Gaifman, H. G. Mairson, Y. Sagiv, and M. Y. Vardi. Undecidable optimization problems for database logic programs. In *Proc. of LICS 87*, pages 106–115, 1987.

[Gottlob and Schwentick, 2011] G. Gottlob and T. Schwentick. Rewriting ontological queries into small nonrecursive datalog programs. In *Proc. of DL 2011*, volume 745 of *CEUR-WS*, 2011.

[Gottlob *et al.*, 2011] G. Gottlob, G. Orsi, and A. Pieris. Ontological queries: Rewriting and optimization. In *Proc. of ICDE 2011*, pages 2–13. IEEE Computer Society, 2011.

[Grohe and Schwandtner, 2009] M. Grohe and G. Schwandtner. The complexity of datalog on linear orders. *Logical Methods in Computer Science*, 5(1), 2009.

[Gutiérrez-Basulto and Klarman, 2012] V. Gutiérrez-Basulto and S. Klarman. Towards a unifying approach to representing and querying temporal data in description logics. In *Proc. of RR 2012*, volume 7497 of *LNCS*, pages 90–105. Springer, 2012.

[Gutierrez *et al.*, 2007] C. Gutierrez, C. A. Hurtado, and A. A. Vaisman. Introducing time into RDF. *IEEE Trans. Knowl. Data Eng.*, 19(2):207–218, 2007.

[König *et al.*, 2012] M. König, M. Leclère, M.-L. Mugnier, and M. Thomazo. A sound and complete backward chaining algorithm for existential rules. In *Proc. of RR 2012*, volume 7497 of *LNCS*, pages 122–138. Springer, 2012.

[Kontchakov *et al.*, 2010] R. Kontchakov, C. Lutz, D. Toman, F. Wolter, and M. Zakharyaschev. The combined approach to query answering in DL-Lite. In *Proc. of KR 2010*. AAAI Press, 2010.

[Kreutzer *et al.*, 2007] S. Kreutzer, M. Otto, and N. Schweikardt. Boundedness of monadic FO over acyclic structures. In *Proc. of ICALP 2007*, pages 571–582, 2007.

[Libkin, 2004] L. Libkin. *Elements Of Finite Model Theory*. Springer, 2004.

[Lutz *et al.*, 2008] C. Lutz, F. Wolter, and M. Zakharyaschev. Temporal description logics: A survey. In *Proc. of TIME 2008*, pages 3–14. IEEE Computer Society, 2008.

[Motik, 2012] B. Motik. Representing and querying validity time in RDF and OWL: A logic-based approach. *J. Web Sem.*, 12:3–21, 2012.

[Pérez-Urbina *et al.*, 2009] H. Pérez-Urbina, B. Motik, and I. Horrocks. A comparison of query rewriting techniques for DL-Lite. In *Proc. of DL 2009*, volume 477 of *CEUR-WS*, 2009.

[Pugliese *et al.*, 2008] A. Pugliese, O. Udrea, and V. S. Subrahmanian. Scaling RDF with time. In *Proc. of WWW 2008*, pages 605–614, 2008.

[Rosati and Almatelli, 2010] R. Rosati and A. Almatelli. Improving query answering over DL-Lite ontologies. In *Proc. of KR 2010*. AAAI Press, 2010.

[Schaerf, 1993] A. Schaerf. On the complexity of the instance checking problem in concept languages with existential quantification. *J. of Intel. Inf. Systems*, 2:265–278, 1993.

[van der Meyden, 2000] R. van der Meyden. Predicate boundedness of linear monadic datalog is in PSPACE. *Int. J. Found. Comput. Sci.*, 11(4):591–612, 2000.