Monitoring Data-aware Temporal Properties

Alessandro Gianola[1] , **Marco Montali**[2] , **Sarah Winkler**[2]
[1]INESC-ID/Instituto Superior Técnico, Universidade de Lisboa, Portugal
[2]Free University of Bozen-Bolzano, Italy
alessandro.gianola@tecnico.ulisboa.pt, {montali, winkler}@inf.unibz.it

Abstract

Dynamic systems in AI are often complex and heterogeneous, so that an internal specification is not accessible and verification techniques such as model checking are not applicable. Monitoring is in such cases an attractive alternative, as it evaluates desirable properties along traces generated by an unknown dynamic system. In this work, we consider anticipatory monitoring of linear-time properties enriched with an arbitrary SMT theory over finite traces (LTL_f^{MT}). Anticipatory monitoring in this setting is highly challenging, as the monitoring state depends on both the trace prefix seen so far and all its possible finite continuations. Under reasonable assumptions on the background theory, we present and formally prove the correctness of a novel foundational framework for monitoring properties in an expressive fragment of LTL_f^{MT}. The framework combines automata-theoretic methods to handle the temporal aspects of the logic, with automated reasoning techniques to address the first-order dimension. Moreover, we identify for the first time decidable fragments of this monitoring problem that are practically relevant as they combine linear arithmetic with uninterpreted functions, which covers e.g. data-aware business processes and dynamic systems operating over a read-only database. Feasibility is witnessed by a prototype implementation and preliminary evaluation.

1 Introduction

Complex, dynamic AI systems increasingly comprise autonomous agents and heterogeneous components with unknown, inaccessible, or opaque internal specifications. This hampers the possibility of ascertaining their safety and trustworthiness through traditional design-time verification methods such as model checking and testing. A remedy in such situations is to keep track of desired properties at runtime, by *monitoring* the executions of the black-box system under scrutiny. A widely known, solid approach to monitoring is that of *runtime verification* (RV), where given a logical property, a corresponding monitor can be automatically constructed, with the guarantee that it will emit a provably correct verdict [Leucker and Schallhart, 2009]. These advantages have fostered the application of RV in various AI contexts, including multiagent systems [Dastani *et al.*, 2018; Alechina *et al.*, 2018], robotics [Begemann *et al.*, 2023], cyber-physical systems [Tracy *et al.*, 2020], and business processes [Ly *et al.*, 2015; Calvanese *et al.*, 2022a]. The vast majority of such approaches specify properties using variants of linear-time temporal logic (LTL).

Leucker and Schallhart (2009) point out two key insights in monitoring: First, since the trace τ seen so far is necessarily of finite length, this calls for a finite-trace semantics for LTL. However, as the system evolves, τ is only the prefix of a (yet unknown) continuation. Different variants of LTL then have to be studied depending on whether such a continuation eventually terminates, or continues forever. In many AI-relevant settings (e.g., planning and business processes [Bacchus and Kabanza, 2000; Baier *et al.*, 2009; De Giacomo and Vardi, 2013; De Giacomo *et al.*, 2014]) the monitored system operates for an unbounded, yet finite, number of steps. This means that also each one of the (infinitely many) possible suffixes of τ has a finite length, making LTL_f (LTL over finite traces [De Giacomo and Vardi, 2013]) a natural candidate logic for monitoring [De Giacomo *et al.*, 2020; De Giacomo *et al.*, 2022; Calvanese *et al.*, 2022a].

A key limitation of many of these approaches is that propositional LTL_f cannot express properties about structured data. However, such properties are ubiquitous in AI systems, as witnessed by the new agentic wave, where agents executing business tasks and processes have to deal with different types of structured data, such as numbers with arithmetic, sets, lists, and uninterpreted functions. Recently, Geatti *et al.* (2022) brought forward LTL_f modulo theories (LTL_f^{MT}) to enrich LTL_f with the full spectrum of theories supported by state-of-the-art SMT [Barrett *et al.*, 2021] solvers, which include all the datatypes mentioned before. The following example illustrates the need for such rich properties:

Example 1. *Consider an online marketplace for concert tickets. A buyer agent dynamically receives ticket offers, and is tasked by its human principal with the goal of selecting the best offer (that is, the ticket with the lowest price) for a given concert (indicated by constant* `myc` *below). Every ticket has two properties: its price, and the concert it refers to.*

Formally, this setting is captured by three datatypes Ticket, $\mathit{Concert}$ *(with* $\texttt{myc} \in \mathit{Concert}$*), and the real num-*

bers $\mathbb{R}$*, a dedicated constant* `undef` *of type* $Ticket$ *and functions* $price\colon Ticket \to \mathbb{R}$ *and* $con\colon Ticket \to Concert$ *that resp. assign to each ticket its price and the concert it is for. Such two functions conceptually model a database table with three columns, where each record stores a ticket id (primary key) with its price and concert. The agent should select the cheapest ticket. An execution consists of a trace where at each time step the values of two variables of type* $Ticket$ *are specified:* t*, the ticket currently offered by the marketplace; and* b*, the ticket currently bookmarked by the agent.*

Since the agent specification is not accessible, its human principal wants to formalise the expected behavior and check whether the agent indeed adheres to it. To this end, the principal constructs a suitable LTL_f^{MT} *formula composed of different parts. The first part* ψ_{change} *captures the situation where the bookmarked ticket must be changed to the currently received one, because it has a lower price. The notation* $\lhd b$ *is used here to refer to the previous value of variable* b*, i.e., the value* b *held in the instant before the current one. With this notation at hand, we set* ψ_{change} *as*

$$con(t) = \texttt{myc} \wedge (\lhd b = \texttt{undef} \vee price(t) < price(\lhd b)) \rightarrow b = t$$

Next, the formula ψ_{keep} *deals with the case where the bookmarked ticket is kept, since the currently received ticket either refers to a concert of no interest, or has a higher price than the bookmarked one. Formally,* ψ_{keep} *is given by*

$$(con(t) \neq \texttt{myc} \vee (\lhd b \neq \texttt{undef} \wedge price(t) > price(\lhd b))) \rightarrow b = \lhd b$$

To bundle everything together, we assume for simplicity that the ticket stream starts at the second instant. We can hence require that b *is initially* `undef`*, while the formulae* ψ_{keep} *and* ψ_{change} *must hold in every instant of the trace, starting from the next one.*[1] *The latter condition is captured using the* LTL_f *operators* X *and* G*, yielding*

$$\psi := (b = \texttt{undef}) \wedge \mathsf{XG}(\psi_{change} \wedge \psi_{keep})$$

Differently from the simpler settings of propositional LTL_f [De Giacomo *et al.*, 2022], or of LTL_f extended with numerical variables [Felli *et al.*, 2023], where a trace is simply a sequence of assignments, dealing with LTL_f^{MT} also requires to keep track of a first-order model M. In Ex. 1, this corresponds to the set of tickets seen in the trace, as well as their prices and concerts. Therefore, different interpretations of the monitoring problem are conceivable: the model underlying a trace can be considered as fixed, or it may be assumed to change in every step. In this work, we adopt the following: given a trace τ with model M, a continuation of τ provides an *extension of* M – a notion that we later anchor precisely to the model-theoretic concept of *embeddings*. This interpretation has the following advantages: (1) the current model, dealing with the trace seen so far, is extended only conservatively, in that facts concerning the previous elements must be kept throughout the trace, and no new fact concerning those elements can be added, but (2) in continuations of the trace, new elements may appear, along with additional facts involving these. This is fully in par with the expected interpretation of settings like that of Ex. 1, where the prices and referred concerts of tickets seen in the past are fixed, while new information on unseen tickets can be acquired later.

[1] We assume for simplicity that if a ticket with the same price appears, the agent keeps the one already bookmarked.

Leucker and Schallhart (2009) also emphasize another central monitoring desideratum, namely the identification of violations in the earliest possible instant, which in turn calls for *anticipatory monitors* [Bartocci *et al.*, 2018]. These monitors return their verdict not only considering the prefix seen so far, but also by reasoning on all its (infinitely many) possible suffixes. The next example makes a clear case for this.

Example 2. *Consider formula* ψ *from Ex. 1. Monitoring* ψ *essentially amounts to check, at every instant, whether the expected relationship between* b *and* t *holds. Consider now*

$$\psi_{\texttt{t123}} := \psi \wedge \mathsf{FG}(b = \texttt{t123})$$

It expresses that at some point the agent selection stabilizes[2] *to the ticket identified by* `t123`*. A monitor checking only the trace prefix seen so far would detect a violation of* $\psi_{\texttt{t123}}$ *only if* ψ *is violated, or at the end of the trace if the selected ticket is not* `t123`*. However, note that the price of ticket* `t123` *must be known, say* $price(\texttt{t123}) = 100$*, and due to the behaviour of the agent, it is clear that* `t123` *will never be selected as soon as the agent finds a ticket with a price below 100 euros. In this respect, an anticipatory monitor for* $\psi_{\texttt{t123}}$ *would immediately recognize that* $\psi_{\texttt{t123}}$ *is permanently violated (i.e., it will stay so independently on how the trace evolves) if in the current instant* b *points to a ticket with a price* < 100*.*

In this paper, we tackle for the first time anticipatory monitoring described in Ex. 2. In fact, *anticipatory monitoring for* LTL_f^{MT} *is a completely open problem*. It is indeed a challenging one, as already in the case where LTL_f is equipped with numeric datatypes with simple forms of arithmetic, it incurs in undecidability [Felli *et al.*, 2023], and therefore relevant monitorable fragments of the logic have to be singled out. We bring a fourfold contribution: (1) We formally introduce the anticipatory monitoring problem for LTL_f^{MT} specifications. (2) We focus on the quantifier-free fragment of LTL_f^{MT}, an expressive logic called dataLTL_f that has been previously studied for process analysis [Gianola *et al.*, 2025] and verification [Gianola *et al.*, 2024] tasks, but never for monitoring. We provide a monitoring technique that combines an automata-theoretic component to deal with the temporal dimension, with an SMT component to tackle the data part. (3) We identify expressive classes of properties combining arithmetic with references to a database, in the style of Ex. 1, where we show that our monitoring task becomes effectively solvable. (4) Feasibility of our method is witnessed by a prototype implementation with preliminary evaluation. Some additional material can be found in an extended version [Gianola *et al.*, 2026].

Related work A rich body of work studies RV of arithmetic temporal properties, notably Lola [D'Angelo *et al.*, 2005] (which also supports anticipation [Hipler *et al.*, 2024]), TeSSLa [Kallwies *et al.*, 2022], and Eagle [Goldberg and Havelund, 2005]. From [Felli *et al.*, 2023], which considers anticipatory monitoring for LTL_f with arithmetic. We borrow

[2] This is captured by the LTL_f operators F and G, where $\mathsf{FG}\gamma$ models that there exists an instant from which γ always holds; it is known that, over finite traces, this is logically equivalent to γ holding in the last instant of the trace [De Giacomo and Vardi, 2013].

in the present work the idea to abstract variable configurations by sets of formulae arranged in a graph, and the resulting approach to identify decidable fragments. However, dealing with richer theories substantially changes both the setting and the techniques needed to construct monitors.

Decker *et al.* (2016) propose an approach for monitoring modulo theories, but use an incomparable logic where variables cannot be compared across states, and models change at each trace instant. In Ex. 1, this would essentially mean that the price and referred concert of already seen tickets non-deterministically change at every instant. Also, no decidable fragments are identified. In another line of work [Schneider *et al.*, 2021; Basin *et al.*, 2015; Lima *et al.*, 2024], monitoring with *metric* first-order temporal logic is considered, but in the non-anticipatory setting. Havelund *et al.* (2020) present a monitoring approach for first-order past time temporal logic without anticipation. This work shares with the present paper that infinitely many assignments are represented by logical formulas, in this case by BDDs. However, in both [Havelund *et al.*, 2020; Basin *et al.*, 2015] the model is supposed to change at every instant, as predicates model events rather than facts.

The logic dataLTL$_f$ we consider here has previously been studied for process analysis and verification. Specifically, [Gianola *et al.*, 2024] verifies data-aware dynamic systems against dataLTL$_f$ properties, using automata-based techniques that are however too weak to be ported to monitoring.

2 Preliminaries

LTL$_f^{MT}$ syntax. We consider a first-order multi-sorted *signature* $\Sigma = \langle \mathcal{S}, \mathcal{P}, \mathcal{F}, V, U \rangle$, where $\mathcal{S}$ is a set of sorts; $\mathcal{P}$ is a set of predicate and $\mathcal{F}$ a set of function symbols over $\mathcal{S}$; V is a set of *data variables*; and U is a set of variables disjoint from V that will be used for quantification; where all variables have a sort in $\mathcal{S}$. below, we consider a fixed signature Σ that contains equality predicates for all sorts in $\mathcal{S}$.

A term t is generated by the following grammar:

$$t := x \mid y \mid f(t_1, \ldots, t_k) \mid \lhd x$$

where $x \in V$ and $y \in U$ are variables, $f \in \mathcal{F}$ is a function symbol of arity k, each t_i, $1 \leq i \leq k$, is a term, and $\lhd$ is the *previous* constructor. The grammar of LTL$_f^{MT}$ formulas over Σ is the following:

$$\alpha := p(t_1, \ldots, t_k)$$
$$\varphi := \alpha \mid \neg\alpha \mid \varphi \vee \varphi' \mid \varphi \wedge \varphi' \mid \exists y.\varphi \mid \forall y.\varphi$$
$$\psi := \top \mid \varphi \mid \psi \vee \psi' \mid \psi \wedge \psi' \mid \mathsf{X}\psi \mid \mathsf{X}_w\psi \mid \psi \,\mathsf{U}\, \psi' \mid \psi \,\mathsf{R}\, \psi'$$

where $y \in U$, $p \in \mathcal{P}$ is an k-ary predicate symbol, each t_i is a term, and X, X_w, U, and R are the next, weak next, until, and release temporal operators, respectively. Formulas of type φ as defined above are called first-order (FO) formulas. In the definition of ψ, we force φ to have no free variables from U. We use the usual shortcuts $\mathsf{F}\psi \equiv \top \,\mathsf{U}\, \psi$ and $\mathsf{G}\psi \equiv \bot \,\mathsf{R}\, \psi$. By the grammar above, LTL$_f^{MT}$ formulas are always in negation normal form, and we assume them to be well-typed with regard to the sorts of all the involved symbols. The set of all LTL$_f^{MT}$ formulas is denoted $\mathcal{L}$.

For instance, ψ and $\psi_{\texttt{t123}}$ in Ex. 1 are LTL$_f^{MT}$ formulas over the signature where $V = \{t, b\}$, there are no quantified variables so U can be chosen empty, $\mathcal{P}$ contains the arithmetic predicates, and $\mathcal{F}$ consists of arithmetic functions, equality, *price*, *concert*, `undef`, `myc`, and `t123`.

A FO formula φ is a *state formula* if it does neither contain free variables from U nor $\lhd$ operators. A state formula without free variables is a *sentence*, and a set of sentences is a Σ*-theory* $\mathcal{T}$. It is *universal* if all its sentences are over Σ and have the form $\forall u_1, \ldots, u_l.\, \psi$ with ψ quantifier-free.

LTL$_f^{MT}$ semantics. To define semantics, we use the standard notion of a Σ*-structure* M, which associates each sort $s \in \mathcal{S}$ with a domain s^M, and each predicate $p \in \mathcal{P}$ and function symbol $f \in \mathcal{F}$ with a suitable interpretation p^M and f^M. The equality predicates have the standard interpretation given by the identity relation. The carrier of M, i.e., the union of all domains of sorts in $\mathcal{S}$, is denoted by $|M|$. A total function $\alpha \colon V \to |M|$ is called a *state variable assignment*. We always assume that variables are mapped to an element of their respective domain. A *trace* $\tau = (M, \langle \alpha_0, \ldots, \alpha_{n-1} \rangle)$ consists of a model M and a sequence of state variable assignments α_i with codomain $|M|$. We call n the *length* of τ, denoted $|\tau|$. For two traces $\tau = (M, \langle \alpha_0, \ldots, \alpha_n \rangle)$ and $\tau' = (M', \langle \alpha'_0, \ldots, \alpha'_m \rangle)$ with $M = M'$, their concatenation is given by $\tau \cdot \tau' := (M, \langle \alpha_0, \ldots, \alpha_n, \alpha'_0, \ldots, \alpha'_m \rangle)$.

Given a trace $\tau = (M, \langle \alpha_0, \ldots, \alpha_{n-1} \rangle)$, and a variable evaluation function $\xi \colon U \to |M|$ for the variables in U, the *evaluation* $[t]^i_{\tau,\xi}$ of a term t is *well-defined* if $i < |\tau|$, and either $i > 0$ or t does not contain $\lhd$ operators, namely as

$$[v]^i_{\tau,\xi} = \alpha_i(v) \qquad [\lhd v]^i_{\tau,\xi} = \alpha_{i-1}(v)$$
$$[u]^i_{\tau,\xi} = \xi(u) \qquad [f(t_1, \ldots, t_k)]^i_{\tau,\xi} = f^M([t_1]^i_{\tau,\xi}, \ldots, [t_k]^i_{\tau,\xi})$$

where $v \in V$ and $u \in U$. Satisfaction of a first-order formula φ with respect to an environment ξ in the run τ with $i < |\tau|$, denoted $\tau \models^i_\xi \varphi$, is defined as:

$\tau \models^i_\xi p(t_1, \ldots, t_k)$	if $t_1, \ldots, t_k$ are well-defined and $([t_1]^i_{\tau,\xi}, \ldots, [t_k]^i_{\tau,\xi}) \in p^{M_i}$,
$\tau \models^i_\xi \neg p(t_1, \ldots, t_k)$	if $\tau \not\models^i_\xi p(t_1, \ldots, t_k)$
$\tau \models^i_\xi \varphi_1 \wedge \varphi_2$	if $\tau \models^i_\xi \varphi_1$ and $\tau \models^i_\xi \varphi_2$
$\tau \models^i_\xi \varphi_1 \vee \varphi_2$	if $\tau \models^i_\xi \varphi_1$ or $\tau \models^i_\xi \varphi_2$
$\tau \models^i_\xi \exists y.\, \varphi$	if $\tau \models^i_{\xi[y \mapsto e]} \varphi$ for some $e \in \|M\|$
$\tau \models^i_\xi \forall y.\, \varphi$	if $\tau \models^i_{\xi[y \mapsto e]} \varphi$ for all $e \in \|M\|$

where y is assumed to have sort s, and e is in the domain for s in $|M|$. Satisfaction w.r.t. τ is extended to a general LTL$_f^{MT}$ formula ψ as follows:

$\tau \models^i \varphi$	if $\tau \models^i_\emptyset \varphi$
$\tau \models^i \psi_1 \wedge \psi_2$	if $\tau \models^i \psi_1$ and $\tau \models^i \psi_2$
$\tau \models^i \psi_1 \vee \psi_2$	if $\tau \models^i \psi_1$ or $\tau \models^i \psi_2$
$\tau \models^i \mathsf{X}\psi$	if $i < \|\tau\|-1$ and $\tau \models^{i+1} \psi$
$\tau \models^i \mathsf{X}_w\psi$	if $i \geq \|\tau\|-1$ or $\tau \models^{i+1} \psi$
$\tau \models^i \psi_1 \,\mathsf{U}\, \psi_2$	if there is some j, $i \leq j < \|\tau\|$ such that $\tau \models^j \psi_2$ and $\tau \models^k \psi_1$ for all $i \leq k < j$
$\tau \models^i \psi_1 \,\mathsf{R}\, \psi_2$	if either $\tau \models^j \psi_2$ for all $i \leq j < \|\tau\|$, or there is some j, $i \leq j < \|\tau\|$ such that $\tau \models^j \psi_1$ and $\tau \models^k \psi_2$ for all $i \leq k \leq j$.

Finally, τ *satisfies* ψ, denoted by $\tau \models \psi$, if $\tau \models^0 \psi$ holds.

Example 3. *For φ from Ex. 1, consider a model M where $Concert^M = \{c_0, c_1, c_2, c_3\}$, $\texttt{myc}^M = c_1$, Ticket has carrier $\{\bot, t_1, t_2, \ldots, t_{1000}\}$, $\texttt{undef}^M = \bot$, with an interpretation that satisfies $con(t_2) = con(t_4) = con(t_5) = c_1$ and $con(t_1) = c_2$, $price(t_2) = 100$, and $price(t_4) = 110$, $price(t_5) = 95$. Then the trace $(M, \overline{\alpha})$ satisfies φ, where*

$$\overline{\alpha} = \langle \left\{ \begin{smallmatrix} t=\bot \\ b=\bot \end{smallmatrix} \right\}, \left\{ \begin{smallmatrix} t=t_2 \\ b=100 \end{smallmatrix} \right\}, \left\{ \begin{smallmatrix} t=t_1 \\ b=100 \end{smallmatrix} \right\}, \left\{ \begin{smallmatrix} t=t_4 \\ b=100 \end{smallmatrix} \right\}, \left\{ \begin{smallmatrix} t=t_5 \\ b=95 \end{smallmatrix} \right\} \rangle$$

Our logic differs from [Geatti *et al.*, 2022] in two respects: we define LTL_f^{MT} with a *lookback* operator $\lhd$ instead of *lookahead*, i.e., the logic allows to refer to values of V in the previous trace instant, not in the next. However, every formula with lookahead can be equivalently expressed with lookback [Felli *et al.*, 2023, Lem. 37]. Moreover, we have the strong version of the cross-state operator (but not its weaker version), so that atoms with not well-defined terms are false.

First-order logic. Next, we need to repeat some preliminaries from FO logic and model theory. If φ is a state formula, we simply write $M, \alpha \models \varphi$ instead of $\langle (M, \alpha) \rangle \models^i_\emptyset \varphi$, and if φ is a sentence, we write $M \models \varphi$ for $M, \emptyset \models^0_\emptyset \varphi$. We adopt a common SMT perspective and view a Σ-theory $\mathcal{T}$ as the set of all Σ-structures that satisfy all axioms of $\mathcal{T}$, and write $M \in \mathcal{T}$ if M is a model of $\mathcal{T}$, i.e., $M \models \varphi$ holds for all sentences φ in $\mathcal{T}$. A state formula φ is *$\mathcal{T}$-satisfiable* if there is some $M \in \mathcal{T}$ and state variable assignment $\alpha \colon V \to |M|$ such that $M, \alpha \models \varphi$; whereas φ is $\mathcal{T}$-valid if $M, \alpha \models \varphi$ for all $M \in \mathcal{T}$ and α. Moreover, formulas φ_1 and φ_2 are *$\mathcal{T}$-equivalent*, denoted $\varphi_1 \equiv_\mathcal{T} \varphi_2$, if $\varphi_1 \leftrightarrow \varphi_2$ is $\mathcal{T}$-valid.

A Σ-theory $\mathcal{T}$ has *quantifier elimination* (QE) if for every Σ-formula φ there is a quantifier-free formula φ' that is $\mathcal{T}$-equivalent to φ. As QE is a strong requirement, we exploit in this paper the weaker notion of a theory having a model completion. In this context, an *embedding* is a mapping $\mu \colon M \to M'$ between two Σ-structures M and M' that preserves the truth of Σ-literals. It is easy to see that a composition of embeddings is again an embedding. A universal Σ-theory $\mathcal{T}$ has model completion [Ghilardi, 2004] iff there exists a Σ-theory $\mathcal{T}^*$, i.e., a theory over the same signature, where $\mathcal{T}^* \supseteq \mathcal{T}$, such that (i) every model of $\mathcal{T}$ can be embedded in a model of $\mathcal{T}^*$, and (ii) $\mathcal{T}^*$ has QE.

It follows from this definition that every model of $\mathcal{T}^*$ can be embedded in a model of $\mathcal{T}$. We will later exploit the following properties [Calvanese *et al.*, 2020b, Sec. 2]:

Lemma 4. Let $\mathcal{T}$ have model completion $\mathcal{T}^*$.

(1) Let $M \in \mathcal{T}$, φ be an existential Σ-formula and α an assignment of values in M such that $M, \alpha \models \varphi$. Then there is an embedding $\mu \colon M \to M'$ for some $M' \in \mathcal{T}^*$ such that $M', \mu \circ \alpha \models \varphi$.

(2) Let $M \in \mathcal{T}$, $M' \in \mathcal{T}^*$, $\mu \colon M \to M'$ be an embedding, φ a quantifier-free Σ-formula, and α be an assignment of values in M. If $M', \mu \circ \alpha \models \varphi$ then $M, \alpha \models \varphi$.

For a trace $\tau = (M, \langle \alpha_0, \ldots, \alpha_{n-1} \rangle)$ and an embedding $\mu \colon M \to M'$, let $\mu(\tau) = (M', \langle \mu \circ \alpha_0, \ldots, \mu \circ \alpha_n \rangle)$.

We will sometimes refer to common SMT theories [Barrett *et al.*, 2021]: the theory of equality and uninterpreted functions (EUF) for a given Σ, and linear arithmetic over rationals (LRA), integers (LIA), or both (LIRA). While the arithmetic theories have QE [Presburger, 1929], EUF admits model completion and so do certain *tame* combinations of LIRA and EUF [Calvanese *et al.*, 2020a; Calvanese *et al.*, 2022b]. In the remainder, we make the following assumptions on the theory:

Assumptions 5. (a) Satisfiability of existential $\mathcal{T}$-formulas is decidable; (b) $\mathcal{T}$ is either universal and has a model completion $\mathcal{T}^*$; or has QE, in this case let $\mathcal{T}^* := \mathcal{T}$.

We will show later that these assumptions are satisfied by several theories which are highly relevant in practice.

Monitoring. We now define our main task, namely determining how the satisfaction of a given property changes along a trace. Throughout the paper, we consider *anticipatory* monitoring, taking into account both the trace seen so far and all its finite future continuations. Following the literature [Bauer *et al.*, 2010], we consider the set $RV = \{\text{PS}, \text{CS}, \text{CV}, \text{PV}\}$ of four distinct *monitoring states*: current satisfaction (CS), permanent satisfaction (PS), current violation (CV) and permanent violation (PV). We call a trace τ' is an *extension* of a trace $\tau = (M, \langle \alpha_0, \ldots, \alpha_{n-1} \rangle)$ if there is some M' such that there is an embedding $\mu \colon M \to M'$ and assignments α'_i such that $\tau' = (M', \langle \mu(\alpha_0), \ldots, \mu(\alpha_n), \alpha'_0, \ldots, \alpha'_m \rangle)$.

Definition 6. An LTL_f^{MT} property ψ is in monitoring state $s \in RV$ after a trace τ, written $\tau \models [\![\psi = s]\!]$, if

- $s = \text{CS}$, $\tau \models \psi$ but $\tau' \not\models \psi$ for some extension τ' of τ;
- $s = \text{PS}$, $\tau \models \psi$, and $\tau' \models \psi$ for every extension τ' of τ;
- $s = \text{CV}$, $\tau \not\models \psi$ but $\tau' \models \psi$ for some extension τ' of τ;
- $s = \text{PV}$, $\tau \not\models \psi$, and $\tau' \not\models \psi$ for every extension τ' of τ.

After each trace, a property ψ is in exactly one possible monitoring state. E.g. for the trace τ from Ex. 3 and ψ, $\psi_{\texttt{t123}}$ from Ex. 1 we have $\tau \models [\![\psi = \text{CS}]\!]$: the property is currently satisfied but could be violated in an extension of τ. Considering $\psi_{\texttt{t123}}$, let $\texttt{t123}^M = t_{123}$. If we have $price(t_{123}) = 80$ then $\tau \models [\![\psi_{\texttt{t123}} = \text{CV}]\!]$ as t_{123} could still be selected in the future; but if $price(t_{123}) = 100$ then $\tau \models [\![\psi_{\texttt{t123}} = \text{PV}]\!]$.

Given τ and ψ, the *monitoring problem* is to compute the state $s \in RV$ s.t. $\tau \models [\![\psi = s]\!]$. It is *solvable* if one can construct a procedure for ψ that computes the monitoring state for any given trace. It is known that the monitoring problem is not solvable if $\mathcal{T}$ is the theory of linear arithmetic over the reals [Felli *et al.*, 2023], and hence also not for the much more general logic LTL_f^{MT}.

3 Automata for Properties

In the remainder of the paper we assume a theory $\mathcal{T}$ that satisfies Assumptions 5, and the quantifier-free fragment of LTL_f^{MT}, that is, the logic dataLTL_f from [Gianola *et al.*, 2025; Gianola *et al.*, 2024]. Excluding quantifiers is here an essential restriction when applying reasoning with model completion to the automaton, which is needed to construct the monitors. As e.g. shown in Ex. 1, dataLTL_f still supports expressive queries over the theories, provided that they are formulated referring to the monitored variables.

It is well-known that LTL_f properties can be captured by non-deterministic automata (NFA) [De Giacomo and Vardi, 2013]. Our monitoring approach relies on such an NFA for

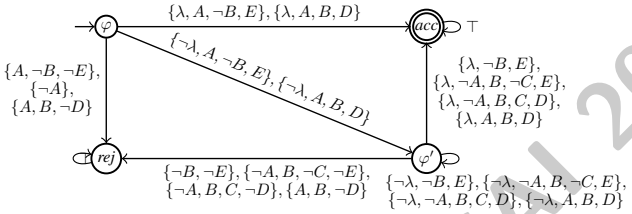
a given LTL_f^{MT} property ψ. While the construction is rather standard, the correctness proofs are more complicated due to the FO setting. We only sketch here our construction, details and proofs can be found in [Gianola *et al.*, 2026]. Let λ be an additional proposition to mark the last element of a trace. Given a dataLTL_f property ψ, let C be the set of FO formulas in ψ and $C^{\pm} = C \cup \{\neg c \mid c \in C\}$ the set of FO formulas in ψ together with their negations. The alphabet of the NFA will be $\Theta = 2^{C^{\pm} \cup \{\lambda, \neg\lambda\}}$, so that a symbol is a set of FO formulas.

The NFA $\mathcal{N}_\psi$ is built using an auxiliary function δ, which has as input a property $\psi \in \mathcal{L} \cup \{\top, \bot\}$ and outputs a set of pairs (ψ', ς) where $\psi' \in \mathcal{L} \cup \{\top, \bot\}$ is again a property and $\varsigma \in \Theta$. Intuitively, $(\psi', \varsigma) \in \delta(\psi)$ means that a word ςw satisfies ψ iff the suffix w satisfies ψ', for its definition see [Gianola *et al.*, 2026]. $\mathcal{N}_\psi$ is then defined as follows:

Definition 7. Given a property $\psi \in \mathcal{L}$, let $\mathcal{N}_\psi = (Q, \Theta, \varrho, q_0, Q_F)$ where $q_0 = \psi$ is the initial state, $Q_F = \{\top, q_+\}$ is the subset of final states, q_- is an additional non-final state, and Q, ϱ are the smallest sets such that $q_0, q_F, q_+, q_- \in Q$ and whenever $q \in Q \setminus \{q_+, q_-\}$ and $(q', \varsigma) \in \delta(q)$ then $q' \in Q$ and

(i) if $\lambda \notin \varsigma$ then $(q, \varsigma, q') \in \varrho$, and
(ii) whenever $\lambda \in \varsigma$, if $q' = \top$ then $(q, \varsigma, q_+) \in \varrho$, and if $q' = \bot$ then $(q, \varsigma \setminus \{\lambda\}, q_-) \in \varrho$.

Example 8. *Consider ψ from Ex. 1. We abbreviate $A := (b = \texttt{undef})$, $B := (con(t) = \texttt{myc})$, $C := (price(t) > price(\lhd b))$, $D := (b = t)$, $E := (b = \lhd b)$, and $\varphi' := \mathsf{G}(\varphi_{change} \wedge \varphi_{keep})$. Then the NFA $\mathcal{N}_\psi$ is as follows:*



We denote by $V^{\lhd}$ the set of variables $V^{\lhd} = \{\lhd v \mid v \in V\}$. A word $w = \varsigma_0, \varsigma_1, \ldots, \varsigma_{n-1} \in \Theta^+$ is *well-formed* if ς_0 contains no variable in $V^{\lhd}$. We next define the notion of *consistency*, which serves to capture that a trace τ satisfies all formulas in a word $w \in \Theta^+$ as above. To that end, for assignments α and α' with domain V, we define the assignment $[\alpha \circledast \alpha']$ with domain $V^{\lhd} \cup V$ as $[\alpha \circledast \alpha'](\lhd v) = \alpha(v)$ and $[\alpha \circledast \alpha'](v) = \alpha'(v)$, for all $v \in V$. For simplicity, we write $M, \alpha \models \varsigma$ instead of $M, \alpha \models \bigwedge \varsigma$.

Definition 9. A well-formed word $\varsigma_0, \varsigma_1, \ldots, \varsigma_{n-1} \in \Theta^+$ is *consistent* with a trace $(M, \langle \alpha_0, \ldots, \alpha_{n-1} \rangle)$ if $M, \alpha_0 \models \varsigma_0$ and $M, [\alpha_{i-1} \circledast \alpha_i] \models \varsigma_i$ for all $0 < i < n$, and moreover $\lambda \in \varsigma_{n-1}$, and $\neg\lambda \in \varsigma_i$ for all $0 \leq i < n-1$.

To state the main result about the NFA $\mathcal{N}_\psi$ below, we need to restrict to properties that use lookback in a valid way, i.e., that do not enforce constraints with lookback variables in the first trace instant. To that end, let an dataLTL_f property ψ have *safe lookback* if no edge from the initial state has a label that contains $V^{\lhd}$. The following is the main result about $\mathcal{N}_\psi$:

Theorem 10. *Given $\psi \in \mathcal{L}$ with safe lookback and trace τ, (1) if $\mathcal{N}_\psi$ accepts a well-formed word $w \in \Theta^+$ consistent with τ, then $\tau \models \psi$, and (2) if $\tau \models \psi$ then there exists a well-formed word $w \in \Theta^+$ consistent with τ and accepted by $\mathcal{N}_\psi$.*

For $w = \varsigma_0, \ldots, \varsigma_{n-1}$, we write $q_0 \xrightarrow{w} q_n$ if there is a sequence of states $q_1, \ldots, q_{n-1}$ such that $(q_i, \varsigma_i, q_{i+1}) \in \varrho$ for all $0 \leq i < n$. We have the following determinism property:

Lemma 11. Given a trace τ there are a unique word $w \in \Theta^*$ and state q s.t. $q_0 \xrightarrow{w} q$ in $\mathcal{N}_\psi$ and w is consistent with τ.

Note that while it is possible to postprocess $\mathcal{N}_\psi$ to an NFA for ψ that does not mention λ [Felli *et al.*, 2023], the resulting automaton requires an explicit determinization step to obtain a property akin to Lem. 11, so we prefer to keep the λ's.

4 Monitoring Technique

We next present our monitoring algorithm. Throughout this section, we assume a dataLTL_f property ψ with NFA $\mathcal{N}_\psi = (Q, \Theta, \varrho, q_0, Q_F)$ as defined in the previous section.

For a formula φ with free variables V, let $\varphi(\overline{V}/\overline{X})$ denote the formula where $\overline{V}$ is replaced by $\overline{X}$ (we denote by $\overline{V}$ a fixed ordering of V as a vector). For a set of constraints C with free variables $V^{\lhd} \cup V$, let $C(\overline{X}/\overline{V^{\lhd}}, \overline{Y}/\overline{V})$ denote the constraints where $\overline{V^{\lhd}}$ is replaced by $\overline{X}$ and $\overline{V}$ by $\overline{Y}$. Moreover, for a set of constraints $C \in \Theta$, we denote by $[C]$ its conjunction where λs are omitted, i.e., $[C] = \bigwedge C \setminus \{\lambda, \neg\lambda\}$; so $[C]$ is a formula with free variables $V^{\lhd} \cup V$.

Definition 12. For a state formula φ and $C \in \Theta$, let $regress(\varphi, C) = \exists \overline{Y}.[C](\overline{V}/\overline{V^{\lhd}}, \overline{Y}/\overline{V}) \wedge \varphi(\overline{Y}/\overline{V})$, where Y is a set of fresh variables of the same size and sorts as V.

Note that since φ has free variables V, also $regress(\varphi, C)$ has free variables V. The *regress* operation defined above allows us to "propagate backwards" the verification obligation expressed by φ through the constraint C: in formal verification, this corresponds to the classic notion of *preimage* when performing backward search in a transition system, cf. e.g. [Abdulla *et al.*, 1996]. We collect these verification obligations in a structure called coreachability graph, which is one of the technical novelties of this paper, defined next:

Definition 13. Given NFA $\mathcal{N}_\psi$ for ψ, the *coreachability graph* $\text{CG}^+(\mathcal{N}_\psi)$ for $\mathcal{N}_\psi$ is a tuple $\langle S, \gamma \rangle$ where S is a set of nodes of the form (q, φ) for $q \in Q$ and φ a quantifier-free Σ-formula with free variables V, and $\gamma \subseteq S \times \Theta \times S$ is a transition relation. It is inductively defined as follows:

(i) for all final states $q \in Q_F$, a node $(q, \top) \in S$, and
(ii) if $s \in S$ is of the form $s = (q, \varphi)$ and there is a transition $q' \xrightarrow{\varsigma} q$ in $\mathcal{N}_\psi$ such that $regress(\varphi, \varsigma)$ is $\mathcal{T}$-satisfiable, there is some node $s' = (q', \varphi') \in S$ such that $\varphi' \equiv_{\mathcal{T}^*} regress(\varphi, \varsigma)$, and $s' \xrightarrow{\varsigma} s$ is in γ.

Note that we use model completion to obtain quantifier-free formulas, which is the main technical novelty of our approach; the fact that all formulas in CGs are quantifier-free allows us to identify decidable fragments in the next section. Note that as $regress(\varphi, \varsigma)$ is existential, by Assumptions 5 some quantifier-free χ s.t. $\chi \equiv_{\mathcal{T}^*} regress(\varphi, \varsigma)$ indeed exists, by applying quantifier elimination in $\mathcal{T}^*$.

Intuitively, if $\text{CG}^+(\mathcal{N}_\psi)$ has a node (q, φ) then this means that from NFA state q a final state can be reached under the

condition that the current variable assignment satisfies φ. It is called '*co*reachability graph' because it is is constructed by *regressing* from the final states of the system, rather than by *forward exploration* from the initial states as customary in standard reachability graphs [Desel and Esparza, 1993; Chiola *et al.*, 1997]. The *non-coreachability graph* $\text{CG}^-(\mathcal{N}_\psi)$ is defined dually, but where in condition (i) of Def. 13 we have $(q, \top) \in S$ for all $q \in Q \setminus Q_F$.

The construction in Def. 13 need not terminate as infinitely many formulas may occur, but we will show in the next section that termination is guaranteed for several relevant classes of properties. For cases where $\text{CG}^+(\mathcal{N}_\psi)$ and $\text{CG}^-(\mathcal{N}_\psi)$ are finite, we define the following formulas which express, intuitively, under which conditions a final (resp. non-final) state is reachable from state q in the NFA.

Definition 14. For ψ with NFA $\mathcal{N}_\psi$ and a state q in $\mathcal{N}_\psi$, let

$$\textsc{ExtSat}(q) = \bigvee\{\varphi \mid (q, \varphi) \text{ is a node in } \text{CG}^+(\mathcal{N}_\psi)\}$$
$$\textsc{ExtViol}(q) = \bigvee\{\varphi \mid (q, \varphi) \text{ is a node in } \text{CG}^-(\mathcal{N}_\psi)\}$$

This leads to the following monitoring procedure, where the trace τ is assumed to have positive length:

1: **procedure** MONITOR(ψ, τ)
2: compute $\mathcal{N}_\psi$
3: $q \leftarrow$ state in $\mathcal{N}_\psi$ such that $q_0 \rightarrow_w^* q$ for some well-formed $w \in \Theta^*$ consistent with τ ▷ cf. Lem. 11
4: let M be the model of τ and α_n its last assignment
5: **if** q is accepting in $\mathcal{N}_\psi$ **then**
6: **return** CS **if** $M, \alpha_n \models \textsc{ExtViol}(q)$ **else** PS
7: **else return** CV **if** $M, \alpha_n \models \textsc{ExtSat}(q)$ **else** PV

Thanks to Thm. 10, Line 5 is equivalent to checking whether the property ψ holds on the prefix τ. If so, the possible monitoring states are restricted to CS and PS; therefore, in Line 7 it suffices to check whether the extension leads to a violation, which is achieved by checking whether α_n satisfies $\textsc{ExtViol}(q)$. If τ does not satisfy ψ, dual reasoning applies.

The NFA, CGs, and formulas $\textsc{ExtSat}$ and $\textsc{ExtViol}$ should only be computed when required but can be cached for later use, to avoid re-computation. This is especially useful when monitoring multiple traces w.r.t. the same property.

Correctness. To prove correctness, we need some additional notation. First of all, *regression constraints* are formulas obtained from nested *regress* operations:

Definition 15. For $w = \varsigma_0, \ldots, \varsigma_{n-1}$ in Θ^*, the *regression constraint* $\textsc{rc}(w)$ is given by $\textsc{rc}(w) = \top$ if $n = 0$, and if $n > 0$ then $\textsc{rc}(w) = \mathit{regress}(\textsc{rc}(\langle \varsigma_1, \ldots, \varsigma_{n-1} \rangle), \varsigma_0)$.

Crucial for our verification technique is the fact that there is a correspondence between satisfying assignments of $\textsc{rc}(w)$, and traces that satisfy the verification obligations given by the constraints in w, as stated next:

Lemma 16. For $w \in \Theta^*$, a model M and assignment α, (1) if $M, \alpha \models \textsc{rc}(w)$ there is a trace τ with model M s.t. $\langle \emptyset \rangle \cdot w$ is consistent with $(M, \langle \alpha \rangle) \cdot \tau$, and (2) if $\langle \emptyset \rangle \cdot w$ is consistent with some $(M, \langle \alpha \rangle) \cdot \tau$ then $M, \alpha \models \textsc{rc}(w)$.

There is a close relationship between paths in CGs and regression constraints, as stated in the next lemma. For a word $w = \varsigma_1, \ldots, \varsigma_n$, we write $(q, \varphi) \rightarrow_w^* (q', \top)$ if there is a path $(q, \varphi) \xrightarrow{\varsigma_1} \cdots\cdots \xrightarrow{\varsigma_n} (q', \top)$ in $\text{CG}^+(\mathcal{N}_\psi)$ or $\text{CG}^-(\mathcal{N}_\psi)$.

Lemma 17. Let G be $\text{CG}^+(\mathcal{N}_\psi)$ (resp. $\text{CG}^-(\mathcal{N}_\psi)$).
(1) If G has a path $(q, \varphi) \rightarrow_w^* (q_f, \top)$ then $q \rightarrow_w^* q_f$ in $\mathcal{N}_\psi$ such that q_f is a final (resp. non-final) state, φ is $\mathcal{T}$-satisfiable, and $\varphi \equiv_{\mathcal{T}^*} \textsc{rc}(w)$.
(2) If $q \rightarrow_w^* q_f$ in $\mathcal{N}_\psi$ such that q_f is final (resp. non-final), and $\textsc{rc}(w)$ is $\mathcal{T}$-satisfiable, there is a path $(q, \varphi) \rightarrow_w^* (q_f, \top)$ in G for some φ such that $\varphi \equiv_{\mathcal{T}^*} \textsc{rc}(w)$.

At this point we are ready to prove our main theorem, which strongly exploits the two properties of model completions: embeddability of $\mathcal{T}$- models into $\mathcal{T}^*$-models, and QE in $\mathcal{T}^*$. The proof can be found in [Gianola *et al.*, 2026].

Theorem 18. *If* $\textsc{monitor}(\psi, \tau)$ *returns* s *then* $\tau \models [\![\psi{=}s]\!]$.

5 Solvability Criteria

In this section, we identify classes of dataLTL$_f$ properties where MONITOR is guaranteed to terminate; we use the terminology that the monitoring task is *solvable* in this case. We focus on properties that combine references to a database with arithmetic, a combination that is very common in practice, in particular in data-aware BPM [Ghilardi *et al.*, 2023] and when observing database-driven systems [Bojańczyk *et al.*, 2013; Calvanese *et al.*, 2013; Deutsch *et al.*, 2018];

To define databases in a symbolical context, we rely on the notion of *static DB schemas* from [Ghilardi *et al.*, 2023], where relational databases with integer and real values are formalized in an algebraic way. Static DB schemas use the combination of two FO theories, $\mathcal{T}_{db}$ and $\mathcal{T}_{ar}$. The theory $\mathcal{T}_{db}$ models read-only DB relations with primary and foreign key constraints by employing function symbols to represent the key dependencies: formally, this can be captured in SMT via constants and functions subject to multi-sorted EUF, as done in our ticketing Ex. 1. The theory $\mathcal{T}_{ar}$ is LIA, LRA or their combination; it is used to formalize arithmetic constraints.

DB-driven theories Let a *DB-LTL$_f$ property* be a property in LTL_f^{MT} where the signature Σ consists of arbitrary relation, and unary function symbols. The associated theory $\mathcal{T}_{db}$ is defined as EUF, possibly extended with a set of ground facts to model an initial DB. This theory enjoys model completion [Calvanese *et al.*, 2020b]. In the case with arithmetic, Σ supports in addition operations from an arithmetic theory $\mathcal{T}_{ar}$ as above, and its theory is $\mathcal{T} := \mathcal{T}_{db} \cup \mathcal{T}_{ar}$. To ensure that $\mathcal{T}$ has model completion, we assume that the signature is *tame*, which intuitively means that no uninterpreted function symbol has an arithmetic domain (see below).

Note that the presence of only finitely many terms guarantees that the procedure MONITOR always terminates. However, in general both the DB and the arithmetic perspective can give rise to infinitely many terms, so in classes of properties where the monitoring task is solvable, both must be suitably restricted. Next, we explore different ways to do so.

I. Acyclic signature. First, we consider DB-LTL$_f$ without arithmetic. Acyclicity of the signature Σ is defined via the *sort graph* of Σ, which has as node set the sorts $\mathcal{S}$, and an edge from s to s' iff there is a function symbol $f\colon s \rightarrow s'$ in Σ (as we have only unary functions); we say that Σ is *acyclic* if the sort graph is so. For instance, the non-arithmetic part of the signature of Ex. 1 is acyclic. In this case, only finitely

many Σ-terms exist [Abadi *et al.*, 2010]. Thus, there are also only finitely many non-equivalent quantifier-free formulas over the finite set of variables V [Calvanese *et al.*, 2020b]. and hence there are only finitely many regression constraints, so that the coreachability graph construction in Def. 13 terminates. Hence, we obtain the following decidability result:

Theorem 19. *If Σ is acyclic, the monitoring task for DB-LTL$_f$ is solvable.*

II. Acyclic signature and monotonicity constraints. Next, we consider DB-LTL$_f$ with arithmetic. To obtain decidability, we make multiple restrictions. First, we assume that the signature Σ is *tame*, i.e., in the sort graph of Σ, sorts rat and int are leaves. In this case, the combined theory $\mathcal{T}$ is known to enjoy model completion [Calvanese *et al.*, 2022b, Thm. 7]. Second, we require that Σ is acyclic. Third, we restrict arithmetic constraints to *monotonicity constraints* (MCs), i.e., constraints of the form $t \odot t'$ where t, t' are Σ-terms of sort rat over free variables V and $\odot$ is one of $=$, $\neq$, $\leq$, or $<$. MCs have been repeatedly considered in the verification literature [Demri and D'Souza, 2007; Felli *et al.*, 2019] and are important in BPM since transition guards of this shape can be learned automatically from data [de Leoni and van der Aalst, 2013]. We call the class of properties that adhere to these restrictions DB-LTL$_f$-MC. For instance, the properties of Ex. 1 are in this class. We obtain the following result, subsuming [Felli *et al.*, 2023, Thm. 20]:

Theorem 20. *The monitoring task is solvable for DB-LTL$_f$-MC properties.*

Proof (idea). As shown in [Gianola *et al.*, 2024], under the assumptions of tame signature and MCs, the procedure TameCombCover [Calvanese *et al.*, 2022b, Sec. 8] has finitely many possible results of the form $\varphi_1(\overline{X}) \wedge \varphi_2(\overline{Y}, \bar{t}(\overline{X}))$, up to equivalence. Hence there are finitely many regression constraints up to equivalence, so the coreachability graph construction in Def. 13 terminates. □

III. Local Finiteness. Next, we consider decidability beyond acyclic signatures for DB-LTL$_f$ without arithmetic, using the concept of local finiteness [Lipparini, 1982]. A Σ-theory $\mathcal{T}$ is called *k-locally finite*, for some $k \in \mathbb{N}$, if for every finite set of variables X, the number of Σ-terms with free variables X is upper-bounded by k, up to $\mathcal{T}$-equivalence. As local finiteness directly ensures that there are only finitely many non-equivalent quantifier-free formulas in $\mathcal{T}^*$, we get:

Theorem 21. *If $\mathcal{T}$ is a k-locally finite Σ-theory that admits model completion, the monitoring task is solvable for any property ψ over Σ.*

One way to ensure in Thm. 21 that $\mathcal{T}$ has a model completion is to restrict axioms in $\mathcal{T}$ to single-variable sentences [Calvanese *et al.*, 2020b]:

Corollary 22. The monitoring task is solvable for a DB-LTL$_f$ property ψ over Σ and if $\mathcal{T}$ is a k-locally finite Σ-theory axiomatized by single-variable axioms.

One can also obtain decidability by replacing acyclicity in the combination result of Thm. 20 by local finiteness, using in the proof local finiteness to ensure that up to equivalence there are only finitely many formulas without arithmetic.

IV. Bounded lookback. The above criteria achieve decidability mostly by syntactic restrictions. However, finite coreachability graphs can also be achieved by controlling interaction of constraints in properties, as it was already shown for the arithmetic case [Felli *et al.*, 2023]: *Bounded lookback* expresses, intuitively, that the verification obligations in an LTL$_f^{MT}$ property can be evaluated in a sliding window and depend on only boundedly many steps from the past; it generalizes the feedback freedom property [Damaggio *et al.*, 2012]. Bounded lookback is defined via *computation graphs*: Let $w = \langle \varsigma_0, \ldots, \varsigma_{n-1} \rangle \in \Theta^*$ such that $q \rightarrow_w^* q'$ in $\mathcal{N}_\psi$, for some states q, q'. For each $0 \leq i < n$, let V_i be a fresh set of variables of the same size and sorts as V, and $\mathcal{V} = \bigcup_{i=0}^{n-1} V_i$. Consider the formulas $\varsigma_0(V/V_0)$, i.e., the formula obtained from ς_0 by systematically replacing V with V_0; and $\varsigma_i(V^{\lhd}/V_{i-1}, V/V_i)$, i.e., the formula obtained from ς_i by replacing $V^{\lhd}$ with V_{i-1} and V with V_i, for $0 < i < n$. The *computation graph* G_w of w is the undirected graph with nodes $\mathcal{V}$ and an edge from $u \in \mathcal{V}$ to $v \in \mathcal{V}$ iff the two variables occur in a common literal of $\varsigma_0(V/V_0) \wedge \bigwedge_{i \geq 1} \varsigma_i(V^{\lhd}/V_{i-1}, V/V_i)$. Finally, $[G_w]$ denotes the graph obtained from G_w by collapsing all edges corresponding to equality literals $u = v$ for $u, v \in \mathcal{V}$.

Definition 23. The property ψ has *K-bounded lookback* if for all states q, q' in the NFA $\mathcal{N}_\psi$ and all $w \in \Theta^*$ such that $q \rightarrow_w^* q'$, all acyclic paths in $[G_w]$ have length at most K.

E.g., $(x = f(y)) \mathsf{U}\, P(\lhd x, x)$ has 1-bounded lookback, but the property φ in Ex. 1 does not because it requires to compare prices of unboundedly many tickets. We show next that the monitoring task for bounded lookback properties is solvable. The proof shows that every formula occurring in a coreachability graph comes from a finite set of formulas since it has bounded quantifier depth and a finite vocabulary.

Theorem 24. *For every $K > 0$, the monitoring task is solvable for K-bounded-lookback properties.*

6 Conclusions

We have studied for the first time anticipatory monitoring of data-aware temporal properties expressed in LTL$_f^{MT}$, and identified expressive fragments of the logic for which monitors can be effectively constructed. Specifically, we show monitoring to be solvable for DB-LTL$_f$, an expressive class of first-order temporal properties where the data part combines arithmetic and references to a database. Our approach is fully implemented in the tool MONTHE, which employs CVC5 [Barbosa *et al.*, 2022] for dealing with data (SMT checks and quantifier elimination). Source code and web interface are available via *https://monitoring.adatool.dev*. The tool takes as input a property, and a trace (given by a sequence of assignments and a set of atoms to define the model); as output, MONTHE reports the monitoring state. Preliminary experiments are reported in the appendix, leaving for future work a more extensive empirical evaluation. Another interesting continuation is to consider dynamic systems where the data component *partially changes* over time. This is essential to capture controlled updates over a database, like updating the price of a ticket in our running example.

Acknowledgments

M. Montali was partially supported by the NextGenerationEU FAIR PE0000013 project MAIPM (CUP C63C22000770006). This work was also partly supported by Portuguese national funds through Fundação para a Ciência e a Tecnologia, I.P. (FCT) under projects UID/50021/2025 (DOI: https://doi.org/10.54499/UID/50021/2025) and UID/PRR/50021/2025 (DOI: https://doi.org/10.54499/UID/PRR/50021/2025).

References

[Abadi *et al.*, 2010] Aaron Abadi, Alexander Rabinovich, and Mooly Sagiv. Decidable fragments of many-sorted logic. *J. Symb. Comput.*, 45(2):153–172, 2010.

[Abdulla *et al.*, 1996] Parosh Aziz Abdulla, Karlis Cerans, Bengt Jonsson, and Yih-Kuen Tsay. General decidability theorems for infinite-state systems. In *Proc. LICS 1996*, pages 313–321, 1996.

[Alechina *et al.*, 2018] Natasha Alechina, Joseph Y. Halpern, Ian A. Kash, and Brian Logan. Incentive-compatible mechanisms for norm monitoring in open multi-agent systems (extended abstract). In *Proc. 27th IJCAI*, pages 5543–5547, 2018.

[Bacchus and Kabanza, 2000] Fahiem Bacchus and Froduald Kabanza. Using temporal logics to express search control knowledge for planning. *Artif. Intell.*, 116(1-2):123–191, 2000.

[Baier *et al.*, 2009] Jorge A. Baier, Fahiem Bacchus, and Sheila A. McIlraith. A heuristic search approach to planning with temporally extended preferences. *Artif. Intell.*, 173(5-6):593–618, 2009.

[Barbosa *et al.*, 2022] Haniel Barbosa, Clark W. Barrett, Martin Brain, Gereon Kremer, Hanna Lachnitt, Makai Mann, Abdalrhman Mohamed, Mudathir Mohamed, Aina Niemetz, Andres Nötzli, Alex Ozdemir, Mathias Preiner, Andrew Reynolds, Ying Sheng, Cesare Tinelli, and Yoni Zohar. cvc5: A versatile and industrial-strength SMT solver. In *Proc. 28th TACAS*, volume 13243 of *LNCS*, pages 415–442. Springer, 2022.

[Barrett *et al.*, 2021] Clark W. Barrett, Roberto Sebastiani, Sanjit A. Seshia, and Cesare Tinelli. Satisfiability modulo theories. In *Handbook of Satisfiability - Second Edition*, volume 336 of *Frontiers in Artificial Intelligence and Applications*, pages 1267–1329. IOS Press, 2021.

[Bartocci *et al.*, 2018] Ezio Bartocci, Yliès Falcone, Adrian Francalanza, and Giles Reger. Introduction to runtime verification. In *Lectures on Runtime Verification - Introductory and Advanced Topics*, volume 10457 of *LNCS*, pages 1–33. Springer, 2018.

[Basin *et al.*, 2015] David A. Basin, Felix Klaedtke, Samuel Müller, and Eugen Zalinescu. Monitoring metric first-order temporal properties. *J. ACM*, 62(2):15:1–15:45, 2015.

[Bauer *et al.*, 2010] Andreas Bauer, Martin Leucker, and Christian Schallhart. Comparing LTL semantics for runtime verification. *J. Logic and Comput.*, 20(3):651–674, 2010.

[Begemann *et al.*, 2023] Marian Johannes Begemann, Hannes Kallwies, Martin Leucker, and Malte Schmitz. TeSSLa-ROS-Bridge - runtime verification of robotic systems. In *Proc. ICTAC 2023*, volume 14446 of *LNCS*, pages 388–398, 2023.

[Bojańczyk *et al.*, 2013] Mikołaj Bojańczyk, Luc Segoufin, and Szymon Toruńczyk. Verification of database-driven systems via amalgamation. In *Proc. 32nd PODS*, pages 63–74, 2013.

[Calvanese *et al.*, 2013] Diego Calvanese, Giuseppe De Giacomo, and Marco Montali. Foundations of data-aware process analysis: a database theory perspective. In *Proc. 32nd PODS*, pages 1–12, 2013.

[Calvanese *et al.*, 2020a] Diego Calvanese, Silvio Ghilardi, Alessandro Gianola, Marco Montali, and Andrey Rivkin. Combined covers and Beth definability. In *Proc. 10th IJCAR*, volume 12166 of *LNCS*, pages 181–200, 2020.

[Calvanese *et al.*, 2020b] Diego Calvanese, Silvio Ghilardi, Alessandro Gianola, Marco Montali, and Andrey Rivkin. Smt-based verification of data-aware processes: a model-theoretic approach. *Math. Struct. Comput. Sci.*, 30(3):271–313, 2020.

[Calvanese *et al.*, 2022a] Diego Calvanese, Giuseppe De Giacomo, Marco Montali, and Fabio Patrizi. Verification and monitoring for first-order LTL with persistence-preserving quantification over finite and infinite traces. In *Proc. 31st IJCAI*, pages 2553–2560, 2022.

[Calvanese *et al.*, 2022b] Diego Calvanese, Silvio Ghilardi, Alessandro Gianola, Marco Montali, and Andrey Rivkin. Combination of uniform interpolants via Beth definability. *J. Automat. Reason.*, 66(3):409–435, 2022.

[Chiola *et al.*, 1997] Giovanni Chiola, Claude Dutheillet, Giuliana Franceschinis, and Serge Haddad. A symbolic reachability graph for coloured petri nets. *Theor. Comput. Sci.*, 176(1-2):39–65, 1997.

[Damaggio *et al.*, 2012] Elio Damaggio, Alin Deutsch, and Victor Vianu. Artifact systems with data dependencies and arithmetic. *ACM Trans. Database Syst.*, 37(3):22:1–22:36, 2012.

[D'Angelo *et al.*, 2005] Ben D'Angelo, Sriram Sankaranarayanan, César Sánchez, Will Robinson, Bernd Finkbeiner, Henny B. Sipma, Sandeep Mehrotra, and Zohar Manna. LOLA: runtime monitoring of synchronous systems. In *Proc. 12th TIME*, pages 166–174, 2005.

[Dastani *et al.*, 2018] Mehdi Dastani, Paolo Torroni, and Neil Yorke-Smith. Monitoring norms: a multi-disciplinary perspective. *Knowl. Eng. Rev.*, 33:1–22, 2018.

[De Giacomo and Vardi, 2013] Giuseppe De Giacomo and Moshe Y. Vardi. Linear temporal logic and linear dynamic logic on finite traces. In *Proc. 23rd IJCAI*, pages 854–860, 2013.

[De Giacomo *et al.*, 2014] Giuseppe De Giacomo, Riccardo de Masellis, and Marco Montali. Reasoning on LTL on finite traces: Insensitivity to infiniteness. In *Proc. 28th AAAI*, pages 1027–1033, 2014.

[De Giacomo *et al.*, 2020] Giuseppe De Giacomo, Marco Favorito, Luca Iocchi, Fabio Patrizi, and Alessandro Ronca. Temporal logic monitoring rewards via transducers. In *Proc. 17th KR*, pages 860–870, 2020.

[De Giacomo *et al.*, 2022] Giuseppe De Giacomo, Riccardo De Masellis, Fabrizio Maria Maggi, and Marco Montali. Monitoring constraints and metaconstraints with temporal logics on finite traces. *ACM Trans. Softw. Eng. Methodol.*, 31(4):1–44, 2022.

[de Leoni and van der Aalst, 2013] Massimiliano de Leoni and Wil M. P. van der Aalst. Data-aware process mining: discovering decisions in processes using alignments. In *Proc. 13th Symposium on Applied Computing*, pages 1454–1461. ACM, 2013.

[Decker *et al.*, 2016] Normann Decker, Martin Leucker, and Daniel Thoma. Monitoring modulo theories. *Int. J. Softw. Tools Technol. Transf.*, 18(2):205–225, 2016.

[Demri and D'Souza, 2007] Stéphane Demri and Deepak D'Souza. An automata-theoretic approach to constraint LTL. *Inform. Comput.*, 205(3):380–415, 2007.

[Desel and Esparza, 1993] Jörg Desel and Javier Esparza. Shortest paths in reachability graphs. In *Proc. of the 14th International Conference on Application and Theory of Petri Nets*, volume 691 of *LNCS*, pages 224–241. Springer, 1993.

[Deutsch *et al.*, 2018] Alin Deutsch, Richard Hull, Yuliang Li, and Victor Vianu. Automatic verification of database-centric systems. *ACM SIGLOG News*, 5(2):37–56, 2018.

[Felli *et al.*, 2019] Paolo Felli, Massimiliano de Leoni, and Marco Montali. Soundness verification of decision-aware process models with variable-to-variable conditions. In *Proc. 19th ACSD*, pages 82–91. IEEE, 2019.

[Felli *et al.*, 2023] Paolo Felli, Marco Montali, Fabio Patrizi, and Sarah Winkler. Monitoring arithmetic temporal properties on finite traces. In *Proc. 37th AAAI*, pages 6346–6354, 2023.

[Geatti *et al.*, 2022] Luca Geatti, Alessandro Gianola, and Nicola Gigante. Linear temporal logic modulo theories over finite traces. In *Proc. 31st IJCAI*, pages 2641–2647, 2022.

[Ghilardi *et al.*, 2023] Silvio Ghilardi, Alessandro Gianola, Marco Montali, and Andrey Rivkin. Safety verification and universal invariants for relational action bases. In *Proc. 32nd IJCAI*, pages 3248–3257, 2023.

[Ghilardi, 2004] Silvio Ghilardi. Model-theoretic methods in combined constraint satisfiability. *J. Automat. Reason.*, 33(3-4):221–249, 2004.

[Gianola *et al.*, 2024] Alessandro Gianola, Marco Montali, and Sarah Winkler. Linear-time verification of data-aware processes modulo theories via covers and automata. In *Proc. 38th AAAI*, pages 10525–10534. AAAI Press, 2024.

[Gianola *et al.*, 2025] Alessandro Gianola, Marco Montali, and Sarah Winkler. SMT techniques for data-aware process mining. *Künstliche Intell.*, 39(3):221–237, 2025.

[Gianola *et al.*, 2026] Alessandro Gianola, Marco Montali, and Sarah Winkler. Monitoring data-aware temporal properties (extended version), 2026. https://doi.org/10.48550/arXiv.2605.14666.

[Goldberg and Havelund, 2005] Allen Goldberg and Klaus Havelund. Automated runtime verification with Eagle. In *Proc. 3rd MSVVEIS*. INSTICC Press, 2005.

[Havelund *et al.*, 2020] Klaus Havelund, Doron Peled, and Dogan Ulus. First-order temporal logic monitoring with bdds. *Formal Methods Syst. Des.*, 56(1):1–21, 2020.

[Hipler *et al.*, 2024] Raik Hipler, Hannes Kallwies, Martin Leucker, and César Sánchez. General anticipatory runtime verification. In *Proc. 36th CAV*, volume 14682 of *LNCS*, pages 133–155, 2024.

[Kallwies *et al.*, 2022] Hannes Kallwies, Martin Leucker, Malte Schmitz, Albert Schulz, Daniel Thoma, and Alexander Weiss. TeSSLa - an ecosystem for runtime verification. In *Proc. 22nd Runtime Verification*, volume 13498 of *LNCS*, pages 314–324, 2022.

[Leucker and Schallhart, 2009] Martin Leucker and Christian Schallhart. A brief account of runtime verification. *J. Log. Algebraic Methods Program.*, 78(5):293–303, 2009.

[Lima *et al.*, 2024] Leonardo Lima, Jonathan Julián Huerta y Munive, and Dmitriy Traytel. Explainable online monitoring of metric first-order temporal logic. In *Proc. 30th TACAS*, volume 14570 of *LNCS*, pages 288–307, 2024.

[Lipparini, 1982] Paolo Lipparini. Locally finite theories with model companion. *Atti della Accademia Nazionale dei Lincei. Classe di Scienze Fisiche, Matematiche e Naturali. Rendiconti, Serie 8*, 72, 1982.

[Ly *et al.*, 2015] Linh Thao Ly, Fabrizio Maria Maggi, Marco Montali, Stefanie Rinderle-Ma, and Wil M. P. van der Aalst. Compliance monitoring in business processes: Functionalities, application, and tool-support. *Inf. Syst.*, 54:209–234, 2015.

[Presburger, 1929] Mojżesz Presburger. Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. In *Comptes Rendus du I congres de Mathem. des Pays Slaves*, pages 92–101, 1929.

[Schneider *et al.*, 2021] Joshua Schneider, David A. Basin, Frederik Brix, Srdan Krstic, and Dmitriy Traytel. Scalable online first-order monitoring. *Int. J. Softw. Tools Technol. Transf.*, 23(2):185–208, 2021.

[Tracy *et al.*, 2020] Tommy Tracy, Lucas M. Tabajara, Moshe Y. Vardi, and Kevin Skadron. Runtime verification on FPGAs with LTLf specifications. In *Proc. FMCAD 2020*, pages 36–46, 2020.