

Do You Need Infinite Time?

Alessandro Artale, Andrea Mazzullo and Ana Ozaki

KRDB Research Centre – Free University of Bozen–Bolzano

{artale, mazzullo, ozaki}@inf.unibz.it

Abstract

Linear temporal logic over finite traces is used as a formalism for temporal specification in automated planning, process modelling and (runtime) verification. In this paper, we investigate first-order temporal logic over finite traces, lifting some known results to a more expressive setting. Satisfiability in the two-variable monodic fragment is shown to be EXPSPACE-complete, as for the infinite trace case, while it decreases to NEXPTIME when we consider finite traces bounded in the number of instants. This leads to new complexity results for temporal description logics over finite traces. We further investigate satisfiability and equivalences of formulas under a model-theoretic perspective, providing a set of semantic conditions that characterise when the distinction between reasoning over finite and infinite traces can be blurred. Finally, we apply these conditions to planning and verification.

1 Introduction

Since the introduction of *linear temporal logic* (LTL), several propositional and first-order LTL-based formalisms have been developed for applications such as automated planning [Bacchus and Kabanza, 2000; Baier and McIlraith, 2006], process modelling [van der Aalst and Pesic, 2006; Maggi *et al.*, 2011] and verification of programs [Manna and Pnueli, 1995]. Related research in knowledge representation has focused on decidable fragments of first-order temporal logic, $T_{\mathcal{U}}\mathcal{QL}$ [Hodkinson *et al.*, 2000; Gabbay *et al.*, 2003], and in particular on *temporal description logics* [Wolter and Zakharyashev, 1998; Artale and Franconi, 2005] that combine LTL operators with description logics (DLs). These logics usually lie within the two-variable monodic fragment of $T_{\mathcal{U}}\mathcal{QL}$, $T_{\mathcal{U}}\mathcal{QL}_{\square}^2$, obtained by restricting to formulas having at most two variables, and so that the temporal operators are applied only to subformulas with at most one free variable. The complexity of the satisfiability problem ranges from EXPSPACE-complete, for $T_{\mathcal{U}}\mathcal{QL}_{\square}^2$ [Gabbay *et al.*, 2003; Hodkinson *et al.*, 2003], down to NEXPTIME- or EXPTIME-complete, for temporal extensions of \mathcal{ALC} without temporalised roles and with restrictions on the application of temporal operators [Lutz *et al.*, 2008;

Baader *et al.*, 2012], or even in NP or NLOGSPACE, by modifying further both the temporal and the DL components, as in some temporal extensions of *DL-Lite* for conceptual data modelling [Artale *et al.*, 2014].

Besides the usual LTL semantics defined on infinite linear structures, attention has been devoted also to *finite traces*, which are linear temporal structures with only a finite number of time points [Cerrito *et al.*, 1999; De Giacomo and Vardi, 2013; Fionda and Greco, 2016]. Indeed, the finiteness of the time dimension is a fairly natural restriction. In automated planning, or when modelling business processes with a declarative formalism, we consider finite action plans and terminating services, often within a given temporal bound [Bauer and Haslum, 2010; De Giacomo *et al.*, 2014a; De Giacomo *et al.*, 2014b; Camacho *et al.*, 2017]. In runtime verification, only the current finite behaviour of the system is taken into account. Infinite models are then considered to check whether a given requirement is satisfied in the infinite extensions of this finite trace [Giannakopoulou and Havelund, 2001; Bauer *et al.*, 2010]. Although some recent work has considered temporal extensions of DLs in the context of runtime verification [Baader and Lippmann, 2014] and business process modelling [van der Aalst *et al.*, 2017], the proposals developed so far are based on the usual infinite trace semantics and are limited in expressivity.

This work focuses on first-order temporal logic on finite traces [Cerrito *et al.*, 1999], defined by extending the first-order language with temporal operators interpreted on finite traces. We obtain the following main results.

- The complexity of $T_{\mathcal{U}}\mathcal{QL}_{\square}^2$ remains EXPSPACE-complete on finite traces, but lowers down to NEXPTIME if we restrict to traces with a bound k on the number of instants. Moreover, it has the bounded model property.
- Similar results (including the bounded model property) also hold for the temporal DL $T_{\mathcal{U}}\mathcal{ALC}$ on finite traces, where the complexity further reduces to EXPTIME if only global axioms and finite traces with a fixed bound on instants are allowed.
- We establish semantic and syntactic conditions which characterise when the distinction between reasoning on finite and infinite traces can be completely blurred, providing connections with planning and verification.

Concerning this last point, several approaches have been con-

sidered to preserve satisfiability of formulas from the finite to the infinite case, so to reuse algorithms developed for the infinite case [Bauer and Haslum, 2010; De Giacomo *et al.*, 2014b]. Here, we determine conditions that preserve satisfiability also in the other direction, from infinite to finite traces. This opens interesting research directions towards the application of efficient finite traces reasoners [Li *et al.*, 2014] to the infinite case. We provide a uniform framework for these notions, bridging finite and infinite trace semantics. Full proofs are available in our technical report [Artale *et al.*, 2019].

2 First-order Temporal Logics

The *first-order temporal language*, $T_{\mathcal{U}}\mathcal{QL}$ [Gabbay *et al.*, 2003], is obtained by extending the usual first-order language with the temporal operator *until* (\mathcal{U}) interpreted over discrete linear structures, called *traces*.

Syntax The *alphabet* of $T_{\mathcal{U}}\mathcal{QL}$ consists of countably infinite and pairwise disjoint sets of *predicates* \mathbf{N}_P (with $\text{ar}(P) \in \mathbb{N}$ being the arity of $P \in \mathbf{N}_P$), *constants* (or *individual names*) \mathbf{N}_I , and *variables* \mathbf{Var} ; the *logical operators* \neg, \wedge ; the *existential quantifier* \exists , and the *temporal operator* \mathcal{U} (*until*). The *formulas* of $T_{\mathcal{U}}\mathcal{QL}$ are of the form:

$$\varphi, \psi ::= P(\bar{\tau}) \mid \neg\varphi \mid \varphi \wedge \psi \mid \exists x\varphi \mid \varphi \mathcal{U} \psi,$$

where $P \in \mathbf{N}_P$, $\bar{\tau} = (\tau_1, \dots, \tau_{\text{ar}(P)})$ is a tuple of *terms*, i.e., constants or variables, and $x \in \mathbf{Var}$. Formulas without the until operator are called *non-temporal*. We write $\varphi(x_1, \dots, x_m)$ to indicate that the free variables of a formula φ are in $\{x_1, \dots, x_m\}$. For $p \in \mathbb{N}$, the *p-variable fragment* of $T_{\mathcal{U}}\mathcal{QL}$, denoted by $T_{\mathcal{U}}\mathcal{QL}^p$, consists of $T_{\mathcal{U}}\mathcal{QL}$ formulas with at most p variables ($T_{\mathcal{U}}\mathcal{QL}^0$ is simply propositional *LTL*). The *monodic fragment* of $T_{\mathcal{U}}\mathcal{QL}$, denoted by $T_{\mathcal{U}}\mathcal{QL}_{\square}$, consists of formulas such that all subformulas of the form $\varphi \mathcal{U} \psi$ have at most one free variable.

Semantics A *first-order temporal interpretation* is a structure $\mathfrak{M} = (\Delta^{\mathfrak{M}}, (I_n)_{n \in \mathfrak{T}})$, where \mathfrak{T} is an interval of the form $[0, \infty)$ or $[0, l]$, with $l \in \mathbb{N}$, and each I_n is a classical first-order interpretation with domain $\Delta^{\mathfrak{M}}$ (or simply Δ): we have $P^{I_n} \subseteq \Delta^{\text{ar}(P)}$, for each $P \in \mathbf{N}_P$, and $a^{I_i} = a^{I_j} \in \Delta$ for all $a \in \mathbf{N}_I$ and $i, j \in \mathbb{N}$, i.e., constants are *rigid designators* (with fixed interpretation, denoted simply by a^I). The stipulation that all time points share the same domain Δ is called the *constant domain assumption* (meaning that objects are not created or destroyed over time), and it is the most general choice in the sense that increasing, decreasing, and varying domains can all be reduced to it [Gabbay *et al.*, 2003]. An *assignment* in \mathfrak{M} is a function \mathfrak{a} from terms to Δ : $\mathfrak{a}(\tau) = \mathfrak{a}(x)$, if $\tau = x$, and $\mathfrak{a}(\tau) = a^I$, if $\tau = a \in \mathbf{N}_I$ (given a tuple of m terms $\bar{\tau} = (\tau_1, \dots, \tau_m)$, we set $\mathfrak{a}(\bar{\tau}) = (\mathfrak{a}(\tau_1), \dots, \mathfrak{a}(\tau_m))$). *Satisfaction* of a formula φ in \mathfrak{M} at time point $n \in \mathfrak{T}$ under assignment \mathfrak{a} (written $\mathfrak{M}, n \models^{\mathfrak{a}} \varphi$) is inductively defined as:

$$\begin{aligned} \mathfrak{M}, n \models^{\mathfrak{a}} P(\bar{\tau}) & \text{ iff } \mathfrak{a}(\bar{\tau}) \in P^{I_n}, \\ \mathfrak{M}, n \models^{\mathfrak{a}} \neg\varphi & \text{ iff } \text{not } \mathfrak{M}, n \models^{\mathfrak{a}} \varphi, \\ \mathfrak{M}, n \models^{\mathfrak{a}} \varphi \wedge \psi & \text{ iff } \mathfrak{M}, n \models^{\mathfrak{a}} \varphi \text{ and } \mathfrak{M}, n \models^{\mathfrak{a}} \psi, \\ \mathfrak{M}, n \models^{\mathfrak{a}} \exists x\varphi & \text{ iff } \mathfrak{M}, n \models^{\mathfrak{a}'} \varphi \text{ for some assignment } \mathfrak{a}' \\ & \text{ that can differ from } \mathfrak{a} \text{ only on } x, \\ \mathfrak{M}, n \models^{\mathfrak{a}} \varphi \mathcal{U} \psi & \text{ iff } \exists m \in \mathfrak{T}, m > n : \mathfrak{M}, m \models^{\mathfrak{a}} \psi \text{ and} \\ & \forall i \in (n, m) : \mathfrak{M}, i \models^{\mathfrak{a}} \varphi. \end{aligned}$$

We say that φ is *satisfied* in \mathfrak{M} (and \mathfrak{M} is a *model* of φ), writing $\mathfrak{M} \models \varphi$, if $\mathfrak{M}, 0 \models^{\mathfrak{a}} \varphi$, for some \mathfrak{a} . Moreover, φ is said to be *satisfiable* if it is satisfied in some \mathfrak{M} . A formula φ *logically implies* a formula ψ if every \mathfrak{M} that satisfies φ satisfies also ψ , and we write $\varphi \models \psi$. We say that φ and ψ are *equivalent*, writing $\varphi \equiv \psi$, if $\varphi \models \psi$ and $\psi \models \varphi$.

In the following, we call *finite trace* a first-order temporal interpretation with $\mathfrak{T} = [0, l]$, often denoted by $\mathfrak{F} = (\Delta^{\mathfrak{F}}, (\mathcal{F}_n)_{n \in [0, l]})$, while *infinite traces*, based on $\mathfrak{T} = [0, \infty)$, will be denoted by $\mathfrak{I} = (\Delta^{\mathfrak{I}}, (\mathcal{I}_n)_{n \in [0, \infty)})$. We say that a $T_{\mathcal{U}}\mathcal{QL}$ formula φ is *satisfiable on infinite, finite, or k-bounded traces*, if it is satisfied in a trace in the class of infinite, finite, or finite traces with at most $k \in \mathbb{N}, k > 0$ (given in binary) time points, respectively.

Let $\mathfrak{F} = (\Delta^{\mathfrak{F}}, (\mathcal{F}_n)_{n \in [0, l]})$ and $\mathfrak{I} = (\Delta^{\mathfrak{I}}, (\mathcal{I}_n)_{n \in [0, \infty)})$ be, respectively, a finite and an infinite trace s.t. $\Delta^{\mathfrak{F}} = \Delta^{\mathfrak{I}}$ (writing Δ) and $a^{\mathcal{F}} = a^{\mathcal{I}}$, for all $a \in \mathbf{N}_I$. We denote by $\mathfrak{F} \cdot \mathfrak{I} = (\Delta^{\mathfrak{F} \cdot \mathfrak{I}}, (\mathcal{F} \cdot \mathcal{I}_n)_{n \in [0, \infty)})$ the *extension* of \mathfrak{F} with \mathfrak{I} , defined as the infinite trace with $\Delta^{\mathfrak{F} \cdot \mathfrak{I}} = \Delta$, $a^{\mathcal{F} \cdot \mathcal{I}} = a^{\mathcal{F}}$, for all $a \in \mathbf{N}_I$, and for $P \in \mathbf{N}_P, n \in \mathbb{N}$:

$$P^{\mathcal{F} \cdot \mathcal{I}_n} = \begin{cases} P^{\mathcal{F}_n}, & \text{if } n \in [0, l] \\ P^{\mathcal{I}_{n-(l+1)}}, & \text{if } n \in [l+1, \infty). \end{cases}$$

In addition to the standard Boolean equivalences, we will use the following equivalences for formulas: $\perp \equiv \exists x(P(x) \wedge \neg P(x))$, $\top \equiv \neg \perp$; $\circ\varphi \equiv \circ^1\varphi \equiv \perp \mathcal{U} \varphi$; $\circ^q\varphi \equiv \circ\circ^{q-1}\varphi$, with $q > 1$; $\diamond\varphi \equiv \top \mathcal{U} \varphi$; $\square\varphi \equiv \neg \diamond \neg \varphi$; $\diamond^+\varphi \equiv \varphi \vee \diamond\varphi$; and $\circ^+\varphi \equiv \neg \diamond^+ \neg \varphi$. Over finite traces, *last* $\equiv \neg \circ \top$ is satisfiable in the *last* time point.

3 Satisfiability over Finite Traces

In the following we show how to reduce the formula satisfiability problem over finite traces to the same problem over infinite traces. Similar to the encoding proposed in [De Giacomo and Vardi, 2013] for propositional *LTL*, to capture the finiteness of the temporal dimension, we introduce a fresh predicate E (standing for the *end of time*) with the following properties: (i) there is a least one instant before the end of time; (ii) the end of time comes for all objects; (iii) the end of time comes at the same time for every object; (iv) the end of time never goes away. We axiomatise these properties as follows:

$$\begin{aligned} \psi_{f_1} &= \forall x \neg E(x) && \text{(Point (i)),} \\ \psi_{f_2} &= (\forall x \neg E(x)) \mathcal{U} (\forall x E(x)) && \text{(Points (ii), (iii)),} \\ \psi_{f_3} &= \square \forall x (E(x) \rightarrow \circ E(x)) && \text{(Point (iv)).} \end{aligned}$$

We now characterise models satisfying the *end of time formula* $\psi_f = \psi_{f_1} \wedge \psi_{f_2} \wedge \psi_{f_3}$. Let $\mathfrak{F} = (\Delta, (\mathcal{F}_n)_{n \in [0, l]})$ and $\mathfrak{I} = (\Delta, (\mathcal{I}_n)_{n \in [0, \infty)})$ be, respectively, a finite and an infinite trace with the same domain Δ and such that $a^{\mathcal{F}} = a^{\mathcal{I}}$, for all $a \in \mathbf{N}_I$. We denote by $\mathfrak{F} \cdot_E \mathfrak{I}$ the *end extension* of \mathfrak{F} with \mathfrak{I} , defined as the extension $\mathfrak{F} \cdot \mathfrak{I}$, for all $P \in \mathbf{N}_P \setminus \{E\}$, such that:

$$E^{\mathcal{F} \cdot_E \mathcal{I}_n} = \begin{cases} \emptyset, & \text{if } n \in [0, l] \\ \Delta, & \text{if } n \in [l+1, \infty) \end{cases}$$

Clearly, the extension of E characterises the satisfiability of ψ_f . We formalise this in the next lemma.

Lemma 1. For every infinite trace \mathcal{J} , $\mathcal{J} \models \psi_f$ iff $\mathcal{J} = \mathfrak{F} \cdot_E \mathcal{J}'$, for some finite trace \mathfrak{F} and some infinite trace \mathcal{J}' .

We now introduce a translation \cdot^\dagger for $T_U\mathcal{QL}$ formulas, used together with the end of time formula, ψ_f , to capture satisfiability on finite traces. More formally, a $T_U\mathcal{QL}$ formula φ is satisfiable on finite traces if and only if its translation φ^\dagger is satisfied in an infinite trace that also satisfies the formula ψ_f . The translation \cdot^\dagger is defined as:

$$\begin{aligned} (P(\bar{\tau}))^\dagger &\mapsto P(\bar{\tau}), & (\neg\varphi)^\dagger &\mapsto \neg\varphi^\dagger, \\ (\varphi \wedge \psi)^\dagger &\mapsto \varphi^\dagger \wedge \psi^\dagger, & (\exists x\varphi)^\dagger &\mapsto \exists x\varphi^\dagger, \\ (\varphi \mathcal{U} \psi)^\dagger &\mapsto \varphi^\dagger \mathcal{U} (\psi^\dagger \wedge \psi_{f_1}). \end{aligned}$$

Lemma 2 states the correctness of \cdot^\dagger .

Lemma 2. Let $\mathfrak{F} \cdot_E \mathcal{J}$ be an end extension of a finite trace \mathfrak{F} . For every $T_U\mathcal{QL}$ formula φ and every assignment \mathfrak{a} , $\mathfrak{F} \models^{\mathfrak{a}} \varphi$ iff $\mathfrak{F} \cdot_E \mathcal{J} \models^{\mathfrak{a}} \varphi^\dagger$.

From the previous lemmas, we obtain a reduction of the $T_U\mathcal{QL}$ satisfiability problem to the same problem for $T_U\mathcal{QL}$.

Theorem 3. Let φ be a $T_U\mathcal{QL}$ formula. Then φ is satisfiable on finite traces iff $\varphi^\dagger \wedge \psi_f$ is satisfiable on infinite traces.

We use the translation to transfer the EXPSPACE upper bound for $T_U\mathcal{QL}_{\square}^2$ over infinite traces to finite traces [Hodkinson *et al.*, 2003]. The lower bound can be proved using similar ideas as those used to prove hardness of $T_U\mathcal{QL}_{\square}^2$.

Theorem 4. $T_U\mathcal{QL}_{\square}^2$ satisfiability on finite traces is EXPSPACE-complete.

We now study $T_U\mathcal{QL}_{\square}^2$ satisfiability on traces with at most k time points, with k given in binary, as part of the input. We show that in this case the complexity of the satisfiability problem decreases from EXPSPACE to NEXPTIME. Hardness follows from the fact that: (1) one can translate \mathcal{ALC} -LTL with rigid concepts to $T_U\mathcal{QL}_{\square}^2$; and (2) satisfiability in \mathcal{ALC} -LTL with rigid concepts on k -bounded traces is NEXPTIME-hard, as for infinite traces [Baader *et al.*, 2012, Lemma 6.2]. For the upper bound, we resort to a classical abstraction of models called *quasimodels* [Gabbay *et al.*, 2003]. One can show that there is a model with at most k time points iff there is a quasimodel with a sequence of states (sets of subformulas with certain constraints) of length at most k . Then, our upper bound is obtained by guessing an exponential size sequence of states which serves as a certificate for the existence of a quasimodel (and therefore a model) for the input formula.

Theorem 5. $T_U\mathcal{QL}_{\square}^2$ satisfiability on k -bounded traces is NEXPTIME-complete.

We end this section by establishing that $T_U\mathcal{QL}_{\square}^2$ on finite traces enjoys the bounded trace property and the bounded domain property. If there is a finite trace which satisfies a $T_U\mathcal{QL}_{\square}^2$ formula φ then there is a finite trace with at most k time points, with k double exponentially large w.r.t. the size of φ . This bound follows from the fact that (1) if there is a quasimodel for φ then there is a quasimodel for φ where there is no repetition of states [Gabbay *et al.*, 2003], except for the last state; and (2) the fact that the length of the finite non-repeating sequence of states is correlated with the number

of time points in a finite trace. The number of distinct states of a $T_U\mathcal{QL}_{\square}^2$ formula on finite traces is the same as in the infinite case, where it is known to be double exponential [Gabbay *et al.*, 2003]. This directly implies that $T_U\mathcal{QL}_{\square}^2$ enjoys the *bounded trace property*.

Theorem 6. $T_U\mathcal{QL}_{\square}^2$ satisfiability of φ on finite traces implies $T_U\mathcal{QL}_{\square}^2$ satisfiability of φ on k -bounded traces, with k double exponential in $|\varphi|$.

We now establish that a $T_U\mathcal{QL}_{\square}^2$ formula has the *bounded domain property*: if it is satisfiable, then there is a model with the size of the domain exponential in k (meaning that it is double exponential in the binary representation of k).

Theorem 7. $T_U\mathcal{QL}_{\square}^2$ satisfiability of φ on k -bounded traces implies the existence of a model with domain size exponential in k and φ .

Since $T_U\mathcal{QL}_{\square}^2$ satisfiability on finite traces implies satisfiability on k -bounded traces, for some $k > 0$, the formula

$$\square^+ \forall x (P(x) \rightarrow \square^+ (\neg P(x) \wedge \exists y R(x, y) \wedge P(y))), \quad (*)$$

which only admits models with an infinite domain [Lutz *et al.*, 2008], is unsatisfiable over finite traces.

4 Finite vs. Infinite Traces

While certain formulas, such as $\square\top$, are satisfiable both on finite and infinite traces, others, e.g., $\diamond last$ and $\square^+\square\top$, are only satisfiable on finite traces and on infinite traces, respectively. One then wonders, when does satisfiability on finite and infinite traces coincide, so that solvers can simply stop trying to build the lasso of an infinite trace when a finite trace is built? A similar question can be posed for the problem of equivalences between formulas. For example, $\diamond\square(\varphi \vee \psi)$ and $(\diamond\square\varphi) \vee (\diamond\square\psi)$ are equivalent on finite traces but not on infinite traces [Bauer and Haslum, 2010]. Moreover, $\square^+\diamond^+\varphi$ and $\diamond^+\square^+\varphi$ are not equivalent on infinite traces, whereas on finite traces they are both equivalent to $\diamond^+(last \wedge \varphi)$ [De Giacomo and Vardi, 2013]. Conversely, \perp and *last* are only equivalent on infinite traces.

In this section we address these questions and investigate the distinction between reasoning on finite and on infinite traces. More specifically, we propose semantic properties which guarantee that formula satisfiability and equivalences between formulas are preserved, and thus, the distinction can be blurred. Given a finite trace \mathfrak{F} , we define the set of *extensions* of \mathfrak{F} as the set $Ext(\mathfrak{F}) = \{\mathcal{J} \mid \mathcal{J} = \mathfrak{F} \cdot \mathcal{J}', \text{ for some } \mathcal{J}'\}$. Instead, given an infinite trace \mathcal{J} , the set of *prefixes* of \mathcal{J} is the set $Pre(\mathcal{J}) = \{\mathfrak{F} \mid \mathcal{J} = \mathfrak{F} \cdot \mathcal{J}', \text{ for some } \mathcal{J}'\}$. For a $T_U\mathcal{QL}$ formula φ and a quantifier $Q \in \{\exists, \forall\}$, we say that φ is F_Q if, for all finite traces \mathfrak{F} and all assignments \mathfrak{a} , it satisfies the *finite trace property*:

$$\mathfrak{F} \models^{\mathfrak{a}} \varphi \Leftrightarrow Q\mathcal{J} \in Ext(\mathfrak{F}). \mathcal{J} \models^{\mathfrak{a}} \varphi,$$

and, similarly, φ is I_Q if, for all infinite traces \mathcal{J} and all assignments \mathfrak{a} , it satisfies the *infinite trace property*:

$$\mathcal{J} \models^{\mathfrak{a}} \varphi \Leftrightarrow Q\mathfrak{F} \in Pre(\mathcal{J}). \mathfrak{F} \models^{\mathfrak{a}} \varphi.$$

Examples of formulas satisfying F_{\exists} and I_{\forall} are formulas of the form $\square^+\varphi$, where φ is a formula without temporal operators

Property	Formulas
F_{\exists}	$\diamond^+last \vee \diamond P(x);$ $\diamond^+last \vee \diamond P(x) \vee (*).$
F_{\forall}	$\diamond^+P(x);$ $\diamond^+P(x) \vee (*).$
l_{\exists}	$\square \circ T \vee last;$ $\square \circ T \vee \circ last.$
l_{\forall}	$\square^+P(x) \vee \diamond^+(P(x) \wedge last);$ $\square^+P(x) \vee \diamond^+(P(x) \wedge \circ last).$

Table 1: Formulas satisfying exactly one of the finite/infinite trace properties (Formula (*) is from Section 3).

(formulas of the form $\square\varphi$ are also l_{\forall} but not F_{\exists} because of, e.g., $\square\perp$). On the other hand, the properties F_{\forall} and l_{\exists} capture formulas of the form $\diamond^+\varphi$, where φ is again a formula without temporal operators (formulas of the form $\diamond\varphi$ are also l_{\exists} but not F_{\forall} because of, e.g., $\diamond T$).

The semantic properties, F_Q and l_Q , capture different classes of $T_{\mathcal{U}}\mathcal{QL}$ formulas, as we illustrate in Table 1. We formalise this statement with the following theorem.

Theorem 8. *Let $T_{\mathcal{U}}\mathcal{QL}(P)$ denote the set of $T_{\mathcal{U}}\mathcal{QL}$ formulas which satisfy property P . The sets $T_{\mathcal{U}}\mathcal{QL}(F_{\exists})$, $T_{\mathcal{U}}\mathcal{QL}(F_{\forall})$, $T_{\mathcal{U}}\mathcal{QL}(l_{\exists})$, and $T_{\mathcal{U}}\mathcal{QL}(l_{\forall})$ are mutually incomparable.*

One can also restrict to the ‘one directional’ version of the above properties. We denote by $F_{\circ Q}$ and $l_{\circ Q}$, where $\circ \in \{\Rightarrow, \Leftarrow\}$, the corresponding ‘ \Rightarrow ’ and ‘ \Leftarrow ’ directions of the F_Q and l_Q properties, respectively. On the relationship between these one directional properties, we have the following. Given two different quantifiers Q and Q' (among \exists and \forall), a $T_{\mathcal{U}}\mathcal{QL}$ formula φ is $F_{\Rightarrow Q}$ if, and only if, its negation $\neg\varphi$ is $F_{\Leftarrow Q'}$. Similarly, φ is $l_{\Rightarrow Q}$ if, and only if, $\neg\varphi$ is $l_{\Leftarrow Q'}$. Moreover, if a $T_{\mathcal{U}}\mathcal{QL}$ formula φ is $F_{\Rightarrow\forall}$, then it is also $F_{\Rightarrow\exists}$, and if φ is $l_{\Rightarrow\forall}$, it is $l_{\Rightarrow\exists}$ as well. However, the sets of formulas satisfying $F_{\Rightarrow\forall}$ and $l_{\Rightarrow\exists}$ are incomparable. Indeed, the formula $\square^+\circ T$ is $F_{\Rightarrow\forall}$ and not $l_{\Rightarrow\exists}$. On the other hand, the sets of formulas satisfying $l_{\Rightarrow\exists}$ and $F_{\Rightarrow\forall}$ are also incomparable. To see this, consider $\diamond last$, which is $l_{\Rightarrow\exists}$ but not $F_{\Rightarrow\forall}$.

In Theorem 3, we have proved that $T_{\mathcal{U}}\mathcal{QL}$ formulas interpreted on finite traces can be translated into equisatisfiable formulas on infinite traces. Such translation is not always needed, since for some classes of formulas satisfiability is already preserved. Indeed, for a $T_{\mathcal{U}}\mathcal{QL}$ formula φ : if φ is $F_{\Rightarrow\exists}$, then it is satisfiable on finite traces only if it is satisfiable on infinite traces; moreover, if φ is $l_{\Rightarrow\exists}$, then it is satisfiable on infinite traces only if it is satisfiable on finite ones.

We now consider the problem of formula equivalence, by showing under which semantic properties equivalence between formulas can be blurred. Given $T_{\mathcal{U}}\mathcal{QL}$ formulas φ and ψ , we write $\varphi \equiv_{\exists} \psi$ if φ and ψ are equivalent on infinite traces, and $\varphi \equiv_{\exists}^f \psi$ if they are equivalent on finite traces. The following theorem provides sufficient conditions to preserve formula equivalence from the infinite to the finite case (cf. the notion of *LTL compliance* in [Bauer et al., 2010]).

Theorem 9. *Given $T_{\mathcal{U}}\mathcal{QL}$ formulas φ and ψ , $\varphi \equiv_{\exists} \psi$ implies $\varphi \equiv_{\exists}^f \psi$ whenever both φ and ψ are (1) F_{\exists} ; or (2) F_{\forall} ; or (3) $F_{\Rightarrow\exists}$ and $l_{\Rightarrow\forall}$.*

Theorem 9 does not hold for formulas that satisfy only l_{\exists} or l_{\forall} . Consider the formulas $\square \circ T \vee last$ and $\square \circ T \vee \circ last$, which are l_{\exists} . These formulas are equivalent only on infinite traces. Also, $\square^+P(x) \vee \diamond^+(P(x) \wedge last)$ and $\square^+P(x) \vee \diamond^+(P(x) \wedge \circ last)$ are l_{\forall} , and equivalent on infinite but not on finite traces. This example also shows that the condition $l_{\Rightarrow\forall}$ alone is not sufficient for Theorem 9. Moreover, $F_{\Rightarrow\exists}$ alone is also not sufficient. To see this, consider, e.g., $\square^+\diamond T \vee (P(x) \wedge \diamond last)$ and $\square^+\diamond T \vee (\diamond last)$, which are $F_{\Rightarrow\exists}$ but are equivalent only over infinite traces. We now present sufficient conditions to preserve equivalences from the finite to the infinite case.

Theorem 10. *Given $T_{\mathcal{U}}\mathcal{QL}$ formulas φ and ψ , $\varphi \equiv_{\exists} \psi$ implies $\varphi \equiv_{\exists}^f \psi$ whenever both φ and ψ are (1) l_{\exists} ; or (2) l_{\forall} ; or (3) $F_{\Rightarrow\forall}$ and $l_{\Rightarrow\exists}$.*

We point out that F_{\exists} or F_{\forall} are not sufficient to ensure that formula equivalence on finite traces implies formula equivalence on infinite traces. To illustrate this, consider for example the formulas $\Phi = \diamond^+last \vee \diamond P(x)$, and the union of Φ and Formula (*) from Section 3. These formulas are F_{\exists} , however, they are only equivalent on finite traces. Moreover, if we take $\diamond^+P(x)$ and the union of $\diamond^+P(x)$ with Formula (*), we have that they are both F_{\forall} , though equivalent only on finite traces. This example also shows that the condition $F_{\Rightarrow\forall}$ alone is not sufficient for Theorem 10. We now argue that $l_{\Rightarrow\exists}$ alone is also not sufficient. To see this, consider, e.g., $(P(x) \wedge \square^+\diamond T) \vee \diamond last$ and $\square^+\diamond T \vee \diamond last$, which are $l_{\Rightarrow\exists}$ but are equivalent only on finite traces.

From Theorems 9 and 10 we have that if both φ and ψ are F_{\exists} or F_{\forall} , and l_{\exists} or l_{\forall} $T_{\mathcal{U}}\mathcal{QL}$ formulas, then, $\varphi \equiv_{\exists} \psi$ iff $\varphi \equiv_{\exists}^f \psi$. In particular, the above examples show that if, from a given pair of conditions F_Q and $l_{Q'}$, we remove any of the two properties, then formula equivalences on finite and infinite traces may not coincide. We now analyse syntactic features of the properties introduced so far, providing a class of formulas satisfying them.

Theorem 11. *All non-temporal $T_{\mathcal{U}}\mathcal{QL}$ formulas satisfy the finite/infinite trace properties F_{\exists} , F_{\forall} , l_{\exists} , and l_{\forall} .*

\diamond^+ -formulas φ, ψ are built according to (with $P \in \mathcal{N}_P$):

$$\diamond^+\varphi \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \exists x\varphi \mid P(\bar{\tau}) \mid \neg P(\bar{\tau})$$

We now show that the language generated by the grammar rule for \diamond^+ -formulas contains only formulas which are F_{\forall} and l_{\exists} . We call \diamond -formulas the set of formulas generated by the result of further allowing $\diamond\varphi$ in the grammar rule for \diamond^+ -formulas; and call $\diamond^+\forall$ -formulas the result of allowing $\forall x\varphi$ in the grammar rule for \diamond^+ -formulas.

Theorem 12. *All \diamond^+ -formulas are F_{\forall} and l_{\exists} . Moreover, all $\diamond^+\forall$ -formulas are F_{\forall} and all \diamond -formulas are l_{\exists} .*

The results of Theorem 12 are tight in the sense that we cannot extend the grammar rule for \diamond^+ -formulas with $\forall x\varphi$; and we cannot extend the grammar rule for $\diamond^+\forall$ -formulas with $\diamond\varphi$. Simple counterexamples are $\forall x\diamond^+P(x)$ and $\diamond T$, which are not l_{\exists} and F_{\forall} , respectively. To see that $\forall x\diamond^+P(x)$ is not l_{\exists} consider the model with an infinite (and countable) domain, where one element is in P exactly at time point $n \in \mathbb{N}$, another one exactly at time point $n + 1$ and so on. There is no finite prefix where $\forall x\diamond^+P(x)$ holds.

\square^+ -formulas φ, ψ are built according to (with $P \in \mathbf{N}_P$):

$$\square^+ \varphi \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \forall x \varphi \mid P(\vec{\tau}) \mid \neg P(\vec{\tau})$$

We call \square -formulas the set of formulas generated by the result of further allowing $\square\varphi$ in the grammar rule for \square^+ -formulas; and call $\square^+\exists$ -formulas the result of allowing $\exists x\varphi$ in the grammar rule for \square^+ -formulas.

Theorem 13. *All \square^+ -formulas are F_{\exists} and I_{\forall} . Moreover, all $\square^+\exists$ -formulas are F_{\exists} and all \square -formulas are I_{\forall} .*

The results of Theorem 13 are also tight in the sense that we cannot extend the grammar rule for \square^+ -formulas with $\exists x\varphi$; and we cannot extend the grammar rule for $\square^+\exists$ -formulas with $\square\varphi$. Simple counterexamples are $\exists x\square^+\neg P(x)$ and $\square\perp$, which are not I_{\forall} and F_{\exists} , respectively. To see that $\exists x\square^+\neg P(x)$ is not I_{\forall} , consider again the model described above with an infinite (and countable) domain, where each element is in P at a specific time point $n \in \mathbb{N}$. The formula $\exists x\square^+\neg P(x)$ holds in every finite prefix but it does not hold in this infinite trace.

It follows from our results that there is no distinction between reasoning on finite and infinite traces whenever a formula is either a \diamond^+ - or a \square^+ -formula. As already pointed out, $\diamond^+\square^+\varphi$ and $\square^+\diamond^+\varphi$ are only equivalent on finite traces, and so, the distinction between reasoning on finite and infinite traces cannot be blurred for the class of formulas that allow both \diamond^+ and \square^+ .

5 Applications

The problem considered so far, to determine when the differences between finite and infinite traces can be safely blurred, is of interest for several applications. Here we show how it can be related to planning and verification. Moreover, for knowledge representation scenarios, we introduce temporal DLs on finite traces, providing complexity results for the satisfiability problem.

Planning. In automated planning, the sequence of states generated by actions is usually finite [Cerrito and Mayer, 1998; Bauer and Haslum, 2010; De Giacomo and Vardi, 2013; De Giacomo *et al.*, 2014b]. To reuse temporal logics based on infinite traces for specifying plan constraints, one approach, developed by De Giacomo *et al.* 2014b for *LTL*, is based on the notion of *insensitivity to infiniteness*. This property is meant to capture those formulas that can be equivalently interpreted on infinite traces, provided that, from a certain instant, these traces satisfy an end event forever and falsify all other atomic propositions. The motivation for this comes from the fact that propositional letters represents atomic tasks/actions that cannot be performed anymore after the end of a process.

In order to lift this notion of insensitivity to our first-order temporal setting, and to provide a characterisation analogous to the propositional one, we introduce the following definitions. Let $\mathfrak{F} = (\Delta^{\mathfrak{F}}, (\mathcal{F}_n)_{n \in [0, l]})$ be a finite trace, and let $\mathfrak{E} = (\Delta^{\mathfrak{E}}, (\mathcal{E}_n)_{n \in [0, \infty)})$ be the infinite trace such that $\Delta^{\mathfrak{E}} = \Delta^{\mathfrak{F}}$ (we write just Δ), $a^{\mathfrak{E}} = a^{\mathfrak{F}}$ for all $a \in \mathbf{N}_I$, and for all $P \in \mathbf{N}_P \setminus \{E\}$, $P^{\mathfrak{E}^n} = \emptyset$, while $E^{\mathfrak{E}^n} = \Delta$, where $n \in [0, \infty)$. The end extension (see Section 2) of \mathfrak{F} with \mathfrak{E} , $\mathfrak{F} \cdot_E \mathfrak{E}$, will be called the *insensitive extension of \mathfrak{F}* . A $T_{\mathcal{U}}\mathcal{QL}$ formula φ is *insensitive to infiniteness* (or simply *insensitive*) if, for every

finite trace \mathfrak{F} and all assignments \mathfrak{a} , $\mathfrak{F} \models^{\mathfrak{a}} \varphi$ iff $\mathfrak{F} \cdot_E \mathfrak{E} \models^{\mathfrak{a}} \varphi$. Clearly, all insensitive $T_{\mathcal{U}}\mathcal{QL}$ formulas are also $F_{\Rightarrow \exists}$.

Now, let Σ be a finite subset of \mathbf{N}_P . Assume w.l.o.g. that the $T_{\mathcal{U}}\mathcal{QL}$ formulas we mention in this subsection have predicates in Σ , and that Σ contains the end of time predicate E . Recalling the definition of ψ_f , we define $\chi_f = \psi_f \wedge \chi_{f_1}$, with

$$\chi_{f_1} = \square \forall x \forall \bar{y} (E(x) \rightarrow \bigwedge_{P \in \Sigma \setminus \{E\}} \neg P(x, \bar{y})).$$

The next characterisation result extends Theorem 4 in [De Giacomo *et al.*, 2014b] to $T_{\mathcal{U}}\mathcal{QL}$.

Theorem 14. *A $T_{\mathcal{U}}\mathcal{QL}$ formula φ is insensitive to infiniteness iff $\chi_f \models \varphi \leftrightarrow \varphi^{\dagger}$ holds on infinite traces.*

Insensitive formulas allow us to blur the distinction between finite and infinite traces as soon as infinite traces satisfy χ_f . Thus, we can check satisfiability of insensitive $T_{\mathcal{U}}\mathcal{QL}_{\square}^2$ (or other decidable languages) formulas on finite traces by using satisfiability algorithms for the infinite case without the need to apply the \cdot^{\dagger} translation.

We now analyse some syntactic features of this property. Firstly, non-temporal $T_{\mathcal{U}}\mathcal{QL}$ formulas, are insensitive. Moreover, this property is preserved under non-temporal operators. We generalise Theorem 5 in [De Giacomo *et al.*, 2014b] in our setting as follows.

Theorem 15. *Let φ, ψ be insensitive $T_{\mathcal{U}}\mathcal{QL}$ formulas. Then $\neg\varphi$, $\exists x\varphi$, and $\varphi \wedge \psi$ are insensitive.*

Concerning temporal operators, in [De Giacomo *et al.*, 2014b] it is shown how several standard temporal patterns derived from the declarative process modelling language *DECLARE* [van der Aalst and Pesic, 2006] are insensitive. On the other hand, negation affects the insensitivity of temporal formulas. For instance, given a $T_{\mathcal{U}}\mathcal{QL}$ formula $P(x)$, we have that $\diamond^+P(x)$ is insensitive while $\diamond^+\neg P(x)$ is not. Dually, $\square^+\neg P(x)$ is insensitive, while $\square^+P(x)$ is not. Therefore, if a $T_{\mathcal{U}}\mathcal{QL}$ formula φ is insensitive, it cannot be concluded that formulas of the form $\diamond^+\varphi$ or $\square^+\varphi$ are insensitive.

Finally, as a consequence of Theorem 14, we show that insensitivity is sufficient to ensure that if formulas are equivalent on infinite traces then they are equivalent on finite traces.

Theorem 16. *Let φ and ψ be insensitive $T_{\mathcal{U}}\mathcal{QL}$ formulas. Then $\varphi \equiv_{\exists} \psi$ implies $\varphi \equiv_{\mathfrak{F}} \psi$.*

However, the viceversa does not hold. Consider, e.g., Formula (*), which is trivially insensitive: this formula is equivalent to \perp only on finite traces. We can obtain the converse direction using Theorem 10. For instance, $\diamond^+P(x) \vee \diamond^+R(x)$ and $\diamond^+(P(x) \vee R(x))$ are insensitive and I_{\exists} formulas for which equivalence on finite and infinite traces coincides.

Verification. In this section we show how our comparison between finite and infinite traces can be related to the literature on temporal logics for verification. We point out some connections between the finite/infinite trace properties and: (i) the definition of safety in *LTL* on infinite traces; (ii) some notions related to the construction of monitoring functions in runtime verification.

(i) *Safety.* Recall that a safety property intuitively asserts that *bad things* never happen during the execution of a program. In verification, *LTL* is often used as a specification

language for such properties, and the notion of safety is defined accordingly on infinite traces [Sistla, 1994]. Let φ be a $T_U\mathcal{QL}$ formula asserting that some “bad thing” never happens. According to the literature, we say that φ denotes a safety property if, whenever φ does not hold for an infinite run of a program, then it must be violated already after a finite execution. That is, the infinite trace contains a finite prefix in which the bad thing (the violation of φ) has already happened. More formally, we say that a $T_U\mathcal{QL}$ formula φ *expresses a safety property* iff, for an infinite trace \mathcal{J} and an assignment α :

$$(\forall \mathcal{F} \in \text{Pre}(\mathcal{J}). \mathcal{F} \models^\alpha \varphi) \Rightarrow \mathcal{J} \models^\alpha \varphi.$$

In other words, safety is captured in $T_U\mathcal{QL}$ by $\mathbb{L}_{\Leftarrow\forall}$ formulas. In particular, all \square^+ -formulas of Section 4 are \mathbb{L}_{\forall} and thus they express safety properties.

(ii) *Runtime verification.* We recall that in runtime verification the task is to evaluate a property with respect to the current history (which is finite at each given instant) of a dynamic system, and to check whether this property is satisfied in all its possible future evolutions [Bauer *et al.*, 2010; Baader and Lippmann, 2014; De Giacomo *et al.*, 2014a]. Here we discuss the relationship between our semantic conditions and the *maxims* for runtime verification introduced by Bauer *et al.* 2010, which relate finite trace semantics to the infinite case. The authors suggest that every semantics to be used in runtime verification should satisfy the following requirements.

- *Impartiality:* never evaluate to true or false a formula on a finite trace, if one of its infinite extensions can possibly change its value.
- *Anticipation:* if a formula has the same truth value on every infinite extension of a finite trace, then it is equally evaluated also on that finite prefix.

Impartiality cannot be guaranteed for $T_U\mathcal{QL}$: $\square^+P(x)$ is an example of a formula violating this maxim. On the other hand, $\diamond\top$ is a formula that violates anticipation. Impartiality, as formalised in [Bauer *et al.*, 2010], is captured by $\mathbb{F}_{\Rightarrow\forall}$ and $\mathbb{F}_{\Leftarrow\exists}$ properties. Instead, the formal version of anticipation corresponds to properties $\mathbb{F}_{\Leftarrow\forall}$ and $\mathbb{F}_{\Rightarrow\exists}$ in our setting. Therefore, any set of $T_U\mathcal{QL}$ formulas satisfying both impartiality and anticipation should belong to the intersection of \mathbb{F}_{\forall} and \mathbb{F}_{\exists} formulas. Concerning the possibility to syntactically characterise these formulas, we have that, due to Theorems 12 and 13, impartiality and anticipation are not guaranteed to be preserved for \diamond^+ - or \square^+ -formulas.

Temporal Description Logics. We now consider temporal description logics. We define the temporal language $T_U\mathcal{ALC}$ as a temporal extension of the description logic \mathcal{ALC} [Gabbay *et al.*, 2003]. Let $\mathbb{N}_C, \mathbb{N}_R \subseteq \mathbb{N}_P$ be, respectively, sets of unary and binary predicates called *concept* and *role names*. A $T_U\mathcal{ALC}$ concept is an expression of the form:

$$C, D ::= A \mid \neg C \mid C \sqcap D \mid \exists R.C \mid C \cup D,$$

where $A \in \mathbb{N}_C$ and $R \in \mathbb{N}_R$. A $T_U\mathcal{ALC}$ axiom is either a *concept inclusion* (CI) of the form $C \sqsubseteq D$, or an *assertion*, α , of the form $A(a)$ or $R(a, b)$, where C, D are $T_U\mathcal{ALC}$ concepts, $A \in \mathbb{N}_C$, $R \in \mathbb{N}_R$, and $a, b \in \mathbb{N}_I$. $T_U\mathcal{ALC}$ formulas have the form:

$$\varphi, \psi ::= \alpha \mid C \sqsubseteq D \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \mathcal{U} \psi.$$

Since a $T_U\mathcal{ALC}$ formula can be mapped into an equisatisfiable $T_U\mathcal{QL}_{\square}^2$ formula [Gabbay *et al.*, 2003] we can transfer the upper bounds of Theorems 4 and 5 to $T_U\mathcal{ALC}$ and $T_U\mathcal{ACC}$; the lower bounds can be obtained in a similar way as in the mentioned theorems. Moreover, from Theorems 6 and 7, we obtain immediately that $T_U\mathcal{ALC}$ on finite traces has both the bounded trace and domain properties.

We now consider the $T_U\mathcal{ALC}$ satisfiability problem on k -bounded traces *restricted to global CIs*, defined as the fragment of $T_U\mathcal{ALC}$ in which formulas can only be of the form $\mathcal{T} \wedge \square(\mathcal{T}) \wedge \phi$, where \mathcal{T} is a conjunction of CIs and ϕ does not contain CIs. The EXPTIME upper bound we provide has a rather challenging proof that uses a form of *type elimination* [Gabbay *et al.*, 2003; Lutz *et al.*, 2008; Gutiérrez-Basulto *et al.*, 2016], but in a setting where the number of time points is bounded by a natural number $k > 0$. The main challenge in solving this problem when the number of time points is arbitrarily large but finite is mainly due to the presence of *last* sub-formulas (i.e., formulas of the form $\neg\bigcirc\top$) that can hold just in the last instant of the model. The complexity is tight since satisfiability in \mathcal{ALC} is already EXPTIME-hard [Gabbay *et al.*, 2003].

Theorem 17. *$T_U\mathcal{ALC}$ satisfiability is EXPSPACE-complete on finite traces, and NEXPTIME-complete on k -bounded traces. Moreover, $T_U\mathcal{ALC}$ satisfiability on k -bounded traces restricted to global CIs is EXPTIME-complete.*

6 Conclusion

We investigated first-order temporal logic over finite traces, studying satisfiability of its two-variable monodic fragment, $T_U\mathcal{QL}_{\square}^2$. While being EXPSPACE-complete over arbitrary finite traces, it lowers down to NEXPTIME in case of $T_U\mathcal{QL}_{\square}^2$, interpreted over traces with at most k time points. Similar results have been shown for a temporal extension of the description logic \mathcal{ALC} , with $T_U\mathcal{ALC}$ restricted to global CIs being EXPTIME-complete. Moreover, in an effort to systematically clarify the correlations between finite vs. infinite reasoning we introduced various semantic conditions that allow to formally specify when it is possible to blur the distinction between finite and infinite traces. Grammars for $T_U\mathcal{QL}$ formulas satisfying some of these conditions have been provided as well. In particular, we have shown that for \diamond^+ - and \square^+ -formulas, equivalence over finite and infinite traces coincide. Some notions used in planning (particularly, insensitivity to infiniteness [De Giacomo *et al.*, 2014b]) and verification have been lifted to the first-order setting, and related to our semantic conditions for blurring the distinction between reasoning over finite and infinite traces.

As future work, we plan to apply the semantic conditions to study the specific case where infinite extensions of finite traces are obtained by repeating the last instant forever [Bauer and Haslum, 2010], as well as to the analysis of monitoring functions for runtime verification [Bauer *et al.*, 2010; Baader and Lippmann, 2014; De Giacomo *et al.*, 2014a]. It would also be interesting to study the precise complexity of the satisfiability problem for $T_U\mathcal{ALC}$ with just global CIs.

References

- [van der Aalst and Pesic, 2006] Wil van der Aalst and Maja Pesic. Decserflow: Towards a truly declarative service flow language. In *WS-FM*, pages 1–23, 2006.
- [van der Aalst *et al.*, 2017] Wil van der Aalst, Alessandro Artale, Marco Montali, and Simone Tritini. Object-centric behavioral constraints: Integrating data and declarative process modelling. In *DL*, volume 1879, 2017.
- [Artale and Franconi, 2005] Alessandro Artale and Enrico Franconi. Temporal description logics. In *Handbook of Temporal Reasoning in Artificial Intelligence*, pages 375–388. Elsevier, 2005.
- [Artale *et al.*, 2014] Alessandro Artale, Roman Kontchakov, Vladislav Ryzhikov, and Michael Zakharyashev. A cookbook for temporal conceptual data modelling with description logics. *ACM Trans. Comput. Log.*, 15(3):25:1–25:50, 2014.
- [Artale *et al.*, 2019] Alessandro Artale, Andrea Mazzullo, and Ana Ozaki. Do you need infinite time? Technical Report 01, Free University of Bozen-Bolzano, 2019. Available at <https://www.inf.unibz.it/krdp/KRDB%20files/tech-reports/KRDB19-01.pdf>.
- [Baader and Lippmann, 2014] Franz Baader and Marcel Lippmann. Runtime verification using the temporal description logic \mathcal{ALC} -LTL revisited. *J. Applied Logic*, 12(4):584–613, 2014.
- [Baader *et al.*, 2012] Franz Baader, Silvio Ghilardi, and Carsten Lutz. LTL over description logic axioms. *ACM Trans. Comput. Log.*, 13(3), 2012.
- [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 and McIlraith, 2006] Jorge A. Baier and Sheila A. McIlraith. Planning with first-order temporally extended goals using heuristic search. In *AAAI*, pages 788–795, 2006.
- [Bauer and Haslum, 2010] Andreas Bauer and Patrik Haslum. LTL goal specifications revisited. In *ECAI*, pages 881–886, 2010.
- [Bauer *et al.*, 2010] Andreas Bauer, Martin Leucker, and Christian Schallhart. Comparing LTL semantics for runtime verification. *J. Log. Comput.*, 20(3):651–674, 2010.
- [Camacho *et al.*, 2017] Alberto Camacho, Eleni Triantafyllou, Christian J. Muise, Jorge A. Baier, and Sheila A. McIlraith. Non-deterministic planning with temporally extended goals: LTL over finite and infinite traces. In *AAAI*, pages 3716–3724, 2017.
- [Cerrito and Mayer, 1998] Serenella Cerrito and Marta Cialdea Mayer. Bounded model search in linear temporal logic and its application to planning. In *TABLEAUX*, pages 124–140, 1998.
- [Cerrito *et al.*, 1999] Serenella Cerrito, Marta Cialdea Mayer, and Sébastien Praud. First order linear temporal logic over finite time structures. In *LPAR*, pages 62–76, 1999.
- [De Giacomo and Vardi, 2013] Giuseppe De Giacomo and Moshe Y. Vardi. Linear temporal logic and linear dynamic logic on finite traces. In *IJCAI*, pages 854–860, 2013.
- [De Giacomo *et al.*, 2014a] Giuseppe De Giacomo, Riccardo De Masellis, Marco Grasso, Fabrizio Maria Maggi, and Marco Montali. Monitoring business metaconstraints based on LTL and LDL for finite traces. In *BPM*, pages 1–17, 2014.
- [De Giacomo *et al.*, 2014b] Giuseppe De Giacomo, Riccardo De Masellis, and Marco Montali. Reasoning on LTL on finite traces: Insensitivity to infiniteness. In *AAAI*, pages 1027–1033, 2014.
- [Fionda and Greco, 2016] Valeria Fionda and Gianluigi Greco. The complexity of LTL on finite traces: Hard and easy fragments. In *AAAI*, pages 971–977, 2016.
- [Gabbay *et al.*, 2003] Dov M. Gabbay, Agi Kurucz, Frank Wolter, and Michael Zakharyashev. *Many-dimensional Modal Logics: Theory and Applications*, volume 148. Elsevier, 2003.
- [Giannakopoulou and Havelund, 2001] Dimitra Giannakopoulou and Klaus Havelund. Automata-based verification of temporal properties on running programs. In *ASE*, pages 412–416, 2001.
- [Gutiérrez-Basulto *et al.*, 2016] Víctor Gutiérrez-Basulto, Jean Christoph Jung, and Ana Ozaki. On metric temporal description logics. In *ECAI*, pages 837–845, 2016.
- [Hodkinson *et al.*, 2000] Ian M. Hodkinson, Frank Wolter, and Michael Zakharyashev. Decidable fragment of first-order temporal logics. *Annals of Pure and Applied Logic*, 106(1-3):85–134, 2000.
- [Hodkinson *et al.*, 2003] Ian M. Hodkinson, Roman Kontchakov, Agi Kurucz, Frank Wolter, and Michael Zakharyashev. On the computational complexity of decidable fragments of first-order linear temporal logics. In *TIME-ICTL*, pages 91–98, 2003.
- [Li *et al.*, 2014] Jianwen Li, Lijun Zhang, Geguang Pu, Moshe Y. Vardi, and Jifeng He. LTL^f Satisfiability Checking. In *ECAI*, pages 513–518, 2014.
- [Lutz *et al.*, 2008] Carsten Lutz, Frank Wolter, and Michael Zakharyashev. Temporal description logics: A survey. In *TIME*, pages 3–14, 2008.
- [Maggi *et al.*, 2011] Fabrizio Maria Maggi, Marco Montali, Michael Westergaard, and Wil M. P. van der Aalst. Monitoring business constraints with linear temporal logic: An approach based on colored automata. In *BPM*, pages 132–147, 2011.
- [Manna and Pnueli, 1995] Zohar Manna and Amir Pnueli. *Temporal verification of reactive systems - Safety*. Springer, 1995.
- [Sistla, 1994] A. Prasad Sistla. Safety, liveness and fairness in temporal logic. *Formal Asp. Comput.*, 6(5):495–512, 1994.
- [Wolter and Zakharyashev, 1998] Frank Wolter and Michael Zakharyashev. Temporalizing description logics. In *FroCoS*, pages 104–109, 1998.