Data Complexity and Rewritability of Ontology-Mediated Queries in Metric Temporal Logic under the Event-Based Semantics

Vladislav Ryzhikov[1], **Przemysław Andrzej Wałęga**[2] and **Michael Zakharyaschev**[1]
[1]Birkbeck, University of London, UK
[2]University of Oxford, U.K and University of Warsaw, Poland
{vlad, michael}@dcs.bbk.ac.uk, przemyslaw.walega@cs.ox.ac.uk

Abstract

We investigate the data complexity of answering queries mediated by metric temporal logic ontologies under the event-based semantics assuming that data instances are finite timed words timestamped with binary fractions. We identify classes of ontology-mediated queries answering which can be done in AC^0, NC^1, L, NL, P, and coNP for data complexity, provide their rewritings to first-order logic and its extensions with primitive recursion, transitive closure or datalog, and establish lower complexity bounds.

1 Introduction

We are concerned with the following problem: given a formula Π of metric temporal logic *MTL* and an atomic proposition A, is it possible to construct a query $\boldsymbol{Q}(x)$ in some standard query language such that, for any data instance $\mathcal{D}$ of atoms timestamped with binary fractions and any timestamp t from $\mathcal{D}$, we have $\Pi, \mathcal{D} \models A(t)$ iff $\boldsymbol{Q}(t)$ is true in $\mathcal{D}$?

MTL was originally designed for modelling and reasoning about real-time systems [Koymans, 1990; Alur and Henzinger, 1993; Bouyer *et al.*, 2018]. Recently, combinations of *MTL* with description logics have been suggested as temporal ontology languages [Gutiérrez-Basulto *et al.*, 2016b; Baader *et al.*, 2017]. Datalog with *MTL*-operators was used by [Brandt *et al.*, 2018] for practical ontology-based access to log data aiming to facilitate detection of events in asynchronous systems based on sensor measurements. For example, a Siemens turbine has a coast down if the rotor speed was below 1500 in the previous 30 seconds, while no more than 2 minutes before that the speed was above 6600 for 30 seconds. The event 'coast down' can be encoded by the following *MTL*-formula, where $\diamondminus_{(r,s]}\varphi$ ($\boxminus_{(r,s]}\varphi$) is true at a timestamp t if φ holds at some (respectively, all) t' with $r < t - t' \leq s$:

$$\boxminus_{(0,30s]}\mathsf{speed}_{<1500} \land \diamondminus_{(0,2m]}\boxminus_{(0,30s]}\mathsf{speed}_{>6600} \to \mathsf{cdown}.$$

To find when a coast down occurred, a Siemens engineer can now simply execute the query $\mathsf{cdown}(x)$ mediated by this formula. Answering *datalogMTL* queries in the streaming setting was considered by [Wałęga *et al.*, 2019].

The underpinning idea of classical ontology-based data access (OBDA) [Calvanese *et al.*, 2007; Xiao *et al.*, 2018] is a reduction of ontology-mediated query (OMQ) answering to standard database query evaluation. As known from descriptive complexity [Immerman, 1999], the existence of such reductions, or *rewritings*, is closely related to the data complexity of OMQ answering, which is by now well understood for atemporal OMQs both uniformly (for all OMQs in a given language) and non-uniformly (for individual OMQs) [Gottlob *et al.*, 2014; Bienvenu and Ortiz, 2015; Bienvenu *et al.*, 2014; Lutz and Sabellek, 2017].

Temporal ontology and query languages have attracted attention of datalog and description logic communities since the 1990s; see [Baudinet *et al.*, 1993; Chomicki and Toman, 1998; Lutz *et al.*, 2008; Artale *et al.*, 2017] for surveys. In recent years, the proliferation of temporal data from various sources and its importance for analysing the behaviour of complex systems and decision making in all economic sectors have intensified research into formalisms that can be used for querying temporal databases and streaming data [Soylu *et al.*, 2017; Beck *et al.*, 2018; Ronca *et al.*, 2018]. OBDA with atemporal ontologies and query languages with linear temporal logic *LTL* operators has been in use since [Baader *et al.*, 2013; Özçep and Möller, 2014]. Rewritability and data complexity of OMQs in the description logics *DL-Lite* and $\mathcal{EL}$ extended with *LTL* operators were considered in [Artale *et al.*, 2015; Gutiérrez-Basulto *et al.*, 2016a].

Here, we investigate the (uniform) rewritability and data complexity problems for basic OMQs given in metric temporal logic *MTL*, assuming that data instances are finite sets of atoms timestamped by dyadic rationals and that *MTL* is interpreted under the event-based semantics where atoms refer to events (state changes) rather than to states themselves [Ouaknine and Worrell, 2008]. *MTL* is more succinct, expressive, and versatile compared to *LTL*, being able to model both synchronous (discrete) and asynchronous (real-time) settings.

First, we observe that answering arbitrary *MTL*-OMQs is coNP-complete for data complexity (in contrast to NC^1-completeness for *LTL*-OMQs). OMQs in the Horn fragment *hornMTL* are P-complete and rewritable to datalog(FO), which extends datalog with FO-formulas built from EDB predicates; in fact, we establish P-hardness already for the fragment $\mathit{coreMTL}^{\boxminus}$ of *hornMTL* with binary rules (like in *OWL 2 QL*) and box operators only. OMQs in $\mathit{coreMTL}^{\diamondminus}$ turn out to be FO(TC)-rewritable (FO with transitive closure) and NL-hard. We then classify *MTL*-OMQs by the type of ranges ϱ constraining their temporal operators $\diamondminus_\varrho$ and $\boxminus_\varrho$: infinite (r, ∞) and $[r, \infty)$, punctual $[r, r]$, and arbitrary non-punctual ϱ. We show that OMQs of the first type are FO-rewritable and can be answered in AC^0. OMQs of the second type are FO(RPR)-rewritable (FO with relational primitive recursion) and NC^1-complete. For the third type, we obtain an NL upper bound with rewritability to FO(TC) and NC^1 lower bound; for *hornMTL*-OMQs of this type, the results are improved to L with rewritability to FO(DTC) (FO with deterministic closure).

The omitted proofs can be found in [Ryzhikov *et al.*, 2019].

2 *MTL* Ontology-Mediated Queries

In the context of event monitoring, we consider a 'past' variant of *MTL*, which is a propositional modal logic with constrained operators $\diamondminus_\varrho$ 'sometime in the past within range ϱ' and $\boxminus_\varrho$ 'always in the past within range ϱ,' interpreted over finite timed words under the event-based semantics. We assume that timestamps in timed words are given as non-negative dyadic rational numbers (finite binary fractions), the set of which is denoted by $\mathbb{Q}_2^{\geq 0}$. The ranges ϱ in $\diamondminus_\varrho$ and $\boxminus_\varrho$ are non-empty intervals with end-points in $\mathbb{Q}_2^{\geq 0} \cup \{\infty\}$.

An *MTL-program*, Π, is a finite set of *rules* of the form

$$\vartheta_1 \wedge \dots \wedge \vartheta_k \to \vartheta_{k+1} \vee \dots \vee \vartheta_{k+l}, \qquad (1)$$

where each ϑ_i takes the form A, $\diamondminus_\varrho A$, or $\boxminus_\varrho A$, for an atomic proposition A. We denote the empty $\wedge$ by $\top$ (truth) and empty $\vee$ by $\bot$ (falsehood). Using fresh atoms, every *MTL*-formula can be transformed to an equivalent (in the sense of giving the same answers to queries) *MTL*-program.

An *MTL*-program is called a *hornMTL-program* if, in all of its rules (1), $l \leq 1$ and ϑ_{k+1} is an atom. As usual, ϑ_{k+1} is called the *head* of the rule and $\vartheta_1 \wedge \dots \wedge \vartheta_k$ its *body*. A *hornMTL*-program is a *coreMTL-program* if $k + l \leq 2$. An *MTL*- (*hornMTL*- or *coreMTL*-) *ontology-mediated query* (OMQ) takes the form $\boldsymbol{q} = (\Pi, A)$, where Π is an *MTL*- (resp., *hornMTL*- or *coreMTL*-) program and A an atom.

Intuitively, a *data instance*, $\mathcal{D}$, can be thought of as a word $\boldsymbol{A}_0(\bar{0}), \dots, \boldsymbol{A}_k(\bar{k})$ with timestamps $\bar{0} < \dots < \bar{k}$, $\bar{i} \in \mathbb{Q}_2^{\geq 0}$, where each $\boldsymbol{A}_i$ is the set of atoms that are true at $\bar{i}$. Formally, we represent $\mathcal{D}$ as the FO-structure

$$\mathcal{D} \;=\; (\Delta, <, \Theta, \mathsf{bit}_{in}, \mathsf{bit}_{fr}, A_1^{\mathcal{D}}, \dots, A_p^{\mathcal{D}}), \qquad (2)$$

with domain $\Delta = \{0, \dots, \ell\}$ ordered by $<$, *timestamps* $\Theta = \{0, \dots, k\}$, $1 \leq k \leq \ell$, and subsets $A_i^{\mathcal{D}} \subseteq \Theta$. The ternary predicates bit_{in} and bit_{fr} are such that, for any $n \in \Theta$ and $i \in \Delta$, there are unique $b_i, c_i \in \{0, 1\}$ with $\mathsf{bit}_{in}(n, i, b_i)$ and $\mathsf{bit}_{fr}(n, i, c_i)$. These predicates give the *value* $\bar{n} \in \mathbb{Q}_2^{\geq 0}$ of every timestamp $n \in \Theta$: $\bar{n} = b_\ell \dots b_0.c_\ell \dots c_0$ iff $\mathsf{bit}_{in}(n, i, b_i)$ and $\mathsf{bit}_{fr}(n, i, c_i)$ hold for all $i \leq \ell$. We assume that $\bar{n} < \bar{m}$ if $n < m$. For any $r \in \mathbb{Q}_2^{\geq 0}$, we can define an FO-formula $\mathsf{dist}_{<r}(x, y)$ that holds in $\mathcal{D}$ iff $x, y \in \Theta$ and $0 \leq \bar{x} - \bar{y} < r$, its variants $\mathsf{dist}_{>r}(x, y)$, $\mathsf{dist}_{=r}(x, y)$, etc. Using these, we can further define FO-formulas $\mathsf{in}_\varrho(x, y)$ for $\bar{x} - \bar{y} \in \varrho$, $\mathsf{suc}(x, y)$ for 'x is an immediate successor of y in $\mathcal{D}$', and FO-expressible constants $\min = 0$ and $\max = k$.

An *event-based interpretation* over $\mathcal{D}$ is a structure

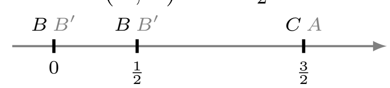
$$\mathcal{I} \;=\; (\Delta, <, \Theta, \mathsf{bit}_{in}, \mathsf{bit}_{fr}, A_1^{\mathcal{I}}, \dots, A_p^{\mathcal{I}}), \;\; A_i^{\mathcal{D}} \subseteq A_i^{\mathcal{I}} \subseteq \Theta,$$

where the Boolean connectives are interpreted as usual and

$$(\diamondminus_\varrho A)^{\mathcal{I}} \;=\; \{t \in \Theta \mid \exists t' \in \Theta\, (\mathsf{in}_\varrho(t, t') \wedge t' \in A^{\mathcal{I}})\},$$
$$(\boxminus_\varrho A)^{\mathcal{I}} \;=\; \{t \in \Theta \mid \forall t' \in \Theta\, (\mathsf{in}_\varrho(t, t') \to t' \in A^{\mathcal{I}})\}.$$

An interpretation $\mathcal{I}$ over $\mathcal{D}$ is a *model* of an *MTL*-program Π and $\mathcal{D}$ if, for any rule (1) in Π and any $t \in \Theta$, whenever $t \in \vartheta_i^{\mathcal{I}}$ for all i, $1 \leq i \leq k$, then $t \in \vartheta_{k+j}^{\mathcal{I}}$ for some j, $1 \leq j \leq l$. We call $\mathcal{D}$ and Π *consistent* if there is a model of Π and $\mathcal{D}$.

Henceforth, we write $\mathsf{ts}(\mathcal{D})$ for the set Θ of timestamps in (2) and often informally identify $t \in \mathsf{ts}(\mathcal{D})$ with its value $\bar{t}$. We call $t \in \mathsf{ts}(\mathcal{D})$ (and so $\bar{t}$) a *certain answer* to $\boldsymbol{q} = (\Pi, A)$ over $\mathcal{D}$ if $t \in A^{\mathcal{I}}$ for every model $\mathcal{I}$ of $\mathcal{D}$ and Π. The *OMQ answering problem* for $\boldsymbol{q}$ is to decide, given $\mathcal{D}$ and $t \in \mathsf{ts}(\mathcal{D})$, whether t is a certain answer to $\boldsymbol{q}$ over $\mathcal{D}$. To illustrate, consider $\Pi = \{\boxminus_{[0,2)} B \to B', \ \diamondminus_{[1,1]} B' \to A\}$, $\mathcal{D}_1 = \{B(0), B(1/2), C(3/2)\}$ and $\mathcal{D}_2 = \{B(0), C(3/2)\}$. Then $3/2$ is a certain answer to (Π, A) over $\mathcal{D}_1$, but there are no certain answers to (Π, A) over $\mathcal{D}_2$:



We are interested in the *data complexity* of OMQ answering, that is, regard $\mathcal{D}$ as the only input to the problem and assume $\boldsymbol{q}$ to be fixed.

Let $\mathcal{L}$ be a query language over FO-structures (2). An OMQ $\boldsymbol{q}$ is said to be *$\mathcal{L}$-rewritable* if there is an $\mathcal{L}$-query $\boldsymbol{Q}(x)$, called an *$\mathcal{L}$-rewriting* of $\boldsymbol{q}$, such that, for any data instance $\mathcal{D}$, a timestamp $t \in \mathsf{ts}(\mathcal{D})$ is a certain answer to $\boldsymbol{q}$ over $\mathcal{D}$ iff $\mathcal{D} \models \boldsymbol{Q}(t)$. Our target query languages $\mathcal{L}$ include:

- FO($<$) and its extension FO($<, +$) with the predicate PLUS (e.g., $\exists x\, \mathrm{PLUS}(x, x, \max)$ says that $|\Theta|$ is odd); evaluating such queries is in AC^0 for data complexity;
- FO(RPR), i.e., FO($<$) with relational primitive recursion, which is in NC^1 [Compton and Laflamme, 1990];
- FO(TC) and FO(DTC), i.e., FO($<$) with transitive and deterministic transitive closure, which are in NL and L, respectively [Immerman, 1999];
- datalog(FO), i.e., datalog queries with additional FO-formulas built from EDB predicates in their rule bodies, which are in P [Grädel, 1991].

All of them save datalog(FO) can be implemented in SQL. $\mathcal{L}$-rewritability of an OMQ $\boldsymbol{q}$ means that answering $\boldsymbol{q}$ is in the same data-complexity class as evaluation of $\mathcal{L}$-queries.

Given a *hornMTL*-program Π and a data instance $\mathcal{D}$, we define a set $\mathfrak{C}_{\Pi,\mathcal{D}}$ of pairs of the form (ϑ, t) that contains all answers to OMQs with Π over $\mathcal{D}$. We start by setting $\mathfrak{C} = \mathcal{D}$ and denote by $\mathsf{cl}(\mathfrak{C})$ the result of applying exhaustively and non-recursively the following rules to $\mathfrak{C}$:

- if $\vartheta_1 \wedge \dots \wedge \vartheta_k \to \vartheta$ is in Π and $(\vartheta_i, t) \in \mathfrak{C}$, for all i, $1 \leq i \leq k$, then we add (ϑ, t) to $\mathfrak{C}$;
- if $\diamondminus_\varrho B$ occurs in Π, $(B, t') \in \mathfrak{C}$, and $\mathsf{in}_\varrho(t, t')$ holds for some $t \in \mathsf{ts}(\mathcal{D})$, then we add $(\diamondminus_\varrho B, t)$ to $\mathfrak{C}$;

– if $\boxminus_\varrho B$ occurs in Π, $t \in \mathsf{ts}(\mathcal{D})$ and $(B, t') \in \mathfrak{C}$ for all $t' \in \mathsf{ts}(\mathcal{D})$ with $\mathsf{in}_\varrho(t, t')$, then we add $(\boxminus_\varrho B, t)$ to $\mathfrak{C}$.

It should be clear that there is some $N < \omega$ polynomially depending on Π and $\mathcal{D}$ such that $\mathsf{cl}^N(\mathfrak{C}) = \mathsf{cl}^{N+1}(\mathfrak{C})$. We then set $\mathfrak{C}_{\Pi,\mathcal{D}} = \mathsf{cl}^N(\mathcal{D})$. We can regard $\mathfrak{C}_{\Pi,\mathcal{D}}$ as a (minimal) model of Π and $\mathcal{D}$ with domain $\mathsf{ts}(\mathcal{D})$ in which $t \in B^{\mathfrak{C}_{\Pi,\mathcal{D}}}$ iff $(B, t) \in \mathfrak{C}_{\Pi,\mathcal{D}}$ The proof of the following is standard:

Theorem 1. *For a hornMTL-OMQ (Π, A), (i) Π is inconsistent with $\mathcal{D}$ iff $(\bot, t) \in \mathfrak{C}_{\Pi,\mathcal{D}}$; (ii) a timestamp $t \in \mathsf{ts}(\mathcal{D})$ is a certain answer to a hornMTL-OMQ (Π, A) over $\mathcal{D}$ iff either $\mathfrak{C}_{\Pi,\mathcal{D}} \models A[t]$ or Π is inconsistent with $\mathcal{D}$.*

Note in passing that, as a consequence, we obtain the following reduction of $\mathcal{L}$-rewritability of more general *hornMTL*-OMQs (Π, φ) with *positive FO-queries* φ (built from atoms, $\wedge$, $\vee$, $\forall$, and $\exists$) to $\mathcal{L}$-rewritability of atomic OMQs we deal with in this paper:

Corollary 2. *Let (Π, φ) be a hornMTL-OMQ with a positive FO-query φ. If (Π, A) has an $\mathcal{L}$-rewriting $\boldsymbol{Q}_A(x)$, for every atom A in φ, then the result of simultaneous replacing every $A(x)$ in φ with $\boldsymbol{Q}_A(x)$ is an $\mathcal{L}$-rewriting of (Π, φ).*

3 OMQs with Arbitrary Ranges

We begin by establishing (non-)rewritability and data complexity of answering OMQs in various classes where *arbitrary* ranges in temporal operators are allowed. We denote by *coreMTL*$^{\boxminus}$ (*coreMTL*$^{\diamondminus}$) the restriction of *coreMTL* to the language with operators $\boxminus_\varrho$ (respectively, $\diamondminus_\varrho$) only.

Theorem 3. (i) *Answering MTL-OMQs is* coNP*-complete for data complexity;* (ii) *hornMTL-OMQs are* datalog(FO)*-rewritable, with* coreMTL$^{\boxminus}$*-OMQs being* P*-hard;* (iii) *coreMTL*$^{\diamondminus}$*-OMQs are* FO(TC)*-rewritable and* NL*-hard.*

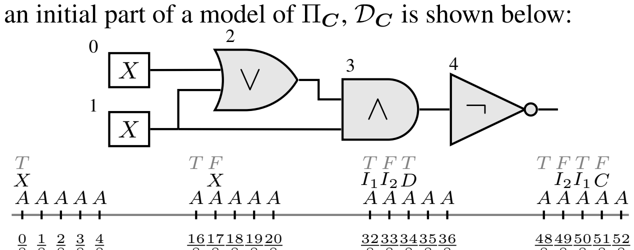
Proof sketch. (i) The membership in coNP is trivial. We establish coNP-hardness by reduction of NP-complete circuit satisfiability [Arora and Barak, 2009]. Let $\boldsymbol{C}$ be a Boolean circuit with N_0-many (two-input) AND, OR and (one-input) NOT gates enumerated by consecutive numbers starting from 0 so that if there is an edge from n to m, then $n < m$. Take the minimal $N = 2^k \geq N_0$ and a data instance $\mathcal{D}_{\boldsymbol{C}}$ with the facts

– $A(2n + i/N)$, if n is a gate and $0 \leq i < N_0$;
– $X(2n + n/N)$, if n is an input gate;
– $N(2n + n/N)$, if n is a NOT gate;
– $D(2n + n/N)$, if n is an OR gate;
– $C(2n + n/N)$, if n is an AND gate;
– $I_0(2n + m/N)$, if n is a NOT gate with input gate m;
– $I_1(2n+m/N)$ and $I_2(2n+k/N)$, if n is an OR or AND gate with input gates m and k.

Let $\Pi_{\boldsymbol{C}}$ be an *MTL*-program with the following rules:

$$X \to T \vee F, \quad \diamondminus_{[2,2]} T \to T, \quad \diamondminus_{[2,2]} F \to F,$$
$$N \wedge \diamondminus_{[0,1]}(I_0 \wedge T) \to F, \quad N \wedge \diamondminus_{[0,1]}(I_0 \wedge F) \to T,$$
$$D \wedge \diamondminus_{[0,1]}(I_1 \wedge T) \to T, \quad D \wedge \diamondminus_{[0,1]}(I_2 \wedge T) \to T,$$
$$C \wedge \diamondminus_{[0,1]}(I_1 \wedge F) \to F, \quad C \wedge \diamondminus_{[0,1]}(I_2 \wedge F) \to F,$$
$$D \wedge \diamondminus_{[0,1]}(I_1 \wedge F) \wedge \diamondminus_{[0,1]}(I_2 \wedge F) \to F,$$
$$C \wedge \diamondminus_{[0,1]}(I_1 \wedge T) \wedge \diamondminus_{[0,1]}(I_2 \wedge T) \to T.$$

Then $\mathbf{C}$ is satisfiable iff the maximal number in $\mathsf{ts}(\mathcal{D})$ is not a certain answer to $(\Pi_{\boldsymbol{C}}, F)$ over $\mathcal{D}_{\boldsymbol{C}}$. An example of $\boldsymbol{C}$ and an initial part of a model of $\Pi_{\boldsymbol{C}}$, $\mathcal{D}_{\boldsymbol{C}}$ is shown below:



(ii) We construct a datalog(FO) rewriting $(\Pi', G(x))$ of a *hornMTL*-OMQ $\boldsymbol{q} = (\Pi, A)$. To begin with, we add to Π the rule $P(x) \to P'(x, x)$ for each P in Π. The other rules in Π' are obtained from the rules in Π by the following transformations. We replace every atom B not under the scope of a temporal operator with $B'(x, x)$ and every $\diamondminus_{[r,s]} B$ with

$$B'(w, z) \wedge \mathsf{dist}_{\geq r}(x, w) \wedge \mathsf{dist}_{\leq s}(x, z)$$

and similarly for other types of ranges ϱ in $\diamondminus_\varrho B$. Intuitively, $\Pi', \mathcal{D} \models B'(x, y)$ iff $(B, t) \in \mathfrak{C}_{\Pi,\mathcal{D}}$, for each $t \in [x, y]$ from $\mathsf{ts}(\mathcal{D})$. We replace every $\boxminus_{[r,s]} B$ in the body of a rule with

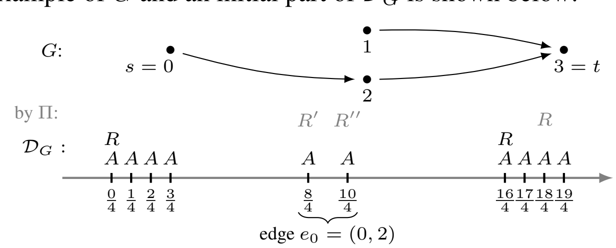
$$B'(w, z) \wedge \mathsf{dist}_{\geq s}(x, w) \wedge \mathsf{dist}_{\leq r}(x, z) \wedge \mathsf{dist}_{\geq (s-r)}(z, w)$$

and similarly for other types of ranges. Finally, we add the following rules to the resulting program:

$$A'(y, z) \wedge (y \leq x \leq z) \to G(x),$$
$$B'(x, y) \wedge B'(z, z) \wedge \mathsf{suc}(y, z) \to B'(x, z).$$

Note that the obtained datalog program Π' contains FO-definable EDB predicates such as $\mathsf{dist}_{\geq r}(x, w)$ and $\mathsf{suc}(y, z)$ in rule bodies. Clearly, t is a certain answer to $\boldsymbol{q}$ over any given data instance $\mathcal{D}$ iff t is an answer to $(\Pi', G(x))$ over $\mathcal{D}$. The proof of hardness via PSA is similar to that in (iii).

(iii) The upper bound can be shown by reduction to FO(TC) via linear datalog(FO). We prove NL-hardness by reduction of the reachability problem in acyclic digraphs. Let G be such a digraph with N_0 vertices enumerated by consecutive natural numbers starting from 0 so that, if there is an edge from n to m, then $n < m$. Let $e_0, \dots, e_{k-1}$ be the lexicographical order of edges. Take the minimal $N = 2^i \geq N_0$ for $i \in \mathbb{N}$. Suppose we want to check whether a vertex t is accessible from s. Let $\mathcal{D}_G$ consist of the atoms $A(4i+n/N)$, for $0 \leq i \leq k$ and a vertex n; $A(2+4i+n/N)$, $A(2+4i+m/N)$, for every edge $e_i = (n, m)$; $R(4i + s/N)$, for $0 \leq i \leq k$. An example of G and an initial part of $\mathcal{D}_G$ is shown below:



Let Π be a *coreMTL*$^{\diamondminus}$ program with the following rules:

$$\diamondminus_{[2,2]} R \to R', \ \diamondminus_{(0,1]} R' \to R'', \ \diamondminus_{[2,2]} R'' \to R, \ \diamondminus_{[4,4]} R \to R.$$

Then $4k + t/N$ is a certain answer to (Π, R) over $\mathcal{D}_G$ iff t is reachable from s in G. ❑

To obtain finer complexity results, we classify *MTL*-OMQs by the type of ranges ϱ in their operators $\diamondminus_\varrho$ and $\boxminus_\varrho$: infinite, punctual, and non-punctual. Let $\langle$ be one of (or [.

4 OMQs with Ranges $\langle r, \infty)$

First, consider OMQs with $\diamondminus_{\langle r,\infty)}$ and $\boxminus_{\langle r,\infty)}$, which resemble *LTL*-operators 'sometime' and 'always in the past'. Using partially-ordered automata, [Artale *et al.*, 2015] showed that *LTL*-OMQs with these operators are FO-rewritable. Although such automata are not applicable now, we establish the same complexity by characterising the structure of models. In the constructions below, it will be convenient to regard $\boxminus_\varrho$ as an abbreviation for $\neg\diamondminus_\varrho\neg$ with Boolean negation $\neg$ and only consider w.l.o.g. OMQs (Π, A) with A occurring in Π.

Theorem 4. *MTL-OMQs with temporal operators of the form $\diamondminus_{\langle r,\infty)}$ and $\boxminus_{\langle r,\infty)}$ only are* FO($<$)*-rewritable.*

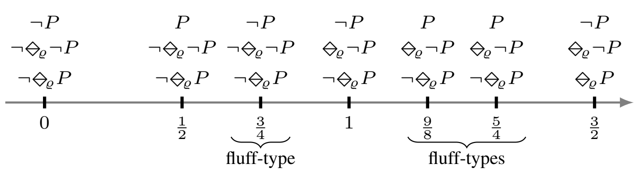
Proof sketch. Let $\boldsymbol{q} = (\Pi, A)$ be an *MTL*-OMQ as specified above. A *simple literal*, σ, for Π takes the form P or $\neg P$, where P is an atom in Π; a *temporal literal*, τ, for Π is of the form $\diamondminus_\varrho\sigma$ or $\neg\diamondminus_\varrho\sigma$ provided that $\diamondminus_\varrho P$ or $\boxminus_\varrho P$ occurs in Π and P is the atom in σ. Let Σ_Π and Ξ_Π be the sets of simple and temporal literals for Π, respectively. A *type* for Π is any maximal set $\boldsymbol{t} \subseteq \Sigma_\Pi \cup \Xi_\Pi$ consistent with Π. The number of different types is $N_\Pi = 2^{O(|\Pi|)}$.

Given a model $\mathcal{I}$ of Π and some $\mathcal{D}$ with $s \in \mathsf{ts}(\mathcal{D})$, denote by $\boldsymbol{t}(s)$ the type of s in $\mathcal{I}$. As the ranges in Π are of the form $\langle r, \infty)$, the model $\mathcal{I}$ has the following *monotonicity property*:

– $\diamondminus_\varrho\sigma \in \boldsymbol{t}(s)$ implies $\diamondminus_\varrho\sigma \in \boldsymbol{t}(s')$ for all $s' > s$ in $\mathcal{I}$;

– $\neg\diamondminus_\varrho\sigma \in \boldsymbol{t}(s)$ implies $\neg\diamondminus_\varrho\sigma \in \boldsymbol{t}(s')$ for all $s' < s$ in $\mathcal{I}$.

We call $\boldsymbol{t}(s)$ in $\mathcal{I}$ an *osteo-type* if there is $\lambda \in \boldsymbol{t}(s)$ such that $\lambda \notin \boldsymbol{t}(s')$, for all $s' < s$. Thus, if $\diamondminus_\varrho\sigma \in \boldsymbol{t}(s')$ in $\mathcal{I}$, there is an osteo-type $\boldsymbol{t}(s) \ni \sigma$ with $\mathsf{in}_\varrho(s', s)$. All osteo-types in $\mathcal{I}$ are pairwise distinct, so the number of them does not exceed N_Π. Non-osteo-types are called *fluff-types*. By monotonicity, any fluff-type $\boldsymbol{t}(s')$ has the same temporal literals as its nearest osteo-type $\boldsymbol{t}(s)$, for $s < s'$. For example, in the model of the program $\Pi = \{\boxminus_\varrho P \wedge \diamondminus_\varrho P \wedge P \to \bot\}$, $\varrho = [1, \infty)$, shown below, there are three fluff-types: $\boldsymbol{t}(3/4)$, $\boldsymbol{t}(9/8)$, and $\boldsymbol{t}(5/4)$.



We now define an FO-sentence Φ_Π such that any given data instance $\mathcal{D}$ is consistent with Π iff Φ_Π holds in the FO-structure $\mathcal{D}$. Let $\mathfrak{D}_\Pi$ be the set of sequences $\bar{\boldsymbol{t}} = (\boldsymbol{t}_1, \dots, \boldsymbol{t}_n)$, $1 \le n \le N_\Pi$, of distinct types for Π that satisfy the monotonicity property and such that $\diamondminus_\varrho\sigma \in \boldsymbol{t}_i$ implies $\sigma \in \boldsymbol{t}_j$ for some $j \le i$; for minimal such j, we write $\mathit{wit}(\boldsymbol{t}_i, \boldsymbol{t}_j, \varrho)$. We write $\overline{\mathit{wit}}(\boldsymbol{t}_i, \boldsymbol{t}_j, \varrho)$ if $j \le i$, $\neg\diamondminus_\varrho\sigma \in \boldsymbol{t}(s_i)$ and $\sigma \in \boldsymbol{t}(s_j)$, for some $\diamondminus_\varrho\sigma$. Denote by $\mathfrak{F}^i_{\bar{\boldsymbol{t}}}$ the set of types $\boldsymbol{t}$ for Π sharing the same temporal literals with $\boldsymbol{t}_i$ and such that, for every $\sigma \in \boldsymbol{t}$, there is $\boldsymbol{t}_j \ni \sigma$ with $j \le i$. Finally, for any type $\boldsymbol{t}$, let $\delta_{\boldsymbol{t}}(x) = \bigwedge_{\neg P \in \boldsymbol{t}} \neg P(x)$ (which is true at t in $\mathcal{D}$ iff, for every P in Π, whenever $P(t) \in \mathcal{D}$ then $P(t) \in \boldsymbol{t}$). Now, we set

$$\Phi_\Pi = \bigvee_{\bar{\boldsymbol{t}} \in \mathfrak{D}_\Pi} \exists x_1, \dots, x_n \big[(x_1 = \min) \wedge \bigwedge_{1 \le i \le n} \delta_{\boldsymbol{t}_i}(x_i) \wedge \bigwedge_{\mathit{wit}(\boldsymbol{t}_i, \boldsymbol{t}_j, \varrho)} \mathsf{in}_\varrho(x_i, x_j) \wedge \bigwedge_{\overline{\mathit{wit}}(\boldsymbol{t}_i, \boldsymbol{t}_j, \varrho)} \neg\mathsf{in}_\varrho(x_i, x_j) \wedge \forall y \bigwedge_{1 \le i \le n} \big((x_i \prec y) \to \bigvee_{\boldsymbol{t} \in \mathfrak{F}^i_{\bar{\boldsymbol{t}}}} (\delta_{\boldsymbol{t}}(y) \wedge \bigwedge_{\overline{\mathit{wit}}(\boldsymbol{t}_i, \boldsymbol{t}_j, \varrho)} \neg\mathsf{in}_\varrho(y, x_j))\big)\big],$$

where $x_i \prec y$ says that x_i is the nearest predecessor of y, which is different from $x_1, \dots, x_n$. An FO($<$)-rewriting of $\boldsymbol{q}$ is the FO formula $\neg\Phi_{\neg A}(x)$, where $\Phi_{\neg A}(x)$ is obtained from Φ_Π by replacing $\delta_{\boldsymbol{t}}(z)$ with $\delta_{\boldsymbol{t}}(z, x)$, which is $\delta_{\boldsymbol{t}}(z)$ if $\neg A \in \boldsymbol{t}$ and $\delta_{\boldsymbol{t}}(z) \wedge (x \neq z)$ otherwise. Clearly, $\Phi_{\neg A}(x)$ holds in $\mathcal{D}$ iff there is a model of Π and $\mathcal{D}$ satisfying $\neg A$ in x. ❑

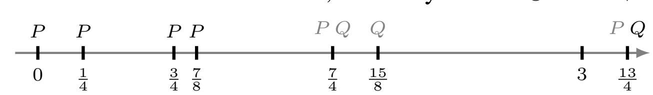
We also mention in passing one more FO-rewritability result (which does not fit our classification). Call an *MTL*-program *range-uniform* if all of its temporal operators have the same constraining range. Let $\rangle$ be one of) or].

Theorem 5. *Range-uniform* coreMTL$^{\diamondminus}$*-OMQs with ranges of the form $\diamondminus_{\langle 0,r\rangle}$ are* FO($<, +$)*-rewritable.*

The proof uses automata with metric constraints that can be viewed as a primitive version of standard timed automata for *MTL* [Alur and Dill, 1994] as they only have one clock c, the clock reset $c := 0$ happens at every transition, and clock constraints are of the form $c \in \varrho$.

5 OMQs with Punctual Ranges $[r, r]$

Operators of the form $\diamondminus_{[r,r]}$ resemble the *LTL* previous time operator $\ominus$. To illustrate an essential difference, consider the program $\Pi = \{\diamondminus_{[1,1]} P \to Q,\ \diamondminus_{[1.5,1.5]} P \wedge Q \to P\}$ and the data instance $\mathcal{D}$ below. In *LTL*, we always derive $\ominus P$ at $n+1$



if P holds at n. In our example, P at $3/4$ implies Q at $7/4$, which together with P at $1/4$ imply P at $7/4$, and eventually the latter P with Q at $13/4$ implies P at $13/4$; independently, P at $7/8$ implies Q at $15/8$.

Theorem 6. *MTL-OMQs with temporal operators of the form $\diamondminus_{[r,r]}$ and $\boxminus_{[r,r]}$ only are* FO(RPR)*-rewritable; answering such OMQs is* NC^1*-complete for data complexity.*

Proof sketch. NC^1-hardness is proved by reduction of *horn-MTL*-OMQs with rules of the form $\ominus P \wedge P' \to Q$, answering which is NC^1-complete [Artale *et al.*, 2015].

To show FO(RPR)-rewritability of a given OMQ $\boldsymbol{q} = (\Pi, A)$, we assume w.l.o.g. that Π does not contain ranges $[0, 0]$. Let R_Π be the set of numbers occurring as endpoints of ranges in Π. We set $\mathbf{1} = \gcd(R_\Pi)$, $\boldsymbol{n} = \mathbf{1} \cdot n$, for $n \in \mathbb{N}$, $\boldsymbol{m} = \max(R_\Pi)$. Thus, in our example above, $\mathbf{1} = 1/2$, $\mathbf{2} = 1$, $\mathbf{3} = 3/2$. We define $cl(\Pi)$ to be the set of simple and temporal literals with atoms from Π and operators $\diamondminus_{\boldsymbol{i}}$ such that $\boldsymbol{i} \in \{\mathbf{1}, \dots, \boldsymbol{n}\}$ and $\diamondminus_{\boldsymbol{n}}$ occurs in Π. By a *type* $\boldsymbol{s}$ for Π we now mean any maximal subset of $cl(\Pi)$ consistent with Π. For types $\boldsymbol{s}$, $\boldsymbol{s}'$ and $\boldsymbol{i} \in \{\mathbf{1}, \dots, \boldsymbol{m}\}$, we write $\boldsymbol{s} \to_{\boldsymbol{i}} \boldsymbol{s}'$ if

– $\sigma \in \boldsymbol{s}$ iff $\diamondminus_{\boldsymbol{i}}\sigma \in \boldsymbol{s}'$, for any $\diamondminus_{\boldsymbol{i}}\sigma \in cl(\Pi)$;

– $\diamondminus_{\boldsymbol{j}}\sigma \in \boldsymbol{s}$ iff $\diamondminus_{\boldsymbol{j}+\boldsymbol{i}}\sigma \in \boldsymbol{s}'$, for $\diamondminus_{\boldsymbol{j}+\boldsymbol{i}}\sigma \in cl(\Pi)$, $\boldsymbol{j} \geq \boldsymbol{1}$.

We say that $(\boldsymbol{s}_0, t_0), \dots, (\boldsymbol{s}_n, t_n)$ is a *run from* t_0 *to* t_n on a data instance $\mathcal{D}$ of the form (2) if $t_i \in \mathsf{ts}(\mathcal{D})$, for $i \leq n$, and

– $\{P \in \Sigma_\Pi \mid t_0 \in P^{\mathcal{D}}\} \subseteq \boldsymbol{s}_0$;

– $\neg\diamondminus_{\boldsymbol{j}}\sigma \in \boldsymbol{s}_0$ for all $\diamondminus_{\boldsymbol{j}}\sigma \in cl(\Pi)$;

– $\bar{t}_{i+1} - \bar{t}_i \in \{\boldsymbol{1}, \dots, \boldsymbol{m}\}$ and if $t_{i+1} > t > t_i$ then $\bar{t} - \bar{t}_i \notin \{\boldsymbol{1}, \dots, \boldsymbol{m}\}$, for any $t \in \mathsf{ts}(\mathcal{D})$;

– $\boldsymbol{s}_i \to_{(\bar{t}_{i+1} - \bar{t}_i)} \boldsymbol{s}_{i+1}$ and $\{P \in \Sigma_\Pi \mid t_{i+1} \in P^{\mathcal{D}}\} \subseteq \boldsymbol{s}_{i+1}$.

Call $t \in \mathsf{ts}(\mathcal{D})$ *initial* if $\bar{t} - \bar{t}' \notin \{\boldsymbol{1}, \dots, \boldsymbol{m}\}$, for all $t' \in \mathsf{ts}(\mathcal{D})$. The next lemma follows directly from the given definitions:

Lemma 7. (i) $(\Pi, \mathcal{D})$ *is consistent iff, for every* $t \in \mathsf{ts}(\mathcal{D})$, *there exists a run on* $\mathcal{D}$ *from some initial* $t' \leq t$ *to* t; (ii) *A timestamp* $t \in \mathsf{ts}(\mathcal{D})$ *is not a certain answer to* $\boldsymbol{q}$ *over* $\mathcal{D}$ *iff* $(\Pi, \mathcal{D})$ *is consistent and there is a run* $(\boldsymbol{s}_0, t_0), \dots, (\boldsymbol{s}_n, t_n)$ *from initial* t_0 *to* $t = t_n$ *on* $\mathcal{D}$ *and* $\neg A \in \boldsymbol{s}_n$.

We first show how to express the existence of a run from x to y specified in (ii) by an FO(RPR)-formula $\mathsf{run}_{\boldsymbol{q}}(x, y)$ over $\mathcal{D}$. First, as divisibility of binary integers by a given number is recognisable by a finite automaton, we can define an FO(RPR)-formula $\mathsf{div}_{\boldsymbol{1}}(u, v)$ that is true iff $\bar{u} - \bar{v} = n\boldsymbol{1}$, for some $n \in \mathbb{N}$. We also have an FO-formula $\mathsf{last}_{\boldsymbol{i}}(u)$ saying that $\boldsymbol{i}$ is minimal among $\{\boldsymbol{1}, \dots, \boldsymbol{m}\}$ with $\bar{u} - \boldsymbol{i} = \bar{v}$, for some $v \in \mathsf{ts}(\mathcal{D})$. Let $Q = \{\boldsymbol{s}_1, \dots, \boldsymbol{s}_n\}$ be the set of all types for Π, and let $Q_0 \subseteq Q$ comprise $\boldsymbol{s}$ with $\neg\diamondminus_{\boldsymbol{j}}\sigma \in \boldsymbol{s}$, for all $\diamondminus_{\boldsymbol{j}}\sigma \in cl(\Pi)$. We define $\mathsf{run}_{\boldsymbol{q}}(x, y)$ as the FO(RPR)-formula

$$\left[\begin{array}{l} R_{\boldsymbol{s}_1}(x, z) \equiv \vartheta_{\boldsymbol{s}_1} \\ \dots \\ R_{\boldsymbol{s}_n}(x, z) \equiv \vartheta_{\boldsymbol{s}_n} \end{array}\right] \bigvee_{\neg A \in \boldsymbol{s} \in Q} R_{\boldsymbol{s}}(x, y) \ \wedge \ \mathsf{div}_{\boldsymbol{1}}(y, x),$$

where $R_{\boldsymbol{s}}(x, z)$, for $\boldsymbol{s} \in Q$, is a *relation variable* and the formula $\vartheta_{\boldsymbol{s}}(x, z, R_{\boldsymbol{s}_1}(x, z-1), \dots, R_{\boldsymbol{s}_n}(x, z-1))$ is a disjunction of the three formulas below if $\boldsymbol{s} \in Q_0$ and a disjunction of the last two of them if $\boldsymbol{s} \notin Q_0$:

$$(x = z) \wedge \delta_{\boldsymbol{s}}(z),$$

$$\neg\mathsf{div}_{\boldsymbol{1}}(z, x) \wedge \exists z' (\mathsf{dist}_{<\boldsymbol{m}}(z, z') \wedge \mathsf{div}_{\boldsymbol{1}}(z', x)) \wedge R_{\boldsymbol{s}}(x, z-1),$$

$$\mathsf{div}_{\boldsymbol{1}}(z, x) \wedge \bigvee_{\substack{\boldsymbol{i} \in \{\boldsymbol{1}, \dots, \boldsymbol{m}\} \\ \boldsymbol{s}' \to_{\boldsymbol{i}} \boldsymbol{s}}} (\delta_{\boldsymbol{s}}(z) \wedge \mathsf{last}_{\boldsymbol{i}}(z) \wedge R_{\boldsymbol{s}'}(x, z-1)),$$

where $z - 1$ is the immediate predecessor of z in $\mathsf{ts}(\mathcal{D})$.

To illustrate, in the context of the example above, the formulas $R_{\boldsymbol{s}} \equiv \vartheta_{\boldsymbol{s}}$ say that $R_{\boldsymbol{s}}(1/4, 1/4)$ holds for the types

$\{\neg\diamondminus_{\boldsymbol{1}}P, \neg\diamondminus_{\boldsymbol{2}}P, \neg\diamondminus_{\boldsymbol{3}}P, P, Q\}$, $\{\neg\diamondminus_{\boldsymbol{1}}P, \neg\diamondminus_{\boldsymbol{2}}P, \neg\diamondminus_{\boldsymbol{3}}P, P, \neg Q\}$.

Then $R_{\boldsymbol{s}}(1/4, 3/4)$ holds for

$\{\diamondminus_{\boldsymbol{1}}P, \neg\diamondminus_{\boldsymbol{2}}P, \neg\diamondminus_{\boldsymbol{3}}P, P, Q\}$, $\{\diamondminus_{\boldsymbol{1}}P, \neg\diamondminus_{\boldsymbol{2}}P, \neg\diamondminus_{\boldsymbol{3}}P, P, \neg Q\}$,

$R_{\boldsymbol{s}}(1/4, 7/8)$ for the same $\boldsymbol{s}$ as $R_{\boldsymbol{s}}(1/4, 3/4)$, $R_{\boldsymbol{s}}(1/4, 7/4)$ for $\boldsymbol{s} = \{\neg\diamondminus_{\boldsymbol{1}}P, \diamondminus_{\boldsymbol{2}}P, \diamondminus_{\boldsymbol{3}}P, P, Q\}$, and so on.

Thus, we obtain the following FO(RPR)-rewriting of $\boldsymbol{q}$

$$\neg\Phi_\Pi \vee \neg\exists y \left(\mathsf{run}_{\boldsymbol{q}}(y, x) \wedge \bigwedge_{\boldsymbol{i} \in \{\boldsymbol{1}, \dots, \boldsymbol{m}\}} \neg\mathsf{last}_{\boldsymbol{i}}(y)\right),$$

where Φ_Π checks the consistency condition of Lemma 7 (i) and can be constructed similarly to $\mathsf{run}_{\boldsymbol{q}}$. ❑

6 OMQs with Non-Punctual Ranges

Unlike the proof of Theorem 6, where the derived facts at t were determined by the data $\mathcal{D}$ at t and the derived facts at the nearest $t' \in \mathsf{ts}(\mathcal{D})$ with $\bar{t}' = \bar{t} - \boldsymbol{i}$, for non-punctual ranges the derived facts at t depend on an unbounded number of timestamps $t' < t$. In the proof of Theorem 8 below, we show that to construct derivations in this case, we can actually keep track of a fixed number (depending only on the given OMQ) of moments $t'_P < t$ where each P was derived.

Theorem 8. (i) *MTL-OMQs whose operators* $\diamondminus_\varrho$ *and* $\boxminus_\varrho$ *have non-punctual* ϱ *are* FO(TC)*-rewritable; answering them is in* NL *and* NC^1*-hard;* (ii) *hornMTL-OMQs of this kind are* FO(DTC)*-rewritable; answering them is in* L *and* NC^1*-hard.*

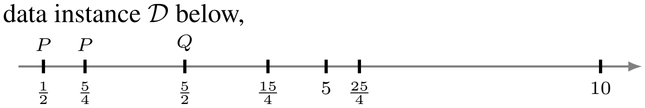
Proof sketch. In both cases, NC^1-hardness can be established as in the proof of Theorem 6 by encoding $\ominus$ with $\diamondminus_{(0,1]}$.

(i) Let $\boldsymbol{q} = (\Pi, A)$ be the given OMQ. For $\varrho = \langle r, q\rangle$ with $q \neq \infty$, let $\varrho^- = \langle 0, q - r\rangle$ and $\varrho^+ = \langle 0, q\rangle$; if $q = \infty$, ϱ^- and ϱ^+ are undefined. Let Σ_Π be the set of all σ with $\diamondminus_\varrho\sigma$ in Π, for some ϱ. For $\sigma \in \Sigma_\Pi$, let ϱ^-_σ (ϱ^+_σ) be the intersection (union) of the defined ϱ^- (ϱ^+) with $\diamondminus_\varrho\sigma$ in Π; if there are no such $\diamondminus_\varrho\sigma$, ϱ^-_σ and ϱ^+_σ are undefined. To illustrate, consider the *hornMTL*-program Π with the rules

$$\diamondminus_{(2,4]}P \to P, \quad \diamondminus_{[1,2)}P \to P, \quad \diamondminus_{[3,\infty]}Q \to Q.$$

Then $\varrho^-_P = (0, 1)$, $\varrho^+_P = [0, 4]$, and ϱ^-_Q, ϱ^+_Q are undefined.

For a data instance $\mathcal{D}$, a *trace of length* ℓ for $t \in \mathsf{ts}(\mathcal{D})$ is a sequence of intervals $[u_0, s_0], \dots, [u_\ell, s_\ell]$ where either $[u_i, s_i] = [*, *]$ (meaning that this interval is undefined) or $u_i, s_i \in \mathsf{ts}(\mathcal{D})$, $u_0 = s_0$, and $u_1 \leq s_1 < u_2 \leq s_2 < \dots < u_\ell \leq s_\ell \leq t$, assuming that $* < u$, for any u. Thus, for the data instance $\mathcal{D}$ below,



$([\frac{1}{2}, \frac{1}{2}], [*, *], [*, *], [\frac{1}{2}, \frac{5}{4}], [\frac{5}{2}, \frac{5}{2}])$ is a trace for $t = 5/2$. Intuitively, such a trace stores the most recent ℓ intervals preceding t where a simple literal holds at some point, with $[u_0, s_0]$ storing the very first point where the literal holds. A tuple $(\boldsymbol{t}, (\boldsymbol{tr}_\sigma)_{\sigma \in \Sigma_\Pi}, t)$ is an *extended type* for $t \in \mathsf{ts}(\mathcal{D})$ if

– $\boldsymbol{t}$ is a *type* for Π (as in the proof of Theorem 4);

– $\boldsymbol{tr}_\sigma$ is a trace for t of length $\ell_\sigma = \lceil |\varrho^+_\sigma| / |\varrho^-_\sigma| \rceil$, where $|\varrho^+_\sigma|$ and $|\varrho^-_\sigma|$ denote the end-points of these intervals; if one of the intervals is undefined, $\ell_\sigma = 0$;

– $\diamondminus_\varrho\sigma \in \boldsymbol{t}$ iff $\mathsf{int}_\varrho(t, u_i, s_i)$, for some $[u_i, s_i]$ in $\boldsymbol{tr}_\sigma$,

where $\mathsf{int}_\varrho(t, u, s)$ is true iff $\{\bar{t} - k \mid k \in \varrho\} \cap [\bar{u}, \bar{s}] \neq \emptyset$ and $u, s \neq *$. In our example, $\ell_P = 4$, $\ell_Q = 0$, and the following triples $(\boldsymbol{t}_i, (\boldsymbol{tr}^i_\sigma)_{\sigma \in \Sigma_\Pi}, t_i)$ are extended types for t_i:

$\boldsymbol{t}_0 = \{P, \neg Q, \neg\diamondminus_{(2,4]}P, \neg\diamondminus_{[1,2)}P, \neg\diamondminus_{[3,\infty)}Q\}$, $t_0 = \frac{1}{2}$,

$\boldsymbol{tr}^0_P = ([\frac{1}{2}, \frac{1}{2}], [*, *], [*, *], [*, *], [\frac{1}{2}, \frac{1}{2}])$, $\boldsymbol{tr}^0_Q = ([*, *])$;

$\boldsymbol{t}_1 = \{P, \neg Q, \neg\diamondminus_{(2,4]}P, \diamondminus_{[1,2)}P, \neg\diamondminus_{[3,\infty)}Q\}$, $t_1 = \frac{5}{4}$,

$\boldsymbol{tr}^1_P = ([\frac{1}{2}, \frac{1}{2}], [*, *], [*, *], [*, *], [\frac{1}{2}, \frac{5}{4}])$, $\boldsymbol{tr}^1_Q = ([*, *])$;

$\boldsymbol{t}_2 = \{P, Q, \diamondminus_{(2,4]}P, \neg\diamondminus_{[1,2)}P, \neg\diamondminus_{[3,\infty)}Q\}$, $t_2 = \frac{5}{2}$,

$\boldsymbol{tr}^2_P = ([\frac{1}{2}, \frac{1}{2}], [*, *], [*, *], [\frac{1}{2}, \frac{5}{4}], [\frac{5}{2}, \frac{5}{2}])$, $\boldsymbol{tr}^2_Q = ([\frac{5}{2}, \frac{5}{2}])$; ...

$\boldsymbol{t}_5 = \{P, Q, \diamondminus_{(2,4]}P, \diamondminus_{[1,2)}P, \neg\diamondminus_{[3,\infty)}Q\},\ t_5 = \frac{25}{4}$,
$\boldsymbol{tr}^5_P = ([\frac{1}{2}, \frac{1}{2}], [\frac{5}{2}, \frac{5}{2}], [\frac{15}{4}, \frac{15}{4}], [5, 5], [\frac{25}{4}, \frac{25}{4}]), \boldsymbol{tr}^5_Q = ([\frac{5}{2}, \frac{5}{2}])$.

Intuitively, an extended type records the simple and temporal literals that hold at t (the type $\boldsymbol{t}$) and also some history of the validity of σ (the traces) justifying the presence of $\diamondminus_\varrho\sigma$ in $\boldsymbol{t}$. As follows from Lemma 9 below, to make correct derivations, this history should keep $\ell_\sigma + 1$ intervals. Note that this bound does not apply if punctual intervals are present in Π, which explains the increase of complexity in Theorem 3.

Lemma 9. *Let $t_0 < \cdots < t_m$ be all the timestamps in $\mathcal{D}$. Then Π and $\mathcal{D}$ are consistent iff there exists a sequence $(\boldsymbol{t}_i, (\boldsymbol{tr}^i_\sigma)_{\sigma\in\Sigma_\Pi}, t_i)$ of extended types for t_i, $0 \le i \le m$, satisfying the following conditions for $\sigma \in \Sigma_\Pi$:*

- *$\{P \in \Sigma_\Pi \mid t_i \in P^{\mathcal{D}}\} \subseteq \boldsymbol{t}_i$;*
- *if $\sigma \notin \boldsymbol{t}_0$, all $[u_j, s_j]$ in $\boldsymbol{tr}^0_\sigma$ are $[*, *]$; if $\sigma \in \boldsymbol{t}_0$, then $[u_0, s_0] = [u_{\ell_\sigma}, s_{\ell_\sigma}] = [t_0, t_0]$ and $[u_j, s_j] = [*, *]$ for $0 < j < \ell_\sigma$;*
- *if $\sigma \notin \boldsymbol{t}_i$ and $i > 0$, then $\boldsymbol{tr}^i_\sigma = \boldsymbol{tr}^{i-1}_\sigma$; if $\sigma \in \boldsymbol{t}_i$, $\boldsymbol{tr}^{i-1}_\sigma = ([u_0, s_0], \dots, [u_{\ell_\sigma}, s_{\ell_\sigma}])$ and $[u, s] = [u_0, s_0]$ when $u_0 \ne *$ and $[u, s] = [t_i, t_i]$ otherwise, then $\boldsymbol{tr}^i_\sigma = ([u, s], [u_1, s_1], \dots, [u_{\ell_\sigma}, t_i])$ if $\bar{t}_i - \bar{s}_{\ell_\sigma} \in \varrho^-_\sigma$, else $\boldsymbol{tr}^i_\sigma = ([u, s], [u_2, s_2], \dots, [u_{\ell_\sigma}, s_{\ell_\sigma}], [t_i, t_i])$.*

We use the characterisation of Lemma 9 to construct an FO(TC)-sentence Φ_Π that is true in $\mathcal{D}$ iff Π and $\mathcal{D}$ are consistent, for any data instance $\mathcal{D}$. Φ_Π contains tuples of variables $\boldsymbol{x} = \boldsymbol{x}_{\sigma_1}, \dots, \boldsymbol{x}_{\sigma_n}$, for $\{\sigma_1, \dots, \sigma_n\} = \Sigma_\Pi$, where $\boldsymbol{x}_\sigma = \boldsymbol{x}_{0\sigma}, \dots, \boldsymbol{x}_{\ell_\sigma\sigma}$ and $\boldsymbol{x}_{i\sigma} = u_{i\sigma}, s_{i\sigma}$ for intervals in traces $\boldsymbol{tr}_\sigma$; $\boldsymbol{x}'$ is the same as $\boldsymbol{x}$ but with primed variables:

$$\Phi_\Pi \ = \ \exists \boldsymbol{x}, \boldsymbol{x}' \,\big(\bigvee_{\boldsymbol{t} \text{ type for } \boldsymbol{q}} \mathsf{first}_{\boldsymbol{t}}(\boldsymbol{x}) \wedge [\mathrm{TC}_{t,\boldsymbol{x},t',\boldsymbol{x}'}\,\xi(t, \boldsymbol{x}, t', \boldsymbol{x}')](\min, \boldsymbol{x}, \max, \boldsymbol{x}')\big).$$

Here, $\mathsf{first}_{\boldsymbol{t}}(\boldsymbol{x})$ is an FO-formula saying that $\boldsymbol{t}$ holds in the first timestamp $(\min)$ of $\mathcal{D}$ and $\boldsymbol{x}$ represents $\boldsymbol{tr}^0_\sigma$ for all σ by encoding $[*, *]$ as the empty interval $[\max, \min]$. The formula $\xi(t, \boldsymbol{x}, t', \boldsymbol{x}')$ under the transitive closure TC says that there is an extended type for t with the trace given by $\boldsymbol{x}$, that t' is the immediate successor of t in $\mathsf{ts}(\mathcal{D})$, and there is an extended type for t' whose trace is given by $\boldsymbol{x}'$. We define it as

$$\xi(t, \boldsymbol{x}, t', \boldsymbol{x}') \ = \ \mathsf{suc}(t', t) \wedge \bigvee_{\boldsymbol{t}' \text{ type for } \boldsymbol{q}} \xi_{\boldsymbol{t}'}(t, \boldsymbol{x}, t', \boldsymbol{x}'),$$

with $\xi_{\boldsymbol{t}'}(t, \boldsymbol{x}, t', \boldsymbol{x}')$ saying that if $(\boldsymbol{t}, (\boldsymbol{tr}_\sigma)_{\sigma\in\Sigma_\Pi}, t)$ is an extended type for t with $(\boldsymbol{tr}_\sigma)_{\sigma\in\Sigma_\Pi}$ given by $\boldsymbol{x}$, then $(\boldsymbol{t}', (\boldsymbol{tr}'_\sigma)_{\sigma\in\Sigma_\Pi}, t')$ can be the next extended type with $(\boldsymbol{tr}'_\sigma)_{\sigma\in\Sigma_\Pi}$ given by $\boldsymbol{x}'$:

$$\begin{aligned}
\xi_{\boldsymbol{t}'}(t, \boldsymbol{x}, t', \boldsymbol{x}') \ = \ & \mathsf{ext}_{\boldsymbol{t}'}(t', \boldsymbol{x}') \wedge \bigwedge_{\sigma \notin \boldsymbol{t}'} (\boldsymbol{x}_\sigma = \boldsymbol{x}'_\sigma) \wedge \\
& \bigwedge_{\sigma \in \boldsymbol{t}'} \big[\big((u_{0\sigma} > s_{0\sigma}) \to (u'_{0\sigma} = t') \wedge (s'_{0\sigma} = t')\big) \wedge \\
& \quad ((u_{0\sigma} \le s_{0\sigma}) \to (\boldsymbol{x}_{0\sigma} = \boldsymbol{x}'_{0\sigma})) \wedge \\
& \quad \big(\mathsf{in}_{\varrho^-_\sigma}(t', s_{\ell_\sigma}) \to \bigwedge_{1 \le i < \ell_\sigma - 1} (\boldsymbol{x}_{i\sigma} = \boldsymbol{x}'_{i\sigma}) \wedge \\
& \qquad (u_{\ell_\sigma\sigma} = u'_{\ell_\sigma\sigma}) \wedge (s'_{\ell_\sigma\sigma} = t')\big) \wedge \\
& \quad \big(\neg\mathsf{in}_{\varrho^-_\sigma}(t', s_{\ell_\sigma}) \to \bigwedge_{1 < i \le \ell_\sigma - 1} (\boldsymbol{x}_{i\sigma} = \boldsymbol{x}'_{i-1\sigma}) \wedge \\
& \qquad (u'_{\ell_\sigma\sigma} = t') \wedge (s'_{\ell_\sigma\sigma} = t')\big)\big].
\end{aligned}$$

The formula $\mathsf{ext}_{\boldsymbol{t}}(t, \boldsymbol{x})$ defines an extended type for t in $\mathcal{D}$:

$$\begin{aligned}
\mathsf{ext}_{\boldsymbol{t}}(t, \boldsymbol{x}) \ = \ & \delta_{\boldsymbol{t}}(t) \wedge \bigwedge_{\diamondminus_\varrho\sigma \in \boldsymbol{t}} \big(\bigvee_{0 \le i \le \ell_\sigma} \mathsf{int}_\varrho(t, u_{i\sigma}, s_{i\sigma})\big) \wedge \\
& \bigwedge_{\diamondminus_\varrho\sigma \notin \boldsymbol{t}} \big(\bigwedge_{0 \le i \le \ell_\sigma} \neg\mathsf{int}_\varrho(t, u_{i\sigma}, s_{i\sigma})\big).
\end{aligned}$$

Finally, $\mathsf{first}_{\boldsymbol{t}}(\boldsymbol{x})$ is $\bot$ if there is $\diamondminus_\varrho\sigma \in \boldsymbol{t}$ and otherwise it is

$$\begin{aligned}
\delta_{\boldsymbol{t}}(\min) \wedge & \bigwedge_{\sigma \notin \boldsymbol{t}} \bigwedge_{0 \le i \le \ell_\sigma} ((u_{i\sigma} = \max) \wedge (s_{i\sigma} = \min)) \wedge \\
& \bigwedge_{\sigma \in \boldsymbol{t}} \big(\bigwedge_{0 < i < \ell_\sigma} (u_{i\sigma} = \max) \wedge (s_{i\sigma} = \min)) \wedge \\
& \quad (u_{0\sigma} = \min) \wedge (s_{0\sigma} = \min) \wedge \\
& \quad (u_{\ell_\sigma\sigma} = \min) \wedge (s_{\ell_\sigma\sigma} = \min)\big),
\end{aligned}$$

saying that the intervals in the initial extended type are set correctly. That Φ_Π is as required follows from Lemma 9. One can now modify Φ_Π to obtain an FO(TC)-rewriting of $\boldsymbol{q}$. ❑

7 Conclusion

In this paper, we made a first step towards understanding the data complexity of answering queries mediated by ontologies with *MTL* operators and their rewritability into standard database query languages. By imposing natural restrictions on the ranges ϱ constraining the operators $\diamondminus_\varrho$ and $\boxminus_\varrho$, and by distinguishing between arbitrary, Horn and core ontologies, we identified classes of *MTL*-OMQs that are rewritable to FO($<$), FO($<, +$), FO(RPR), FO(DTC), FO(TC), and datalog(FO). Unrestricted *MTL*-OMQs were shown to be CONP-hard. The rewritability results look encouraging, though much remains to be done to make our rewritings practical, especially in the presence of more expressive atemporal (description logic or datalog) ontologies.

We can extend our language with constrained operators since $\mathcal{S}_\varrho$. In this case, *hornMTL* remains P-complete (but *coreMTL* becomes P-hard) and Theorem 8 holds, too. We believe that our *hornMTL* can also be extended with $\boxminus_\varrho$ in the rule heads (cf. [Brzoska, 1998]): Theorems 3 (ii) and 8 (i) also hold in this case, but so far we have not managed to prove Theorem 8 (ii) for such rules. Extending *MTL* with future-time operators is also interesting, in which case Theorems 3 and 4 remain to hold. Finally, we are looking into *MTL*-OMQs under the continuous (state-based) semantics, where the techniques developed above do not apply directly.

Acknowledgements

This work was supported by EPSRC UK grant EP/S032282, NCN Poland grant 2016/23/N/HS1/ 02168, and the Foundation for Polish Science (FNP). We would like to thank Stanislav Kikot for his help with the proof of Theorem 5.

References

[Alur and Dill, 1994] R. Alur and D. Dill. A theory of timed automata. *Theor. Com. Sci.*, 126(2):183–235, 1994.

[Alur and Henzinger, 1993] R. Alur and T. Henzinger. Real-time logics: Complexity and expressiveness. *Inf. Comput.*, 104(1):35–77, 1993.

[Arora and Barak, 2009] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[Artale *et al.*, 2015] A. Artale, R. Kontchakov, A. Kovtunova, V. Ryzhikov, F. Wolter, and M. Zakharyaschev. First-order rewritability of temporal ontology-mediated queries. In *Proc. IJCAI*, pages 2706–2712, 2015.

[Artale *et al.*, 2017] A. Artale, R. Kontchakov, A. Kovtunova, V. Ryzhikov, F. Wolter, and M. Zakharyaschev. Ontology-mediated query answering over temporal data: A survey. In *TIME*, vol. 90 of *LIPIcs*, 1:1–1:37, 2017.

[Baader *et al.*, 2013] F. Baader, S. Borgwardt, and M. Lippmann. Temporalizing ontology-based data access. In *Proc. CADE-24*, vol. 7898 of *LNCS*, 330–344, 2013.

[Baader *et al.*, 2017] F. Baader, S. Borgwardt, P. Koopmann, A. Ozaki, and V. Thost. Metric temporal description logics with interval-rigid names. In *Proc. FroCoS*, 60–76, 2017.

[Baudinet *et al.*, 1993] M. Baudinet, J. Chomicki, and P. Wolper. Temporal deductive databases. In *Temporal Databases*, pages 294–320. 1993.

[Beck *et al.*, 2018] H. Beck, M. Dao-Tran, and T. Eiter. LARS: A logic-based framework for analytic reasoning over streams. *Artif. Intell.*, 261:16–70, 2018.

[Bienvenu and Ortiz, 2015] M. Bienvenu and M. Ortiz. Ontology-mediated query answering with data-tractable description logics. *RW*, vol. 9203 of *LNCS*, 218–307, 2015.

[Bienvenu *et al.*, 2014] M. Bienvenu, B. ten Cate, C. Lutz, and F. Wolter. Ontology-based data access: A study through disjunctive datalog, CSP, and MMSNP. *ACM Trans. Database Syst.*, 39(4):33:1–33:44, 2014.

[Bouyer *et al.*, 2018] P. Bouyer, U. Fahrenberg, K. Larsen, N. Markey, J. Ouaknine, and J. Worrell. Model checking real-time systems. In *Handbook of Model Checking*, pages 1001–1046. Springer, 2018.

[Brandt *et al.*, 2018] S. Brandt, E. Kalayci, V. Ryzhikov, G. Xiao, M. Zakharyaschev. Querying log data with metric temporal logic. *J. Artif. Intell. Res.*, 62:829–877, 2018.

[Brzoska, 1998] C. Brzoska. Programming in metric temporal logic. *Theor. Comput. Sci.*, 202(1-2):55–125, 1998.

[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. Autom. Reas.*, 39:385–429, 2007.

[Chomicki and Toman, 1998] J. Chomicki and D. Toman. Temporal logic in information systems. In *Logics for Databases and Information Systems*, pages 31–70. Kluwer, 1998.

[Compton and Laflamme, 1990] K. Compton and C. Laflamme. An algebra and a logic for NC^1. *Inf. Comput.*, 87:240–262, 1990.

[Gottlob *et al.*, 2014] G. Gottlob, G. Orsi, and A. Pieris. Query rewriting and optimization for ontological databases. *ACM Trans. Database Syst.*, 39:25:1–25:46, 2014.

[Grädel, 1991] E. Grädel. On transitive closure logic. In *CSL*, volume 626 of *LNCS*, 149–163. Springer, 1991.

[Gutiérrez-Basulto *et al.*, 2016a] V. Gutiérrez-Basulto, J. Jung, and R. Kontchakov. Temporalized EL ontologies for accessing temporal data: Complexity of atomic queries. In *Proc. IJCAI*, pages 1102–1108, 2016.

[Gutiérrez-Basulto *et al.*, 2016b] V. Gutiérrez-Basulto, J. Jung, and A. Ozaki. On metric temporal description logics. In *ECAI*, volume 285 of *Frontiers in Artificial Intelligence and Applications*, pages 837–845, 2016.

[Immerman, 1999] N. Immerman. *Descriptive Complexity*. Springer, 1999.

[Koymans, 1990] R. Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Systems*, 2(4):255–299, 1990.

[Lutz and Sabellek, 2017] C. Lutz and L. Sabellek. Ontology-mediated querying with the description logic EL: trichotomy and linear datalog rewritability. In *Proc. IJCAI*, pages 1181–1187, 2017.

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

[Ouaknine and Worrell, 2008] J. Ouaknine and J. Worrell. Some recent results in metric temporal logic. In *FORMATS*, pages 1–13, 2008.

[Özçep and Möller, 2014] Ö. Özçep and R. Möller. Ontology based data access on temporal and streaming data. In *RW*, volume 8714 of *LNCS*, pages 279–312, 2014.

[Ronca *et al.*, 2018] A. Ronca, M. Kaminski, B. Cuenca Grau, B. Motik, and I. Horrocks. Stream reasoning in temporal datalog. In *AAAI*, pages 1941–1948, 2018.

[Ryzhikov *et al.*, 2019] V. Ryzhikov, P. Wałęga, and M. Zakharyaschev. Data complexity and rewritability of ontology-mediated queries in metric temporal logic under the event-based semantics (Technical Report). http://arxiv.org/abs/1905.12990, 2019.

[Soylu *et al.*, 2017] A. Soylu, M. Giese, R. Schlatte, E. Jiménez-Ruiz, E. Kharlamov, Ö. Özçep, C. Neuenstadt, and S. Brandt. Querying industrial stream-temporal data: An ontology-based visual approach. *JAISE*, 9(1):77–95, 2017.

[Wałęga *et al.*, 2019] P. Wałęga, B. Cuenca Grau, and M. Kaminski. Reasoning over streaming data in metric temporal datalog. In *AAAI*, 2019.

[Xiao *et al.*, 2018] G. Xiao, D. Calvanese, R. Kontchakov, D. Lembo, A. Poggi, R. Rosati, and M. Zakharyaschev. Ontology-based data access: A survey. In *Proc. IJCAI*, 2018.