On the Data Complexity of Ontology-Mediated Queries with MTL Operators over Timed Words

S. Kikot[1], V. Ryzhikov[2], P. A. Wałęga[1,3], and M. Zakharyaschev[2]

[1] University of Oxford, U.K.
[2] Birkbeck, University of London, U.K.
[3] University of Warsaw, Poland

Abstract. We report on our initial results regarding the data complexity of answering atomic queries mediated by ontologies given in a Horn fragment of the metric temporal logic *MTL*. We adopt the pointwise semantics for *MTL* over dense time. The complexity classes involved are AC^0, NC^1, L, NL, and P.

1 Introduction

Our general concern is detecting events in a complex asynchronous system based on qualitative sensor measurements. More precisely, the system in question has a number of sensors that from time to time and asynchronously send their readings to a database. We may think of such readings as pairs of the form (A, t), where A is a concept name and t a *timestamp* given as a non-negative finite binary fraction such as 101.011. As a formalism to define events we use propositional Horn clauses with operators of the metric temporal logic *MTL* [14,2]. For example, according to Siemens, a 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 *hornMTL* rule, where $\diamondminus_{(n,m]}\varphi$ is assumed to hold true at a timestamp t if φ holds at some previous timestamp t' with $n < t - t' \leq m$:

$$\mathsf{NormalStop} \leftarrow \mathsf{CoastDown1500to200} \wedge \diamondminus_{(0,9m]}\big[\mathsf{CoastDown6600to1500} \wedge \diamondminus_{(0,2m]}\big(\mathsf{MainFlameOff} \wedge \diamondminus_{(0,2m]}\mathsf{ActivePowerOff}\big)\big]$$

(the concepts on the right-hand side can be defined by similar rules).

The use of *hornMTL* and *datalogMTL* ontologies (with both diamond and box operators in rule bodies and only box operators in rule heads) for querying temporal log data was advocated in [9], which also demonstrated experimentally feasibility and efficiency of ontology-based data access (OBDA) with nonrecursive *datalogMTL* queries. An extension of the OBDA platform Ontop to support such queries was suggested in [8]. For a recent survey of temporal OBDA we refer the reader to [5]; surveys of early developments in temporal deductive databases

CEUR-WS.org/Vol-2211/paper-21.pdf

are given in [7,10]. Note also that the satisfiability problem for the description logic $\mathcal{ALC}$ extended with *MTL* operators over $(\mathbb{N}, \leq)$ was considered in [12,6].

In this paper, we start investigating the data complexity of answering ontology-mediated queries (OMQs) with *MTL* operators. As the problem turns out to be quite involved, we restrict ourselves to atomic queries and ontologies in the fragment *hornMTL*$^-$ of *hornMTL* where only past diamond operators are allowed in rule bodies and no temporal operators can be used in rule heads. We distinguish between arbitrary, linear and core (or binary) rules and consider, in particular, the following types of the range ϱ in $\diamondminus_\varrho$: $\langle r, \infty)$, $\langle 0, r\rangle$, and $[r, r]$ where $\langle$ is one of [or (, while $\rangle$ is one of] or). The underlying timeline is $(\mathbb{R}, \leq)$ as we cannot assume that sensor readings come at regular time intervals. There are two standard semantics for *MTL* over $(\mathbb{R}, \leq)$: *continuous* and *pointwise*; see, e.g., [16] and references therein. Here, we only consider the latter, under which both model checking and satisfiability for full *MTL* are decidable but not primitive recursive [15] (over $(\mathbb{Z}, \leq)$, these problems are ExpSpace-complete). As pointed out in [16], under the pointwise semantics, one thinks of atomic propositions in *MTL* as referring to events (corresponding to state changes) rather than to states themselves.

The complexity results we obtain in this paper are collected in the table below, where 'range-uniform' means that all the $\diamondminus_\varrho$ operators before intensional concept names in a given *hornMTL*$^-$ program have the same range ϱ:

rules \ ranges	any	$\langle r, \infty)$	$\langle 0, r\rangle$	$[r, r]$
Horn	$= \mathrm{P}$		$\leq \mathrm{L}$	$\leq \mathrm{L}$
linear	$= \mathrm{NL}$	in AC^0	$\geq \mathrm{NC}^1$	$\geq \mathrm{NC}^1$
core	$\leq \mathrm{NL}$		in AC^0(range-uniform)	$\leq \mathrm{L}$

2 The Horn Fragment of *MTL*

We denote the set of finite binary fractions—aka dyadic rational numbers—by $\mathbb{Q}_2$; the set of non-negative dyadic rationals is denoted by $\mathbb{Q}_2^{\geq 0}$. As well-known, $\mathbb{Q}_2$ is dense in $\mathbb{R}$ and, by Cantor's theorem, $(\mathbb{Q}_2, <)$ is isomorphic to $(\mathbb{Q}, <)$.

By an *interval*, ι, we mean any nonempty subset of the real numbers $\mathbb{R}$ of the form $[t_1, t_2]$, $[t_1, t_2)$, $(t_1, t_2]$ or (t_1, t_2), where $t_1, t_2 \in \mathbb{Q}_2 \cup \{-\infty, \infty\}$ and $t_1 \leq t_2$, excluding $t_1 = t_2 \in \{-\infty, \infty\}$. We identify $(t, \infty]$ with (t, ∞), $[-\infty, t]$ with $(-\infty, t]$, etc. A *range*, ϱ, is an interval with non-negative endpoints.

The *metric temporal logic MTL* [14,2] is a propositional modal logic with box operators indexed by ranges, say $\boxminus_{(0,60]}$, which is interpreted over $(\mathbb{R}, <)$ as 'at every time instant within the previous minute', its future counterpart $\boxplus_{(0,60]}$, and their dual diamond operators $\diamondminus_{(0,60]}$ and $\diamondplus_{(0,60]}$. In this paper, we only consider a fragment of *MTL* that is called *hornMTL*$^-$.

A *hornMTL*$^-$ *program*, $\mathit{\Pi}$, is a finite set of *rules* of the form

$$A \leftarrow B_1 \land \cdots \land B_k, \tag{1}$$

where $k \geq 1$ and the B_i are defined by the grammar

$$B \ ::= \ A \quad | \quad \top \ | \ \ \diamondminus_\varrho B,$$

with A being a concept name or $\bot$. As usual, A is called the *head* of the rule, and $B_1 \wedge \ldots \wedge B_k$ its *body*. A *hornMTL*$^-$ program Π is *linear* if, in each of its rules, at most one of the concepts in the body occurs as a head in Π. A *hornMTL*$^-$ program is *core* if all of its rules are of the form $A \leftarrow B$ or $\bot \leftarrow B_1 \wedge B_2$. A *hornMTL*$^-$ (ontology-mediated) *query* takes the form $(\Pi, A(x))$.

As mentioned in Section 1, we may think of a *data instance* as a finite set $\mathcal{D} = \{(A_1, t_1), \ldots, (A_n, t_n)\}$, where the A_i are concept names from some fixed alphabet Λ and the t_i are (not necessarily ordered) *timestamps* from $\mathbb{Q}_2^{\geq 0}$. When proving some complexity results, we assume that $\mathcal{D}$ is given as the FO-structure

$$\mathfrak{D} \;=\; (\Delta, <, \Omega, \mathtt{bit}_{in}, \mathtt{bit}_{fr}, A_1, \ldots, A_p), \tag{2}$$

in which

- $\Delta = \{0, \ldots, \ell\} \subseteq \mathbb{N}$, where ℓ is the maximum of the number of distinct timestamps in $\mathcal{D}$ and the number of bits in the longest binary fraction in $\mathcal{D}$ (excluding the binary point), and $<$ is the usual order on Δ;
- $\Omega \subseteq \Delta$ is a set of *timestamps*; for every $n \in \Omega$, we set $\bar{n} = b_\ell \cdots b_0.c_0 \cdots c_\ell$ such that $\mathsf{bit}_{in}(i, n, b_i)$ and $\mathsf{bit}_{fr}(i, n, c_i)$ hold, for $i = 0, \ldots, \ell$;
- thus, bit_{in} and bit_{fr} are ternary relations on Δ such that, for any $n \in \Omega$ and $i \in \Delta$, 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 $\mathsf{bit}_{fr}(i, n, c_i)$;
- $A_i \subseteq \Omega$, for $i = 1, \ldots, p$; intuitively, $A_i(n)$ holds iff $A_i(\bar{n}) \in \mathcal{D}$.

For any $d \in \mathbb{Q}_2^{\geq 0}$, one can readily define FO-formulas:

- $\mathsf{dist}_{<d}(x, y)$ that holds in $\mathfrak{D}$ iff $x, y \in \Omega$ and $0 \leq \bar{x} - \bar{y} < d$;
- $\mathsf{dist}_{>d}(x, y)$ that holds in $\mathfrak{D}$ iff $x, y \in \Omega$ and $\bar{x} - \bar{y} > d$;
- their modifications $\mathsf{dist}_{\leq d}(x, y)$ and $\mathsf{dist}_{\geq d}(x, y)$.

It will be convenient to assume that these predicates are also given by $\mathfrak{D}$ for the relevant d. In Theorem 5, we also make the following assumption:

(ord) $\Omega = \{0, \ldots, k\}$, for some $k \leq \ell$, and $n < m \leq k$ implies $\bar{n} < \bar{m}$.

There are two semantics for *MTL* known as *pointwise* and *continuous* [16].

Pointwise semantics. A *pointwise interpretation* is a structure of the form

$$\mathcal{I} = (\mathbb{T}, A_1^{\mathcal{I}}, A_2^{\mathcal{I}}, \ldots),$$

where $\mathbb{T} \neq \emptyset$ is a finite subset of $\mathbb{Q}_2^{\geq 0}$ (timestamps) and $A_i^{\mathcal{I}} \subseteq \mathbb{T}$. We set $B^{\mathcal{I}} = A^{\mathcal{I}}$ if $B = A$, $B^{\mathcal{I}} = \mathbb{T}$ if $B = \top$, $B^{\mathcal{I}} = \emptyset$ if $B = \bot$, and

$$(\diamondminus_\varrho B)^{\mathcal{I}} = \{t \in \mathbb{T} \mid \exists t' \in B^{\mathcal{I}}\, (t - t' \in \varrho)\}.$$

$\mathcal{I}$ is a *model* of a data instance $\mathcal{D}$ if $t \in A^{\mathcal{I}}$ for any $(A, t) \in \mathcal{D}$; $\mathcal{I}$ is a *model* of a *hornMTL*$^-$ program Π if, for any rule (1) in Π and any $t \in \mathbb{T}$, we have $t \in A^{\mathcal{I}}$ whenever $t \in B_i^{\mathcal{I}}$ for $1 \leq i \leq k$. We say that $\mathcal{D}$ and Π are *consistent* if there is a model of $\mathcal{D}$ and Π. We also say that a timestamp t from $\mathcal{D}$ is a *certain* *answer* to a query $(\Pi, A(x))$ over $\mathcal{D}$ if $t \in A^{\mathcal{I}}$, for every model $\mathcal{I}$ of $\mathcal{D}$ and Π. The *query answering problem* for $(\Pi, A(x))$ is to decide, given a data instance $\mathcal{D}$ and a timestamp t in it, whether t is a certain answer to $(\Pi, A(x))$ over $\mathcal{D}$.

Continuous semantics. The only difference from the pointwise case is that a *continuous interpretation* is defined over the reals: $\mathcal{I} = (\mathbb{R}, A_1^{\mathcal{I}}, A_2^{\mathcal{I}}, \dots)$ with $A_i^{\mathcal{I}} \subseteq \mathbb{R}$ and $(\diamondminus_\varrho B)^{\mathcal{I}} = \{t \in \mathbb{R} \mid \exists t' \in B^{\mathcal{I}}\, (t - t' \in \varrho)\}$. To illustrate, suppose that $\Pi = \{A \leftarrow \diamondminus_{(0,1]}\diamondminus_{(0,1]}B\}$ and $\mathcal{D} = \{(B, 0), (C, 2)\}$. Then (Π, A) has no answers over $\mathcal{D}$ under the former semantics (because 1 is not a timestamp in $\mathcal{D}$), but 2 is an answer under the latter one.

In this paper, we only consider the pointwise semantics and leave the continuous one for future work.

Normal form. A program is in *normal form* if its rules have one of the forms:

$$P_0 \leftarrow \diamondminus_{\varrho_1'} P_1' \wedge \dots \wedge \diamondminus_{\varrho_\ell'} P_m', \tag{3}$$

$$P_0 \leftarrow \diamondminus_{\varrho_1} P_1 \wedge \dots \wedge \diamondminus_{\varrho_k} P_k \wedge \diamondminus_{\varrho_1'} P_1' \wedge \dots \wedge \diamondminus_{\varrho_\ell'} P_m', \tag{4}$$

where the P_i' are from the data alphabet Λ, the P_i do not belong to Λ (and so cannot occur in data instances), and $0 \notin \varrho_i$ for any i (although there may be, say $\varrho_i' = [0, 0]$). Every *hornMTL*$^-$ program can be transformed to a program in normal form with the same answers. We illustrate this claim by an example.

Example 1. Let $\Pi = \{P_1' \leftarrow \diamondminus_{[0,d]} P_0' \wedge Q_0',\ P_0' \leftarrow \diamondminus_{(0,e)} P_1' \wedge \diamondminus_{[0,f]} Q_1'\}$, where the P_i' are in Λ. By introducing fresh concept names P_0, P_1, we convert Π to

$$P_1 \leftarrow \diamondminus_{[0,d]} P_0 \wedge Q_0',\ P_0 \leftarrow \diamondminus_{(0,e)} P_1 \wedge \diamondminus_{[0,f]} Q_1',\ P_0 \leftarrow P_0',\ P_1 \leftarrow P_1'.$$

To get rid of $[0, d]$, we further transform the program to

$$P_1 \leftarrow P_0 \wedge Q_0',\ P_1 \leftarrow \diamondminus_{(0,d]} P_0 \wedge Q_0',\ P_0 \leftarrow \diamondminus_{(0,e)} P_1 \wedge \diamondminus_{[0,f]} Q_1',\ P_0 \leftarrow P_0',\ P_1 \leftarrow P_1'.$$

Now, P_0 in the first rule is not in the scope of $\diamondminus_\varrho$ (Q_0' can be regarded as a shorthand for $\diamondminus_{[0,0]} Q_0'$). So we transform the rule using obvious derivations to obtain the following program in normal form:

$$P_1 \leftarrow P_0' \wedge Q_0',\ P_1 \leftarrow \diamondminus_{(0,e)} P_1 \wedge \diamondminus_{[0,f]} Q_1' \wedge Q_0',\ P_1 \leftarrow \diamondminus_{(0,d]} P_0 \wedge Q_0',$$
$$P_0 \leftarrow \diamondminus_{(0,e)} P_1 \wedge \diamondminus_{[0,f]} Q_1',\ P_0 \leftarrow P_0',\ P_1 \leftarrow P_1'.$$

A *hornMTL*$^-$ query $(\Pi, A(x))$ is in *normal form* if Π is in normal form and $A \notin \Lambda$. Clearly, every query can be converted to a one in normal form and having the same answers.

3 The Data Complexity of Answering *hornMTL*$^-$ Queries

It is not hard to see that consistency checking and answering *hornMTL*$^-$ queries can be reduced to consistency checking and answering monadic datalog queries over $\mathfrak{D}$. For example, the rule $A \leftarrow \diamondminus_{(r,s]} B$ can be replaced with the monadic datalog rule

$$A(x) \leftarrow B(y) \wedge \mathsf{dist}_{>r}(x, y) \wedge \mathsf{dist}_{\leq s}(x, y),$$

where $\mathsf{dist}_{>r}$ and $\mathsf{dist}_{\leq s}$ are extensional predicates given by the data instance. It follows that consistency checking and answering *hornMTL*$^-$ queries can be done in polynomial time for data complexity; for linear *hornMTL*$^-$ queries, this can be done in NL. The next theorem establishes a matching lower bound, which is in sharp contrast to *hornLTL* queries that are in NC^1 for data complexity [4].

Theorem 1. (i) *Consistency checking and answering hornMTL*$^-$ *queries is* P-*complete for data complexity.*

(ii) *Consistency checking and answering linear hornMTL*$^-$ *queries is* NL-*complete for data complexity.*

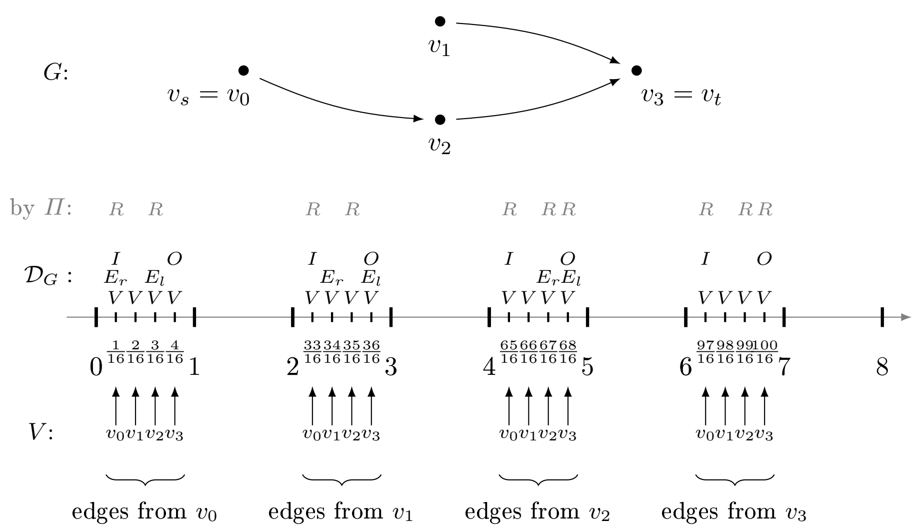
Proof. We only show the lower bound in (ii); the construction in (i) is similar.

The proof is by reduction of the reachability problem in acyclic digraphs. Let $G = (V, E)$ be such a digraph with a set $V = \{v_0, \dots, v_{n-1}\}$ of vertices and a set $E \subseteq V \times V$ of edges such that whenever $(v_i, v_j) \in E$ then $i < j$ (we can always represent G in this way due to its acyclicity). Suppose that the task is to check whether $v_t \in V$ is reachable from $v_s \in V$ in G.

We construct a data instance $\mathcal{D}_G$ with concepts V, E_r, E_l, I, O, which encodes G on a linear order. Namely, $\mathcal{D}_G$ contains the following pairs ($i, j, k \in \mathbb{N}$):

- $(V, 2k + \frac{i}{2^n})$, for $0 \leq k < n$ and $1 \leq i \leq n$;
- $(E_r, 2i + \frac{i+1}{2^n})$, for $(v_i, v_j) \in E$;
- $(E_l, 2i + \frac{j+1}{2^n})$, for $(v_i, v_j) \in E$;
- $(I, 2k + \frac{i+1}{2^n})$, for $0 \leq k < n$ and $v_s = v_i$;
- $(O, 2k + \frac{i+1}{2^n})$, for $0 \leq k < n$ and $v_t = v_i$.

An example of a graph and the corresponding data instance are shown below:



Let Π be a linear *hornMTL*$^-$ program with the following rules:

$$R \leftarrow I, \quad R \leftarrow E_l \wedge \diamondminus_{[0,1]}(E_r \wedge R), \quad R \leftarrow V \wedge \diamondminus_{[2,2]} R, \quad P \leftarrow R \wedge O, \quad \bot \leftarrow P.$$

Note that all numbers occurring in $\mathcal{D}_G$ and Π belong to $\mathbb{Q}_2$. For the atoms implied by Π, see the previous picture. One can show that Π and $\mathcal{D}_G$ are consistent iff v_t is not reachable from v_s.

4 *hornMTL*$^-$ Queries with $\diamondminus_{\langle r,\infty)}$

In this section, we show that if all temporal operators in a *hornMTL*$^-$ program Π are of the form $\diamondminus_{\langle r,\infty)}$, then answering $(\Pi, A(x))$ can be done in AC^0; in other words, $(\Pi, A(x))$ is FO-rewritable. To this end, we require the canonical models for *hornMTL*$^-$ programs, which can be defined as follows.

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

(horn) if $A \leftarrow B_1 \wedge \ldots \wedge B_k$ is in Π and $(B_i, t) \in \mathcal{C}$, for $i = 1, \ldots, k$, then we add (A, t) to $\mathcal{C}$;
($\diamondminus_\varrho$) if $\diamondminus_\varrho B$ occurs in Π, $(B, t') \in \mathcal{C}$, and $t - t' \in \varrho$, for a timestamp t in $\mathcal{D}$, then we add $(\diamondminus_\varrho B, t)$ to $\mathcal{C}$.

It should be clear that there is some $N < \omega$ (polynomially depending on Π and $\mathcal{D}$) such that $\mathsf{cl}^N(\mathcal{C}) = \mathsf{cl}^{N+1}(\mathcal{C})$. We then set $\mathfrak{C}_{\Pi,\mathcal{D}} = \mathsf{cl}^N(\mathcal{D})$.

Theorem 2. *A timestamp t from $\mathcal{D}$ is a certain answer to $(\Pi, A(x))$ over $\mathcal{D}$ iff $(A, t) \in \mathfrak{C}_{\Pi,\mathcal{D}}$.*

As known from [3], if we use *LTL* diamonds in place of the *MTL* ones in *hornMTL*$^-$ programs, then all such queries are FO-rewritable and in AC^0 for data complexity. In fact, almost the same argument shows the following:

Theorem 3. *Consistency checking and answering hornMTL$^-$ queries with temporal operators of the form $\diamondminus_{\langle r,\infty)}$ are in AC^0 for data complexity.*

Proof. Let $(\Pi, A(x))$ be a *hornMTL*$^-$ query with temporal operators of the form $\diamondminus_{\langle r,\infty)}$. It is easy to see that constructing $\mathfrak{C}^p_{\Pi,\mathcal{D}}$ needs at most $|\Pi|^2$ applications of the cl operator to $\mathcal{D}$ (because each rule ($\diamondminus_{\langle r,\infty)}$) needs to be applied at most once). Thus, we can construct an FO-rewriting of $(\Pi, A(x))$ using iteratively the standard FO-translation of, say, $\diamondminus_{[r,\infty)} B$ into $\exists t' \, (\mathsf{dist}_{\geq r}(t, t') \wedge B(t'))$.

Example 2. An FO-rewriting for the query $(\Pi, A(x))$ with the program Π comprising two rules $A \leftarrow \diamondminus_{[2,\infty)} C$, $C \leftarrow \diamondminus_{[1,\infty)} A$ looks as follows:

$$\exists y \left[A(x) \vee \big(C(y) \wedge \mathsf{dist}_{\geq 2}(x, y) \big) \vee \big(A(y) \wedge \mathsf{dist}_{\geq 3}(x, y) \big) \right].$$

When the ranges ϱ in $\diamondminus_\varrho$ are different from $\langle r, \infty)$, the technique above does not work any more and the complexity landscape changes significantly.

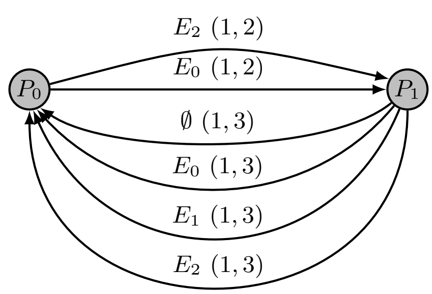
5 Metric Automata for Linear *hornMTL*$^-$

Our technical tool for studying the data complexity of linear *hornMTL*$^-$ queries is automata with metric constraints that are defined for programs in normal form. These automata can be viewed as a primitive version of standard timed automata for *MTL* [1] as we only have one clock c, the clock reset $c := 0$ happens at every transition, and the clock constraints are of the simple form $c \in \varrho$.

A (nondeterministic) *metric automaton* is a quadruple $\mathcal{A} = (S, S_0, \Sigma, \delta)$, where $S \neq \emptyset$ is a set of *states*, Σ a *tape alphabet*, δ a *transition relation*, and S_0 is a nonempty set of pairs of the form (q, e), where $e \in \Sigma$, $q \in S$. The transition relation δ is a set of instructions of the form $q \xrightarrow{\varrho}_e q'$ with $q, q' \in S$, $e \in \Sigma$ and a range ϱ. The automaton $\mathcal{A}$ takes as input *timed words* $\sigma = (e_0, t_0), \dots, (e_n, t_n)$, where the t_i are timestamps with $t_{i-1} < t_i$. A *run* over σ is a sequence $q_0, \dots, q_m$ such that $(q_0, e_0) \in S_0$, $q_{i-1} \xrightarrow{\varrho_i}_{e_i} q_i$ is in δ and $t_i - t_{i-1} \in \varrho_i$, for $0 < i \leq n$.

Let Π be a *linear hornMTL*$^-$ program in normal form. We denote the conjunctions $\diamondminus_{\varrho'_1} P'_1 \land \dots \land \diamondminus_{\varrho'_\ell} P'_m$ (with data concept names P'_i) that occur in Π by ε, possibly with subscripts. Thus, since Π is linear, rules (4) in Π are of the form $P \leftarrow \varepsilon \land \diamondminus_\varrho Q$. Let $E_\Pi = \{\varepsilon_1, \dots, \varepsilon_q\}$ be the set of all such ε occurring in Π. We define a metric automaton $\mathcal{A}_\Pi$ for Π as follows. The set S of its states comprises the head concept names in Π, and $\Sigma = 2^{E_\Pi}$. The transition relation δ comprises $Q \xrightarrow{\varrho}_E P$ such that $P \leftarrow \varepsilon \land \diamondminus_\varrho Q$ is in Π and $\varepsilon \in E$. Finally, S_0 is the set of all pairs (P, ε) such that a rule $P \leftarrow \varepsilon$ of the form (3) is in Π.

Example 3. For $\Pi = \{P_0 \leftarrow \diamondminus_{[0,1]} P'_0,\ P_1 \leftarrow \diamondminus_{(1,2)} P_0 \land P'_1,\ P_0 \leftarrow \diamondminus_{(1,3)} P_1\}$, the metric automaton $\mathcal{A}_\Pi$ is depicted below, where $P'_0, P'_1 \in \Lambda$, $E_0 = \{P'_1\}$, $E_1 = \{\diamondminus_{[0,1]} P'_0\}$, $E_2 = \{P'_1, \diamondminus_{[0,1]} P'_0\}$, and $S_0 = \{(P_0, \diamondminus_{[0,1]} P'_0)\}$.



We represent any data instance $\mathcal{D}$ as a timed word $\sigma_\mathcal{D}$. For t_i occurring in $\mathcal{D}$, let $E(t_i)$ be the maximal set of ε from Π that hold at t_i in $\mathcal{D}$, and let $\sigma_\mathcal{D} = \big((E(t_1), t_1), \dots, (E(t_n), t_n)\big)$. The corresponding FO-structure from Section 2 can take the form

$$\sigma_\mathfrak{D} \;=\; (\Delta, <, \Omega, \mathtt{bit}_{in}, \mathtt{bit}_{fr}, E_1, \dots, E_k), \tag{5}$$

where $\{E_1, \dots, E_k\} = 2^{E_\Pi}$. It is not hard to see that there is an FO-translation of (2) to (5).

Example 4. A data instance $\mathcal{D}$ and its representation as $\sigma_{\mathcal{D}}$ are shown below:

	0	1	1.5	4	4.5	5	6.5
$\mathcal{D}:$	P'_0	Q'	P'_1	P'_0	P'_1	P'_1	Q'
$\sigma_{\mathcal{D}}:$	$\emptyset$	E_1	E_0	$\emptyset$	E_2	E_2	$\emptyset$

Theorem 4. *For any linear hornMTL⁻ query $(\Pi, A(x))$, a timestamp t_i is a certain answer over a data instance $\mathcal{D}$ iff there exist a subword $\sigma'_{\mathcal{D}}$ of $\sigma_{\mathcal{D}}$ with the last timestamp t_i and a run of $\mathcal{A}_\Pi$ over $\sigma'_{\mathcal{D}}$ that ends with A.*

Example 5. Let $(\Pi, P_1(x))$ be the query with Π from Example 3. Then, for $\sigma_{\mathcal{D}}$ from Example 4, we have the run P_0, P_1, P_0, P_1 on

$$(E_1, 1), (E_0, \frac{3}{2}), (\emptyset, 4), (E_2, 5),$$

and so 5 is a certain answer to the query over $\mathcal{D}$ from Example 4.

One could define metric automata as classical timed automata; however, Theorem 4 does not use them in the standard way as it requires runs on subwords. Whether and how such runs can be captured by timed automata remains to be clarified. We now use the obtained automaton characterisation of certain answers to linear queries to give better complexity bounds for two classes of linear programs with restricted temporal ranges than the NL bound of Theorem 1 (*ii*).

6 *hornMTL⁻* Queries with $\diamondminus_{\langle 0,r\rangle}$

We say that a *hornMTL⁻* program Π in normal form is *range-uniform* if every (intensional) concept name $P \notin \Lambda$ occurs in Π in the scope of $\diamondminus_\varrho$, for some *fixed* range ϱ. By a *coreMTL⁻* program we mean a core *hornMTL⁻* program in normal form. Rules (4) in such a program take the form $P \leftarrow \diamondminus_\varrho Q$.

6.1 Range-uniform *coreMTL⁻* queries with $\diamondminus_{\langle 0,r\rangle}$

Let Π be a range-uniform *coreMTL⁻* program and $\mathcal{A}_\Pi = (S, S_0, \Sigma, \delta)$ the corresponding metric automaton. For each (q_0, e_0) in S_0, we define a classical finite automaton $\mathcal{A}^A_{q_0,e_0} = (S, \Sigma, \{q_0\}, \delta, \{A\})$, where S, Σ, and δ are as in $\mathcal{A}_\Pi$ (note that all transitions take the form $q \xrightarrow{\langle 0,r\rangle}_\emptyset q'$), and q_0 and A are unique initial and final states, respectively. Thus, $\mathcal{A}^A_{q_0,e_0}$ is a *unary* (i.e., over singleton alphabet) automaton. It is known that each such automaton has an equivalent automaton in normal form [11,17], where cycles can be only disjoint. More precisely, there is a number of arithmetic progressions $a_i + b_i\mathbb{N} = \{a_i + b_i \cdot m \mid m \in \mathbb{N}\}$ such that a word $\emptyset^n$ is accepted by $\mathcal{A}^A_{q_0,e_0}$ iff $n \in \bigcup_i a_i + b_i\mathbb{N}$. This characterisation allows us to obtain the following:

Theorem 5. *Consistency checking and answering range-uniform coreMTL⁻ queries with temporal operators of the form $\diamondminus_{\langle 0,r\rangle}$ are in* AC^0 *for data complexity provided that the input data instances satisfy* **(ord)**.

Example 6. To illustrate the theorem, consider the query $(\mathit{\Pi}, S_1(x))$ with

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

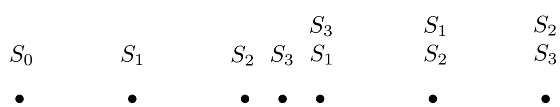
The automaton $\mathcal{A}_{S_0,B}$ (which is in normal form) is shown in the picture below on the right. Using it, we construct the following FO-rewriting $\varphi(x)$ of $(\mathit{\Pi}, S_1(x))$:

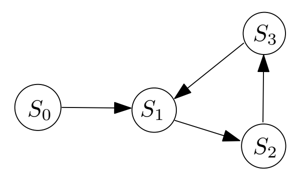
$$\varphi(x) = \exists x' \big[B(x') \land \forall y \big((x' < y \leq x) \to \exists y' \,\mathsf{dist}_{<d}(y, y') \land \\ (\varphi_1(x', x) \lor \varphi_2(x', x) \lor \varphi_3(x', x))\big)\big],$$

where

- $\varphi_1(x', x) = (x - x') \in 1 + 3\mathbb{N}$;
- $\varphi_2(x', x) = (x - x') \in 2 + 3\mathbb{N} \land \exists x_1\,((x' < x_1 \leq x) \land \varphi_{+1}(x_1, x'))$;
- $\varphi_3(x', x) = (x - x') \in 3 + 3\mathbb{N} \land \varphi_{1+1+1}(x', x) \lor \varphi_{1+2}(x', x)$;
- $\varphi_{1+2}(x', x) = \exists x_1\,((x' < x_1 \leq x) \land \varphi_{+2}(x_1, x'))$;
- $\varphi_{1+1+1}(x', x) = \exists x_1, x_2\,((x' < x_1 < x_2 \leq x) \land \varphi_{+1}(x_1, x') \land \varphi_{+1}(x_2, x') \land ((x_2 - x_1) > 1))$;
- $\varphi_{+k}(z, x') = \mathsf{dist}_{<d}(z, z - k - 1) \land ((z - k - 1) \geq x')$, 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 $\geq d$. 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 [13], whereas FO with $<$ only is not enough. For example, $(x - x') \in 1 + 3\mathbb{N}$ is expressed by the formula

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

Also, $\varphi(x)$ above is a correct rewriting only if evaluated over $\mathcal{D}$ satisfying **(ord)**.

6.2 *hornMTL*$^-$ queries with $\diamondminus_{\langle 0,r\rangle}$

We next turn to *hornMTL*$^-$ programs Π with ranges ϱ of the form $\langle 0, r\rangle$.

Theorem 6. *Consistency checking and answering hornMTL*$^-$ *queries with temporal operators of the form* $\diamondminus_{\langle 0,r\rangle}$ *are in* L *for data complexity.*

Example 7. We illustrate the log-space algorithm used in the proof of Theorem 6 by means of a program Π with the following rules:

$$P_2 \leftarrow \diamondminus_{(0,2]} P_2 \wedge \diamondminus_{(0,1]} P_1, \quad P_1 \leftarrow \diamondminus_{(0,3]} P_2, \quad P_2 \leftarrow P_2'.$$

For this Π, we can scan an input word $\mathcal{D}$ with the help of three pointers π, π_1 and π_2. Intuitively, π will point to the last processed timestamp and π_1 (π_2) to the last timestamp where P_1 (respectively, P_2) holds. Consider the word $\mathcal{D}$ shown in the picture below. Before the algorithm starts, we assume that π, π_1, and π_2 do not point anywhere. First, we set π to the first point t_0 (with the timestamp 0) and read P_2'. Because P_2 holds there by the last rule in Π, we set π_2 to t_0. Next, we move π to t_1 where we read Q' (not present in Π). Here, we check whether t_0 pointed to by π_2 and t_1 pointed to by π are such that $t_1 - t_0 \leq 3$ (which can be done in L using any subtraction algorithm). Since it is the case, we set π_1 to t_1 to reflect the meaning of the second rule in Π, and we do not need to update π_2. Next, we move π to t_2. We check that the difference between the timestamps pointed to by π and π_1 does not exceed 3 to verify whether the second rule in Π applies. Similarly, we check that the difference between π and π_1 does not exceed 1 and the difference between π and π_2 does not exceed 2 to verify whether the first rule in Π applies. As a result, we set both π_1 and π_2 to t_2. A complete run of our algorithm on $\mathcal{D}$ is shown below.

	0	1	2	5	5.5	6
$\mathcal{D}$:	P_2'	Q'	Q'	Q'	P_2'	Q'
run:	π_2	π_1	π_1, π_2	π_1	π_2	π_1, π_2
derived:	P_2	P_1	P_1, P_2	P_1	P_2	P_1, P_2

To decide whether, say, t_5 is a certain answer to $(\Pi, P_1(x))$, we only need to check whether we move π_2 to t_5 when we move π there.

Note that the algorithm above works correctly only if the ranges are of the form $\langle 0, r\rangle$. Intuitively, resetting π_i to π every time a rule with P_i in the head applies does not provide correct answers for other forms of ranges.

The exact complexity for the queries in Theorem 6 remains open. At the moment, we only have the following result, which is proved by reduction of *hornLTL* queries with rules of the form $Q \leftarrow \bigcirc_P P \wedge P'$, where $\bigcirc_P$ is the 'previous moment of time' operator, that were shown to be NC^1-complete in [4]. In this reduction, we use the axioms $Q \leftarrow \diamondminus_{(0,1]} P \wedge P'$ to encode the axioms above. Then every *hornLTL* data instance of the form $P_0'(0), P_1'(1), \ldots, P_k'(k)$ is translated to a *hornMTL* data instance by means of an FO-reduction using the standard predicate BIT(*num*, *bit*).

Theorem 7. *Consistency checking and answering hornMTL*$^-$ *queries with temporal operators of the form* $\diamondminus_{\langle 0,r\rangle}$ *are* NC^1*-hard for data complexity.*

7 *hornMTL*$^-$ Queries with $\diamondminus_{[r,r]}$

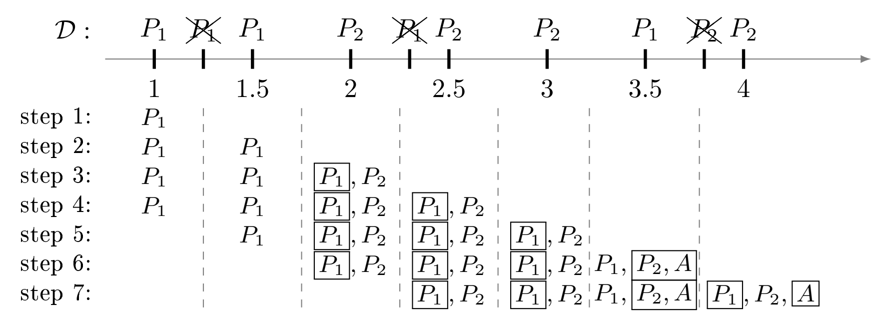
Theorem 8. *Consistency checking and answering hornMTL*$^-$ *queries with temporal operators of the form* $\diamondminus_{[r,r]}$ *are in* L *for data complexity.*

We illustrate the idea of the proof by a concrete example.

Example 8. Suppose $\Pi = \{P_1 \leftarrow \diamondminus_{[1,1]} P_1,\ A \leftarrow P_1 \wedge \diamondminus_{[1.5,1.5]} P_2\}$ and

$$\mathcal{D} = \{(P_1, 1), (P_1, 1.25), (P_1, 1.5), (P_2, 2), (P_1, 2.375), (P_2, 2.5), (P_2, 3), (P_1, 3.5), (P_2, 3.625), (P_2, 4)\}.$$

Let $num(\Pi)$ be the set of numbers in Π. To check whether 4 is a certain answer to $(\Pi, A(x))$, we compute $d = gcd(num(\Pi)) = 0.5$ and $k = \frac{max(num(\Pi))}{d} = 3$. Observe first that the algorithm can ignore those facts in $\mathcal{D}$ with a timestamp t for which $4 - t$ is not divisible by d, that is, we can omit $(P_1, 1.25)$, $(P_1, 2.375)$, and $(P_2, 3.625)$ as they have no influence on the facts derived at 4. Next, notice that to derive a fact at some t, it suffices to check what facts hold at the instants $t, t-d, \ldots, t-kd$. Hence, for each concept name in Π, it suffices to use $k+1 = 4$ pointers storing the last 4 timestamps where this concept holds. The consecutive steps of our algorithm are shown below, where symbols in boxes represent facts derived by the rules in Π:



The algorithm uses $k+1$-many log-space pointers for each concept name in Π, where k only depends on Π, and a single pointer indicating the currently processed timestamp. As a result, the algorithm is in L for data complexity.

We note that the best known lower bound for this language is NC^1, which can be shown using a reduction similar to that in the proof of Theorem 7. The algorithm illustrated above cannot be used to show an NC^1 upper bound because it must ignore some facts in $\mathcal{D}$.

Acknowledgements

This work was supported by the NCN grant 2016/23/N/HS1/02168 and the Foundation for Polish Science (FNP).

References

1. Alur, R., Dill, D.L.: A theory of timed automata. Theoretical Computer Science 126(2), 183 – 235 (1994)
2. Alur, R., Henzinger, T.A.: Real-time logics: Complexity and expressiveness. Inf. Comput. 104(1), 35–77 (1993)
3. Artale, A., Kontchakov, R., Wolter, F., Zakharyaschev, M.: Temporal description logic for ontology-based data access. In: Proc. of the 23rd Int. Joint Conf. on Artificial Intelligence, IJCAI 2013. IJCAI/AAAI (2013)
4. Artale, A., Kontchakov, R., Kovtunova, A., Ryzhikov, V., Wolter, F., Zakharyaschev, M.: First-order rewritability of temporal ontology-mediated queries. In: Proc. of the 24th Int. Joint Conf. on Artificial Intelligence, IJCAI 2015. pp. 2706–2712. IJCAI/AAAI (2015)
5. Artale, A., Kontchakov, R., Kovtunova, A., Ryzhikov, V., Wolter, F., Zakharyaschev, M.: Ontology-mediated query answering over temporal data: A survey (invited talk). In: TIME. LIPIcs, vol. 90, pp. 1:1–1:37. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2017)
6. Baader, F., Borgwardt, S., Koopmann, P., Ozaki, A., Thost, V.: 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. pp. 60–76 (2017)
7. Baudinet, M., Chomicki, J., Wolper, P.: Temporal deductive databases. In: Temporal Databases, pp. 294–320 (1993)
8. Brandt, S., Kalayci, E.G., Ryzhikov, V., Xiao, G., Zakharyaschev, M.: A framework for temporal ontology-based data access: A proposal. In: New Trends in Databases and Information Systems - ADBIS 2017 Short Papers and Workshops, AMSD, BigNovelTI, DAS, SW4CH, DC, Nicosia, Cyprus, September 24-27, 2017, Proceedings. Communications in Computer and Information Science, vol. 767, pp. 161–173. Springer (2017)
9. Brandt, S., Kalaycı, E.G., Ryzhikov, V., Xiao, G., Zakharyaschev, M.: Querying log data with metric temporal logic. Journal of Artificial Intelligence Research 62, 829–877 (2018)
10. Chomicki, J., Toman, D.: Temporal logic in information systems. In: Logics for Databases and Information Systems. pp. 31–70. Kluwer (1998)
11. Chrobak, M.: Finite automata and unary languages. Theor. Comp. Sci. 47(2), 149–158 (1986)
12. Gutiérrez-Basulto, V., Jung, J.C., Ozaki, A.: On metric temporal description logics. In: 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). pp. 837–845 (2016)
13. Immerman, N.: Descriptive Complexity. Springer (1999)
14. Koymans, R.: Specifying real-time properties with metric temporal logic. Real-Time Systems 2(4), 255–299 (1990)
15. Ouaknine, J., Worrell, J.: On the decidability and complexity of metric temporal logic over finite words. Logical Methods in Computer Science 3(1) (2007)
16. Ouaknine, J., Worrell, J.: 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. pp. 1–13 (2008)
17. To, A.W.: Unary finite automata vs. arithmetic progressions. Inf. Process. Lett. 109(17), 1010–1014 (2009)