RZ 3702 (# 99712) 02/11/08 (Revised version: 09/29/2008)
Computer Science 27 pages

Research Report

Runtime Monitoring of Metric First-order Temporal Properties (Revised Version)

David Basin[1], Felix Klaedtke[1], Samuel Müller[1,2], and Birgit Pfitzmann[3]

[1]Information Security Group, Department of Computer Science
Swiss Federal Institute of Technology - ETH Zurich
8092 Zurich, Switzerland

[2]IBM Research GmbH, Zurich Research Laboratory
8803 Rüschlikon, Switzerland

[3]IBM Thomas J. Watson Research Center
Hawthorne, NY 10532-1596, USA

Research
Almaden · Austin · Beijing · Delhi · Haifa · T.J. Watson · Tokyo · Zurich

Runtime Monitoring of Metric First-order Temporal Properties

David Basin[1], Felix Klaedtke[1], Samuel Müller[1,2], and Birgit Pfitzmann[3]

[1] ETH Zurich, Switzerland
[2] IBM Zurich Research Lab, Switzerland
[3] IBM Watson Research Lab, USA

Abstract. We introduce a novel approach to the runtime monitoring of complex system properties. In particular, we present an online algorithm for a safety fragment of metric first-order temporal logic (MFOTL) that is considerably more expressive than the logics supported by prior monitoring methods. Our approach, based on automatic structures, allows the unrestricted use of negation, universal and existential quantification over infinite domains, and the arbitrary nesting of both past and bounded future operators. Moreover, we show how to use and optimize our approach for the common case where structures consist of only finite relations, over possibly infinite domains. Under an additional restriction, we prove that the space consumed by our monitor is polynomially bounded by the cardinality of the data appearing in the processed prefix of the temporal structure being monitored.

1 Introduction

Runtime monitoring [1] is an approach to verifying system properties at execution time by using an online algorithm to check whether a system trace satisfies a temporal property. Whereas novel application areas such as compliance or business activity monitoring [13, 15] require expressive property specification languages, current monitoring techniques are restricted in the properties they can handle. They either support properties expressed in propositional temporal logics and thus cannot cope with variables ranging over infinite domains [6, 16, 21, 24, 30], do not provide both universal and existential quantification [5, 12, 17, 24–26] or only in restricted ways [5, 29, 31], do not allow arbitrary quantifier alternation [5, 23], cannot handle unrestricted negation [8, 23, 28, 31], do not provide quantitative temporal operators [23, 26], or cannot simultaneously handle both past and future temporal operators [8, 23–25, 27, 28].

In this paper, we present a runtime monitoring approach for an expressive safety fragment of metric first-order temporal logic (MFOTL) [8] that overcomes most of these limitations. The fragment consists of formulae of the form $\Box\,\phi$, where ϕ is bounded, i.e., its temporal operators refer only finitely into the future. Our monitor uses automatic structures [7] to finitely represent infinite structures, which allows for the unrestricted use of negation and quantification in monitored formulae. Moreover, our monitor supports the arbitrary nesting of both (metric) past and bounded future operators. This means that complex properties can be specified more naturally than with only past operators.[4]

In a nutshell, our monitor works as follows: Given a MFOTL formula $\Box\,\phi$ over a signature S, where ϕ is bounded, we first transform ϕ into a first-order formula $\hat{\phi}$ over an extended signature $\hat{S}$, obtained by augmenting S with auxiliary predicates for every temporal subformula in ϕ. Our monitor then incrementally processes a temporal structure (D, τ) over S and determines for each time point i those elements in (D, τ) that violate ϕ. This is achieved by incrementally constructing a collection of automata that finitely represent the (possibly infinite) interpretations of the auxiliary predicates and by evaluating the transformed first-order formula $\neg\hat{\phi}$ over the extended $\hat{S}$-structure

[4] It is unknown whether the past-only fragment of MFOTL is as expressive as the fragment with both past and bounded future operators and whether formulae in the past-only fragment can be expressed as succinctly as those in the future-bounded fragment.

at every time point. In doing so, our monitor discards any information not required for evaluating $\neg\hat{\phi}$ at the current and future time points.

We also show how to adapt our monitoring approach to the common case where all relations are required to be finite and hence relational databases can serve as an alternative to automata. Under the additional (realistic) restriction that time increases after at most a fixed number of time points, our incremental construction ensures that our monitor requires only polynomial space in the cardinality of the data appearing in the processed prefix of the monitored temporal structure. This is in contrast to complexity results for other approaches, such as the logical data expiration technique proposed for 2-FOL [31]. While this logic is at least as expressive as MFOTL, the space required for monitoring (syntactically-restricted) 2-FOL formulae is non-elementary in the cardinality of the data in the processed prefix.

Overall, we see our contributions as follows. First, the presented monitor admits a substantially more expressive logic than previous monitoring approaches. In particular, by supporting arbitrary bounded MFOTL formulae, it significantly extends Chomicki's dynamic integrity checking approach for temporal databases [8]. Second, we extend runtime monitoring to automatic structures, which allows for the unrestricted use of negation and quantification in monitored formulae. Third, for the restricted setting where all relations are finite, we show how to implement our monitor using relational databases. Here, we extend the rewrite procedure of [11] to handle a larger class of temporal formulae. We then prove that, under an additional restriction, the space consumed by our monitor is polynomially bounded in the cardinality of the data appearing in the processed prefix of a monitored temporal structure. Finally, our work shows how to effectively combine ideas from different, but related areas, including database theory, runtime monitoring, model checking, and model theory.

2 Metric First-order Temporal Logic

In this section, we introduce metric first-order temporal logic (MFOTL) [8], which extends propositional metric temporal logic [4, 20] in a standard way. In the forthcoming sections, we present a method for monitoring requirements formalized within MFOTL.

Syntax and Semantics. Let $\mathbb{I}$ be the set of nonempty intervals over $\mathbb{N}$. We often write an interval in $\mathbb{I}$ as $[c, d)$, where $c \in \mathbb{N}$, $d \in \mathbb{N} \cup \{\infty\}$, and $c < d$, i.e., $[c, d) := \{a \in \mathbb{N} \,|\, c \leq a < d\}$. A *signature* S is a tuple $(\mathsf{C}, \mathsf{R}, a)$, where C is a finite set of constant symbols, R is a finite set of predicates disjoint from C, and the function $a : \mathsf{R} \to \mathbb{N}$ associates each predicate $r \in \mathsf{R}$ with an arity $a(r) \in \mathbb{N}$. For the rest of this paper, V denotes a countably infinite set of variables, where we assume that $\mathsf{V} \cap (\mathsf{C} \cup \mathsf{R}) = \emptyset$, for every signature $S = (\mathsf{C}, \mathsf{R}, a)$. In the following, let $S = (\mathsf{C}, \mathsf{R}, a)$ be a signature.

Definition 1. *The* formulae *over S are inductively defined: (i) For $t, t' \in \mathsf{V} \cup \mathsf{C}$, $t \approx t'$ and $t \prec t'$ are formulae. (ii) For $r \in \mathsf{R}$ and $t_1, \dots, t_{a(r)} \in \mathsf{V} \cup \mathsf{C}$, $r(t_1, \dots, t_{a(r)})$ is a formula. (iii) For $x \in \mathsf{V}$, if θ and θ' are formulae then $(\neg\theta)$, $(\theta \wedge \theta')$, and $(\exists x.\, \theta)$ are formulae. (iv) For $I \in \mathbb{I}$, if θ and θ' are formulae then $(\CIRCLE_I\, \theta)$, $(\bigcirc_I\, \theta)$, $(\theta\, \mathcal{S}_I\, \theta')$, and $(\theta\, \mathcal{U}_I\, \theta')$ are formulae.*

To define the semantics of MFOTL, we need the following notions: A *(first-order) structure D* over S consists of a domain $|D| \neq \emptyset$ and interpretations $c^D \in |D|$ and $r^D \subseteq |D|^{a(r)}$, for each $c \in \mathsf{C}$ and $r \in \mathsf{R}$. A *temporal (first-order) structure* over S is a pair $(\bar{D}, \bar{\tau})$, where $\bar{D} = (D_0, D_1, \dots)$ is a sequence of structures over S and $\bar{\tau} = (\tau_0, \tau_1, \dots)$ is a sequence of natural numbers (time stamps), where:

1. The sequence $\bar{\tau}$ is monotonically increasing (i.e., $\tau_i \leq \tau_{i+1}$, for all $i \geq 0$) and makes progress (i.e., for every $i \geq 0$, there is some $j > i$ such that $\tau_j > \tau_i$).

2. D has constant domains, i.e., $|D_i| = |D_{i+1}|$, for all $i \geq 0$. We denote the domain by $|D|$ and require that $|D|$ is linearly ordered by the relation $<$.
3. Each constant symbol $c \in \mathsf{C}$ has a rigid interpretation, i.e., $c^{D_i} = c^{D_{i+1}}$, for all $i \geq 0$. We denote the interpretation of c by c^D.

A *valuation* is a mapping $v : \mathsf{V} \to |D|$. We abuse notation by applying a valuation v also to constant symbols $c \in \mathsf{C}$, with $v(c) = c^D$. For a valuation v, a variable vector $\bar{x} = (x_1, \dots, x_n)$, and $\bar{d} = (d_1, \dots, d_n) \in |D|^n$, $v[\bar{x}/\bar{d}]$ is the valuation that maps x_i to d_i, for i such that $1 \leq i \leq n$, and the valuation of the other variables is unaltered.

Definition 2. *Let (D, τ) be a temporal structure over S, with $D = (D_0, D_1, \dots)$ and $\tau = (\tau_0, \tau_1, \dots)$, θ a formula over S, v a valuation, and $i \in \mathbb{N}$. We define the relation $(D, \tau, v, i) \models \theta$ as follows:*

$$\begin{array}{lll}
(D,\tau,v,i) \models t \approx t' & \textit{iff} & v(t) = v(t') \\
(D,\tau,v,i) \models t \prec t' & \textit{iff} & v(t) < v(t') \\
(D,\tau,v,i) \models r(t_1,\dots,t_{a(r)}) & \textit{iff} & (v(t_1),\dots,v(t_{a(r)})) \in r^{D_i} \\
(D,\tau,v,i) \models (\neg\theta_1) & \textit{iff} & (D,\tau,v,i) \not\models \theta_1 \\
(D,\tau,v,i) \models (\theta_1 \wedge \theta_2) & \textit{iff} & (D,\tau,v,i) \models \theta_1 \textit{ and } (D,\tau,v,i) \models \theta_2 \\
(D,\tau,v,i) \models (\exists x.\, \theta_1) & \textit{iff} & (D,\tau,v[x/d],i) \models \theta_1, \textit{ for some } d \in |D| \\
(D,\tau,v,i) \models (\bullet_I\, \theta_1) & \textit{iff} & i > 0,\ \tau_i - \tau_{i-1} \in I,\ \textit{and } (D,\tau,v,i-1) \models \theta_1 \\
(D,\tau,v,i) \models (\bigcirc_I\, \theta_1) & \textit{iff} & \tau_{i+1} - \tau_i \in I \textit{ and } (D,\tau,v,i+1) \models \theta_1 \\
(D,\tau,v,i) \models (\theta_1\, \mathcal{S}_I\, \theta_2) & \textit{iff} & \textit{for some } j \leq i,\ \tau_i - \tau_j \in I,\ (D,\tau,v,j) \models \theta_2, \\
 & & \textit{and } (D,\tau,v,k) \models \theta_1,\ \textit{for all } k \in [j+1, i+1) \\
(D,\tau,v,i) \models (\theta_1\, \mathcal{U}_I\, \theta_2) & \textit{iff} & \textit{for some } j \geq i,\ \tau_j - \tau_i \in I,\ (D,\tau,v,j) \models \theta_2, \\
 & & \textit{and } (D,\tau,v,k) \models \theta_1,\ \textit{for all } k \in [i, j)
\end{array}$$

Note that the temporal operators are augmented with lower and upper bounds. A temporal formula is only satisfied if it is satisfied within the bounds given by the temporal operator, which are relative to the current time stamp τ_i.

Terminology and Notation. As syntactic sugar, we use the standard boolean connectives like $(\theta_1 \vee \theta_2) := (\neg((\neg\theta_1) \wedge (\neg\theta_2)))$ and $(\theta_1 \to \theta_2) := ((\neg\theta_1) \vee \theta_2)$, the universal quantifier $(\forall x.\, \theta) := (\neg(\exists x.\, \neg\theta)$, and the temporal operators $(\blacklozenge_I\, \theta) := (\mathit{true}\ \mathcal{S}_I\ \theta)$, $(\blacksquare_I\, \theta) := (\neg(\blacklozenge_I(\neg\theta)))$, $(\lozenge_I\, \theta) := (\mathit{true}\mathcal{U}_I\theta)$, and $(\square_I\, \theta) := (\neg(\lozenge_I(\neg\theta)))$, where $I \in \mathbb{I}$ and *true* abbreviates $(\exists x.\, x = x)$. The non-metric variants of the temporal operators are easily defined, e.g., $(\square\, \theta) := (\square_{[0,\infty)}\, \theta)$. We use standard conventions concerning the binding strength of operators to omit parentheses. For example, $\neg$ binds stronger than $\wedge$, which binds stronger than $\vee$, which in turn binds stronger than $\exists$. Moreover, temporal operators bind weaker than Boolean connectives and quantifiers.

We call formulae of the form $t \approx t'$, $t \prec t'$, and $r(t_1, \dots, t_{a(r)})$ *atomic*, and formulae with no temporal operators *first-order*. The outermost connective (i.e., Boolean connective, quantifier, or temporal operator) occurring in a formula θ is called the *main connective* of θ. A formula that has a temporal operator as its main connective is a *temporal* formula. A formula θ is *bounded* if the interval I of every temporal operator $\mathcal{U}_I$ occurring in θ is finite.

MFOTL denotes the set of MFOTL formulae and FOL the set of first-order formulae. For $\theta \in \mathsf{MFOTL}$, we define its immediate temporal subformulae $\mathit{tsub}(\theta)$ to be:

$$\mathit{tsub}(\theta) := \begin{cases} \mathit{tsub}(\alpha) & \text{if } \theta = \neg\alpha \text{ or } \theta = \exists x.\alpha, \\ \mathit{tsub}(\alpha) \cup \mathit{tsub}(\beta) & \text{if } \theta = \alpha \wedge \beta, \\ \{\theta\} & \text{if } \theta \text{ is a temporal formula,} \\ \emptyset & \text{otherwise.} \end{cases}$$

For example, for $\theta := (\bullet\, \alpha) \wedge ((\bigcirc\, \beta)\, \mathcal{S}_{[1,9)}\, \gamma)$, we have that $\mathit{tsub}(\theta) = \{\bullet\, \alpha, (\bigcirc\, \beta)\, \mathcal{S}_{[1,9)}\, \gamma\}$.

If $\theta \in \mathsf{MFOTL}$ has the free variables given by the vector $\bar{x} = (x_1, \dots, x_n)$, we define the set of satisfying assignments at time instance i as

$$\theta^{(D,\tau,i)} := \big\{\bar{d} \in |D|^n \,\big|\, (D, \tau, v[\bar{x}/\bar{d}], i) \models \theta, \text{ for some valuation } v\big\}\,.$$

If the formula θ is in FOL, we write $(D_i, v) \models \theta$ instead of $(D, \tau, v, i) \models \theta$ and θ^{D_i} for $\theta^{(D,\tau,i)}$. Note that $(D_i, v) \models \theta$ agrees with the standard definition of satisfaction in first-order logic.

3 Monitoring by Reduction to First-order Queries

To effectively monitor MFOTL formulae, we restrict both the formulae and the temporal structures under consideration. We discuss these restrictions in §3.1 and describe monitoring in §3.2–§3.5.

3.1 Restrictions

Throughout this section, let (D, τ) be a temporal structure over the signature $S = (\mathsf{C}, \mathsf{R}, a)$ and ψ the formula to be monitored. We make the following restrictions on ψ and D. First, we require ψ to be of the form $\Box\, \phi$, where ϕ is bounded. It follows that ψ describes a safety property [3]. Note though that not all safety properties can be expressed by formulae of this form [9]. This is in contrast to propositional linear temporal logic, where every safety property can be expressed as $\Box\, \beta$, where β contains only past-time operators [22].

Second, we require that each structure in D is automatic [18]. Roughly speaking, this means that each structure in D can be finitely represented by a collection of automata over finite words. Let us briefly recall some background on automatic structures [7, 18]. Let Σ be an alphabet and # a symbol not in Σ. The *convolution* of the words $w_1, \dots, w_k \in \Sigma^*$ with $w_i = w_{i1} \cdots w_{i\ell_i}$ is the word

$$w_1 \otimes \cdots \otimes w_k := \begin{bmatrix} w'_{11} \\ \vdots \\ w'_{k1} \end{bmatrix} \cdots \begin{bmatrix} w'_{1\ell} \\ \vdots \\ w'_{k\ell} \end{bmatrix} \in \big((\Sigma \cup \{\#\})^k\big)^*\,,$$

where $\ell = \max\{\ell_1, \dots, \ell_k\}$ and $w'_{ij} = w_{ij}$, for $j \leq \ell_i$ and $w'_{ij} = \#$ otherwise. The padding symbol # is added to the words w_i to ensure that all of them have the same length.

Definition 3. *A structure A over a signature $S = (\mathsf{C}, \mathsf{R}, a)$ is* automatic *if there is a regular language $\mathcal{L}_{|A|} \subseteq \Sigma^*$ and a surjective function $\nu : \mathcal{L}_{|A|} \to |A|$ such that the language $\mathcal{L}_{\approx} := \{u \otimes v \,|\, u, v \in \mathcal{L}_{|A|} \text{ with } \nu(u) = \nu(v)\}$ is regular and, for each relation $r^A \subseteq |A|^{a(r)}$ with $r \in \mathsf{R}$, the language $\mathcal{L}_r := \{w_1 \otimes \cdots \otimes w_{a(r)} \,|\, w_1, \dots, w_{a(r)} \in \mathcal{L}_{|D|} \text{ with } (\nu(w_1), \dots, \nu(w_{a(r)})) \in r^A\}$ is regular.*

An *automatic representation* of the automatic structure A consists of (i) the function $\nu : \mathcal{L}_{|A|} \to |A|$, (ii) a family of words $(w_c)_{c \in \mathsf{C}}$ with $w_c \in \mathcal{L}_{|A|}$ and $\nu(w_c) = c^A$, for all $c \in \mathsf{C}$, and (iii) a collection $(\mathcal{A}_{|A|}, \mathcal{A}_{\approx}, (\mathcal{A}_r)_{r \in \mathsf{R}})$ of automata that recognize the languages $\mathcal{L}_{|A|}$, $\mathcal{L}_{\approx}$, and $\mathcal{L}_r$, for all $r \in \mathsf{R}$. In the following, we assume that for an automatic structure, we always have an automatic representation for it at hand. A relation $r^A \subseteq |A|^k$ is *regular* if the language $\{u_1 \otimes \cdots \otimes u_k \,|\, u_1, \dots, u_k \in \mathcal{L}_{|A|}$ with $(\nu(u_1), \dots, \nu(u_k)) \in r\}$ is regular. Note that an automaton reads the components of the convolution of a representative of $\bar{a} \in |A|^k$ synchronously.

In addition to the requirement that each structure in D is automatic, we require that D has a constant domain representation. This means that the domain of each D_i is represented by the same regular language $\mathcal{L}_{|D|}$ and each word in $\mathcal{L}_{|D|}$ represents the same element in $|D|$, i.e., each automatic representation has the same function $\nu : \mathcal{L}_{|D|} \to |D|$. Finally, we assume that $|D| = \mathbb{N}$ and that $<$ is the standard ordering on $\mathbb{N}$. This is without loss of generality whenever the function

ν is injective, i.e., every element in $|D|$ has only one representative in $\mathcal{L}_{|D|}$. See Appendix A.1 for details. Furthermore, note that every automatic structure has an automatic representation in which the function ν is injective [18].

Remark 4. Let us state some properties of automatic structures that we will use later. First, for a first-order formula θ, we can effectively construct an automaton that represents the set θ^{D_i}. This follows from the closure properties of regular languages. Second, some basic arithmetical relations are first-order definable in the structure $(\mathbb{N}, <)$ and thus regular. In particular, the successor relation $\{(x, y) \in \mathbb{N}^2 \mid y = x + 1\}$ is regular, since the formula $x \prec y \wedge \neg\exists z. x \prec z \wedge z \prec y$ defines it. It is also easy to see that $\{(x, y) \in \mathbb{N}^2 \mid x + d \leq y\}$ is regular, for any $d \in \mathbb{N}$.

Example 5. Before presenting our monitoring method, we give two examples of system properties expressed in the MFOTL fragment that our monitor can handle. First, the property "whenever the program variable *in* stores the input x, within 5 time units x must be stored in the program variable *out*" can be expressed by $\Box\, \forall x.\, in(x) \to \Diamond_{[0,6)}\, out(x)$. Second, the property "the value of the program variable v increases by 1 in each step from an initial value 0 until it becomes 5, then it stays constant" can be formalized as $\Box(\neg(\bullet\, true) \to v(0)) \wedge (\exists i.\, v(i) \wedge i \prec 5 \to \bigcirc\, v(i{+}1)) \wedge (v(5) \to \bigcirc\, v(5))$. Note that we use relations that are singletons to model program variables.

3.2 Overview of the Monitoring Method

To monitor the formula $\Box\, \phi$ over a temporal structure (D, τ), we incrementally build a sequence of structures $\hat{D}_0, \hat{D}_1, \ldots$ over an extended signature $\hat{S}$. The extension depends on the temporal subformulae of ϕ. For each time point i, we determine the elements that violate ϕ by evaluating a transformed formula $\neg\hat{\phi} \in \mathsf{FOL}$ over $\hat{D}_i$. Observe that with future operators, we usually cannot do this yet when time point i occurs. Our monitor, which we present in §3.5, therefore maintains a list of unevaluated subformulae for past time points. In the following, we first describe how we extend S and transform ϕ. Afterwards, we explain how we incrementally build $\hat{D}_i$. Finally, we present our monitor and prove its correctness. Omitted proofs are given in Appendix A.2.

3.3 Signature Extension and Formula Transformation

In addition to the predicates in R, the extended signature $\hat{S}$ contains an auxiliary predicate p_α for each temporal subformula α of ϕ. For subformulae of the form $\beta\, \mathcal{S}_I\, \gamma$ and $\beta\, \mathcal{U}_I\, \gamma$, we introduce further predicates, which store information that allow us to incrementally update the auxiliary relations.

Definition 6. *Let $\hat{S} := (\hat{\mathsf{C}}, \hat{\mathsf{R}}, \hat{a})$ be the signature with $\hat{\mathsf{C}} := \mathsf{C}$ and*

$$\begin{aligned}\hat{\mathsf{R}} := \mathsf{R} \cup \{&p_\alpha \mid \alpha \textit{ temporal subformula of } \phi\} \cup \\ \{&r_\alpha \mid \alpha \textit{ subformula of } \phi \textit{ of the form } \beta\, \mathcal{S}_I\, \gamma \textit{ or } \beta\, \mathcal{U}_I\, \gamma\} \cup \\ \{&s_\alpha \mid \alpha \textit{ subformula of } \phi \textit{ of the form } \beta\, \mathcal{U}_I\, \gamma\}\,.\end{aligned}$$

For $r \in \mathsf{R}$, let $\hat{a}(r) := a(r)$. If α is a temporal subformula with n free variables, then $\hat{a}(p_\alpha) := n$, and $\hat{a}(r_\alpha) := n + 1$ and $\hat{a}(s_\alpha) := n + 2$, if r_α and s_α exist. We assume that $p_\alpha, r_\alpha, s_\alpha \notin \mathsf{C} \cup \mathsf{R} \cup \mathsf{V}$.

We transform MFOTL formulae over the signature S into first-order formulae over the extended signature $\hat{S}$ as follows.

Definition 7. *For $\theta \in \mathsf{MFOTL}$ with the vector of free variables $\bar{x}$, we define*

$$\hat{\theta} := \begin{cases} \neg\hat{\beta} & \text{if } \theta = \neg\beta, \\ \hat{\beta} \wedge \hat{\gamma} & \text{if } \theta = \beta \wedge \gamma, \\ \exists y.\, \hat{\beta} & \text{if } \theta = \exists y.\, \beta, \\ p_\theta(\bar{x}) & \text{if } \theta \text{ is a temporal formula,} \\ \theta & \text{is an atomic formula.} \end{cases}$$

We assume throughout this section, without loss of generality, that each subformula of ϕ has the vector of free variables $\bar{x} = (x_1, \dots, x_n)$. The formula transformation has the following properties, which are easily shown by an induction over the formula structure.

Lemma 8. *Let θ be a subformula of ϕ. For all $i \in \mathbb{N}$, the following properties hold:*
(i) If $p_\alpha^{\hat{D}_i} = \alpha^{(D,\tau,i)}$ for all $\alpha \in tsub(\theta)$, then $\hat{\theta}^{\hat{D}_i} = \theta^{(D,\tau,i)}$.
(ii) If $p_\alpha^{\hat{D}_i}$ is regular for all $\alpha \in tsub(\theta)$, then $\hat{\theta}^{\hat{D}_i}$ is regular.

3.4 Incremental Extended Structure Construction

We now show how the auxiliary relations in the $\hat{D}_i$s are incrementally constructed. Their instantiations are computed recursively both over time and over the formula structure, where evaluations of subformulae may also be needed from future time points. We later show that this is well-defined and can be evaluated incrementally.

For $c \in \mathsf{C}$ and $r \in \mathsf{R}$, we define $c^{\hat{D}_i} := c^{D_i}$ and $r^{\hat{D}_i} := r^{D_i}$. We address the auxiliary relations for each type of main temporal operator separately.

Previous and Next. For $\alpha = \bullet_I \beta$ with $I \in \mathbb{I}$, we define

$$p_\alpha^{\hat{D}_i} := \begin{cases} \hat{\beta}^{\hat{D}_{i-1}} & \text{if } i > 0 \text{ and } \tau_i - \tau_{i-1} \in I, \\ \emptyset & \text{otherwise.} \end{cases}$$

Intuitively, a tuple $\bar{a}$ is in $p_\alpha^{\hat{D}_i}$ if $\bar{a}$ satisfies β at the previous time point $i-1$ and the difference of the two successive time stamps is in the interval I given by the metric temporal operator $\bullet_I$.

Lemma 9. *Let $\alpha = \bullet_I \beta$. For $i > 0$, if $p_\delta^{\hat{D}_{i-1}}$ is regular and $p_\delta^{\hat{D}_{i-1}} = \delta^{(D,\tau,i-1)}$ for all $\delta \in tsub(\beta)$, then $p_\alpha^{\hat{D}_i}$ is regular and $p_\alpha^{\hat{D}_i} = \alpha^{(D,\tau,i)}$. Moreover, $p_\alpha^{\hat{D}_0}$ is regular and $p_\alpha^{\hat{D}_0} = \alpha^{(D,\tau,0)}$.*

Proof. For $i = 0$, the lemma obviously holds. For $i > 0$, the regularity of $p_\alpha^{\hat{D}_i}$ follows from the assumption that the relations $p_\delta^{\hat{D}_{i-1}}$ are regular and Lemma 8(ii). The equality of the two sets follows from Lemma 8(i) and the semantics of the temporal operator $\bullet_I$. ⊣

For $\alpha = \bigcirc_I \beta$ with $I \in \mathbb{I}$, we define

$$p_\alpha^{\hat{D}_i} := \begin{cases} \hat{\beta}^{\hat{D}_{i+1}} & \text{if } \tau_{i+1} - \tau_i \in I, \\ \emptyset & \text{otherwise.} \end{cases}$$

Note that the definition of $p_\alpha^{\hat{D}_i}$ depends on the relations of the next structure D_{i+1} and on the auxiliary relations for $\delta \in tsub(\beta)$ of the next extended structure $\hat{D}_{i+1}$. Hence, the monitor instantiates $p_\alpha^{\hat{D}_i}$ with a delay of at least one time step.

The following lemma is proved similarly to Lemma 9.

Lemma 10. *Let $\alpha = \bigcirc_I \beta$. If $p_\delta^{\hat{D}_{i+1}}$ is regular and $p_\delta^{\hat{D}_{i+1}} = \delta^{(D,\tau,i+1)}$ for all $\delta \in tsub(\beta)$, then $p_\alpha^{\hat{D}_i}$ is regular and $p_\alpha^{\hat{D}_i} = \alpha^{(D,\tau,i)}$.*

Since and Until. We first address the past-time operator $\mathcal{S}_I$ with $I = [c, d) \in \mathbb{I}$. Assume that $\alpha = \beta \, \mathcal{S}_I \, \gamma$. We start with the initialization and update of the auxiliary relations for r_α. We define

$$r_\alpha^{\hat{D}_0} := \hat{\gamma}^{\hat{D}_0} \times \{0\} \,,$$

and for $i > 0$, we define

$$r_\alpha^{\hat{D}_i} := \big(\hat{\gamma}^{\hat{D}_i} \times \{0\}\big) \cup \big\{(\bar{a}, y) \in \mathbb{N}^{n+1} \,\big|\, \bar{a} \in \hat{\beta}^{\hat{D}_i},\ y < d, \text{ and } (\bar{a}, y') \in r_\alpha^{\hat{D}_{i-1}}, \text{ for } y' = y - \tau_i + \tau_{i-1}\big\} \,.$$

Intuitively, a pair $(\bar{a}, y)$ is in $r_\alpha^{\hat{D}_i}$ if $\bar{a}$ satisfies α at the time point i independent of the lower bound c, where the "age" y indicates how long ago the formula γ was satisfied by $\bar{a}$. If $\bar{a}$ satisfies γ at the time point i, it is added to $r_\alpha^{\hat{D}_i}$ with the age 0. For $i > 0$, we additionally update the tuples $(\bar{a}, y) \in r_\alpha^{\hat{D}_{i-1}}$. First, $\bar{a}$ must satisfy β at the time point i. Second, the age is adjusted by the difference of the time stamps τ_{i-1} and τ_i. Third, the new age must be less than d, otherwise it is too old to satisfy α.

The arithmetic constraint $y' = y - \tau_i + \tau_{i-1}$ in the definition of $r_\alpha^{\hat{D}_i}$ for $i > 0$ is first-order definable in D, see Remark 4. Note that $\tau_i + \tau_{i-1}$ is a constant value. Now it is not hard to see that $r_\alpha^{\hat{D}_i}$ is regular if all its components are regular.

With the relation $r_\alpha^{\hat{D}_i}$, we can determine the elements that satisfy α at the time point i. We define

$$p_\alpha^{\hat{D}_i} := \big\{\bar{a} \in \mathbb{N}^n \,\big|\, (\bar{a}, y) \in r_\alpha^{\hat{D}_i}, \text{ for some } y \geq c\big\} \,.$$

Lemma 11. *Let $\alpha = \beta \, \mathcal{S}_{[c,d)} \, \gamma$. Assume that $p_\delta^{\hat{D}_j}$ is regular and $p_\delta^{\hat{D}_j} = \delta^{(D,\tau,j)}$, for all $j \leq i$ and $\delta \in tsub(\beta) \cup tsub(\gamma)$. Then the following properties hold:*

(i) The relation $r_\alpha^{\hat{D}_i}$ is regular and for all $\bar{a} \in \mathbb{N}^n$ and $y \in \mathbb{N}$,

$$(\bar{a}, y) \in r_\alpha^{\hat{D}_i} \quad \textit{iff} \quad \begin{array}{c}\textit{there is a } j \in [0, i+1) \textit{ such that } y = \tau_i - \tau_j < d \,, \bar{a} \in \gamma^{(D,\tau,j)} \,, \\ \textit{and } \bar{a} \in \beta^{(D,\tau,k)}, \textit{ for all } k \in [j+1, i+1) \,.\end{array}$$

(ii) The relation $p_\alpha^{\hat{D}_i}$ is regular and $p_\alpha^{\hat{D}_i} = \alpha^{(D,\tau,i)}$.

Note that the definition of $r_\alpha^{\hat{D}_i}$ only depends on the relation $r_\alpha^{\hat{D}_{i-1}}$, if $i > 0$, and on the relations in $\hat{D}_i$ for which the corresponding predicates occur in the subformulae of $\hat{\beta}$ or $\hat{\gamma}$. Furthermore, the definition of $p_\alpha^{\hat{D}_i}$ only depends on $r_\alpha^{\hat{D}_i}$.

We now address the bounded future-time operator $\mathcal{U}_I$ with $I = [c, d) \in \mathbb{I}$ and $d \in \mathbb{N}$. Assume that $\alpha = \beta \, \mathcal{U}_I \, \gamma$. For all $i \in \mathbb{N}$, let $\ell_i := \max\{j \in \mathbb{N} \,|\, \tau_{i+j} - \tau_i < d\}$. We call ℓ_i the lookahead offset at time point i. For convenience, let $\ell_{-1} := 0$. To instantiate the relation $p_\alpha^{\hat{D}_i}$, only the relations $p_\delta^{\hat{D}_i}, \ldots, p_\delta^{\hat{D}_{i+\ell_i}}$ are relevant, where $\delta \in tsub(\beta) \cup tsub(\gamma)$. The definition of $p_\alpha^{\hat{D}_i}$ is based on the auxiliary relations $r_\alpha^{\hat{D}_i}$ and $s_\alpha^{\hat{D}_i}$, which we first show how to initialize and update.

We define $r_\alpha^{\hat{D}_i}$ as the union of the sets N_r and U_r. N_r contains the tuples that are new in the sense that they are obtained from data at the time points $i+\ell_{i-1}, \ldots, i+\ell_i$; U_r contains the updated data from the time points $i, \ldots, i + \ell_{i-1} - 1$. Formally, we define

$$N_r := \big\{(\bar{a}, j) \in \mathbb{N}^{n+1} \,\big|\, \ell_{i-1} \leq j \leq \ell_i,\ \bar{a} \in \hat{\gamma}^{\hat{D}_{i+j}}, \text{ and } \tau_{i+j} - \tau_i \geq c\big\}$$

$$U_r := \begin{cases} \big\{(\bar{a}, j) \in \mathbb{N}^{n+1} \,\big|\, (\bar{a}, j+1) \in r_\alpha^{\hat{D}_{i-1}} \text{ and } \tau_{i+j} - \tau_i \geq c\big\} & \text{if } i > 0, \\ \emptyset & \text{otherwise.} \end{cases}$$

Intuitively, $r_\alpha^{\hat{D}_i}$ stores the tuples satisfying the formula $\Diamond_I \, \gamma$ at the time point i, where each tuple in $r_\alpha^{\hat{D}_i}$ is augmented by the index relative to i where the tuple satisfies γ.

Similarly to $r_\alpha^{\hat{D}_i}$, the relation $s_\alpha^{\hat{D}_i}$ is the union of a set N_s for the new elements and a set U_s for the updates. These two sets are defined as

$$N_s := \big\{(\bar{a}, j, j') \in \mathbb{N}^{n+2} \,\big|\, \ell_{i-1} \le j \le j' \le \ell_i \text{ and } \bar{a} \in \hat{\beta}^{\hat{D}_{i+k}},\ \text{for all } k \in [j, j'+1)\big\}$$

and $U_s := \emptyset$ if $i = 0$, and

$$\begin{aligned} U_s := &\big\{(\bar{a}, j, j') \in \mathbb{N}^{n+2} \,\big|\, (\bar{a}, j+1, j'+1) \in s_\alpha^{\hat{D}_{i-1}}\big\} \cup \\ &\big\{(\bar{a}, j, j') \in \mathbb{N}^{n+2} \,\big|\, (\bar{a}, j+1, \ell_{i-1}) \in s_\alpha^{\hat{D}_{i-1}} \text{ and } (\bar{a}, \ell_{i-1}, j') \in N_s\big\} \end{aligned}$$

otherwise. Intuitively, $s_\alpha^{\hat{D}_i}$ stores the tuples and the bounds of the interval (relative to i) in which β is satisfied.

With the relations $r_\alpha^{\hat{D}_i}$ and $s_\alpha^{\hat{D}_i}$ at hand, we define

$$p_\alpha^{\hat{D}_i} := \big\{\bar{a} \in \mathbb{N}^n \,\big|\, (\bar{a}, j) \in r_\alpha^{\hat{D}_i} \text{ and } (\bar{a}, 0, j') \in s_\alpha^{\hat{D}_i},\ \text{for some } j \le j'+1\big\}\,.$$

Lemma 12. *Let $\alpha = \beta\, \mathcal{U}_I\, \gamma$. Assume that $p_\delta^{\hat{D}_k}$ is regular and $p_\delta^{\hat{D}_k} = \delta^{(D,\tau,k)}$, for all $k \le i + \ell_i$ and $\delta \in tsub(\beta) \cup tsub(\gamma)$. Then the following properties hold:*

(i) The relation $r_\alpha^{\hat{D}_i}$ is regular and for all $\bar{a} \in \mathbb{N}$ and $j \in \mathbb{N}$,

$$(\bar{a}, j) \in r_\alpha^{\hat{D}_i} \qquad \text{iff} \qquad \bar{a} \in \gamma^{(D,\tau,i+j)} \text{ and } \tau_{i+j} - \tau_i \in I\,.$$

(ii) The relation $s_\alpha^{\hat{D}_i}$ is regular and for all $\bar{a} \in \mathbb{N}^n$ and $j, j' \in \mathbb{N}$,

$$(\bar{a}, j, j') \in s_\alpha^{\hat{D}_i} \qquad \text{iff} \qquad j \le j',\ \tau_{i+j'} - \tau_i < d, \text{ and } \bar{a} \in \beta^{(D,\tau,i+k)},\ \text{for all } k \in [j, j'+1)\,.$$

(iii) The relation $p_\alpha^{\hat{D}_i}$ is regular and $p_\alpha^{\hat{D}_i} = \alpha^{(D,\tau,i)}$.

Example 13. We illustrate the described transformations and constructions by revisiting the formula

$$\Box\, \forall x. in(x) \to \Diamond_{[0,6)}\, out(x)$$

given in Example 5. We observe that the formula is defined over a signature $S = (\mathsf{C}, \mathsf{R}, a)$, where R contains the unary predicates *in* and *out*.

As a first step, let us remove some syntactic sugar. We obtain the formula

$$\Box\, \neg\exists x. in(x) \land \neg\big(\exists y. y \approx y\, \mathcal{U}_{[0,6)}\, out(x)\big).$$

In order to detect violations of this formula, we negate it to obtain $\Diamond\, \theta$ with

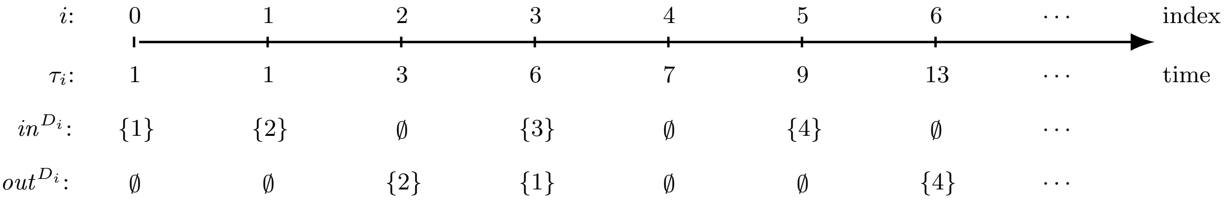
$$\theta := \exists x. in(x) \land \neg\big(\exists y. y \approx y\, \mathcal{U}_{[0,6)}\, out(x)\big).$$

We extend the signature S according to Definition 6. Note that θ contains one temporal subformula, namely $\alpha := \exists y. y \approx y\, \mathcal{U}_{[0,6)}\, out(x)$. Hence, the extended signature $\hat{S}$ is obtained from S by adding the auxiliary predicates p_α, r_α, and s_α to R. According to Definition 7, we transform θ into the first-order formula

$$\hat{\theta} := \exists x. in(x) \land \neg p_\alpha(x).$$

For each time point i, we incrementally build the auxiliary relations $p_\alpha^{\hat{D}_i}$, $r_\alpha^{\hat{D}_i}$, and $s_\alpha^{\hat{D}_i}$ such that the auxiliary predicate p_α is satisfied by exactly those elements that satisfy α at i.

To illustrate how the auxiliary relations are built, let us consider the temporal structure given in Figure 1. Observe that to build the relations $r_\alpha^{\hat{D}_i}$, for $i \ge 0$, we require the relations out^{D_j} with $i \le j \le \ell_i$. Recall that ℓ_i is the lookahead offset at time point i. For example, at $i = 0$ we have that $\ell_0 = 3$ because $\tau_3 - \tau_0 < 6$ and $\tau_4 - \tau_0 = 6$. Hence, to build $r_\alpha^{\hat{D}_0}$, we need to take into account the relations out^{D_0}, out^{D_1}, out^{D_2}, and out^{D_3}. Moreover, as the subformula $\exists y. y \approx y$ is always true we do not depend on any relations to build the relations $s_\alpha^{\hat{D}_i}$, for all $i \ge 0$. For $i = 0$, we thus have $r_\alpha^{\hat{D}_0} :=$



i:	0	1	2	3	4	5	6	$\cdots$ index
τ_i:	1	1	3	6	7	9	13	$\cdots$ time
in^{D_i}:	$\{1\}$	$\{2\}$	$\emptyset$	$\{3\}$	$\emptyset$	$\{4\}$	$\emptyset$	$\cdots$
out^{D_i}:	$\emptyset$	$\emptyset$	$\{2\}$	$\{1\}$	$\emptyset$	$\emptyset$	$\{4\}$	$\cdots$

Fig. 1. Temporal structure used in Example 13.

$\{(1,3),(2,2)\}$ and $s_\alpha^{\hat{D}_0} = \big(\mathbb{N}\times\{0\}\times\{0\}\big)\cup\big(\mathbb{N}\times\{0,1\}\times\{1\}\big)\cup\big(\mathbb{N}\times\{0,1,2\}\times\{2\}\big)\cup\big(\mathbb{N}\times\{0,1,2,3\}\times\{3\}\big)$. The first component of a pair in $r_\alpha^{\hat{D}_i}$ denotes an element occurring in a relation out^{D_j}, with $i \leq j \leq \ell_i$. The second component stores the difference $j - i$ between the respective indices. For example, the pair $(1,3)$ in $r_\alpha^{\hat{D}_0}$ means that the element 1 occurs in out^{D_3}. Similarly, while the first component of a triple in $s_\alpha^{\hat{D}_i}$ denotes the elements for which the subformula $\exists y.y \approx y$ is satisfied, the second and the third components denote the bounds of the interval relative to i for which that element satisfies the formula. Because the subformula $\exists y.y \approx y$ is satisfied by any $a \in \mathbb{N}$, the relation $s_\alpha^{\hat{D}_0}$ contains all tuples (a, j, j') with $a \in \mathbb{N}$ as their first as well as all $j, j' \in \mathbb{N}$ with $0 \leq j \leq j' \leq \ell_0$ as their second and third components. We obtain $p_\alpha^{\hat{D}_0} := \{1,2\}$ by projecting out the first component of the tuples $(a,j) \in r_\alpha^{\hat{D}_0}$ and $(a,0,j') \in s_\alpha^{\hat{D}_0}$ for which the condition $j \leq j'+1$ is satisfied.

We obtain $r_\alpha^{\hat{D}_1}$ from $r_\alpha^{\hat{D}_0}$ by updating the tuples already contained in $r_\alpha^{\hat{D}_0}$ and by possibly adding new tuples to $r_\alpha^{\hat{D}_1}$. Because $\ell_1 = 2$, we must not take any further relations into account and simply update the index component of the tuples that are already contained in $r_\alpha^{\hat{D}_0}$. We thus get $r_\alpha^{\hat{D}_1} := \{(1,2),(2,1)\}$. Similarly, we obtain $s_\alpha^{\hat{D}_1}$ by decreasing the relative indices of the tuples already contained in $s_\alpha^{\hat{D}_0}$ by one. Hence, we have that $s_\alpha^{\hat{D}_1} := \big(\mathbb{N}\times\{0\}\times\{0\}\big) \cup \big(\mathbb{N}\times\{0,1\}\times\{1\}\big) \cup \big(\mathbb{N}\times\{0,1,2\}\times\{2\}\big)$ and $p_\alpha^{\hat{D}_1} := \{1,2\}$.

In addition to updating those tuples already contained in $r_\alpha^{\hat{D}_1}$, to obtain $r_\alpha^{\hat{D}_2}$ we must also take tuples from additional relations into account. Particularly, because $\ell_2 = 2$ also the tuples in out^{D_4} must be considered. As $out^{D_4} = \emptyset$, no new elements are added though. As a result, we get $r_\alpha^{\hat{D}_2} := \{(1,1),(2,0)\}$ by decrementing the indices of the tuples stored in $r_\alpha^{\hat{D}_1}$. Moreover, we have $s_\alpha^{\hat{D}_2} := s_\alpha^{\hat{D}_1}$ and thus obtain $p_\alpha^{\hat{D}_2} := \{1,2\}$.

For $i = 3$, we decrease the second component of the tuples in $r_\alpha^{\hat{D}_2}$ to obtain $r_\alpha^{\hat{D}_3} := \{(1,0)\}$. Note that the tuple $(2,0)$ contained in $r_\alpha^{\hat{D}_2}$ is not carried over to $r_\alpha^{\hat{D}_3}$ because from the viewpoint of $i = 3$ the element 2 occurs in the past. Moreover, we have that $s_\alpha^{\hat{D}_3} := s_\alpha^{\hat{D}_2}$ and thus $p_\alpha^{\hat{D}_3} := \{1\}$. Because $3 \in in^{D_3}$ but $3 \notin p_\alpha^{\hat{D}_3}$, the formula $\forall x.in(x) \rightarrow \Diamond_{[0,6)}\, out(x)$ is violated at $i = 3$.

3.5 Monitor and Correctness

Figure 2 presents the monitor $\mathcal{M}(\phi)$. Without loss of generality, it assumes that each temporal subformula occurs only once in ϕ. In the following, we outline its operation.

The monitor uses two counters i and q. The counter i is the index of the current element (D_i, τ_i) in the input sequence $(D_0,\tau_0),(D_1,\tau_1),\ldots$, which is processed sequentially. Initially, i is 0 and it is incremented at the end of each loop iteration (lines 4–16). The counter $q \leq i$ is the index of the next time point q (possibly in the past, from the point of view of i) for which we evaluate $\neg\hat{\phi}$ over the structure $\hat{D}_q$. The evaluation is delayed until the relations $p_\alpha^{\hat{D}_q}$ for $\alpha \in tsub(\phi)$ are all

```
1: $i \leftarrow 0$        % current index in input sequence $(D_0,\tau_0),(D_1,\tau_1),\dots$
2: $q \leftarrow 0$        % index of next query evaluation in sequence $(D_0,\tau_0),(D_1,\tau_1),\dots$
3: $Q \leftarrow \{(\alpha, 0, \mathit{waitfor}(\alpha)) \mid \alpha$ temporal subformula of $\phi\}$
4: loop
5:     Carry over constants and relations of $D_i$ to $\hat{D}_i$.
6:     for all $(\alpha, j, \emptyset) \in Q$ do        % respect ordering of subformulae
7:         Build relations for $\alpha$ in $\hat{D}_j$ (e.g., build $r_\alpha^{\hat{D}_j}$ and $p_\alpha^{\hat{D}_j}$ if $\alpha = \beta\,\mathcal{S}_I\,\gamma$).
8:         Discard auxiliary relations for $\alpha$ in $\hat{D}_{j-1}$ if $j-1 \geq 0$ (e.g., discard $r_\alpha^{\hat{D}_{j-1}}$ if $\alpha = \beta\mathcal{S}_I\gamma$).
9:         Discard relations $p_\delta^{\hat{D}_j}$, where $\delta$ is a temporal subformula of $\alpha$.
10:    while all relations $p_\alpha^{\hat{D}_q}$ are built for $\alpha \in \mathit{tsub}(\phi)$ do
11:        Output valuations violating $\phi$ at time point $q$, i.e., output $(\neg\hat{\phi})^{\hat{D}_q}$ and $q$.
12:        Discard structure $\hat{D}_{q-1}$ if $q-1 \geq 0$.
13:        $q \leftarrow q+1$
14:    $Q \leftarrow \{(\alpha, i+1, \mathit{waitfor}(\alpha)) \mid \alpha$ temporal subformula of $\phi\} \cup$
           $\{(\alpha, j, \bigcup_{\theta \in \mathit{update}(S, \tau_{i+1}-\tau_i)} \mathit{waitfor}(\theta)) \mid (\alpha, j, S) \in Q$ and $S \neq \emptyset\}$
15:    $i \leftarrow i+1$        % process next element in input sequence $(D_{i+1},\tau_{i+1})$
16: end loop
```

Fig. 2. Monitor $\mathcal{M}(\phi)$

instantiated (lines 10–13). Furthermore, the monitor uses the list[5] Q to ensure that the auxiliary relations of $\hat{D}_0, \hat{D}_1, \dots$ are built at the right time: if $(\alpha, j, \emptyset)$ is an element of Q at the beginning of a loop iteration, enough time has elapsed to build the relations for the temporal subformula α of the structure $\hat{D}_j$. The monitor initializes Q in line 3. The function *waitfor*, defined below, extracts the subformulae that cause a delay of the formula evaluation.

$$\mathit{waitfor}(\theta) := \begin{cases} \mathit{waitfor}(\beta) & \text{if } \theta = \neg\beta,\ \theta = \exists x.\,\beta, \text{ or } \theta = \bullet_I\,\beta, \\ \mathit{waitfor}(\beta) \cup \mathit{waitfor}(\gamma) & \text{if } \theta = \beta \wedge \gamma \text{ or } \theta = \beta\,\mathcal{S}_I\,\gamma, \\ \{\theta\} & \text{if } \theta = \circ_I\,\beta \text{ or } \theta = \beta\,\mathcal{U}_I\,\gamma, \\ \emptyset & \text{otherwise.} \end{cases}$$

The list Q is updated in line 14 before we increment i and start a new loop iteration. For the update, we use the function *update* that is defined as follows for a formula set U and $\Delta \in \mathbb{N}$:

$$\begin{aligned} \mathit{update}(U, \Delta) := {} & \{\beta \mid \circ_I\,\beta \in U\} \cup \{\beta\,\mathcal{U}_{[\max\{0, c-\Delta\}, d-\Delta)}\,\gamma \mid \beta\,\mathcal{U}_{[c,d)}\,\gamma \in U, \text{ with } d-\Delta > 0\} \cup \\ & \{\beta \mid \beta\,\mathcal{U}_{[c,d)}\,\gamma \in U \text{ or } \gamma\,\mathcal{U}_{[c,d)}\,\beta \in U, \text{ with } d-\Delta \leq 0\} \end{aligned}$$

The update adds a new tuple $(\alpha, i+1, \mathit{waitfor}(\alpha))$ to Q, for each temporal subformula α of ϕ, and it removes the tuples of the form $(\alpha, j, \emptyset)$ from Q. Moreover, for tuples (α, j, S) with $S \neq \emptyset$, the set S is updated using the functions *waitfor* and *update* by taking into account the elapsed time to the next time point, i.e. $\tau_{i+1} - \tau_i$.

In lines 6–9, we build the relations for which enough time has elapsed, i.e., the auxiliary relations for α in $\hat{D}_j$ with $(\alpha, j, \emptyset) \in Q$. Since a tuple $(\alpha', j, \emptyset)$ does not occur before a tuple $(\alpha, j, \emptyset)$ in Q, where α is a subformula of α', the relations in $\hat{D}_j$ for α are built before those for α'. To build the relations, we use the incremental constructions described earlier in this section. We thus discard certain relations after we have built the relations for α in $\hat{D}_j$ to reduce space consumption. For instance, if $j > 0$ and $\alpha = \beta\,\mathcal{S}_I\,\gamma$, we discard the relation $r_\alpha^{\hat{D}_{j-1}}$, and we discard $r_\alpha^{\hat{D}_{j-1}}$ and $s_\alpha^{\hat{D}_{j-1}}$ when $\alpha = \beta\,\mathcal{U}_I\,\gamma$.

[5] We abuse notation by using set notation for lists. Moreover, we assume that Q is ordered in that (α, j, S) occurs before (α', j', S'), whenever α is a proper subformula of α', or $\alpha = \alpha'$ and $j < j'$.

In lines 10–13, the valuations violating ϕ at time point q are output together with q, for all q where the relations $p_\alpha^{\hat{D}_q}$ of all immediate temporal subformulae α of ϕ have been built. After an output, the remainder of the extended structure $\hat{D}_{q-1}$ is discarded and q is incremented by 1.

Theorem 14. *The monitor $\mathcal{M}(\phi)$ from Figure 2 has the following properties:*

- *(i) Whenever $\mathcal{M}(\phi)$ outputs $(\neg\hat{\phi})^{\hat{D}_q}$, then $(\neg\hat{\phi})^{\hat{D}_q} = (\neg\phi)^{(D,\tau,q)}$. Furthermore, the set $(\neg\hat{\phi})^{\hat{D}_q}$ is effectively constructable and finitely representable.*
- *(ii) For every $n \in \mathbb{N}$, $\mathcal{M}(\phi)$ eventually sets the counter q to n in some loop iteration.*

4 MFOTL Monitoring with Finite Relations

Throughout this section, we shall assume that all relations are finite. In this case, relational databases provide an alternative to automata for implementing the monitor $\mathcal{M}(\phi)$. When representing relations as finite tables, however, we inherit standard problems from database theory. For a restricted class of formulae, we provide solutions that build upon and extend the work [8,10,11] from the area of temporal databases. Moreover, we analyze the space requirements of monitoring such formulae.

Handling Finite Relations. A *temporal database* is a temporal structure (D, τ) over a signature $S = (\mathsf{C}, \mathsf{R}, a)$, where the domain $|D|$ is infinitely countable and each relation r^{D_i} is finite, for each $r \in \mathsf{R}$ and $i \in \mathbb{N}$ [8, 11]. Our constructions from §3.4 do not work when the auxiliary relations are required to be finite. In particular, Lemmas 9–12 become invalid when replacing "regular" by "finite." The constructed relations are still regular but possibly infinite. In the following, we explain why our constructions fail and sketch our solution. Further details are given in Appendix B.

Consider the formula $p(x) \wedge \blacklozenge_I \neg q(x)$. The subformula $\blacklozenge_I \neg q(x)$ is problematic because, at each time point, our monitor stores the elements that satisfy $\blacklozenge_I \neg q(x)$ in an auxiliary relation. This relation is infinite when the relations for q are finite. However, the entire formula is unproblematic since, at each time point, an element satisfying $\blacklozenge_I \neg q(x)$ must also be in the relation for the predicate p, which is finite. To handle the subformula $\blacklozenge_I \neg q(x)$, we build on work from database theory on domain independence (e.g., [14]), where a similar problem arises with queries containing negation and quantification. A standard solution attempts to rewrite queries so that the quantified variables range only over finitely many elements (see [2]). We generalize this solution for first-order queries to MFOTL formulae whereby we try to rewrite the given MFOTL formula ϕ so that all temporal subformulae and their direct subformulae have only finitely many satisfying valuations. For instance, we rewrite $p(x) \wedge \blacklozenge_I \neg q(x)$ to the equivalent formula $p(x) \wedge \blacklozenge_I \big(\neg q(x) \wedge \lozenge_I\, p(x)\big)$, whose temporal subformulae each have only finitely many satisfying elements. Hence, the transformed formula can be handled by our monitor if the interval I is finite. Recall the requirement that future-time operators are bounded.

After rewriting the formula ϕ, we check, based on the syntax of the result ψ, if each $\theta \in \{\alpha \mid \alpha = \psi, \alpha \text{ is a temporal subformula of } \psi, \text{ or } \alpha \text{ is a direct subformula of a temporal subformula of } \psi\}$ is *temporal domain independent.* If ψ passes this check, we know that ψ can be handled by our monitor for finite relations. Otherwise, no conclusions can be drawn. The notion of temporal domain independence is a natural generalization of the standard notion of domain independence (see [2]). For the ease of exposition, we omit its definition here and refer to Definition 18 in Appendix B.1. Future-bounded temporal domain independent formulae have properties similar to first-order domain independent formulae. For instance, the set $\theta^{(D,\tau,i)}$ is finite, for a future-bounded formula θ, a temporal database (D, τ), and $i \in \mathbb{N}$. Moreover, the set $\theta^{(D,\tau,i)}$ contains only data tuples whose elements are constants or which appear in a finite prefix of (D, τ). The length of the prefix depends on i, τ, and α.

For the remainder of this section, let us assume that the given formula ϕ, all temporal subformulae of ϕ, and all direct subformulae of temporal subformulae of ϕ are temporal domain independent.

Space Requirements. In the following, we analyze our monitor's space requirements. First, observe that the counters q and i of $\mathcal{M}(\phi)$ grow arbitrarily large when processing the sequence $(D_0, \tau_0), (D_1, \tau_1), \dots$. This problem can be partly overcome by replacing these two counters with a single counter that stores $i - q$, i.e., the distance between the current time and the time when the last evaluation of $\neg\hat{\phi}$ took place. Still, $i - q$ can become arbitrarily large if ϕ contains a temporal subformula of the form $\beta\, \mathcal{U}_I\, \gamma$ and the number of time points with the same time stamp in (D, τ) is unbounded, i.e., $\{j' - j \mid \tau_j = \tau_{j+1} = \dots = \tau_{j'} \text{ with } j \leq j'\}$ is infinite. Note that tuples stored in Q also contain indices of the sequence (D, τ). These indices must also be made relative to q.

A problem related to the above one is that the difference between time stamps can be arbitrarily large. If ϕ contains a subformula of the form $\alpha = \beta\, \mathcal{S}_{[c,\infty)}\, \gamma$, the auxiliary relations $r_\alpha^{\hat{D}_0}, r_\alpha^{\hat{D}_1}, \dots$ may need to store tuples whose last component grows with each i. To overcome this problem, we slightly modify the incremental construction of $r_\alpha^{\hat{D}_i}$ for $i > 0$. Namely, the "age" of a tuple $(\bar{a}, y) \in r_\alpha^{\hat{D}_{i-1}}$ is only increased when it is less than c. For this special case, we define

$$r_\alpha^{\hat{D}_i} := \big(\hat{\gamma}^{\hat{D}_i} \times \{0\}\big) \cup \big\{(\bar{a}, \min\{c, z + \tau_i - \tau_{i-1}\}) \in \mathbb{N}^{n+1} \,\big|\, \bar{a} \in \hat{\beta}^{\hat{D}_i} \text{ and } (\bar{a}, z) \in r_\alpha^{\hat{D}_{i-1}}\big\}\,.$$

This new construction ensures that the size of the last component of the tuples in $r_\alpha^{\hat{D}_i}$ is bounded. Similar to Lemma 11, we can prove that this construction has the desired properties. In the following, we assume that the monitor $\mathcal{M}(\phi)$ uses this modified construction.

We can further optimize our construction of $r_\alpha^{\hat{D}_i}$ by removing redundant tuples: whenever $(\bar{a}, y), (\bar{a}, y') \in r_\alpha^{\hat{D}_i}$ with $y < y'$, we remove $(\bar{a}, y)$ from $r_\alpha^{\hat{D}_i}$. Reducing the size of the auxiliary relations together with other optimizations like formula rewriting [10] should substantially improve the practical performance of the monitor. However, these optimizations are irrelevant for the following worst-case analysis and we therefore do not further discuss them here. More details on optimizations are given in Appendix B.3.

We now analyze the sizes of the auxiliary relations stored by our monitor in each loop iteration. We first introduce the following abstract notion for analyzing the resources consumed by monitors in general. Let C be a class of temporal structures over the signature $S = (\mathsf{C}, \mathsf{R}, a)$ and let $\mathit{pre}(C)$ denote the set of nonempty finite prefixes of the temporal structures in C.

Definition 15. *Let $f, g : \mathit{pre}(C) \to \mathbb{N}$ and $s : \mathbb{N} \to \mathbb{N}$ be functions. We write $f \lhd^s g$ if $f(\bar{D}, \bar{\tau}) < s(g(\bar{D}, \bar{\tau}))$, for all $(\bar{D}, \bar{\tau}) \in \mathit{pre}(C)$.*

In our context, the function $f : \mathit{pre}(C) \to \mathbb{N}$ measures the consumption of a particular resource (e.g., storage) of a monitor after it has processed the finite prefix $(\bar{D}, \bar{\tau})$. The function $g : \mathit{pre}(C) \to \mathbb{N}$ measures the size of the prefix $(\bar{D}, \bar{\tau})$. Intuitively, $f \lhd^s g$ means that, at any time point, the resource consumption (measured by f) of the monitor is bounded by the function $s : \mathbb{N} \to \mathbb{N}$ with respect to the size of the processed prefix (measured by g) of an input from C.

In our analysis of the monitor $\mathcal{M}(\phi)$, we use the following concrete functions f and g. Let $(\bar{D}, \bar{\tau}) \in \mathit{pre}(C)$ with $\bar{D} = (D_0, \dots, D_i)$ and $\bar{\tau} = (\tau_0, \dots, \tau_i)$.

- We define $g(\bar{D}, \bar{\tau}) := |\mathit{adom}(\bar{D})|$, where $\mathit{adom}(\bar{D})$ is the so-called active domain of $(\bar{D}, \bar{\tau})$, i.e.,
$$\mathit{adom}(\bar{D}) := \{c^{D_0} \mid c \in \mathsf{C}\} \cup \textstyle\bigcup_{0 \leq k \leq i} \bigcup_{r \in \mathsf{R}} \{d_j \mid (d_1, \dots, d_{a(r)}) \in r^{D_k} \text{ and } 1 \leq j \leq a(r)\}\,.$$
Note that g only counts the number of elements of $\bar{D}$ that are constants or that occur in some of $\bar{D}$'s relations. It ignores the sizes of these elements, the number of times an element appears in $\bar{D}$, and where an element occurs. Moreover, it ignores the time stamps in $\bar{\tau}$.

- We define $f(\bar{D},\bar{\tau})$ to be the sum of the cardinalities of the relations for $r \in \hat{\mathsf{R}}$ stored by $\mathcal{M}(\phi)$ after the $(i+1)$st loop iteration, having processed the input $(D_0,\tau_0),(D_1,\tau_1),\dots,(D_i,\tau_i)$.

Observe that $f \lhd^s g$ is a desirable property of a monitor. Intuitively, it says that the amount of data stored by the monitor does not depend on how long the monitor has been running but only on the number of domain elements that appeared so far. Furthermore, the stored data is bounded by the function s. We remark that the property of a (polynomially) bounded history encoding [8] can be formalized as $f \lhd^s g$, for some (polynomial) $s : \mathbb{N} \to \mathbb{N}$.

Theorem 16. *Let C be a class of temporal databases. Assume that there is some $\ell \in \mathbb{N}$ such that $\max\{j \mid \tau_i = \tau_{i+1} = \ldots = \tau_{i+j}\} < \ell$, for all $(D,\tau) \in C$ and all $i \in \mathbb{N}$. Then, we have that $f \lhd^s g$, where $s : \mathbb{N} \to \mathbb{N}$ is a polynomial of degree $\max\{a(r) \mid r \in \hat{\mathsf{R}}\}$.*

The proof of Theorem 16 is given in Appendix B.4.

Note that if such a bound ℓ on the sequence τ of time stamps does not exist, we cannot guarantee any upper bound on f. To see this, consider the formula $\phi = (p(x)\,\mathcal{U}_{[0,1)}\,q(x)) \wedge (p'(x)\,\mathcal{U}_{[0,2)}\,q'(x))$ and a temporal database (D,τ), where the relations for p, p', q, and q' are all singletons and equal. We have that $g(\bar{D},\bar{\tau}) = 1$, for all finite prefixes $(\bar{D},\bar{\tau})$ of (D,τ). However, if τ contains a sequence $\tau_i,\dots,\tau_{i+j}$ with $\tau_i = \tau_{i+1} = \cdots = \tau_{i+j-1} = 1 - \tau_{i+j}$, we have that $f(\bar{D},\bar{\tau}) \geq j$, for the prefix $(\bar{D},\bar{\tau})$ with $\bar{D} = (D_0,\dots,D_{i+j})$ and $\bar{\tau} = (\tau_0,\dots,\tau_{i+j})$. The reason for this is that our monitor stores at least the nonempty relations $p^{\hat{D}_i}_{p(x)\mathcal{U}_{[0,1)}q(x)},\dots,p^{\hat{D}_{i+j-1}}_{p(x)\mathcal{U}_{[0,1)}q(x)}$ at the end of the $(i+j+1)$st loop iteration. If we can choose j arbitrarily large, we can exceed $s(g(\bar{D},\bar{\tau}))$, for any $s : \mathbb{N} \to \mathbb{N}$.

In [31], Toman describes the use of two-sorted first-order logic (2-FOL) to query temporal databases. In this work, he presents a data-expiration technique to remove irrelevant data with respect to the given property. Note that our incremental constructions in §3.4 can also be seen as a data-expiration technique, since the constructions remove irrelevant data. A monitoring approach using 2-FOL based on Toman's work is bounded by a function with a stack of exponentials [31]. The height of the stack is given by the quantifier depth of the given 2-FOL formula. The polynomial upper bound of the presented MFOTL monitor suggest that MFOTL is better suited for monitoring than 2-FOL whenever the property that needs to be monitored is expressible as MFOTL formula that can be handled by the monitor $\mathcal{M}(\phi)$.

Finally, we remark that it is open whether Theorem 16 can be carried over to temporal structures with possibly infinite relations and automatic representations. To begin with, it is not clear how to define the function g that measures the size of the automatic representations. In particular, g should be independent of the length of the prefix. Moreover, establishing tight upper bounds on the size of automata for automatic structures is difficult and only a few results for specific automatic structures exist (see [19]).

5 Conclusion and Future Work

We have presented an automata-based monitoring approach for an expressive fragment of a metric first-order temporal logic. The use of automata substantially generalizes both the kinds of structures and the class of formulae that can be monitored. Moreover, it eliminates the limitations that arise in databases, where relations must be finite. An interesting question here is to what extent the use of automatic structures can be carried over to other monitoring approaches, thereby solving the problems they have with infinite relations.

One direction for future work is to explore whether our approach can be used to monitor temporal first-order logics that have an interval-based semantics instead of a point-based semantics, or a combined interval and point-based semantics, which is useful for modeling state and event predicates. Another direction is to conduct a refined complexity analysis for our algorithm with automatic structures and to validate our results by implementation and testing. In particular, we plan to design and evaluate data structures and algorithms for efficiently incrementally updating relations, which is at the heart of our monitoring algorithm.

References

1. *Proceedings of the 1st to 8th Workshop on Runtime Verification*, 2001–2008.
2. S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases.* Addison-Wesley, 1995.
3. B. Alpern and F. B. Schneider. Defining liveness. *Inf. Process. Lett.*, 21(4):181–185, 1985.
4. R. Alur and T. Henzinger. Logics and models of real time: A survey. In *Proceedings of the REX Workshop on Real Time: Theory in Practice*, volume 600 of *Lect. Notes in Comput. Sci.*, pages 74–106, 1991.
5. H. Barringer, A. Goldberg, K. Havelund, and K. Sen. Rule-based runtime verification. In *Proceedings of the 5th International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI)*, volume 2937 of *Lect. Notes in Comput. Sci.*, pages 44–57, 2004.
6. A. Bauer, M. Leucker, and C. Schallhart. Monitoring of real-time properties. In *Proceedings of the 26th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, volume 4337 of *Lect. Notes in Comput. Sci.*, pages 260–272, 2006.
7. A. Blumensath and E. Grädel. Finite presentations of infinite structures: Automata and interpretations. *Theory Comput. Syst.*, 37(6):641–674, 2004.
8. J. Chomicki. Efficient checking of temporal integrity constraints using bounded history encoding. *ACM Trans. Database Syst.*, 20(2):149–186, 1995.
9. J. Chomicki and D. Niwiński. On the feasibility of checking temporal integrity constraints. *J. Comput. Syst. Sci.*, 51(3):523–535, 1995.
10. J. Chomicki and D. Toman. Implementing temporal integrity constraints using an active DBMS. *IEEE Trans. on Knowl. and Data Eng.*, 7(4):566–582, 1995.
11. J. Chomicki, D. Toman, and M. H. Böhlen. Querying ATSQL databases with temporal logic. *ACM Trans. Database Syst.*, 26(2):145–178, 2001.
12. B. D'Angelo, S. Sankaranarayanan, C. Snchez, W. Robinson, B. Finkbeiner, H. B. Sipma, S. Mehrotra, and Z. Manna. LOLA: Runtime monitoring of synchronous systems. In *Proceedings of the 12th International Symposium on Temporal Representation and Reasoning (TIME)*, pages 166–174, 2005.
13. N. Dinesh, A. Joshi, I. Lee, and O. Sokolsky. Checking traces for regulatory conformance. In *8th Workshop on Runtime Verification (RV)*, Lect. Notes in Comput. Sci., 2008.
14. R. Fagin. Horn clauses and database dependencies. *J. ACM*, 29(4):952–985, 1982.
15. C. Giblin, A. Y. Liu, S. Müller, B. Pfitzmann, and X. Zhou. Regulations expressed as logical models (REALM). In *Proceedings of the 18th Annual Conference on Legal Knowledge and Information Systems (JURIX)*, volume 134 of *Frontiers in Artificial Intelligence and Applications*, pages 37–48, 2005.
16. K. Havelund and G. Rosu. Efficient monitoring of safety properties. *Int. J. Softw. Tools Technol. Transf.*, 6(2):158–173, 2004.
17. J. Håkansson, B. Jonsson, and O. Lundqvist. Generating online test oracles from temporal logic specifications. *Int. J. Softw. Tools Technol. Transf.*, 4(4):456–471, 2003.
18. B. Khoussainov and A. Nerode. Automatic presentations of structures. In *Proceedings of the International Workshop on Logical and Computational Complexity (LCC)*, volume 960 of *Lect. Notes in Comput. Sci.*, pages 367–392, 1995.
19. F. Klaedtke. On the automata size for Presburger arithmetic. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 110–119, 2004.
20. R. Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Systems*, 2(4):255–299, 1990.
21. K. J. Kristoffersen, C. Pedersen, and H. R. Andersen. Runtime verification of timed LTL using disjunctive normalized equation systems. *Electr. Notes Theor. Comput. Sci.*, 89(2), 2003.
22. O. Lichtenstein, A. Pnueli, and L. D. Zuck. The glory of the past. In *Proceedings of the Conference on Logic of Programs*, volume 193 of *Lect. Notes in Comput. Sci.*, pages 196–218, 1985.
23. U. W. Lipeck and G. Saake. Monitoring dynamic integrity constraints based on temporal logic. *Inf. Syst.*, 12(3):255–269, 1987.
24. O. Maler, D. Nickovic, and A. Pnueli. From MITL to timed automata. In *Proceedings of the 4th International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS)*, volume 4202 of *Lect. Notes in Comput. Sci.*, pages 274–289, 2006.

25. D. Nickovic and O. Maler. AMT: A property-based monitoring tool for analog systems. In *Proceedings of the 5th International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS)*, volume 4763 of *Lect. Notes in Comput. Sci.*, pages 304–319, 2007.
26. M. Roger and J. Goubault-Larrecq. Log auditing through model-checking. In *Proceedings of the 14th IEEE Computer Security Foundations Workshop (CSFW)*, pages 220–234, 2001.
27. G. Rosu and K. Havelund. Rewriting-based techniques for runtime verification. *Autom. Softw. Eng.*, 12(2):151–197, 2005.
28. A. P. Sistla and O. Wolfson. Temporal triggers in active databases. *IEEE Trans. Knowl. Data Eng.*, 7(3):471–486, 1995.
29. O. Sokolsky, U. Sammapun, I. Lee, and J. Kim. Run-time checking of dynamic properties. *Electr. Notes Theor. Comput. Sci.*, 144(4):91–108, 2006.
30. P. Thati and G. Rosu. Monitoring algorithms for metric temporal logic specifications. *Electr. Notes Theor. Comput. Sci.*, 113:145–162, 2005.
31. D. Toman. Logical data expiration. In J. Chomicki, R. van der Meyden, and G. Saake, editors, *Logics for Emerging Applications of Databases*, pages 203–238. Springer, 2003.

A Details for Monitoring Temporal Automatic Structures

In this section, we provide additional details to §3.

A.1 Basic Arithmetic in Automatic Structures

We now show that under the restrictions made in §3.1, we can define different basic arithmetic relations in the first-order logic over an automatic structure in D.

We start by showing that the elements in $|D|$ can be linearly ordered by a regular relation. Let $L \subseteq \Sigma^*$ be the regular language that represents the domain $|D|$, where Σ is some finite alphabet. Without loss of generality, we assume a linear order $\prec_{\text{alph}}$ on Σ. We lift $\prec_{\text{alph}}$ to linearly order the elements in Σ^*. For $w, w' \in \Sigma^*$, we define $w \prec_* w'$ iff $|w| < |w'|$, or $|w| = |w'|$ and $w \prec_{\text{lex}} w'$, where $|u|$ denotes the length of a word $u \in \Sigma^*$ and $\prec_{\text{lex}}$ is the lexicographical ordering on Σ^* with respect to the ordering $\prec_{\text{alph}}$ on the alphabet Σ.

It is easy to see that $\prec_*$ can be recognized by an automaton by reading the letters of w and w' synchronously. More formally, the language $O := \{w \otimes w' \,|\, w \prec_* w'\}$ is regular. Here, $u \otimes v$ is the *convolution* of the words $u, v \in \Sigma^*$. See [7], for further details. $O \cap L$ is a regular language, which we can use to order the elements in $|D|$. For $a, b \in |D|$, we define $a <_* b$ iff $u \prec_* v$, where u and v represent a and b, respectively. Recall that we assume that the domain representation is injective, i.e., every element in $|D|$ has a unique representative in L.

Obviously, the ordering $<_*$ is regular and $(|D|, <_*)$ is isomorphic to $(\mathbb{N}, <)$. In the following, we assume that $|D| = \mathbb{N}$ and $<_* = <$.

We already have seen that we can define the relation $\text{succ} := \{(x,y) \in \mathbb{N}^2 \,|\, y = x+1\}$ in $\text{FO}(\mathbb{N}, <)$. For $d \in \mathbb{N}$, the formula

$$\exists z_0. \ldots \exists z_d.\, x \approx z_0 \wedge y \approx z_d \wedge \bigwedge_{0 \le i < d} \text{succ}(z_i, z_{i+1})$$

defines the relation $\{(x,y) \in \mathbb{N}^2 \,|\, x + d = y\}$. From these relations, it follows that the relations $\{(x,y) \in \mathbb{N}^2 \,|\, x - y = d\}$, $\{(x,y) \in \mathbb{N}^2 \,|\, x - y \le d\}$, and $\{(x,y) \in \mathbb{N}^2 \,|\, |x-y| \le d\}$ are all regular.

A.2 Proof Details of §3

In this subsection, we provide the omitted proofs from §3. We start with Lemmas 11 and 12, which establish the correctness of the constructions for the temporal operators $\mathcal{S}_I$ and $\mathcal{U}_I$, respectively. Then, we provide the proof of Theorem 14, which establishes the correctness of the monitor $\mathcal{M}(\phi)$ presented in Fig. 2.

Throughout this section, v_0 denotes any valuation. Moreover, we use Lemma 8 without explicitly referring to it.

Proof of Lemma 11. Property (ii) follows immediately from (i) and the definition of $p_\alpha^{\hat{D}_i}$. We prove (i) by induction over i.

Base case $i = 0$: The set $r_\alpha^{\hat{D}_0}$ is regular, since it can be defined by the formula

$$\psi(\bar{x}, y) := \hat{\gamma}(\bar{x}) \wedge \neg\exists z.\ \text{succ}(z, y)\,.$$

Note that, by assumption, the relations for the predicates in $\hat{\gamma}$ are regular.

The equivalence for $i = 0$ follows directly from the definition of $r_\alpha^{\hat{D}_0}$. Note that $\tau_i - \tau_i < d$, since in the definition of the syntax of MFOTL, we require that $I \neq \emptyset$. Hence, $d > 0$.

Step case $i > 0$: We first show that $r_\alpha^{\hat{D}_i}$ is regular. Similar to the base case, it follows that the set $S := \hat{\gamma}^{\hat{D}_i} \times \{0\}$ is regular. The set $T := \{(\bar{a}, y) \in \mathbb{N}^{n+1} \,|\, y < d,\ \bar{a} \in \hat{\beta}^{\hat{D}_i}, \text{ and } (\bar{a}, y') \in r_\alpha^{\hat{D}_{i-1}}, \text{ for } y' = y - \tau_i + \tau_{i-1}\}$ is also regular. It can be expressed by the first-order formula

$$\psi(\bar{x}, y) := y \prec d \wedge \hat{\beta}(\bar{x}) \wedge \exists y'.\, \psi'(\bar{x}, y') \wedge y' + (\tau_i - \tau_{i-1}) \approx y\,,$$

where ψ' is the formula that defines $r_\alpha^{\hat{D}_{i-1}}$, which is regular by the induction hypothesis. Note that d and $\tau_i - \tau_{i-1}$ are constant values and not variables. Since $r_\alpha^{\hat{D}_i}$ is defined as the union of S and T, we conclude that $r_\alpha^{\hat{D}_i}$ is regular.

Now, we show the step case for the other claim.

($\Rightarrow$) If the tuple $(\bar{a}, y)$ is in S, then the claim is obviously true. Assume that $(\bar{a}, y) \in T$. By definition, there is a tuple $(\bar{a}, y')$ in $r_\alpha^{\hat{D}_{i-1}}$ such that $y' = y - \tau_i + \tau_{i-1}$. By the induction hypothesis, we have that

$$\begin{gathered}\exists j \in [0, i) : y' = \tau_{i-1} - \tau_j < d,\ (D, \tau, v_0[\bar{x}/\bar{a}], j) \models \gamma,\ \text{and}\\ \forall k \in [j+1, i) : (D, \tau, v_0[\bar{x}/\bar{a}], k) \models \beta\,.\end{gathered}$$

It follows that $y = y' + \tau_i - \tau_{i-1} = \tau_i - \tau_j$. From the assumption, we conclude that $(D, \tau, v_0[\bar{x}/\bar{a}], k) \models \beta$, for all k with $j < k \leq i$.

($\Leftarrow$) If $j = i$, it follows that $y = 0$. From the assumption and the definition of $r_\alpha^{\hat{D}_i}$, it follows that $(\bar{a}, 0) \in r_\alpha^{\hat{D}_i}$. Assume that $j < i$. By the induction hypothesis, $(\bar{a}, y') \in r_\alpha^{\hat{D}_{i-1}}$ with $y' = y - (\tau_i - \tau_{i-1})$. From the definition of $r_\alpha^{\hat{D}_i}$ and the assumption, we conclude that $(\bar{a}, y) \in r_\alpha^{\hat{D}_i}$.

Proof of Lemma 12. Property (iii) follows immediately from (i), (ii), and the definition of $p_\alpha^{\hat{D}_i}$.

Let us first prove (i), which we do by induction over i.

Base case $i = 0$: We first show that $r_\alpha^{\hat{D}_i}$ is regular. It suffices to show that the set N_r is regular. The regularity of N_r can be seen as follows. For $j \in \mathbb{N}$ with $\ell_{i-1} \leq j \leq \ell_i$, we define $N_r^j := \emptyset$ if $\tau_{i+j} - \tau_i < c$, and $N_r^j := \hat{\gamma}^{\hat{D}_{i+j}} \times \{j\}$ otherwise. Obviously, N_r^j is regular. Since $N_r = \bigcup_{\ell_{i-1} \leq j \leq \ell_i} N_r^j$, we conclude that N_r is regular.

Note that by definition of ℓ_i we have $\tau_{\ell_i} - \tau_j < d$, for every $j \leq \ell_i$. The equivalence follows directly from the definition of N_r.

Step case $i > 0$: We first show that $r_\alpha^{\hat{D}_i}$ is regular. It suffices to show that the sets N_r and U_r are regular. The regularity of N_r can be shown as in the base case. The regularity of U_r can be seen as follows. The set $H := \{j \in \mathbb{N} \,|\, \ell_{i-1} \leq j \leq \ell_i \text{ and } \tau_{i+j} - \tau_j \geq c\}$ is regular, since it is finite. The set U_r can be defined by the formula $h(z) \wedge \exists z'.\, \mathrm{succ}(z, z') \wedge r(\bar{x}, z')$, where h denotes the formula that defines H and r denotes the formula that defines the set $r_\alpha^{\hat{D}_{i-1}}$, which is regular by induction hypothesis.

If $j \geq \ell_{i-1}$, the equivalence follows, similar to the base case, directly from the definition of ℓ_i. In the following, assume $j < \ell_{i-1}$.

($\Rightarrow$) We have that $(\bar{a}, j) \in U_r$. By definition, $(\bar{a}, j+1) \in r_\alpha^{\hat{D}_{i-1}}$ and $\tau_{i+j} - \tau_i \geq c$. By the induction hypothesis, $\bar{a} \in \hat{\gamma}^{\hat{D}_{i-1+j+1}}$ and $\tau_{i-1+j+1} - \tau_{i-1} \in I$. Since the difference of the time stamps from $i-1$ to $i+j$ and from i to $i+j$ decreases, we have that $\tau_{i+j} - \tau_i < d$. We are done since $\tau_{i+j} - \tau_i \geq c$ by the definition of U_r.

($\Leftarrow$) From the definition of ℓ_i it follows that $\tau_{i-1+j+1} - \tau_{i-1} < d$. Hence, $\bar{a} \in \hat{\gamma}^{\hat{D}_{i-1+j+1}}$ and $\tau_{i-1+j+1} - \tau_{i-1} \in I$. By the induction hypothesis, $(\bar{a}, j+1) \in r_\alpha^{\hat{D}_{i-1}}$. From the definition of U_r, we conclude that $(\bar{a}, j) \in r_\alpha^{\hat{D}_i}$.

Let us now prove (ii), which we do again by induction over i.

Base case $i = 0$: We first show that $s_\alpha^{\hat{D}_i}$ is regular. It suffices to show that the set N_s is regular. To see that N_s is regular, let $N_s^{j,j'} := \bigcap_{j\leq k\leq j'} \hat{\beta}^{\hat{D}_{i+k}}$, for $j, j' \in \mathbb{N}$ with $\ell_{i-1} \leq j \leq j' \leq \ell_i$. Obviously, $N_s^{j,j'}$ is regular and $N_s = \bigcup_{\ell_{i-1}\leq j\leq j'\leq \ell_i}(N_s^{j,j'} \times \{(j,j')\})$. We conclude that N_s is regular.

Note that by the definition of ℓ_i we have $\tau_{\ell_i} - \tau_{j'} < d$, for every $j' \leq \ell_i$. The equivalence follows directly from the definition of N_s.

Step case $i > 0$: We first show that $r_\alpha^{\hat{D}_i}$ is regular. It suffices to show that the sets N_s and U_s are regular. The regularity of N_s can be shown as in the base case. The regularity of U_s can be seen as follows. The formula

$$\big(\exists y'.\, \exists z'.\, \mathrm{succ}(y,y') \wedge \mathrm{succ}(z,z') \wedge s(\bar{x},y',z')\big) \vee \\ \big(\exists y'.\, \exists z'.\, \mathrm{succ}(y,y') \wedge z' \approx \ell' \wedge s(\bar{x},y',z') \wedge n(\bar{x},z',z)\big)$$

defines U_r, where n is the formula that defines the set N_s and s is the formula that defines $s_\alpha^{\hat{D}_i - 1}$, which is regular by the induction hypothesis. Note that ℓ_{i-1} is a constant.

If $j' \geq \ell_{i-1}$, the equivalence follows, similar to the base case, directly from the definition of ℓ_i. For $j < \ell_{i-1}$, it suffices to prove that $(\bar{a}, j, j') \in U_s$ iff $j \leq j'$, $\tau_{i+j'} - \tau_i \leq d$, and $\bar{a} \in \hat{\gamma}^{\hat{D}_{i+k}}$, for all $k \in [j, j'+1)$ The proof is similar as for (i). We omit it.

Correctness of the Monitor $\boldsymbol{\mathcal{M}(\phi)}$. For the proof of Theorem 14, we index the program variable Q of the monitor $\mathcal{M}(\phi)$ from Figure 2 by the counter i. That means, Q_i denotes the list when we enter (line 4) the $(i+1)$st loop iteration. Analogously, we index the program variable q with i: q_i is the value when the enter the $(i+1)$st loop iteration.

Proof (Theorem 14). In the following, assume that $(D_0, \tau_0), (D_1, \tau_1), \dots$ is the input sequence of the monitor $\mathcal{M}(\phi)$ and $\phi \in \mathsf{MFOTL}$ is a future-bounded input formula. Moreover, let T be the set of temporal subformulae of ϕ.

We start with some observations about the monitoring algorithm $\mathcal{M}(\phi)$. Let $\alpha \in T$ and $i, j \in \mathbb{N}$.

(1) If $(\alpha, j, S), (\alpha, j, S') \in Q_i$ then $S = S'$. This follows immediately from the initialization (line 3) and the update (line 14) of the list Q.

(2) We have that $(\alpha, i, \mathit{waitfor}(\alpha)) \in Q_i$. This directly follows from the initialization of Q (line 3) and the update of Q (line 14).

(3) If $(\alpha, j, S) \in Q_i$ then there is an integer $i' \geq i$ such that $(\alpha, j, \emptyset) \in Q_{i'}$. This follows from the update of Q (line 14) (in particular, from the application of the functions *waitfor* and *update*), and because the sequence of time stamps τ is monotonically increasing and it makes progress. Note that we only remove a tuple (α, j, S) from Q_i if $S = \emptyset$ and after the relations for α in $\hat{D}_j$ have been built (lines 7 and 14).

From (2) and (3), it follows that for every $\alpha \in T$ and $j \in \mathbb{N}$, we eventually execute line 7, where we build the relations for α in the structure $\hat{D}_j$. From (1), it follows that line 7 is executed at most once for $\alpha \in T$ and $j \in \mathbb{N}$. It follows that for each $q \in \mathbb{N}$, we execute line 9 exactly once in a run of $\mathcal{M}(\phi)$.

Furthermore, observe that the relations $p_\alpha^{\hat{D}_j}$, for $\alpha \in \mathit{tsub}(\phi)$ are only discarded at line 12 of the monitoring algorithm. We conclude that for every value of the counter q, the condition of the while loop (line 10) will eventually become true in some loop iteration. Hence, the counter q will always eventually be increased by 1. We conclude that the second property (ii) of the theorem holds.

We now turn to the property (i) of the theorem. We need the following definition of the *temporal rank* of a formula θ:

$$rank(\theta) := \begin{cases} rank(\theta') & \text{if } \theta = \neg\theta' \text{ or } \theta = \exists x.\, \theta', \\ \max\{rank(\theta'), rank(\theta'')\} & \text{if } \theta = \theta' \wedge \theta'', \\ 1 + rank(\theta') & \text{if } \theta = \bullet_I\, \theta' \text{ or } \theta = \bigcirc_I\, \theta' \\ 1 + \max\{rank(\theta'), rank(\theta'')\} & \text{if } \theta = \theta'\, \mathcal{S}_I\, \theta'' \text{ or } \theta = \theta'\, \mathcal{U}_I\, \theta'', \\ 0 & \text{otherwise.} \end{cases}$$

For the remainder of the proof, let us assume that $(\alpha, j, \emptyset) \in Q_i$. The fact that $q_i \leq j$ holds at the beginning of the $(i+1)$st loop iteration is easily established by an induction over i. In the following, we prove by induction over $rank(\alpha)$ that for the construction of the relations of α in $\hat{D}_j$, the necessary relations (according to the incremental constructions given in §3.4) have been built earlier and have not yet been discarded. From the lemmas in §3.4 about these constructions, it follows that $p_\alpha^{\hat{D}_j} = p_\alpha^{(D,\tau,j)}$ and $p_\alpha^{\hat{D}_j}$ is regular. From this, we then conclude that the monitor $\mathcal{M}(\phi)$ has the property (i) of the theorem.

Base Case: $rank(\alpha) = 1$*.* We make a case split on α's main connective.

- $\alpha = \bullet_I\, \beta$: We have that $tsub(\beta) = \emptyset$. Hence, the construction of the relation $p_\alpha^{\hat{D}_j}$ requires no auxiliary relations. Moreover, if $j > 0$, the atomic relations $r^{\hat{D}_{j-1}}$ with $r \in \mathsf{R}$ have not been discarded. This follows from the fact that $q_i \leq j$ and that a relation $r^{\hat{D}_{j-1}}$ is only discarded in line 12. Observe that in line 12, we discard $\hat{D}_{q-1}$ and not $\hat{D}_q$.
- $\alpha = \beta\, \mathcal{S}_I\, \gamma$: We have that $tsub(\beta) \cup tsub(\gamma) = \emptyset$. Similarly to the above case, the atomic relations $r^{\hat{D}_j}$ for $r \in \mathsf{R}$ have not been discarded. Moreover, for $j > 0$ the auxiliary relation $r_\alpha^{\hat{D}_{j-1}}$ has been built earlier and has not been discarded. The fact that it has been built earlier follows from the ordering of the tuples in the list Q and the fact that there is an $i' \leq i$ such that $(\alpha, j-1, \emptyset) \in Q_{i'}$. The latter fact easily follows from an induction over i by using the initialization and the updates of the list Q.
- $\alpha = \bigcirc_I\, \beta$: Since $tsub(\beta) = \emptyset$, we only have to check that a relation $r^{\hat{D}_{j+1}}$ with $r \in \mathsf{R}$ is available. Because of the initialization and the update of the list Q, we have that $j + 1 = i$ and $(\alpha, j, \emptyset) \in Q_{j+1}$. Thus, the relation $r^{\hat{D}_{j+1}}$ is available.
- $\alpha = \beta\, \mathcal{U}_I\, \gamma$: Let I be the interval $[c, d)$. Since $tsub(\beta) \cup tsub(\gamma) = \emptyset$, it suffices to check that for all $r \in \mathsf{R}$ and $k < \max\{k' \in \mathbb{N} \mid \tau_{j+k'} - \tau_j < d\}$, the relation $r^{\hat{D}_{j+k}}$ is available. This follows from the initialization and the updates of the list Q. As in the case for the since operator, we conclude that the relations $r_\alpha^{\hat{D}_{j-1}}$ and $s_\alpha^{\hat{D}_{j-1}}$ are available.

Step Case: $rank(\alpha) > 1$*.* We make again a case split on the main connective of α.

- $\alpha = \bullet_I\, \beta$: For $j = 0$, there is nothing to prove since $p_\alpha^{\hat{D}_j} = \emptyset$. Let $j > 0$. As in the corresponding case of the base case, we have that the relations $r^{\hat{D}_{j-1}}$ with $r \in \mathsf{R}$ are available.
  Let $\delta \in tsub(\beta)$. There is an $i' \leq i$ such that $(\delta, j-1, \emptyset) \in Q_{i'}$. This fact easily follows from an induction over i by using the initialization and the updates of the list Q. Due to the ordering of the list Q and the induction hypothesis, we have that $p_\delta^{\hat{D}_{j-1}}$ has been built earlier. It has not yet been discarded because this only happens in line 9 or line 12, i.e., after the relation $p_\alpha^{D_j}$ has been built. Note that $q_i \leq j$.
- $\alpha = \beta\, \mathcal{S}_I\, \gamma$: This case uses a similar argumentation as the corresponding case of the base case for the relations $r^{\hat{D}_j}$ with $r \in \mathsf{R}$ and $r_\alpha^{\hat{D}_{j-1}}$. For the relations $p_\delta^{\hat{D}_j}$ with $\delta \in tsub(\beta) \cup tsub(\gamma)$, we use a similar argumentation as in the case for the previous operator.

- $\alpha = \bigcirc_I \beta$: As in the corresponding case of the base case, the relations $r^{\hat{D}_{j+1}}$ with $r \in \mathsf{R}$ are available. Let $\delta \in tsub(\beta)$. There is an $i' \leq i$ such that $(\delta, j+1, \emptyset) \in Q_{i'}$. This fact easily follows from an induction over i by using the induction hypothesis on δ and by using the initialization and the updates of the list Q.
- $\alpha = \beta\, \mathcal{U}_I\, \gamma$: Let I be the interval $[c, d)$ and let $\ell_j := \max\{k' \in \mathbb{N} \mid \tau_{j+k'} - \tau_j < d\}$. Recall that ℓ_j is the lookahead offset at time point j. As in the corresponding case of the base case, the relations $r^{\hat{D}_{j+k}}$ with $r \in \mathsf{R}$ and $k \leq \ell_j$ are available. Moreover, when $j > 0$ we conclude as in the base case that the relations $r_\alpha^{\hat{D}_{j-1}}$ and $s_\alpha^{\hat{D}_{j-1}}$ are available.
  Let $\delta \in tsub(\beta) \cup tsub(\gamma)$. By the induction hypothesis, we have that the relations when building the relations for δ of $\hat{D}_{j+k}$ with $k \leq \ell_j$ are available. It suffices to show that for all $k \leq \ell_j$, there is an $i' \leq i$ such that $(\delta, j+k, \emptyset) \in Q_{i'}$. This follows easily from the initialization and the updates of the list Q. $\dashv$

B Details for MFOTL Monitoring with Finite Relations

In this section, we provide additional details for §4. Recall that a temporal database is a temporal structure (D, τ) over a signature $S = (\mathsf{C}, \mathsf{R}, a)$, where the domain $|D|$ is infinitely countable and each relation r^{D_i} is finite, for each $r \in \mathsf{R}$ and $i \in \mathbb{N}$ [8, 11]. We point out that the relations for $\approx$ and $\prec$ are infinite.

B.1 Non-Strict Operators and Temporal Domain Independence

For reasons to become clear later, we now also provide strict versions of the binary temporal operators $\mathcal{S}_I$ and $\mathcal{U}_I$. The semantics of their strict counterparts $\dot{\mathcal{U}}_I$ and $\dot{\mathcal{S}}_I$ are defined as follows.

Definition 17. *Let (D, τ) be a temporal structure over a signature S, with $D = (D_0, D_1, \dots)$ and $\tau = (\tau_0, \tau_1, \dots)$, β and γ formulae over S, v a valuation, and $i \in \mathbb{N}$. We define:*

$$\begin{array}{lll} (D,\tau,v,i) \models (\beta\, \dot{\mathcal{S}}_I\, \gamma) & \textit{iff} & \textit{for some } j < i,\ \tau_i - \tau_j \in I,\ (D,\tau,v,j) \models \gamma, \\ & & \textit{and } (D,\tau,v,k) \models \beta,\ \textit{for all } k \in [j+1, i+1) \\ (D,\tau,v,i) \models (\beta\, \dot{\mathcal{U}}_I\, \gamma) & \textit{iff} & \textit{for some } j > i,\ \tau_j - \tau_i \in I,\ (D,\tau,v,j) \models \gamma, \\ & & \textit{and } (D,\tau,v,k) \models \beta,\ \textit{for all } k \in [i, j) \end{array}$$

In addition, we need the dual temporal operators $\mathcal{R}_I$ (release) and $\mathcal{T}_I$ (trigger), which are defined as $\beta\, \mathcal{R}_I\, \gamma := \neg(\neg\beta\, \mathcal{U}_I\, \neg\gamma)$ and $\beta\, \mathcal{T}_I\, \gamma := \neg(\neg\beta\, \mathcal{S}_I\, \neg\gamma)$. Moreover, we need the strict versions of the derived operators $\Box_I$, $\Diamond_I$, $\blacksquare_I$, $\blacklozenge_I$, $\mathcal{R}_I$, and $\mathcal{T}_I$, which are derived from $\dot{\mathcal{S}}_I$ and $\dot{\mathcal{U}}_I$ analogously to their non-strict counterparts. We decorate them with a dot to distinguish them from the non-strict versions, e.g., we write $\dot{\blacksquare}_I \phi$. We now define MFOTL^+ to be the set of formulae obtained from Definition 2 and the operators $\dot{\mathcal{S}}_I$ and $\dot{\mathcal{U}}_I$, where the (strict) derived temporal operators $\dot{\Diamond}_I, \dot{\blacklozenge}_I, \dot{\Box}_I, \dot{\blacksquare}_I, \dot{\mathcal{R}}_I$, and $\dot{\mathcal{T}}_I$, the (non-strict) derived temporal operators $\Diamond_I, \blacklozenge_I, \Box_I, \blacksquare_I, \mathcal{R}_I$ and $\mathcal{T}_I$, the derived Boolean connectives $\vee, \rightarrow$, and $\leftrightarrow$, and the universal quantifier $\forall$ are treated as primitives. Note that in the metric case, the strict and the non-strict versions of the operators since and until cannot be derived from each other.

The following function returns the set of direct subformulae of a temporal formula. For a formula $\alpha \in \mathsf{MFOTL}^+$, we define

$$dstf(\alpha) := \begin{cases} \{\beta\} & \text{if } \alpha = \otimes\beta \text{, where } \otimes \text{ is an unary temporal operator,} \\ \{\beta, \gamma\} & \text{if } \alpha = \beta \oplus \gamma \text{, where } \oplus \text{ is a binary temporal operator,} \\ \emptyset & \text{otherwise.} \end{cases}$$

We say that $i \in \mathbb{N}$ is *minimal* for $q \in \mathbb{N}$, the temporal database (D, τ) with $D = (D_0, D_1, \dots)$, and the bounded formula θ if for all valuations v, we have that

$$(D, \tau, v, q) \models \theta \qquad \text{iff} \qquad (D', \tau, v, q) \models \theta\,,$$

where $D' = (D'_0, D'_1, \dots)$ with $D'_j = D_j$, for all $j \leq i$. Since, at any time point, θ can only refer to time points finitely far into the future and τ makes progress, there is always a minimal i, for every given q, (D, τ), and θ. The *active domain* of $\bar{D} = (D_1, \dots, D_i)$ is

$$adom(\bar{D}) := \{c^{D_0} \mid c \in \mathsf{C}\} \cup \bigcup_{0 \leq k \leq i} \bigcup_{r \in \mathsf{R}} \{d_j \mid (d_1, \dots, d_{a(r)}) \in r^{D_k} \text{ and } 1 \leq j \leq a(r)\}\,.$$

We write $\models_{\mathsf{U}}$ to denote the relation $\models$, as defined in Definition 2, but quantification is relativized to the set $\mathsf{U} \subseteq |D|$. In the following, let v_0 be an arbitrary valuation.

Definition 18. *Let θ be a future-bounded formula with the free variables given by the vector $\bar{x} = (x_1, \dots, x_n)$. Moreover, let T be the set of temporal subformulae of θ.*

(i) The formula θ is temporal domain independent *if for all temporal databases (D, τ), all $i, q \in \mathbb{N}$ with $q \leq i$, and all $\mathsf{U}, \mathsf{U}' \subseteq |D|$, we have that*

$$\left\{\bar{d} \in \mathsf{U}^n \,\middle|\, (D, \tau, v_0[\bar{x}/\bar{d}], q) \models_{\mathsf{U}} \theta\right\} = \left\{\bar{d} \in \mathsf{U}'^n \,\middle|\, (D, \tau, v_0[\bar{x}/\bar{d}], q) \models_{\mathsf{U}'} \theta\right\},$$

whenever $adom(\bar{D}) \subseteq \mathsf{U}, \mathsf{U}'$ with $\bar{D} = (D_1, \dots, D_q, \dots, D_i)$ and i is minimal for q, (D, τ), and θ.

(ii) The formula θ is temporal subformula domain independent *(*TSF *domain independent, for short) if (1) is θ temporal domain independent, (2) each $\alpha \in T$ is temporal domain independent, and (3) each $\delta \in dstf(\alpha)$ is temporal subformula independent, for all $\alpha \in T$.*

As noted by Chomicki [8], the semantics of the temporal operators may influence TSF domain independence. For example, by using the strict temporal operator $\dot{\mathcal{S}}_{[0,\infty)}$ a larger set of properties can be specified for which the formulae are TSF domain independent formulae [8]. To illustrate this, consider the formula $p(x, y)\dot{\mathcal{S}}_{[0,\infty)}q(x)$, which is TSF domain independent. Now consider the logically equivalent formula $\bullet_{[0,\infty)}(p(x, y)\, \mathcal{S}_{[0,\infty)}\, q(x)) \wedge p(x, y)$, which is not TSF domain independent. This can be seen as follows. Assume that $(D, \tau, v_0[x/a], 0) \models q(x)$, for some temporal database (D, τ) and $a \in |D|$. Then $(D, \tau, v_0[(x, y)/(a, b)], 0) \models p(x, y)\mathcal{S}_{[0,\infty)}q(x)$, for any $b \in |D|$. Hence, $p(x, y)\mathcal{S}_{[0,\infty)}q(x)$ is not temporal domain independent, which follows from the following lemma.

Lemma 19. *Let θ be a future-bounded formula, (D, τ) a temporal database, and $q \in \mathbb{N}$. If θ is temporal domain independent, then $\theta^{(D,\tau,q)}$ is finite.*

Proof. Let $i \in \mathbb{N}$ be minimal for q, (D, τ), and θ. Let $\bar{D} = (D_0, \dots, D_q, \dots, D_i)$. We have that

$$\left\{\bar{d} \in |D|^n \,\middle|\, (D, \tau, v_0[\bar{x}/\bar{d}], q) \models \theta\right\} = \left\{\bar{d} \in adom(\bar{D})^n \,\middle|\, (D, \tau, v_0[\bar{x}/\bar{d}], q) \models_{adom(\bar{D})} \theta\right\}.$$

Since $adom(\bar{D})$ is finite, we are done. ⊣

B.2 Monitorable MFOTL Formulae

In this subsection, we describe a procedure that analyzes MFOTL formulae and attempts to rewrite them such that they can be monitored by our monitoring approach. Throughout this section, let (D, τ) be a temporal database over the signature $S = (\mathsf{C}, \mathsf{R}, a)$, where we assume that $|D| = \mathbb{N}$ and $<$ is the standard ordering.

By definition, at every time point the interpretations of a bounded TSF domain independent formula $\phi \in \mathsf{MFOTL}^+$, all its temporal subformulae α, and all direct subformulae of temporal subformulae $\delta \in dstf(\alpha)$, for all α, are finite. Under this condition, ϕ can be monitored with our monitoring approach for temporal databases. However, even for a first-order formula it is undecidable whether it is domain independent [2]. In the following, we present a procedure based on rewriting that identifies a subset of the class of TSF domain independent formulae, which we name TSF *safe-range formulae.* Our procedure extends the one for non-metric first-order temporal logic [11] in two ways. First, we extend the procedure to the metric temporal logic MFOTL. Second, we improve the procedure to recognize a larger set of formulae by defining additional rewrite rules for the strict dual temporal operators $\dot{\mathcal{R}}_I$ and $\dot{\mathcal{T}}_I$, with $I \in \mathbb{I}$. We remark that our procedure presumes that dedicated incremental update constructions have been defined for all primitive and derived temporal operators. The constructions are similar to those for $\mathcal{S}_I$ and $\mathcal{U}_I$ given in §3.4.

Overview. We now outline the main steps required to check whether a formula $\phi \in \mathsf{MFOTL}^+$ can be monitored with our method.

1. *Normalization.* In the first step, we transform ϕ into a logically equivalent formula $\phi' \in \mathsf{MFOTL}^+$, where all quantified variables have unique names, some syntactic sugar is unfolded, double negations are removed, and negations are pushed towards the leaves of the subformulae by applying rewrite rules.
2. *Safe-range check.* We then check if the sets of free and range restricted variables in ϕ' coincide and if for all subformulae of the form $\exists x.\, \psi$, where the variable x occurs free in ψ, it holds that x is range restricted in ψ. If this is the case, ϕ' is safe-range and we proceed to step 3. Otherwise, ϕ' is rejected.
3. *Propagation of range restrictions.* Next, we transform ϕ' into a logically equivalent formula $\phi'' \in \mathsf{MFOTL}^+$ by propagating all range restrictions towards all subformulae of ϕ' by applying rewrite rules.
4. TSF *safe-range check.* In the final step, we check if all temporal subformulae of ϕ'', and all their direct subformulae are safe-range. If this is the case, ϕ'' is TSF *safe-range.* Otherwise, ϕ'' is not TSF safe-range and is rejected. Finally, we check that all future-time operators in ϕ'' are bounded. If this is also the case, ϕ'' and can be monitored.

In the following, we describe these steps in detail.

Normalization. In the following, let $fv(\theta)$ denote the set of free variables of the formula θ. In the first step, the input formula $\phi \in \mathsf{MFOTL}^+$ is transformed into the logically equivalent formula $sr(\phi)$ defined in Definition 20.

Definition 20. *We denote as $sr(\phi)$ the formula obtained from $\phi \in \mathsf{MFOTL}^+$*

(1) by renaming the quantified variables using unique names;

(2) by replacing the occurrences of subformulae of the form $\forall x.\, \beta$, $\beta \rightarrow \gamma$, and $\beta \leftrightarrow \gamma$ by $\neg\exists x.\, \neg\alpha$, $\neg\alpha \vee \beta$, and $(\neg\alpha \vee \beta) \wedge (\neg\beta \vee \alpha)$, respectively;

(3) by pushing down negations and removing double negations by applying the following rules as long as possible:

(a) $\neg\neg\alpha \mapsto \alpha$

(b) $\exists x.\alpha \mapsto \alpha$, *if* $x \notin fv(\alpha)$

(c) $\neg(\alpha \vee \beta) \mapsto \neg\alpha \wedge \neg\beta$ *and* $\neg(\alpha \wedge \beta) \mapsto \neg\alpha \vee \neg\beta$

(d) $\neg\dot{\Diamond}_I\alpha \mapsto \dot{\Box}_I\neg\alpha$, $\neg\dot{\Box}_I\alpha \mapsto \dot{\Diamond}_I\neg\alpha$, $\neg\dot{\blacklozenge}_I\alpha \mapsto \dot{\blacksquare}_I\neg\alpha$, *and* $\neg\dot{\blacksquare}_I\alpha \mapsto \dot{\blacklozenge}_I\neg\alpha$

(e) $\neg\Diamond_I\alpha \mapsto \Box_I\neg\alpha$, $\neg\Box_I\alpha \mapsto \Diamond_I\neg\alpha$, $\neg\blacklozenge_I\alpha \mapsto \blacksquare_I\neg\alpha$, *and* $\neg\blacksquare_I\alpha \mapsto \blacklozenge_I\,\neg\alpha$

(f) $\neg \bigcirc_I \alpha \mapsto \bigcirc_I \neg\alpha$

(g) $\neg(\beta\dot{\mathcal{S}}_I\gamma) \mapsto \neg\beta\dot{\mathcal{T}}_I\neg\gamma$ *and* $\neg(\beta\dot{\mathcal{U}}_I\gamma) \mapsto \neg\beta\dot{\mathcal{R}}_I\neg\gamma$

(h) $\neg(\beta\dot{\mathcal{T}}_I\gamma) \mapsto \neg\beta\dot{\mathcal{S}}_I\neg\gamma$ *and* $\neg(\beta\dot{\mathcal{R}}_I\gamma) \mapsto \neg\beta\dot{\mathcal{U}}_I\neg\gamma$

(i) $\neg(\beta\,\mathcal{S}_I\,\gamma) \mapsto \neg\beta\,\mathcal{T}_I\,\neg\gamma$ *and* $\neg(\beta\,\mathcal{U}_I\,\gamma) \mapsto \neg\beta\,\mathcal{R}_I\,\neg\gamma$

(j) $\neg(\beta\,\mathcal{T}_I\,\gamma) \mapsto \neg\beta\,\mathcal{S}_I\,\neg\gamma$ *and* $\neg(\beta\,\mathcal{R}_I\,\gamma) \mapsto \neg\beta\,\mathcal{U}_I\,\neg\gamma$

It is easy to see that $sr(\phi)$ is logically equivalent to ϕ, since every step preserves logical equivalence. Moreover, observe that the rewrite rules are terminating. Finally, note that the Boolean connective $\neg$ cannot be moved over an existential quantifier $\exists$.

Lemma 21. *Let $\phi \in \mathsf{MFOTL}^+$ be a formula, v a valuation, and $i \in \mathbb{N}$. It holds that $(D, \tau, v, i) \models \phi$ iff $(D, \tau, v, i) \models sr(\phi)$.*

Safe-range Check. For $\alpha \in \mathsf{MFOTL}^+$, we define $rr(\alpha)$ as follows, where x, x' range over V and c over C.

$$rr(\alpha) := \begin{cases} \{x\} & \text{if } \alpha = x \approx c,\ c \approx x, \\ & \quad \text{or } \alpha = x \prec c \text{ if } c' \in \mathsf{C} \text{ for all } c'^D < c^D, \\ \{t_i \,|\, t_i \in \mathsf{V} \land 1 \leq i \leq a(r)\}, & \text{if } \alpha = r(t_1, \dots, t_{a(r)}), \\ \emptyset & \text{if } \alpha = \neg\beta, \alpha = c \prec x, \alpha = x \approx x', \alpha = x \prec x', \\ & \quad \text{or } \alpha = x \prec c \text{ if } c' \notin \mathsf{C} \text{ for some } c'^D < c^D, \\ rr(\beta) \setminus \{x\} & \text{if } \alpha = \exists x.\, \beta \text{ and } x \in rr(\beta), \\ rr(\beta) & \text{if } \alpha = \otimes\beta \text{ with } \otimes \in \{\bullet_I, \dot{\blacklozenge}_I, \blacklozenge_I, \dot{\blacksquare}_I, \blacksquare_I\} \\ & \qquad\qquad\qquad \cup \{\bigcirc_I, \dot{\lozenge}_I, \lozenge_I, \dot{\square}_I, \square_I\}, \\ rr(\beta) \cup rr(\gamma) & \text{if } \alpha = \beta \land \gamma,\ \alpha = \beta \dot{\mathcal{S}}_I \gamma,\ \text{or } \alpha = \beta \dot{\mathcal{U}}_I \gamma, \\ rr(\beta) \cap rr(\gamma) & \text{if } \alpha = \beta \lor \gamma, \alpha = \beta \dot{\mathcal{T}}_I \gamma, \text{ or } \alpha = \beta \dot{\mathcal{R}}_I \gamma, \\ rr(\gamma) & \text{if } \alpha = \beta\, \mathcal{S}_I\, \gamma, \alpha = \beta\, \mathcal{U}_I\, \gamma, \alpha = \beta\, \mathcal{T}_I\, \gamma, \\ & \quad \text{or } \alpha = \beta\, \mathcal{R}_I\, \gamma. \end{cases}$$

Note that rr is only applied to formulae $sr(\phi)$, where $\phi \in \mathsf{MFOTL}^+$. In particular, this means that universal quantifiers and Boolean connectives like → have been replaced according to Definition 20.

Definition 22. *The formula α is called* safe-range *if $rr(\alpha) = fv(\alpha)$ and for every subformula of α of the form $\exists x.\beta$, it holds that $x \in fv(\beta)$ implies $x \in rr(\beta)$.*

Because of the following lemma, we do not check whether ϕ is safe-range but we check whether the normalized formula $sr(\phi)$ is safe-range.

Lemma 23. *Let $\phi \in \mathsf{MFOTL}^+$ be safe-range. Then, $sr(\phi)$ is also safe-range.*

Proof. Follows directly from Definition 2 and Definition 20. Note that $rr(\neg\beta) = \emptyset$ for any $\beta \in \mathsf{MFOTL}^+$. Hence, pulling the negation inwards increases the chances that a formula becomes safe-range. $\dashv$

Because of their less favorable properties in terms of restricting the ranges of free variables and because of brevity, in the following we refrain from presenting rewrite rules for the non-strict operators $\mathcal{S}_I$, $\mathcal{U}_I$, $\mathcal{T}_I$, and $\mathcal{R}_I$. An extension to the non-strict versions is straightforward.

Propagation of Range Restrictions. We now describe how range restrictions can be propagated towards the subformulae of a safe-range formula.To this end, the following logical equivalences allow us to move range-restricting subformulae between the left- and right-hand sides of the $\dot{\mathcal{S}}_I$ and $\dot{\mathcal{U}}_I$ operators and to move a range-restricting formula into the scope of a temporal connective.

Lemma 24. *Let $\alpha, \beta, \gamma \in \mathsf{MFOTL}^+$. The following logical equivalences hold:*

1. $\alpha \dot{\mathcal{U}}_I \beta \equiv \alpha \dot{\mathcal{U}}_I (\blacklozenge_I \alpha \land \beta)$

1.	$\alpha \wedge (\beta \vee \gamma) \mapsto (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$	(push range restriction into $\vee$)
2.	$\alpha \wedge (\beta \dot{\mathcal{R}}_I \gamma) \mapsto \alpha \wedge ((\beta \wedge \blacklozenge_I\, \alpha) \dot{\mathcal{R}}_I (\gamma \wedge \blacklozenge_I\, \alpha)$	(push range restriction into $\dot{\mathcal{R}}_I$)
3.	$\alpha \wedge (\beta \dot{\mathcal{T}}_I \gamma) \mapsto \alpha \wedge ((\beta \wedge \lozenge_I\, \alpha) \dot{\mathcal{T}}_I (\gamma \wedge \lozenge_I\, \alpha)$	(push range restriction into $\dot{\mathcal{T}}_I$)
4.	$\alpha \wedge \exists x.\beta \mapsto \alpha \wedge \exists x.(\alpha \wedge \beta)$	(push range restriction into $\exists$)
5.	$\alpha \wedge \neg\beta \mapsto \alpha \wedge \neg(\alpha \wedge \beta)$	(push range restriction into $\neg$)
6.	$(\alpha \wedge \gamma)\dot{\mathcal{S}}_I\beta \mapsto (\alpha \wedge \gamma)\dot{\mathcal{S}}_I(\dot{\lozenge}_I\alpha \wedge \beta)$	(distribute from left to right in $\dot{\mathcal{S}}_I$)
7.	$(\alpha \wedge \gamma)\dot{\mathcal{U}}_I\beta \mapsto (\alpha \wedge \gamma)\dot{\mathcal{U}}_I(\dot{\blacklozenge}_I\alpha \wedge \beta)$	(distribute from left to right in $\dot{\mathcal{U}}_I$)
8.	$\beta\dot{\mathcal{S}}_I(\alpha \wedge \gamma) \mapsto (\dot{\blacklozenge}_I\alpha \wedge \beta)\dot{\mathcal{S}}_I(\alpha \wedge \gamma)$	(distribute from right to left in $\dot{\mathcal{S}}_I$)
9.	$\beta\dot{\mathcal{U}}_I(\alpha \wedge \gamma) \mapsto (\dot{\lozenge}_I\alpha \wedge \beta)\dot{\mathcal{U}}_I(\alpha \wedge \gamma)$	(distribute from right to left in $\dot{\mathcal{U}}_I$)
10.	$\alpha \wedge (\beta\dot{\mathcal{S}}_I\gamma) \mapsto \alpha \wedge ((\lozenge_I\, \alpha \wedge \beta)\dot{\mathcal{S}}_I\gamma)$	(push into $\dot{\mathcal{S}}_I$, left side)
11.	$\alpha \wedge (\beta\dot{\mathcal{U}}_I\gamma) \mapsto \alpha \wedge ((\blacklozenge_I\, \alpha \wedge \beta)\dot{\mathcal{U}}_I\gamma)$	(push into $\dot{\mathcal{U}}_I$, left side)
12.	$\alpha \wedge (\gamma\dot{\mathcal{S}}_I\beta) \mapsto \alpha \wedge (\gamma\dot{\mathcal{S}}_I(\dot{\lozenge}_I\alpha \wedge \beta)$	(push into $\dot{\mathcal{S}}_I$, right side)
13.	$\alpha \wedge (\gamma\dot{\mathcal{U}}_I\beta) \mapsto \alpha \wedge (\gamma\dot{\mathcal{U}}_I(\dot{\blacklozenge}_I\alpha \wedge \beta)$	(push into $\dot{\mathcal{U}}_I$, right side)
14.	$\alpha \wedge \dot{\blacklozenge}_I\beta \mapsto \alpha \wedge \dot{\blacklozenge}_I(\dot{\lozenge}_I\alpha \wedge \beta)$	(push into $\dot{\blacklozenge}_I$)
15.	$\alpha \wedge \dot{\lozenge}_I\beta \mapsto \alpha \wedge \dot{\lozenge}_I(\dot{\blacklozenge}_I\alpha \wedge \beta)$	(push into $\dot{\lozenge}_I$)
16.	$\alpha \wedge \dot{\blacksquare}_I\beta \mapsto \alpha \wedge \dot{\blacksquare}_I(\dot{\lozenge}_I\alpha \wedge \beta)$	(push into $\dot{\blacksquare}_I$)
17.	$\alpha \wedge \dot{\square}_I\beta \mapsto \alpha \wedge \dot{\square}_I(\dot{\blacklozenge}_I\alpha \wedge \beta)$	(push into $\dot{\square}_I$)
18.	$\alpha \wedge \blacklozenge_I\, \beta \mapsto \alpha \wedge \blacklozenge_I(\lozenge_I\, \alpha \wedge \beta)$	(push into $\blacklozenge_I$)
19.	$\alpha \wedge \lozenge_I\, \beta \mapsto \alpha \wedge \lozenge_I(\blacklozenge_I\, \alpha \wedge \beta)$	(push into $\lozenge_I$)
20.	$\alpha \wedge \blacksquare_I\, \beta \mapsto \alpha \wedge \blacksquare_I(\lozenge_I\, \alpha \wedge \beta)$	(push into $\blacksquare_I$)
21.	$\alpha \wedge \square_I\, \beta \mapsto \alpha \wedge \square_I(\blacklozenge_I\, \alpha \wedge \beta)$	(push into $\square_I$)
22.	$\alpha \wedge \bullet_I\, \beta \mapsto \alpha \wedge \bullet_I(\bigcirc_I\, \alpha \wedge \beta)$	(push into $\bullet_I$)
23.	$\alpha \wedge \bigcirc_I\, \beta \mapsto \alpha \wedge \bigcirc_I(\bullet_I\, \alpha \wedge \beta)$	(push into $\bigcirc_I$)

Fig. 3. Rewrite rules for ra. The above rules are used when x is a variable that is range restricted in the subformula α (i.e., $x \in rr(\alpha)$) and free but not range restricted in the subformula β (i.e., $x \in fv(\beta) \setminus rr(\beta)$).

2. $\alpha\dot{\mathcal{U}}_I\beta \equiv (\alpha \wedge \dot{\lozenge}_I\beta)\dot{\mathcal{U}}_I\beta$
3. $\alpha\dot{\mathcal{S}}_I\beta \equiv \alpha\dot{\mathcal{S}}_I(\dot{\lozenge}_I\alpha \wedge \beta)$
4. $\alpha\dot{\mathcal{S}}_I\beta \equiv (\alpha \wedge \dot{\blacklozenge}_I\beta)\dot{\mathcal{S}}_I\beta$
5. $\alpha \wedge (\beta\dot{\mathcal{U}}_I\gamma) \equiv \alpha \wedge (\beta\dot{\mathcal{U}}_I(\dot{\blacklozenge}_I\alpha \wedge \gamma))$
6. $\alpha \wedge (\beta\dot{\mathcal{U}}_I\gamma) \equiv \alpha \wedge ((\beta \wedge \blacklozenge_I\, \alpha)\dot{\mathcal{U}}_I\gamma)$
7. $\alpha \wedge (\beta\dot{\mathcal{S}}_I\gamma) \equiv \alpha \wedge (\beta\dot{\mathcal{S}}_I(\dot{\lozenge}_I\alpha \wedge \gamma))$
8. $\alpha \wedge (\beta\dot{\mathcal{S}}_I\gamma) \equiv \alpha \wedge ((\beta \wedge \lozenge_I\, \alpha)\dot{\mathcal{S}}_I\gamma)$

Proof. Follows directly from Definitions 2 and 17. Note that the equivalences 6. and 8. make use of the *non-strict* operators $\blacklozenge_I$ and $\lozenge_I$. ⊣

Definition 25. *Let ϕ be a safe-range formula. We denote as $ra(\phi)$ the result of applying the rules of commutativity and associativity, and the rules stated in Fig. 3, starting from the top-most main connective.*

Note that some of the rewrite rules given in Fig. 3 may lead to formulae that cannot be monitored. For example, consider rule 6, which describes how to rewrite a formula of the form $(\alpha \wedge \gamma)\dot{\mathcal{S}}_I\beta$ into the logically equivalent formula $(\alpha \wedge \gamma)\dot{\mathcal{S}}_I(\dot{\lozenge}_I\alpha \wedge \beta)$. If we have $I = [c, d)$, where $d \in \mathbb{N}$, there is no problem and the rewritten formula can be handled using our monitor. If, however, we have $I = [c, \infty)$, the rewritten formula, while preserving logical equivalence, is not bounded any more and thus cannot be monitored by our approach. Generally speaking, whenever a formula containing an unbounded past operator (i.e., $I = [0, \infty)$) is rewritten, the rewrite rules in Fig. 3 may generate a logically equivalent formula that is no longer bounded. Because of this, the rules 6, 10, 12, 14, 16, 18, and 20 require that $I = [c, d)$, where $d \in \mathbb{N}$.

Lemma 26. *Let $\phi \in \mathsf{MFOTL}^+$ be a safe-range formula, v a valuation, and $i \in \mathbb{N}$. We have that*

$$(D, \tau, v, i) \models \phi \qquad \textit{iff} \qquad (D, \tau, v, i) \models ra(sr(\phi))\,.$$

Proof. Follows from Definitions 2 and 17, Lemma 24 and standard equivalences for first-order logic. $\dashv$

TSF Safe-range Check. A safe-range formula $\theta \in \mathsf{MFOTL}^+$ is TSF *safe-range* if each temporal subformula α of θ and each direct subformula $\delta \in \mathit{dstf}(\alpha)$ is safe-range.

For a safe-range formula $\phi \in \mathsf{MFOTL}^+$, we check if if $ra(\phi)$ is TSF safe-range. This can be done by examining if each temporal subformula α of $ra(\phi)$ and each direct subformula $\delta \in \mathit{dstf}(\alpha)$ is safe-range and bounded.

Lemma 27. *Let ϕ be a bounded formula in MFOTL^+. If ϕ is* TSF *safe-range then it is* TSF *domain independent.*

Proof. Assume that $\delta \in \mathsf{MFOTL}^+$ is bounded. It is straightforward to see by induction over the formula structure that if δ is safe-range, then only finitely many valuations satisfy δ. Hence, δ is temporal domain independent. $\dashv$

B.3 Optimizations

In the following, we present details of the optimizations for the auxiliary relations as briefly sketched in §4.

Deletion of redundant data tuples. To minimize the size of the relations, we can optimize our incremental structure construction by removing redundant data tuples from auxiliary relations. For example, consider the formula $\alpha = \beta\, \mathcal{S}_{[0,\infty)}\, \gamma$ and assume that $\bar{a}$ satisfies γ and β at all time points. In this case, the relation for r_α in $\hat{D}_i$ stores the tuples $(\bar{a}, y)$, with $y = \tau_i - \tau_j$ for all $j \leq i$. However, it suffices to store only one of these tuples.

More generally, for $\alpha = \beta \mathcal{S}_I \gamma$, we can optimize the construction as follows. If $(\bar{a}, y), (\bar{a}, y') \in r_\alpha^{\hat{D}_i}$ with $y, y' \in I$ and $y > y'$, then we can remove $(\bar{a}, y)$ from $r_\alpha^{\hat{D}_i}$. This follows by an easy inductive argument. Since $y, y' \in I$, both tuples satisfy the condition of our construction so that $\bar{a}$ is put into the relation $p_\alpha^{\hat{D}_i}$. Moreover, if the updated version of $(\bar{a}, y)$ is in $r_\alpha^{\hat{D}_{i+1}}$, then also the updated version of $(\bar{a}, y')$ is in $r_\alpha^{\hat{D}_{i+1}}$, and we have that $y + \tau_{i+1} - \tau_i > y' + \tau_{i+1} - \tau_i$. Again, both updated tuples satisfy the condition such that $\bar{a}$ is put into the relation $p_\alpha^{\hat{D}_{i+1}}$.

Similar optimizations apply to the case where $\alpha = \beta\, \mathcal{U}_I\, \gamma$. There the relation $s_\alpha^{\hat{D}_i}$ may contain redundant elements. Namely, if $(\bar{a}, j_1, j_1'), (\bar{a}, j_2, j_2') \in s_\alpha^{\hat{D}_i}$ with $[j_1, j_1') \subsetneq [j_2, j_2')$ then we can remove $(\bar{a}, j_1, j_1')$ from $s_\alpha^{\hat{D}_i}$. After removing such elements, $s_\alpha^{\hat{D}_i}$ only contains tuples where the intervals given by the last coordinates of an element in $s_\alpha^{\hat{D}_i}$ is maximal. When filtering out these elements, we need to adjust the update of $s_\alpha^{\hat{D}_i}$ slightly, since we can no longer ignore elements of the form $(\bar{a}, 0, j') \in s_\alpha^{\hat{D}_{i-1}}$. Note that the optimization has removed the element $(\bar{a}, 1, j')$ from $s_\alpha^{\hat{D}_{i-1}}$.

Assume that $I = [c, d)$. Another optimization is to remove a tuple $(\bar{a}, j, j')$ in $s_\alpha^{\hat{D}_i}$ if $\tau_{i+j'+1} - \tau_i \geq c$ and $j' < \ell_i$. Here, ℓ_i is the lookahead offset at time instant i, i.e., $\ell_i := \max\{j \in \mathbb{N} \,|\, \tau_{i+j} - \tau_i < d\}$. Note that the semantics of the $\mathcal{U}_I$ operator requires that $\bar{a}$ has to satisfy γ at the time instant $i + j'$. Since this time instant is too close at time instant i, the timing constraint given by the I is violated. We remark that we cannot remove the element $(\bar{a}, j, j')$ if $j' = \ell_i$ since we might need the information that $\bar{a}$ satisfies β in the time instants $i+j, \ldots, i+\ell_i$ when updating the relation for s_α.

Finally, we can use the relation $r_\alpha^{\hat{D}_i}$ to eliminate elements in $s_\alpha^{\hat{D}_i}$. Namely, we can eliminate $(\bar{a}, j, j')$ in $s_\alpha^{\hat{D}_i}$ if $j' < \ell_i$ and when there is no $(\bar{a}, k) \in r_\alpha^{\hat{D}_i}$ with $j \leq k$ and $k - 1 \leq j'$.

Dedicated constructions for derived operators. Another kind of optimization is to tune the above definitions for the auxiliary predicates for certain kinds of formulae. For instance, if $\alpha = \Diamond_I \gamma$ (i.e., $\alpha = \mathit{true}\, \mathcal{U}_I\, \gamma$) then we do not need the auxiliary relations for s_α at all. Furthermore, some of the tuples can be removed from $r_\alpha^{\hat{D}_i}$. Namely, if $(\bar{a}, j)$ and $(\bar{a}, j')$ are in $r_\alpha^{\hat{D}_i}$, with $j < j'$, then $(\bar{a}, j)$ can be removed from $r_\alpha^{\hat{D}_i}$. This can be seen by an argument similar to the one we gave when optimizing relations that handle the operator $\mathcal{S}_I$.

Algebraic optimization. The rewriting techniques given for past-only first-order temporal logic in [10] can be extended to MFOTL. Thereby, we can reduce the number of auxiliary relations created from an input formula and also minimize their arity. For example, by rewriting the formula $\exists x. \bigcirc_I \beta$ to $\bigcirc_I \exists x.\beta$ we reduce the arity of the auxiliary relation $p^{\hat{D}_i}_{\bigcirc_I \exists x.\beta}$ by one. Similarly, by rewriting the formula $\bigcirc_I \CIRCLE_I \beta$ to β we can minimize the number of auxiliary relations created. Under certain conditions, formulae containing nested metric operators with different intervals can also be rewritten. For example, if $c \geq d'$, the formula $\Diamond_{[c,d)} \blacklozenge_{[c',d')} p(x)$ can be rewritten to $\Diamond_{[c-d',d-c')} p(x)$.

Context-based optimization. We can also reduce the number of tuples stored in the auxiliary relations by analyzing the contexts in which the relations are used and by then restricting the definitions of the auxiliary relations with appropriate *magic conditions* as described in [10]. For example, for a formula $x \prec 10 \wedge (q(x) \vee \beta(x)\, \mathcal{U}_I\, \gamma(x))$, we can adapt the definitions of the auxiliary relations $r^{\hat{D}_i}_{\beta(x)\mathcal{U}_I\gamma(x)}$ and $s^{\hat{D}_i}_{\beta(x)\mathcal{U}_I\gamma(x)}$ such that only those elements that satisfy the condition $x \prec 10$ are stored. Note that the availability of future operators in MFOTL requires slight modifications of the definitions used in [10].

B.4 Proof Details of Theorem 16

Recall that ϕ is a TSF domain independent formula (see Definition 18) and (D, τ) a temporal database over the signature $S = (\mathsf{C}, \mathsf{R}, a)$, i.e., all relations for $r \in \mathsf{R}$ are finite.

Lemma 28. *Let α be a temporal subformula of ϕ or a direct subformula of a temporal subformula of ϕ. Assume that $\mathcal{M}(\phi)$ constructs the relation $p_\alpha^{\hat{D}_j}$ in the $(i + 1)$st loop iteration on the input (D, τ). Then, $\alpha^{(\hat{D},\tau,j)} \subseteq \mathit{adom}(\bar{D})^n$, where $\bar{D} = (D_0, \dots, D_i)$ and n is the number of free variables of α.*

Proof. Since ϕ is TSF domain independent, we have that α is temporal domain independent. From the correctness of the monitor $\mathcal{M}(\phi)$ follows that i is minimal for j, (D, τ), and α. Similar as in Lemma 19, we show that $\alpha^{(D,\tau,j)} \subseteq \mathit{adom}(\bar{D})^n$. ⊣

Proof (Theorem 16). Throughout the proof, we assume that the monitor $\mathcal{M}(\phi)$ processes the temporal structure (D, τ) from the class C. Let us first make the following observation. If $q = 0$, the monitor stores in the $(i+1)$st iteration at most the relations $r^{\hat{D}_q}, r^{\hat{D}_{q+1}}, \dots, r^{\hat{D}_i}$, for $r \in \hat{\mathsf{R}}$. If $q > 0$, the monitor might additionally store the relations $r^{\hat{D}_{q-1}}$, for $r \in \hat{\mathsf{R}}$. Without loss of generality, we assume that $q > 0$. We now proceed as follows. (1) We establish an upper bound on $i - q$. (2) We establish an upper bound on the cardinalities of the relations $r^{\hat{D}_j}$ with $q - 1 \leq j \leq i$.

Let us start with (1). Recall that ℓ is the bound on the number of equal time stamps in the sequence τ. Namely, we have that $\max\{j \mid \tau_k = \tau_{k+1} = \dots = \tau_{k+j}\} < \ell$, for all $k \in \mathbb{N}$. The bound of $i - q$ depends on ℓ and the future-time operators that occur in ϕ. First, observe that since there are at most ℓ equal time stamps in τ, the monitor postpones the evaluation of a formula $p(x)\, \mathcal{U}_{[c,d)}\, q(x)$ at the time point i by at most $\ell \cdot d$ time steps. Furthermore, note that a formula of the form $\bigcirc_I p(x)$

is postponed by one time step. Taking the nesting of subformulae with temporal operators into account, we define

$$\mathit{maxwaitfor}(\theta) := \begin{cases} \mathit{maxwaitfor}(\beta) & \text{if } \theta = \neg\beta,\ \theta = \exists x.\, \beta, \text{ or } \theta = \CIRCLE_I\, \beta, \\ \max\{\mathit{maxwaitfor}(\beta), \mathit{maxwaitfor}(\gamma)\} & \text{if } \theta = \beta \wedge \gamma \text{ or } \theta = \beta\, \mathcal{S}_I\, \gamma, \\ 1 + \mathit{maxwaitfor}(\beta) & \text{if } \theta = \bigcirc_I\, \beta, \\ \ell d + \max\{\mathit{maxwaitfor}(\beta), \mathit{maxwaitfor}(\gamma)\} & \text{if } \theta = \beta\, \mathcal{U}_{[c,d)}\, \gamma, \\ 0 & \text{otherwise.} \end{cases}$$

From the initialization of the list Q and the updates of Q in each loop iteration, it follows by induction that $i - q \leq \mathit{maxwaitfor}(\phi)$.

Let us now turn to (2). We define m as $\max\{a(r) \,|\, r \in \mathsf{R}\}$ and k as the maximum of $1 + \mathit{maxwaitfor}(\phi)$ and the maximal upper bound of an interval of a since operator in ϕ, where we set the maximum of the upper bound of an interval $[c, \infty)$ in a since operator to c.

Assume that the monitor constructs a relation $r^{\hat{D}_j}$ for $r \in \hat{\mathsf{R}}$ at time point i, i.e., in the $(i+1)$st loop iteration. In the following, we give an upper bound on the cardinality of $r^{\hat{D}_j}$. First note that the data elements $d \in |D|$ that occur in $r^{\hat{D}_j}$ also occur in $\mathit{adom}(\bar{D})$, where $\bar{D} = (D_0, \dots, D_i)$ and $\bar{\tau} = (\tau_0, \dots, \tau_i)$. This follows from Lemma 28.

- $r \in \mathsf{R}$ or $r = p_\alpha$, where α is a temporal formula: We have that $|r^{\hat{D}_j}| \leq |\mathit{adom}(\bar{D})|^m$.
- $r = r_\alpha$, where α has the form $\beta\, \mathcal{S}_I\, \gamma$: Recall that in this case $r^{\hat{D}_j}$ consists of tuples of the form $(\bar{a}, y)$ with $y \leq k$. Note that $y \leq k$ holds because our optimized construction for the temporal operator $\mathcal{S}_{[c,\infty)}$. We have that $|r^{\hat{D}_j}| \leq |\mathit{adom}(\bar{D})|^m \cdot k$.
- $r = r_\alpha$, where α has the form $\beta\, \mathcal{U}_I\, \gamma$: Recall that in this case $r^{\hat{D}_j}$ consists of tuples of the form $(\bar{a}, j)$ with $j \leq k$. We have that $|r^{\hat{D}_j}| \leq |\mathit{adom}(\bar{D})|^m \cdot k$.
- $r = s_\alpha$, where α has the form $\beta\, \mathcal{U}_I\, \gamma$: Recall that in this case $r^{\hat{D}_j}$ consists of tuples of the form $(\bar{a}, j, j')$ with $j, j' \leq k$. We have that $|r^{\hat{D}_j}| \leq |\mathit{adom}(\bar{D})|^m \cdot k^2$.

We conclude that $|r^{\hat{D}_j}| \leq |\mathit{adom}(\bar{D})|^m \cdot k^2$, for all $r \in \hat{\mathsf{R}}$.

Form this and the upper bound on $i - q$, we obtain that $f(\bar{D}, \bar{\tau}) \leq |\mathit{adom}(\bar{D})|^m \cdot |\hat{\mathsf{R}}| \cdot k^3$. Since k only depends on ℓ and ϕ, we have that $f \lhd^s g$, for $s(x) := c \cdot x^m$ with the constant $c := |\hat{\mathsf{R}}| \cdot k^3$. ⊣