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

Vladislav Ryzhikov[1] , **Przemysław Andrzej Wałęga**[2] , **Michael Zakharyaschev**[1]

[1]Birkbeck, University of London, U.K.
[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 ontologies given in metric temporal logic *MTL* under the event-based semantics, assuming that data instances are finite timed words with binary fractions as timestamps. We identify various classes of ontology-mediated queries answering which can be done in AC^0, NC^1, L, NL, P, and CONP, provide 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 by binary fractions and any timestamp t from $\mathcal{D}$, we have $\Pi, \mathcal{D} \models A(t)$ iff $\boldsymbol{Q}(t)$ holds in $\mathcal{D}$?

MTL was originally designed for modelling and reasoning about real-time systems [Koymans, 1990; Alur and Henzinger, 1993]; for a survey see [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; Mehdi *et al.*, 2017] for practical ontology-based access to temporal log data aiming to facilitate detection and monitoring complex events in asynchronous systems based on sensor measurements. For example, a Siemens gas turbine has a normal stop if the rotor speed coasts down from 1500 to 200, which was preceded by another coast down from 6600 to 1500 some time in the previous 9 minutes, at most 2 minutes before which the main flame was off, while the active power was off earlier within another 2 minutes. The event 'normal stop' can be encoded by the following *MTL*-formula, where $\diamondminus_{(r,s]}\varphi$ is true at a timestamp t if φ holds at a timestamp t' with $r < t - t' \leq s$:

$$\mathsf{CoastDown}_{1500\text{-}200} \wedge \diamondminus_{(0,9m]}\big(\mathsf{CoastDown}_{6600\text{-}1500} \wedge \diamondminus_{(0,2m]}\big(\mathsf{FlameOff} \wedge \diamondminus_{(0,2m]}\mathsf{PowerOff}\big)\big) \to \mathsf{NormalStop}.$$

Now, to find when a normal stop occurred, a service engineer can simply run the query $\mathsf{NormalStop}(x)$ mediated by an *MTL*-ontology with formulas such as the one above, whose atoms are related to sensor data by appropriate mappings. Answering *datalogMTL* queries in the streaming setting has been 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; cf. [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 *hornMTL* (without temporal operators in the heads) are P-complete and rewritable to datalog(FO) extending datalog with FO-formulas built from EDB predicates; in fact, we establish P-hardness already for the fragment *coreMTL*$^{\boxminus}$ of *hornMTL* with binary rules and box operators only. OMQs in *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 and FO(DTC) (FO with deterministic closure). Note that all of our rewritings save datalog(FO) can be implemented in SQL.

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*

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

where each ϑ_i takes the form A, $\diamondminus_\varrho A$, or $\boxminus_\varrho A$ with 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 easily 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 atomic proposition. 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 called a *coreMTL-program* if $k + l \leq 2$, that is, all of its rules are of the form $\vartheta_1 \to \vartheta_2$ or $\vartheta_1 \wedge \vartheta_2 \to \bot$. An *MTL*- (*hornMTL*- or *coreMTL*-) *ontology-mediated query* (OMQ for short) takes the form $\boldsymbol{q} = (\Pi, A)$, where Π is an *MTL*- (respectively, *hornMTL*- or *coreMTL*-) program and A an atomic proposition.

Formally, a *data instance*, $\mathcal{D}$, is an FO-structure of the form

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

with domain $\varDelta = \{0, \dots, \ell\}$ strictly linearly ordered by $<$, *timestamps* $\varTheta = \{0, \dots, k\}$, for $1 \leq k \leq \ell$, and subsets $A_i^{\mathcal{D}} \subseteq \varTheta$. The ternary predicates bit_{in} and bit_{fr} are such that, for any $n \in \varTheta$ and $i \in \varDelta$, there is a unique $b_i \in \{0, 1\}$ and a unique $c_i \in \{0, 1\}$ with $\mathsf{bit}_{in}(i, n, b_i)$ and $\mathsf{bit}_{fr}(i, n, c_i)$. These predicates provide the *value* $\bar{n} \in \mathbb{Q}_2^{\geq 0}$ of every timestamp $n \in \varTheta$, viz. $\bar{n} = b_\ell \cdots b_0.c_0 \cdots c_\ell$ iff $\mathsf{bit}_{in}(i, n, b_i)$ and $\mathsf{bit}_{fr}(i, n, c_i)$ hold, for $i \leq \ell$. We assume that $\bar{n} < \bar{m}$ if $n < m$. Intuitively, $\mathcal{D}$ represents the timed word $\boldsymbol{A}_0(\bar{0}), \dots, \boldsymbol{A}_k(\bar{k})$ with timestamp values $\bar{0} < \dots < \bar{k}$ and sets $\boldsymbol{A}_i$ of atoms that hold true at $\bar{i}$. 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 $x, y \in \varTheta$, $0 \leq \bar{x} - \bar{y} < r$, its variants $\mathsf{dist}_{>r}(x, y)$, $\mathsf{dist}_{=r}(x, y)$, etc. (see the full version). Using these, we can further define FO-formulas $\mathsf{in}_\varrho(x, y)$ saying that $\bar{x} - \bar{y} \in \varrho$, $\mathsf{suc}(x, y)$ 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 any structure

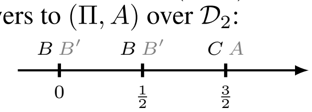
$$\mathcal{I} \;=\; (\varDelta, <, \varTheta, \mathsf{bit}_{in}, \mathsf{bit}_{fr}, A_1^{\mathcal{I}}, \dots, A_p^{\mathcal{I}}), \;\; A_i^{\mathcal{D}} \subseteq A_i^{\mathcal{I}} \subseteq \varTheta,$$

where the Boolean connectives are interpreted as usual and

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

Since $\boxminus_\varrho A$ is equivalent to $\neg \diamondminus_\varrho \neg A$, we often treat $\boxminus_\varrho$ as an abbreviation and assume that programs Π contain only $\diamondminus_\varrho$. An interpretation $\mathcal{I}$ is a *model* of an *MTL*-program Π and $\mathcal{D}$ if, for any rule (1) in Π and any $t \in \varTheta$, 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 $\mathcal{D}$.

From now on, we write $\mathsf{ts}(\mathcal{D})$ for the set $\varTheta$ of timestamps in (2). We call $t \in \mathsf{ts}(\mathcal{D})$ 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 $\mathcal{D}_2$:



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 $|\varTheta|$ 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.

OMQ answering for $\boldsymbol{q} = (\Pi, A)$ is clearly FO-reducible to inconsistency checking. Indeed, given $\mathcal{D}$ and $t \in \mathsf{ts}(\mathcal{D})$, let $\Pi' = \Pi \cup \{A \wedge B \to \bot\}$, for a fresh B, and $\mathcal{D}' = \mathcal{D} \cup \{B(t)\}$. Then t is a certain answer to $\boldsymbol{q}$ over $\mathcal{D}$ iff Π' and

$\mathcal{D}'$ are inconsistent. (Note that checking if an *MTL*-program Π is consistent with *some* $\mathcal{D}$ is decidable but not primitive recursive [Ouaknine and Worrell, 2007].)

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 $\vartheta_1 \land \dots \land \vartheta_n \to \vartheta$ is in Π and $(\vartheta_i, t) \in \mathfrak{C}$, for all i, $1 \le i \le k$, 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 $\mathfrak{C}_{\Pi,\mathcal{D}} = \mathsf{cl}^N(\mathcal{D})$. The proof of the following is standard:

Theorem 1. *A timestamp $t \in \mathsf{ts}(\mathcal{D})$ is a certain answer to a hornMTL-OMQ (Π, A) over $\mathcal{D}$ iff either $(A, t) \in \mathfrak{C}_{\Pi,\mathcal{D}}$ or Π is inconsistent with $\mathcal{D}$, in which case there is $(\bot, t') \in \mathfrak{C}_{\Pi,\mathcal{D}}$.*

Theorem 1 implies that if atomic OMQs (Π, A) with *hornMTL* Π are $\mathcal{L}$-rewritable, then OMQs $(\Pi, \varphi(x))$ with FO-queries $\varphi(x)$ over data instances defined above are also $\mathcal{L}$-rewritable because a rewriting of $(\Pi, \varphi(x))$ can be obtained by replacing each $A(x)$ in φ by the rewriting of (Π, A).

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. By *coreMTL*$^{\boxminus}$ (*coreMTL*$^{\diamondminus}$) we denote the restriction of *coreMTL* to the language with operators $\boxminus_\varrho$ (respectively, $\diamondminus_\varrho$) only.

Theorem 2. (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.*

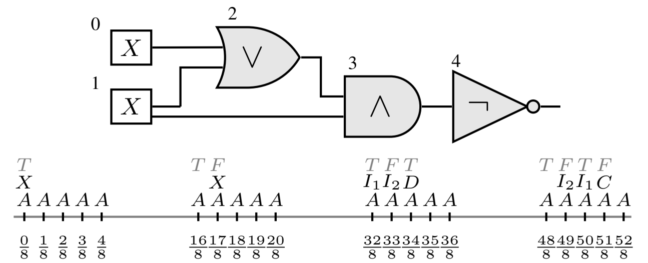
Proof sketch. (i) The membership in coNP is trivial. We establish coNP-hardness by reduction of NP-complete circuit satisfiability [Arora and Barak, 2009]. Let $\boldsymbol{C}$ be a Boolean circuit with N_0-many AND, OR and NOT gates enumerated by consecutive natural numbers starting from 0 so that if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

- $A(2n + i/N)$, if n is a gate and $0 \le i \le n$;
- $X(2n + n/N)$, 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:

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

One can check that $\mathbf{C}$ is satisfiable iff $2(N_0-1)+(N_0-1)/N$ 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:



(iii) Without loss of generality, we assume that, in the disjointness constraints $\vartheta_1 \land \vartheta_2 \to \bot$ occurring in the given *coreMTL*$^{\diamondminus}$-OMQ $\boldsymbol{q} = (\Pi, A)$, the ϑ_i are atomic. First, we straightforwardly translate $\boldsymbol{q}$ with the disjointness constraints removed from Π to linear datalog(FO). Then, we transform the result into an FO(TC)-query $\Psi_A(x)$ [Grädel, 1991]. Now, for every disjointness constraint $B_1 \land B_2 \to \bot$ in Π, we take the sentence $\exists x(\Psi_{B_1}(x) \land \Psi_{B_2}(x))$ and, finally, form a disjunction of $\Psi_A(x)$ with those sentences, which is obviously an FO(TC)-rewriting of $\boldsymbol{q}$.

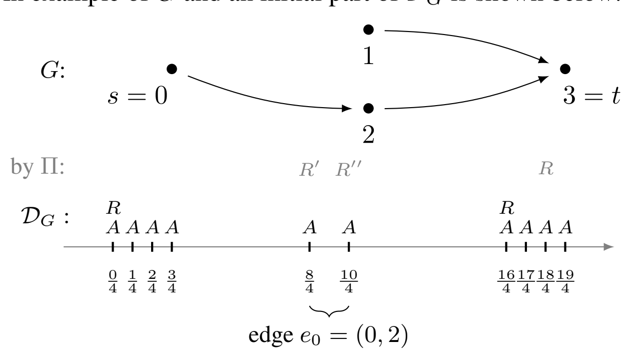
We prove NL-hardness by reduction of the reachability problem in acyclic digraphs. Let G be such a digraph with N_0 vertices enumerated by consecutive natural numbers starting from 0 so that if there is an edge from n to m, then $n < m$. Let $e_0, \dots, e_{k-1}$ be the lexicographical order of edges (i.e., a subsequence of $(0, 1), \dots, (0, N_0 - 1), (1, 0), \dots$). Take the minimal $N = 2^i \ge N_0$ 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) $A(4i + n/N)$, for $0 \le i \le k$ and a vertex n;

(b) $A(2 + 4i + n/N)$, $A(2 + 4i + m/N)$, for every edge $e_i = (n, m)$;

(c) $R(4i + s/N)$, for $0 \le i \le m$.

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. Atoms implied by Π and $\mathcal{D}_G$ in our example are shown in the figure above.

(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 rules $P(x) \to P'(x, x)$ for each P in Π. The other rules of Π' are obtained from the rules of Π by the following transformations. We replace every atom B, which is not under the scope of a temporal operator, with $B'(x, x)$ and we replace 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 $\diamondminus_\varrho B$. 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

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

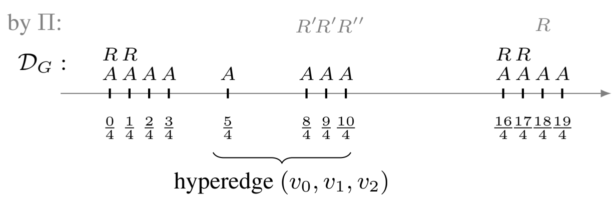
Note that the obtained datalog program Π' contains FO-definable EDB predicates such as $\mathsf{dist}_{\geq r}(x, w)$ and $\mathsf{suc}(y, z)$ in rule bodies. One can readily show that t is a certain answer to $\boldsymbol{q}$ over any given data instance $\mathcal{D}$ iff t is an answer to $(\Pi', G(x))$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

We show P hardness of *coreMTL*$^{\boxminus}$-OMQs by reduction of path system accessibility. Let G be a hypergraph with N_0 vertices enumerated by consecutive natural numbers starting from 0 so that if (m, n, o) is a hyperedge, then $m < n < o$. Let $e_0, \dots, e_{k-1}$ be the lexicographical order of hyperedges. Suppose the problem is to check whether a vertex t is accessible from a set of vertices S, i.e., whether $t \in S$ or there are vertices u, w accessible from S and (u, w, t) is a hyperedge. Let $\mathcal{D}_G$ comprise the atoms in (a) together with

(d) $A(2+4i+m/N)$, $A(2+4i+n/N)$, $A(2+4i+o/N)$, and $A(2+4i+n/N-1)$, for a hyperedge $e_i = (m, n, o)$;

(e) $R(4i + n/N)$, for $0 \leq i \leq m$ and $n \in S$.

For example, for the vertices $0, 1, 2, 3$, hyperedge $(0, 1, 2)$, $S = \{0, 1\}$, and $t = 3$, $\mathcal{D}_G$ looks as follows:



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

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

Then $4k + t/N$ is a certain answer to (Π, R) over $\mathcal{D}_G$ iff t is accessible 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.

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 charactering the structure of models.

Theorem 3. *MTL-OMQs with temporal operators of the form* $\diamondminus_{\langle r,\infty)}$ *and* $\boxminus_{\langle r,\infty)}$ *only are* FO($<$)*-rewritable.*

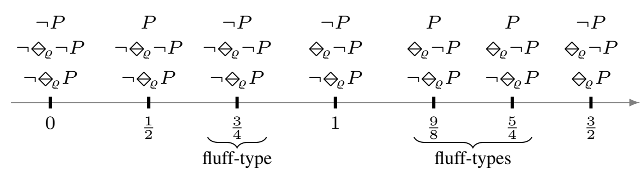
Proof sketch. Let $\boldsymbol{q} = (\Pi, A)$ be an *MTL*-OMQ as specified above. A *simple literal*, σ, for $\boldsymbol{q}$ takes the form P or $\neg P$, where P is an atom in $\boldsymbol{q}$; a *temporal literal*, τ, for $\boldsymbol{q}$ is of the form $\diamondminus_\varrho \sigma$ or $\neg \diamondminus_\varrho \sigma$ 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 $\boldsymbol{q}$, 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 $t \in \mathsf{ts}(\mathcal{D})$, denote by $\boldsymbol{t}(t)$ the type of t in $\mathcal{I}$. As the ranges in Π are of the form $\langle r, \infty)$, the model $\mathcal{I}$ has the following *monotonicity property*:

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

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

We call $\boldsymbol{t}(t)$ in $\mathcal{I}$ an *osteo-type* if there is $\lambda \in \boldsymbol{t}(t)$ such that $\lambda \notin \boldsymbol{t}(t')$, for all $t' < t$. Thus, if $\diamondminus_\varrho \sigma \in \boldsymbol{t}(t')$ in $\mathcal{I}$, there is an osteo-type $\boldsymbol{t}(t) \ni \sigma$ with $\mathsf{in}_\varrho(t', t)$. 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 $\boldsymbol{t}(t')$ has the same temporal literals as its nearest osteo-type $\boldsymbol{t}(t)$, for $t < t'$. 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 $\mathfrak{O}_\Pi$ be the set of sequences $\bar{\boldsymbol{t}} = (\boldsymbol{t}_1, \dots, \boldsymbol{t}_n)$, $1 \leq n \leq N_\Pi$, of distinct types for $\boldsymbol{q}$ that satisfy the monotonicity property and such that $\diamondminus_\varrho \sigma \in \boldsymbol{t}_i$ implies $\sigma \in \boldsymbol{t}_j$ for some $j \leq i$; 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 $j \leq i$, $\neg\diamondminus_\varrho \sigma \in \boldsymbol{t}(t_i)$ and $\sigma \in \boldsymbol{t}(t_j)$, for some $\diamondminus_\varrho \sigma$. Denote by $\mathfrak{F}^i_{\bar{\boldsymbol{t}}}$ the set of types $\boldsymbol{t}$ for $\boldsymbol{q}$ 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 $j \leq i$. Finally, for any type $\boldsymbol{t}$, let $\delta_{\boldsymbol{t}}(x) = \bigwedge_{\neg P \in \boldsymbol{t}} \neg P(x)$. Now, we set

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

where $x_i \prec y$ says that x_i is the nearest predecessor of y, which is different from $x_1, \dots, x_n$.

Suppose $\mathcal{I}$ is a model of Π, $\mathcal{D}$ and $\bar{\boldsymbol{t}} = (\boldsymbol{t}(t_1), \dots, \boldsymbol{t}(t_n))$, for $t_1 < \dots < t_n$, are all the osteo-types in $\mathcal{I}$. This n-tuple of types is in $\mathfrak{O}_\Pi$ and the $\delta_{\boldsymbol{t}(t_i)}(t_i)$ hold true in $\mathcal{I}$ by definition. The $\mathsf{in}_\varrho(t_i, t_j)$ also hold for $\mathit{wit}(\boldsymbol{t}(t_i), \boldsymbol{t}(t_j), \varrho)$ because $\boldsymbol{t}(t_i)$ is the first type in $\mathcal{I}$ witnessing the relevant $\diamondminus_\varrho \sigma$. Similarly, $\mathsf{in}_\varrho(t_i, t_j)$ does not hold in $\mathcal{I}$ for $\overline{\mathit{wit}}(\boldsymbol{t}(t_i), \boldsymbol{t}(t_j), \varrho)$. Finally, let t be any timestamp in $\mathcal{I}$ with $t_i \prec t$. By construction, $\boldsymbol{t}(t)$ is a fluff-type in $\mathfrak{F}^i_{\bar{\boldsymbol{t}}}$ and $\delta_{\boldsymbol{t}(t)}(t)$ holds in $\mathcal{I}$. If $\overline{\mathit{wit}}(\boldsymbol{t}(t_i), \boldsymbol{t}(t_j), \varrho)$, we have $\neg\diamondminus_\varrho \sigma \in \boldsymbol{t}(t_i) \cap \boldsymbol{t}(t)$ and $\sigma \in \boldsymbol{t}(t_j)$, and so $\mathsf{in}_\varrho(t, t_j)$ cannot hold in $\mathcal{I}$. Thus, $\mathcal{D} \models \Phi_\Pi$.

Conversely, suppose Φ_Π holds in $\mathcal{D}$, assigning timestamps t_i to the x_i and associating types $\boldsymbol{t}(t)$ with every $t \in \mathsf{ts}(\mathcal{D})$. Define an interpretation $\mathcal{I}$ by setting

$$P^{\mathcal{I}} = \{t \in \mathsf{ts}(\mathcal{D}) \mid P \in \boldsymbol{t}(t)\},$$

for every atom P. We prove that $\mathcal{I}$ is a model of Π and $\mathcal{D}$. As all the $\boldsymbol{t}(t)$ are types for $\boldsymbol{q}$, it suffices to show that

$$\diamondminus_\varrho \sigma \in \boldsymbol{t}(t) \iff \exists t' \left(\mathsf{in}_\varrho(t, t') \,\wedge\, \sigma \in \boldsymbol{t}(t')\right).$$

Suppose $\diamondminus_\varrho \sigma \in \boldsymbol{t}(t)$. If $t = t_i$, for some i, then $\mathit{wit}(\boldsymbol{t}_i, \boldsymbol{t}_j, \varrho)$, for some $j \le i$, and so $\sigma \in \boldsymbol{t}(t_j)$ and $\mathsf{in}_\varrho(t_i, t_j)$ holds in $\mathcal{I}$. If $t_i \prec t$, then $\diamondminus_\varrho \sigma \in \boldsymbol{t}(t_i)$, and we can use the previous argument as $\mathsf{in}_\varrho(t_i, t_j)$ implies $\mathsf{in}_\varrho(t, t_j)$.

Conversely, suppose $\diamondminus_\varrho \sigma \notin \boldsymbol{t}(t)$. Then $\neg\diamondminus_\varrho \sigma \in \boldsymbol{t}(t)$. Consider first the case $t = t_i$. Suppose $t' \le t_i$ with $\sigma \in \boldsymbol{t}(t')$. Then $\sigma \in \boldsymbol{t}(t_j)$ for some $t_j \le t'$, and so $\overline{\mathit{wit}}(\boldsymbol{t}_i, \boldsymbol{t}_j, \varrho)$ and $\neg\mathsf{in}_\varrho(t_i, t_j)$, whence $\neg\mathsf{in}_\varrho(t, t_j)$ and $\neg\mathsf{in}_\varrho(t, t')$. Now, let $t \notin \{t_1, \dots, t_n\}$. Then $t_i \prec t$ for some i (because of $\min x_1$). Suppose $t' \le t$ with $\sigma \in \boldsymbol{t}(t')$. If $t' < t_i$, then $\sigma \in \boldsymbol{t}(t_j)$ for some $t_j \le t'$, and so $\overline{\mathit{wit}}(\boldsymbol{t}_i, \boldsymbol{t}_j, \varrho)$ and $\neg\mathsf{in}_\varrho(t, t_j)$, whence $\neg\mathsf{in}_\varrho(t, t')$. If $t' = t_i$, then, by the last conjunct of Φ_Π, we have $\neg\mathsf{in}_\varrho(t, t_i)$. Finally, if $t_i < t' \le t$, then $\sigma \in \boldsymbol{t}(t_j)$, for some $t_j \le t_i$, and we are done again.

An $\mathrm{FO}(<)$-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 $\delta_{\boldsymbol{t}}(z) \wedge (x \ne z)$ 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 and *MTL*-program *range-uniform* if all of its temporal operators have the same constraining range.

Theorem 4. *Range-uniform* core*MTL*$^{\diamondminus}$*-OMQs with ranges of the form* $\diamondminus_{\langle 0, r\rangle}$ *are* $\mathrm{FO}(<, +)$*-rewritable.*

Proof sketch. We illustrate the proof by a concrete example. Consider the OMQ (Π, S_1) with

$$\Pi = \{S_0 \leftarrow B,\ S_1 \leftarrow \diamondminus_{(0,d)} S_0,\ S_2 \leftarrow \diamondminus_{(0,d)} S_1,\ S_3 \leftarrow \diamondminus_{(0,d)} S_2,\ S_1 \leftarrow \diamondminus_{(0,d)} S_3\}.$$

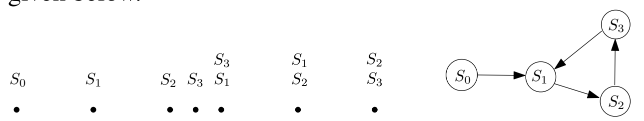
With such an OMQ, we can naturally associate an automaton $\mathcal{A}_{S_0,B}$ (in normal form) shown in the picture below on the right. Using it, we construct the following FO-rewriting $\boldsymbol{Q}(x)$ of (Π, S_1):

$$\exists x' \left[B(x') \wedge \forall y \left((x' < y \le x) \to \exists y'\, \mathsf{dist}_{<d}(y, y') \wedge (\varphi_1(x', x) \vee \varphi_2(x', x) \vee \varphi_3(x', x))\right)\right],$$

where

- $\varphi_1(x', x) = (x - x') \in 1 + 3\mathbb{N}$;
- $\varphi_2(x', x) = (x - x') \in 2 + 3\mathbb{N} \wedge \exists x_1 \left((x' < x_1 \le x) \wedge \varphi_{+1}(x_1, x')\right)$;
- $\varphi_3(x', x) = (x - x') \in 3 + 3\mathbb{N} \wedge \varphi_{1+1+1}(x', x) \vee \varphi_{1+2}(x', x)$;
- $\varphi_{1+2}(x', x) = \exists x_1 \left((x' < x_1 \le x) \wedge \varphi_{+2}(x_1, x')\right)$;
- $\varphi_{1+1+1}(x', x) = \exists x_1, x_2 \left((x' < x_1 < x_2 \le x) \wedge \varphi_{+1}(x_1, x') \wedge \varphi_{+1}(x_2, x') \wedge ((x_2 - x_1) > 1)\right)$;
- $\varphi_{+k}(z, x') = \mathsf{dist}_{<d}(z, z - k - 1) \wedge ((z - k - 1) \ge x')$, for $k = 1, 2$.

Intuitively, to derive S_1 at x, we need a point x' with $B(x')$ in the data and a sequence of points y between x' and x without gaps of length $\ge d$. An example of such a data instance is given below.



Note how we maintain the 'stack of states' with the elements at its bottom alternating in a cycle between S_1, S_2, and S_3. Note also that the states go in decreasing order when we scan the stack from bottom to top. So we use the formulas $\varphi_k(x', x)$ to express that S_1 is inferred at x on level k of the stack. The formula $\varphi_{+k}(z, x')$ says that the height of the stack increases by k because of a cluster of $k + 2$ points within the segment of size $< d$ ending with z. The formulas $\varphi_{1+2}(x', x)$ and $\varphi_{1+1+1}(x', x)$ express two ways of increasing the height of the stack from 1 to 3. It is to be emphasised that properties of x and x' such as $(x - x') \in 1 + 3\mathbb{N}$ can be expressed by FO-formulas using the predicate $\mathrm{PLUS}(\mathit{num}1, \mathit{num}2, \mathit{sum})$ or $\mathrm{BIT}(\mathit{num}, \mathit{bit})$, which gives a binary representation of every object num in the domain of an FO-structure [Immerman, 1999], whereas FO with $<$ only is not enough. For example, $(x - x') \in 1 + 3\mathbb{N}$ is expressed by the formula

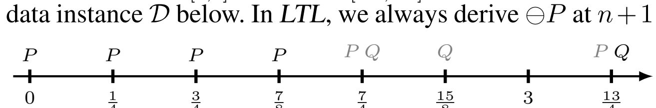
$$\varphi_1(x', x) \;=\; \exists z, z', z'', y \left((x = y + 1) \wedge \mathrm{PLUS}(z, z, z') \wedge \mathrm{PLUS}(z', z, z'') \wedge \mathrm{PLUS}(x', z'', y)\right).$$

❑

The proof given in the full version 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 last P with Q at $13/4$ implies P at $13/4$; independently, P at $7/8$ imply Q at $15/8$.

Theorem 5. *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 $\boldsymbol{q}$ 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 $\diamondminus_{\boldsymbol{j}+\boldsymbol{i}}\sigma \in cl(\Pi)$, $\boldsymbol{j} \geq \mathbf{1}$.

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 $\diamondminus_{\boldsymbol{j}}\sigma \in cl(\Pi)$;

– $\bar{t}_{i+1} - \bar{t}_i \in \{\mathbf{1}, \dots, \boldsymbol{m}\}$ and if $t_{i+1} > t > t_i$ then $\bar{t} - \bar{t}_i \notin \{\mathbf{1}, \dots, \boldsymbol{m}\}$, 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 $\bar{t} - \bar{t}' \notin \{\mathbf{1}, \dots, \boldsymbol{m}\}$ for all $t' \in \mathsf{ts}(\mathcal{D})$. The next lemma follows directly from the given definitions:

Lemma 6. (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 $\mathsf{div}_{\mathbf{1}}(u, v)$ that holds true iff $\bar{u} - \bar{v} = n\mathbf{1}$, for some $n \in \mathbb{N}$ (see the full version). We also have an FO-formula $\mathsf{last}_{\boldsymbol{i}}(u)$ saying that $\boldsymbol{i}$ is minimal among $\{\mathbf{1}, \dots, \boldsymbol{m}\}$ 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 $\boldsymbol{q}$, 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

$$\begin{bmatrix} R_{\boldsymbol{s}_1}(x,z) \equiv \vartheta_{\boldsymbol{s}_1} \\ \dots \\ R_{\boldsymbol{s}_n}(x,z) \equiv \vartheta_{\boldsymbol{s}_n} \end{bmatrix} \bigvee_{\neg A \in \boldsymbol{s} \in Q} R_{\boldsymbol{s}}(x, y) \ \wedge\ \mathsf{div}_{\mathbf{1}}(y, x),$$

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 $\boldsymbol{s} \notin Q_0$:

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

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

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

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

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

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

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

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

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

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

where Φ_Π checks the consistency condition of Lemma 6 (i) and can be constructed similarly to $\mathsf{run}_{\boldsymbol{q}}$. ❑

6 OMQs with Non-Punctual Ranges

Unlike the proof of Theorem 5, 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 7 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 7. (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.*

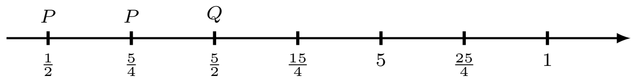
Proof sketch. In both cases, NC^1-hardness can be established as in the proof of Theorem 5 by encoding $\ominus$ with $\diamondminus_{(0,1]}$.

(i) Let $\boldsymbol{q} = (\Pi, A)$ be the given OMQ. As before, we assume w.l.o.g. that Π does not contain ranges $[0,0]$. 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 $u_1 \leq s_1 < u_2 \leq s_2 < \cdots < u_\ell \leq s_\ell \leq t$, 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 when 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 3);

– $\boldsymbol{tr}_\sigma$ 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 $u, v \neq *$. 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:

$\boldsymbol{t}_0 = \{P, \neg Q, \neg \diamondminus_{(2,4]} P, \neg \diamondminus_{[1,2)} P, \neg \diamondminus_{[3,\infty)} Q\}, t_0 = \frac{1}{2}$,
$\boldsymbol{tr}^0_P = ([\frac{1}{2}, \frac{1}{2}], [*,*], [*,*], [*,*], [\frac{1}{2}, \frac{1}{2}])$, $\boldsymbol{tr}^0_Q = ([*,*])$;
$\boldsymbol{t}_1 = \{P, \neg Q, \neg \diamondminus_{(2,4]} P, \diamondminus_{[1,2)} P, \neg \diamondminus_{[3,\infty)} Q\}, t_1 = \frac{5}{4}$,
$\boldsymbol{tr}^1_P = ([\frac{1}{2}, \frac{1}{2}], [*,*], [*,*], [*,*], [\frac{1}{2}, \frac{5}{4}])$, $\boldsymbol{tr}^1_Q = ([*,*])$;
$\boldsymbol{t}_2 = \{P, Q, \diamondminus_{(2,4]} P, \neg \diamondminus_{[1,2)} P, \neg \diamondminus_{[3,\infty)} Q\}$, $t_2 = \frac{5}{2}$,
$\boldsymbol{tr}^2_P = ([\frac{1}{2}, \frac{1}{2}], [*,*], [*,*], [\frac{1}{2}, \frac{5}{4}], [\frac{5}{2}, \frac{5}{2}])$, $\boldsymbol{tr}^2_Q = ([\frac{5}{2}, \frac{5}{2}]); \dots$

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

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 8 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 justifies the increase of complexity in Theorem 2.

Lemma 8. *Let $t_0 < \dots < t_m$ 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, $0 \leq i \leq m$, 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 $\sigma \in \boldsymbol{t}_i$, $\boldsymbol{tr}^{i-1}_\sigma = ([u_0, s_0], \dots, [u_{\ell_\sigma}, t_{\ell_\sigma}])$ and $[u, s] = [u_0, s_0]$ when $u_0 \neq *$ 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 $\boldsymbol{tr}^i_\sigma = ([u, s], [u_2, s_2], \dots, [u_{\ell_\sigma}, s_{\ell_\sigma}], [t_i, t_i])$.*

Proof sketch. $(\Rightarrow)$ For a model $\mathcal{I}$ of Π and $\mathcal{D}$, we define $(\boldsymbol{t}_i, (\boldsymbol{tr}^i_\sigma)_{\sigma \in \Sigma_\Pi}, t_i)$ with $\boldsymbol{t}_i = \boldsymbol{t}(t_i)$ as follows. If there is a minimal $t_j \leq t_i$ with $\sigma \in \boldsymbol{t}(t_j)$, we set $[u_0, s_0] = [t_j, t_j]$ in $\boldsymbol{tr}^i_\sigma$; otherwise $[u_0, s_0] = [*,*]$. Consider a maximal interval $[u, s]$, $s \leq t_i$, such that $\sigma \in \boldsymbol{t}(u) \cap \boldsymbol{t}(s)$ and, for any $t \in [u, s)$, there is $t' \in \mathsf{ts}(\mathcal{D})$ with $\sigma \in \boldsymbol{t}(t')$ and $\bar{t}' - \bar{t} \in \varrho_\sigma^-$. Suppose there are k such intervals. Let $k' = \min\{k, \ell_\sigma\}$. We define $\boldsymbol{tr}^i_\sigma$ by making its last intervals equal to $[u_{\ell_\sigma - k' + 1}, s_{\ell_\sigma - k' + 1}], \dots, [u_{\ell_\sigma}, s_{\ell_\sigma}]$, making its 0-th interval equal to $[u_0, s_0]$, and making all the remaining intervals equal to $[*,*]$. One can check that $(\boldsymbol{t}_i, (\boldsymbol{tr}^i_\sigma)_{\sigma \in \Sigma_\Pi}, t_i)$ are as required.

$(\Leftarrow)$ Given a sequence $(\boldsymbol{t}_i, (\boldsymbol{tr}^i_\sigma)_{\sigma \in \Sigma_\Pi}, t_i)$, $1 \leq i \leq m$, we construct an interpretation $\mathcal{I}$ by making σ true at t_i in $\mathcal{I}$ iff $\sigma \in \boldsymbol{t}_i$ for each simple literal σ from $\boldsymbol{q}$. The conditions on these extended types ensure that $\mathcal{I}$ is a model of Π and $\mathcal{D}$. ❑

We use the characterisation of Lemma 8 to construct an FO(TC)-sentence Φ_Π that holds 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:

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

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 $\boldsymbol{x}'$:

$$\begin{aligned} \xi_{\boldsymbol{t}'}(t, \boldsymbol{x}, t', \boldsymbol{x}') = {} & \mathsf{ext}_{\boldsymbol{t}'}(t', \boldsymbol{x}') \wedge \bigwedge_{\sigma \notin \boldsymbol{t}'} (\boldsymbol{x}_\sigma = \boldsymbol{x}'_\sigma) \wedge \\ & \bigwedge_{\sigma \in \boldsymbol{t}'} \Big[\big((x_{0\sigma} > y_{0\sigma}) \to (x'_{0\sigma} = t') \wedge (y'_{0\sigma} = t') \big) \wedge \\ & \quad \big((x_{0\sigma} \leq y_{0\sigma}) \to (\boldsymbol{x}_{0\sigma} = \boldsymbol{x}'_{0\sigma}) \big) \wedge \\ & \quad \Big(\mathsf{in}_{\varrho_\sigma^-}(t', y_{\ell_\sigma}) \to \bigwedge_{1 \leq i < \ell_\sigma - 1} (\boldsymbol{x}_{i\sigma} = \boldsymbol{x}'_{i\sigma}) \wedge \\ & \qquad (x_{\ell_\sigma \sigma} = x'_{\ell_\sigma \sigma}) \wedge (y'_{\ell_\sigma \sigma} = t') \Big) \wedge \\ & \quad \Big(\neg \mathsf{in}_{\varrho_\sigma^-}(t', y_{\ell_\sigma}) \to \bigwedge_{1 < i \leq \ell_\sigma - 1} (\boldsymbol{x}_{i\sigma} = \boldsymbol{x}'_{i-1\sigma}) \wedge \\ & \qquad (x'_{\ell_\sigma \sigma} = t') \wedge (y'_{\ell_\sigma \sigma} = t') \Big) \Big]. \end{aligned}$$

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

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

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

$$\delta_{\boldsymbol{t}}(\min) \wedge \bigwedge_{\sigma \notin \boldsymbol{t}} \bigwedge_{0 \le i \le \ell_\sigma} ((x_{i\sigma} = \max) \wedge (y_{i\sigma} = \min) \wedge \bigwedge_{\sigma \in \boldsymbol{t}} \bigwedge_{0 \le i \le \ell_\sigma} (x_{i\sigma} = \max) \wedge (x_{0\sigma} = \min) \wedge (y_{0\sigma} = \min) \wedge (x_{\ell_\sigma \sigma} = \min) \wedge (y_{\ell_\sigma \sigma} = \min))$$

saying that the intervals in the initial extended type are set correctly. That Φ_Π is as required follows from Lemma 8.

As previously, to obtain the rewriting $\Phi_{\boldsymbol{q}}(x)$ we need a formula $\Phi_{\neg A}(x)$ that holds true on $\mathcal{D}$ iff there exists a model of $(\Pi, \mathcal{D})$ such that $\neg A$ is true at x in this model. We take $\Phi_{\neg A}(x)$ equal to the conjunction of Φ_Π and

$$\exists \boldsymbol{x}, \boldsymbol{x}', \boldsymbol{x}'', t_1 \big(((x = 0) \wedge \bigvee_{\substack{\boldsymbol{t} \text{ type for } \Pi \\ \neg A \in \boldsymbol{t}}} \mathsf{first}_{\boldsymbol{t}}(\boldsymbol{x})) \vee (\mathsf{first}_{\boldsymbol{t}}(\boldsymbol{x}) \wedge [\mathrm{TC}_{t, \boldsymbol{x}, t', \boldsymbol{x}'} \theta(t, \boldsymbol{x}, t', \boldsymbol{x}')](\min, \boldsymbol{x}, t_1, \boldsymbol{x}') \wedge \mathsf{suc}(x, t_1) \wedge \bigvee_{\substack{\boldsymbol{t}' \text{ type for } \Pi \\ \neg A \in \boldsymbol{t}'}} \theta_{\boldsymbol{t}'}(t_1, \boldsymbol{x}', x, \boldsymbol{x}'')) \big).$$

The negation of $\Phi_{\neg A}(x)$ is the required rewriting $\Phi_{\boldsymbol{q}}(x)$.

(ii) It will be convenient to assume a restricted version of our normal form (1), where $\diamondminus_{\varrho}$ operators do not occur with $\varrho = [0, r\rangle$. Every *hornMTL* program Π can be converted to this form by replacing $\diamondminus_{[0,r\rangle} A$ by $A \vee \diamondminus_{(0,r\rangle} A$ and then expressing, e.g., $A \vee \diamondminus_{(0,r\rangle} A \to B$ by a pair of rules $A \to B$, $\diamondminus_{(0,r\rangle} A \to B$. (For each $\neg \diamondminus_{[0,r\rangle} \neg A$ we substitute it by $A \wedge \neg \diamondminus_{(0,r\rangle} \neg A$.) Let $(\boldsymbol{tr}_\sigma)_{\sigma \in \Sigma_\Pi}$ be a trace for $t' \in \mathsf{ts}(\mathcal{D})$ and Λ be the set $\diamondminus_{\varrho}\sigma$ from Π such that $int_\varrho(t, u_i, s_i)$, for some $[u_i, s_i]$ in $\boldsymbol{tr}_\sigma$. Let Δ be a set of P from Π. We will call a type $\boldsymbol{t}$ for Π *minimal for* t *w.r.t.* Δ *and* $(\boldsymbol{tr}_\sigma)_{\sigma \in \Sigma_\Pi}$ if every ϑ from Π is in $\boldsymbol{t}$ iff ϑ is in the closure of Λ and Δ under rules (1). We say that a type $\boldsymbol{t}$ is *minimal initial w.r.t.* Δ if it is minimal for some (every) t w.r.t. an empty trace with all $[u_j, s_j]$ in $\boldsymbol{tr}_\sigma$ are $[*, *]$ for all $\sigma \in \Sigma_\Pi$.

Lemma 9. *Let $t_0 < \dots < t_m$ be 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 conditions of Lemma 8 and such that: (i) $\boldsymbol{t}_0$ is minimal initial w.r.t. $\mathcal{D}(t_0)$, (ii) $\boldsymbol{t}_i$ is the minimal for t_i w.r.t. $\mathcal{D}(t_i)$ and $(\boldsymbol{tr}^{i-1}_\sigma)_{\sigma \in \Sigma_\Pi}$.*

Note that each type $\boldsymbol{t}_0$ in the lemma above is uniquely determined by $\mathcal{D}$ and so is the trace $(\boldsymbol{tr}^0_\sigma)_{\sigma \in \Sigma_\Pi}$. Then, type $\boldsymbol{t}_1$ is uniquely determined by $(\boldsymbol{tr}^0_\sigma)_{\sigma \in \Sigma_\Pi}$ and $\mathcal{D}$ and so is $(\boldsymbol{tr}^1_\sigma)_{\sigma \in \Sigma_\Pi}$, etc. Therefore, we can replace in Φ_Π the formula $\xi(t, \boldsymbol{x}, t', \boldsymbol{x}')$ by $\xi'(t, \boldsymbol{x}, t', \boldsymbol{x}')$ such that for given values of t and $\boldsymbol{x}$, there are *unique* values of t' and $\boldsymbol{x}'$ for which $\xi'(t, \boldsymbol{x}, t', \boldsymbol{x}')$ holds. ❑

7 Conclusion

This paper makes 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, which are rewritable to FO($<$), FO($<, +$), their extensions 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 ontologies (description logics or datalog) and more complex (say, conjunctive) queries.

We can extend our language with constrained operators since $\mathcal{S}_\varrho$. In this case, *hornMTL* remains P-complete (though *coreMTL* becomes P-hard) and Theorem 7 holds too. We believe that our *hornMTL* can also be extended with $\boxminus_{\varrho}$ in rule heads: Theorem 2 (ii) holds in this case too, but so far we have not managed to prove Theorem 7 (ii) for such rules. Extending *MTL* with future-time operators is also interesting, in which case Theorem 3 remains to hold. Finally, we are looking into *MTL*-OMQs under the continuous semantics, where the techniques developed in this paper do not apply.

References

[Alur and Dill, 1994] Rajeev Alur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183 – 235, 1994. http://www.sciencedirect.com/science/article/pii/0304397594900108.

[Alur and Henzinger, 1993] Rajeev Alur and Thomas A. 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, New York, USA, 1st edition, 2009.

[Artale *et al.*, 2015] Alessandro Artale, Roman Kontchakov, Alisa Kovtunova, Vladislav Ryzhikov, Frank Wolter, and Michael Zakharyaschev. First-order rewritability of temporal ontology-mediated queries. In *Proc. of the 24th Int. Joint Conf. on Artificial Intelligence, IJCAI 2015*, pages 2706–2712. IJCAI/AAAI, 2015.

[Artale *et al.*, 2017] Alessandro Artale, Roman Kontchakov, Alisa Kovtunova, Vladislav Ryzhikov, Frank Wolter, and Michael Zakharyaschev. Ontology-mediated query answering over temporal data: A survey (invited talk). In *TIME*, volume 90 of *LIPIcs*, pages 1:1–1:37. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[Baader *et al.*, 2013] F. Baader, S. Borgwardt, and M. Lippmann. Temporalizing ontology-based data access. In *Proc. of the 24th Int. Conf. on Automated Deduction, CADE-24*, volume 7898 of *LNCS*, pages 330–344. Springer, 2013.

[Baader *et al.*, 2017] Franz Baader, Stefan Borgwardt, Patrick Koopmann, Ana Ozaki, and Veronika Thost. Metric temporal description logics with interval-rigid names. In *Frontiers of Combining Systems - 11th International Symposium, FroCoS 2017, Brasília, Brazil, September 27-29, 2017, Proceedings*, pages 60–76, 2017.

[Baudinet *et al.*, 1993] Marianne Baudinet, Jan Chomicki, and Pierre Wolper. Temporal deductive databases. In *Temporal Databases*, pages 294–320. 1993.

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

[Bienvenu and Ortiz, 2015] Meghyn Bienvenu and Magdalena Ortiz. Ontology-mediated query answering with data-tractable description logics. In Wolfgang Faber and Adrian Paschke, editors, *Reasoning Web. Web Logic Rules - 11th International Summer School 2015, Berlin, Germany, July 31 - August 4, 2015, Tutorial Lectures*, volume 9203 of *Lecture Notes in Computer Science*, pages 218–307. Springer, 2015.

[Bienvenu *et al.*, 2014] Meghyn Bienvenu, Balder ten Cate, Carsten Lutz, and Frank 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] Patricia Bouyer, Uli Fahrenberg, Kim Guldstrand Larsen, Nicolas Markey, Joël Ouaknine, and James Worrell. Model checking real-time systems. In Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem, editors, *Handbook of Model Checking.*, pages 1001–1046. Springer, 2018.

[Brandt *et al.*, 2018] Sebastian Brandt, Elem Güzel Kalayci, Vladislav Ryzhikov, Guohui Xiao, and Michael Zakharyaschev. Querying log data with metric temporal logic. *J. Artif. Intell. Res.*, 62:829–877, 2018.

[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. Reasoning*, 39(3):385–429, 2007.

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

[Compton and Laflamme, 1990] Kevin J. Compton and Claude Laflamme. An algebra and a logic for nc^1. *Inf. Comput.*, 87(1/2):240–262, 1990.

[Gottlob *et al.*, 2014] Georg Gottlob, Giorgio Orsi, and Andreas Pieris. Query rewriting and optimization for ontological databases. *ACM Trans. Database Syst.*, 39(3):25:1–25:46, 2014.

[Grädel, 1991] Erich Grädel. On transitive closure logic. In Egon Börger, Gerhard Jäger, Hans Kleine Büning, and Michael M. Richter, editors, *Computer Science Logic, 5th Workshop, CSL '91, Berne, Switzerland, October 7-11, 1991, Proceedings*, volume 626 of *Lecture Notes in Computer Science*, pages 149–163. Springer, 1991.

[Gutiérrez-Basulto *et al.*, 2016a] Víctor Gutiérrez-Basulto, Jean Christoph Jung, and Roman Kontchakov. Temporalized EL ontologies for accessing temporal data: Complexity of atomic queries. In Subbarao Kambhampati, editor, *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 1102–1108. IJCAI/AAAI Press, 2016.

[Gutiérrez-Basulto *et al.*, 2016b] Víctor Gutiérrez-Basulto, Jean Christoph Jung, and Ana Ozaki. On metric temporal description logics. In Gal A. Kaminka, Maria Fox, Paolo Bouquet, Eyke Hüllermeier, Virginia Dignum, Frank Dignum, and Frank van Harmelen, editors, *ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016)*, volume 285 of *Frontiers in Artificial Intelligence and Applications*, pages 837–845. IOS Press, 2016.

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

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

[Lutz and Sabellek, 2017] Carsten Lutz and Leif Sabellek. Ontology-mediated querying with the description logic EL: trichotomy and linear datalog rewritability. In Carles Sierra, editor, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 1181–1187. ijcai.org, 2017.

[Lutz *et al.*, 2008] C. Lutz, F. Wolter, and M. Zakharyaschev. Temporal description logics: A survey. In *Proc. of the 15th Int. Symposium on Temporal Representation and Reasoning (TIME 2008)*, pages 3–14, 2008.

[Mehdi *et al.*, 2017] Gulnar Mehdi, Evgeny Kharlamov, Ognjen Savkovic, Guohui Xiao, Elem Guzel Kalayci, Sebastian Brandt, Ian Horrocks, Mikhail Roshchin, and Thomas A. Runkler. Semantic rule-based equipment diagnostics. In *The 16th International Semantic Web Conference, Part II*, volume 10588 of *Lecture Notes in Computer Science*, pages 314–333. Springer, 2017.

[Ouaknine and Worrell, 2007] Joël Ouaknine and James Worrell. On the decidability and complexity of metric temporal logic over finite words. *Logical Methods in Computer Science*, 3(1), 2007.

[Ouaknine and Worrell, 2008] Joël Ouaknine and James Worrell. Some recent results in metric temporal logic. In *Formal Modeling and Analysis of Timed Systems, 6th International Conference, FORMATS 2008, Saint Malo, France, September 15-17, 2008. Proceedings*, pages 1–13, 2008.

[Özçep and Möller, 2014] Özgür Lütfü Özçep and Ralf Möller. Ontology based data access on temporal and streaming data. In *the 10th Int. Summer School on Reasoning Web, RW 2014*, volume 8714 of *LNCS*, pages 279–312. Springer, 2014.

[Ronca *et al.*, 2018] Alessandro Ronca, Mark Kaminski, Bernardo Cuenca Grau, Boris Motik, and Ian Horrocks.

Stream reasoning in temporal datalog. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1941–1948. AAAI Press, 2018.

[Soylu *et al.*, 2017] Ahmet Soylu, Martin Giese, Rudolf Schlatte, Ernesto Jiménez-Ruiz, Evgeny Kharlamov, Özgür L. Özçep, Christian Neuenstadt, and Sebastian Brandt. Querying industrial stream-temporal data: An ontology-based visual approach. *JAISE*, 9(1):77–95, 2017.

[Wałęga *et al.*, 2019] P.A. Wałęga, Cuenca Grau Bernardo, and Mark Kaminski. Reasoning over streaming data in metric temporal datalog. In *AAAI*. AAAI Press, 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. of the 28th Int. Joint Conf. on Artificial Intelligence (IJCAI)*. IJCAI/AAAI, 2018.

	any	$\langle r, \infty)$	$[r, r]$	$\neq [r, r]$
MTL	$=$ coNP	FO($<$), in AC^0	FO(PRP), $=$ NC^1	FO(TC), $\leq$ NL, $\geq$ NC^1
hornMTL	datalog(FO), = P			FO(DTC), $\leq$ L, $\geq$ NC^1
coreMTL$^{\boxminus}$				
coreMTL$^{\diamondminus}$	FO(TC), $=$ NL	FO(<,+) when range uniform		

A $\mathsf{dist}_{=c}$ and Related Formulas

The formula $\mathsf{dist}_{=c}(y, x)$ is defined as follows. For $c = \infty$, we take the formula

$$\forall j\ (\mathsf{bit}_{in}(x, j, 1) \wedge \mathsf{bit}_{fr}(x, j, 1)),$$

whereas for a constant $c = h/2^k$, we can use

$$\forall j\ \Big(\big(\mathsf{bit}_{in}(x, j, 0) \wedge \mathsf{bit}_{in}^{+h/2^k}(y, j, 0)\big) \vee \big(\mathsf{bit}_{in}(x, j, 1) \wedge \mathsf{bit}_{in}^{+h/2^k}(y, j, 1)\big)\Big) \wedge$$
$$\forall j\ \Big(\big(\mathsf{bit}_{fr}(x, j, 0) \wedge \mathsf{bit}_{fr}^{+h/2^k}(y, j, 0)\big) \vee \big(\mathsf{bit}_{fr}(x, j, 1) \wedge \mathsf{bit}_{fr}^{+h/2^k}(y, j, 1)\big)\Big),$$

where predicates $\mathsf{bit}_{in}^{+h/2^k}(y, j, v)$, saying that v is the j-th bit of the integer part of $y + h/2^k$, and $\mathsf{bit}_{fr}^{+h/2^k}(y, j, v)$, saying that v is the j-th bit of the fractional part of $y + h/2^k$, are defined inductively as follows:

$$\begin{aligned}
\mathsf{bit}_{fr}^{+0/2^k}(y, j, v) &= \mathsf{bit}_{fr}(y, j, v),\\
\mathsf{bit}_{fr}^{+(d+1/2^k)}(y, j, v) &= \exists u\Big((u = \ell - k) \wedge \Big(\big((j \leq u) \wedge \mathsf{bit}_{fr}^{+d}(y, j, v)\big) \vee\\
&\qquad \big((v = 0) \wedge \mathsf{bit}_{fr}^{+d}(y, j, 0) \wedge \exists j'((u < j' < j) \wedge \mathsf{bit}_{fr}^{+d}(y, j', 0))\big) \vee\\
&\qquad \big((v = 0) \wedge \mathsf{bit}_{fr}^{+d}(y, j, 1) \wedge \forall j'((u < j' < j) \to \mathsf{bit}_{fr}^{+d}(y, j', 1))\big) \vee\\
&\qquad \big((v = 1) \wedge \mathsf{bit}_{fr}^{+d}(y, j, 1) \wedge \exists j'((u < j' < j) \wedge \mathsf{bit}_{fr}^{+d}(y, j', 0))\big) \vee\\
&\qquad \big((v = 1) \wedge \mathsf{bit}_{fr}^{+d}(y, j, 0) \wedge \forall j'((u < j' < j) \to \mathsf{bit}_{fr}^{+d}(y, j', 1))\big)\Big)\Big),
\end{aligned}$$

$$\begin{aligned}
\mathsf{bit}_{in}^{+0/2^k}(y, j, v) &= \mathsf{bit}_{in}(y, j, v),\\
\mathsf{bit}_{in}^{+(d+1/2^k)}(y, j, v) &= \exists u\Big((u = \ell - k) \wedge \Big(\\
&\qquad \big((v = 0) \wedge \mathsf{bit}_{in}^{+d}(y, j, 0) \wedge \exists j'(((j' < j) \wedge \mathsf{bit}_{in}^{+d}(y, j', 0)) \vee\\
&\qquad\qquad ((u < j' < j) \wedge \mathsf{bit}_{fr}^{+d}(y, j', 0)))\big) \vee\\
&\qquad \big((v = 0) \wedge \mathsf{bit}_{in}^{+d}(y, j, 1) \wedge \forall j'(((j' < j) \to \mathsf{bit}_{in}^{+d}(y, j', 1)) \wedge\\
&\qquad\qquad (u < j' < j) \to \mathsf{bit}_{fr}^{+d}(y, j', 1))\big) \vee\\
&\qquad \big((v = 1) \wedge \mathsf{bit}_{in}^{+d}(y, j, 0) \wedge \exists j'(((j' < j) \wedge \mathsf{bit}_{in}^{+d}(y, j', 0)) \vee\\
&\qquad\qquad ((u < j' < j) \wedge \mathsf{bit}_{fr}^{+d}(y, j', 0)))\big) \vee\\
&\qquad \big((v = 1) \wedge \mathsf{bit}_{in}^{+d}(y, j, 1) \wedge \forall j'(((j' < j) \to \mathsf{bit}_{in}^{+d}(y, j', 1)) \wedge\\
&\qquad\qquad ((u < j' < j) \to \mathsf{bit}_{fr}^{+d}(y, j', 1)))\big)\Big)\Big).
\end{aligned}$$

Here, ℓ is the last element of the domain Δ and $u = \ell - k$ can be easily defined using $<$ and k.

The formulas $\mathsf{dist}_{<r}(x, y)$, $\mathsf{dist}_{>r}(x, y)$ are defined by the straightforward modifications of $\mathsf{dist}_{=r}$. Using these, we can further define FO-formulas $\mathsf{in}_{\varrho}(x, y)$ and $\mathsf{int}_{\varrho}(t, u, s)$.

B Divisibility and div_c

In this we section, we will investigate how to define the formula $\mathsf{div}_d(x, y)$ which is true iff the difference between $\bar{x}$ and $\bar{y}$ is divisible by d. The formula will be in $FO(RPR)$. First we need to define its FO subformula $\mathsf{dif}_{fr}(i, x, y)$ (respectively, $\mathsf{dif}_{in}(i, x, y)$) such that it is true iff i-th bit of the fractional (integral) part of $x - y$ is 1. We use the column method. Let the fractional part of x (resp. y) be $x_\ell \dots x_0$ (resp. $y_\ell \dots y_0$). The we define b_i, which indicates that from the bit i it was *borrowed* as:

$$b_i = (\neg x_{i-1} \wedge y_{i-1}) \vee (b_{i-1} \wedge (\neg x_{i-1} \vee y_{i-1})).$$

Let $x - y$ be $z_\ell \dots z_0$, which can be defined as:

$$z_i = (b_i \wedge \neg x_i \wedge \neg y_i) \vee (\neg b_i \wedge (x_i \leftrightarrow \neg y_i)).$$

Therefore, we can define the FO formula $b_{fr}(x, y, i)$ which is true iff the substraction from x of y in the fractional parts borrows from the bit i as:

$$\exists j((j < i) \wedge (bit_{fr}(j, x, 0) \wedge bit_{fr}(j, y, 1)) \wedge \forall k((j < k < i) \rightarrow (bit_{fr}(k, x, 0) \vee bit_{fr}(k, y, 1)))),$$

while $b_{in}(x, y, i)$ with the analogous meaning for the integral parts of x and y can be defined as:

$$\big(\exists j((j < i) \wedge (bit_{in}(j, x, 0) \wedge bit_{in}(j, y, 1)) \wedge \forall k((j < k < i) \rightarrow (bit_{in}(k, x, 0) \vee bit_{in}(k, y, 1))))\big) \vee \big(b_{fr}(\ell, x, y,) \wedge ((bit_{fr}(\ell, x, 0) \vee bit_{fr}(\ell, y, 1))) \wedge \forall k((k < i) \rightarrow (bit_{in}(k, x, 0) \vee bit_{in}(k, y, 1)))\big),$$

where ℓ is (a constant for) the last element of the FO-structure domain Δ. Finally, we define $\mathsf{dif}_{fr}(i, x, y)$ as

$$(b_{fr}(x, y, i) \wedge bit_{fr}(i, x, 0) \wedge bit_{fr}(i, y, 0)) \vee (\neg b_{fr}(x, y, i) \wedge (bit_{fr}(i, x, 1) \leftrightarrow bit_{fr}(i, y, 0))),$$

and analogously for $\mathsf{dif}_{in}(i, x, y)$.

To define the required div_d we will make use of the integer divisibility automaton $\mathcal{A}_k = (Q, \{0, 1\}, q_0, q_a, \delta)$, which takes as input a(n inverted) binary representation $z_0 z_1 \dots z_n$ of a number z and reaches its accepting state q_a on it iff z is divisible by k. It is known that such an automaton exists for each integer k. In what follows let $z_\ell z_{\ell-1} \dots z_0$ for $z_i \in \{0, 1\}$ be the sequence such that $z_i = 1$ iff $\mathsf{dif}_{fr}(i, x, y)$ is true. Clearly, this sequence is the binary representation of the fractional part of $x - y$. We analogously define $w_\ell w_{\ell-1} \dots w_0$ through $\mathsf{dif}_{in}(i, x, y)$ which represents the integral part of $x - y$.

Consider an FO formula $reach^n_{q,\mathcal{A}_k}(x, y)$ which is true iff either

- $\ell \geq n$, there exists a run (from q_0) of $\mathcal{A}_k$ on $z_{\ell-n} \dots z_{\ell-1} z_\ell$ resulting in q, and $z_i = 0$ for $i < \ell - n$, or
- $\ell < n$ and there exists a run of $\mathcal{A}_k$ on
$$\underbrace{0 \dots 0}_{n-\ell} z_0 \dots z_\ell.$$

To construct $reach^n_{q,\mathcal{A}_k}$ one needs to consider paths of length bounded by n in $\mathcal{A}_k$, which are finitely many, and therefore the formula is constructable in FO (we leave details to the reader). Let f_d be the number of significant bits in the binary representation of d (e.g., $f_d = 3$ for $d = 10001.101$). We can now prove the following:

Lemma 10. *Let $D = d2^{f_d}$ and $\mathcal{A}_D = (Q, \{0, 1\}, q_0, q_a, \delta)$ the divisibility automaton for D. Then $x - y$ is divisible by d iff there is $q \in Q$ such that $reach^{f_d}_{q,\mathcal{A}_D}(x, y)$ and there is an accepting run of $\mathcal{A}_D$ from q on $w_0 w_1 \dots w_\ell$.*

We will write $q' \rightarrow_w q$ for $w \in \{0, 1\}$ if there is a w transition from q' to q in $\mathcal{A}_D$. To encode the condition of the lemma above for checking the divisibility of $x - y$, we are going to construct an FO(RPR) formula defining the relations R_q for each $q \in Q$. Let α_q be:

$$\begin{aligned} R_q(i, x, y) \equiv & \Big((i = 0) \wedge \mathsf{dif}_{in}(0, x, y) \wedge \bigvee_{q' \rightarrow_1 q} (reach^{f_d}_{q',\mathcal{A}_D}(x, y))\Big) \vee \\ & \Big((i = 0) \wedge \neg\mathsf{dif}_{in}(0, x, y) \wedge \bigvee_{q' \rightarrow_0 q} (reach^{f_d}_{q',\mathcal{A}_D}(x, y))\Big) \vee \\ & \Big(\mathsf{dif}_{in}(i, x, y) \wedge \bigvee_{q' \rightarrow_1 q} R_{q'}(i - 1, x, y)\Big) \vee \\ & \Big(\neg\mathsf{dif}_{in}(i, x, y) \wedge \bigvee_{q' \rightarrow_0 q} R_{q'}(i - 1, x, y)\Big), \end{aligned}$$

which intuitively says that R_q will be true on bit i if either i is 0 and we can transition in $\mathcal{A}_D$ from some q' satisfying $reach$ by reading w_0, or it is known that $R_{q'}$ will be true on bit $i - 1$ for some q' and we can transition from q' to q by reading w_i. The required $\mathsf{div}_d(x, y)$ can finally be defined as:

$$\begin{bmatrix} \alpha_{q_0} \\ \dots \\ \alpha_{q_{|Q|}} \end{bmatrix} R_{q_a}(\ell, x, y),$$

which is an FO(RPR) formula using simultaneous recursion that, intuitively, checks whether the accepting state is reached after reading $w_0 w_1 \dots w_\ell$.