



First-Order Temporal Logic on Finite Traces: Semantic Properties, Decidable Fragments, and Applications

ALESSANDRO ARTALE, Faculty of Engineering, Free University of Bozen-Bolzano, Bolzano, Italy

ANDREA MAZZULLO, Faculty of Engineering, Free University of Bozen-Bolzano, Bolzano, Italy and Department of Information Engineering and Computer Science, University of Trento, Trento, Italy

ANA OZAKI, Department of Informatics, University of Bergen, Bergen, Norway and Department of Informatics, University of Oslo, Oslo, Norway

Formalisms based on temporal logics interpreted over finite strict linear orders, known in the literature as *finite traces*, have been used for temporal specification in automated planning, process modelling, (runtime) verification and synthesis of programs, as well as in knowledge representation and reasoning. In this article, we focus on *first-order temporal logic on finite traces*. We first investigate preservation of equivalences and satisfiability of formulas between finite and infinite traces, by providing a set of semantic and syntactic conditions to guarantee when the distinction between reasoning in the two cases can be blurred. Moreover, we show that the satisfiability problem on finite traces for several decidable fragments of first-order temporal logic is EXPSpace-complete, as in the infinite trace case, while it decreases to NEXPTIME when finite traces bounded in the number of instants are considered. This leads also to new complexity results for temporal description logics over finite traces. Finally, we investigate applications to planning and verification, in particular by establishing connections with the notions of insensitivity to infiniteness and safety from the literature.

CCS Concepts: • **Theory of computation** → **Modal and temporal logics**; • **Computing methodologies** → **Temporal reasoning**;

Additional Key Words and Phrases: First-order temporal logics, temporal description logics, finite traces, automated reasoning

ACM Reference Format:

Alessandro Artale, Andrea Mazzullo, and Ana Ozaki. 2024. First-Order Temporal Logic on Finite Traces: Semantic Properties, Decidable Fragments, and Applications. *ACM Trans. Comput. Logic* 25, 2, Article 13 (April 2024), 43 pages. <https://doi.org/10.1145/3651161>

1 INTRODUCTION

The study of formalisms based on propositional or first-order temporal logics on linear flows of time has found a wide spectrum of applications, ranging from verification of programs and model

Mazzullo acknowledges the support of the MUR PNRR project FAIR - Future AI Research (PE00000013) funded by the NextGenerationEU. Ozaki is supported by the Research Council of Norway, project number 316022.

Authors' addresses: A. Artale, Faculty of Engineering, Free University of Bozen-Bolzano, Bolzano, Italy and Piazza Domenicani, 3, Bolzano 39100, Italy; e-mail: artale@inf.unibz.it; A. Mazzullo, Faculty of Engineering, Free University of Bozen-Bolzano, Bolzano, Italy and Department of Information Engineering and Computer Science, University of Trento, Trento, Italy; Via Sommarive, 9, 38123 Povo TN; e-mail: andrea.mazzullo@unitn.it; A. Ozaki, Department of Informatics, University of Bergen, Bergen, Norway and Department of Informatics, University of Oslo, Oslo, Norway; Gaustadalléen 23B, Oslo, 0373, Norway; e-mail: anaoz@uio.no.



This work is licensed under a [Creative Commons Attribution International 4.0 License](https://creativecommons.org/licenses/by/4.0/).

© 2024 Copyright held by the owner/author(s).

ACM 1529-3785/2024/04-ART13

<https://doi.org/10.1145/3651161>

checking [24, 87, 91], to automated planning [22, 23, 25], process modelling [2, 86], and knowledge representation. In the latter context, several decidable fragments of first-order temporal logic with the linear time operator *until* (\mathcal{U}), denoted $T_{\mathcal{U}}\mathcal{QL}$, have been investigated [54, 75, 76]. *Temporal description logics* (see [4, 8, 20, 84, 98] and references therein), obtained by suitably combining (linear) temporal logic operators with *description logics* (DLs) constructs, are well-known examples of such fragments. These logics usually lie within the *two-variable monodic* fragment of $T_{\mathcal{U}}\mathcal{QL}$, denoted as $T_{\mathcal{U}}\mathcal{QL}_{\square}^2$ obtained by restricting the language to formulas having at most two variables, and so that the temporal operators are applied only to subformulas with at most one free variable. For instance, by using the reflexive temporal operators \diamond^+ , meaning *sometimes in the future*, and \square^+ , meaning *always in the future*, the following formula

$$\forall x(\text{Reviewer}(x) \rightarrow \diamond^+ \square^+ \forall y(\text{Submission}(y) \wedge \text{reviews}(x, y) \rightarrow \diamond^+ \text{Evaluated}(y))), \quad (1)$$

is a $T_{\mathcal{U}}\mathcal{QL}_{\square}^2$ formula stating that every reviewer will reach a (present or future) moment after which all the submissions they review will be eventually evaluated. Other decidable languages considered in the literature are the *monodic monadic* fragment $T_{\mathcal{U}}\mathcal{QL}_{\square}^{mo}$ and the *one-variable* fragment $T_{\mathcal{U}}\mathcal{QL}^1$, with monodic formulas having, respectively, at most unary predicates and at most one variable. The complexity of the satisfiability problem ranges from EXPSpace-complete, for $T_{\mathcal{U}}\mathcal{QL}_{\square}^2$, $T_{\mathcal{U}}\mathcal{QL}_{\square}^{mo}$ and $T_{\mathcal{U}}\mathcal{QL}^1$ [54, 75], down to NEXPTIME- or EXPTIME-complete, for temporal extensions of the DL \mathcal{ALC} without temporalised roles and with restrictions on the application of temporal operators [19, 84]. Recent work on temporal extensions of lightweight DLs in the *DL-Lite* and \mathcal{EL} families [9, 71], used in conceptual data modelling and underlying prominent profiles of the OWL standard, shows that the complexity of reasoning can be even lowered down to NP or NLOGSPACE.

A widely studied semantics for temporal logics is defined on structures based on the strict linear order of the natural numbers [53, 64, 91]. However, linear temporal structures with only a finite number of time points, often called *finite traces*, have been investigated as well [44, 53], receiving a renewed interest in the literature [48, 51, 52]. The finiteness of the time dimension represents indeed a fairly natural restriction for several applications. 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 [28, 41, 46, 47]. In runtime verification only the current finite behaviour of the system is taken into account, while infinite models are considered when checking whether a given requirement is satisfied in some/all infinite extensions of the finite trace [29, 62]. These needs from critical applications of temporal logics can be reflected by a semantics based on finite traces, with formulas having different satisfaction conditions compared to the infinite case. For instance, by using the formula *last* to refer to the last time point of a finite trace, we have that Formula (1) above is equivalent on finite traces to

$$\forall x(\text{Reviewer}(x) \rightarrow \diamond^+ (\text{last} \wedge \forall y(\text{Submission}(y) \wedge \text{reviews}(x, y) \rightarrow \text{Evaluated}(y)))), \quad (2)$$

stating that every reviewer will eventually reach a “deadline”, represented by the *last* formula, when all the submission they review are evaluated. This follows from the fact that, on finite traces, formulas $\diamond^+ \square^+ \varphi$ and $\diamond^+ (\text{last} \wedge \varphi)$ are equivalent [48], and $\text{last} \wedge \forall x(\varphi \rightarrow \diamond^+ \psi)$ is in turn equivalent to $\text{last} \wedge \forall x(\varphi \rightarrow \psi)$.

This work focuses on **first-order temporal logic on finite traces**, defined by extending the first-order language with linear time operators interpreted on finite traces. Part of the results contained in the current article, establishing bridges between finite and infinite traces semantics, has been presented in [11, 12, 14]. We highlight the main differences in the related work section (Section 2.3). We comment on the main results presented in this article while illustrating its

structure. Section 2 is devoted to a discussion of related work in this area. Then, after introducing in Section 3 the required preliminary notions about first-order (linear) temporal logic, we provide the following main contributions. In Section 4, we focus on bridging reasoning on finite and infinite traces semantics. In general, indeed, the sets of formulas equivalent on finite and on infinite traces do not coincide, as witnessed by the following examples: the formula $\diamond^+\square\perp$, forcing the existence of a (present or future) last instant of time, is equivalent to \top only on finite traces, whereas $\square^+\diamond\top$, stating that there is always a future instant after the current one, is equivalent to \top on infinite, but not on finite, traces. We establish here semantic and syntactic conditions to guarantee that two formulas equivalent on finite (respectively, infinite) traces are also equivalent on infinite (respectively, finite) traces. We also devise a semantic criterion and a syntactically defined class of formulas to guarantee preservation of satisfiability from finite to infinite traces.

In Section 5, we study the complexity of reasoning over finite traces for the *two-variable monodic* fragment $T_{\mathcal{U}}QL_{\square}^2$, the *monadic* fragment $T_{\mathcal{U}}QL^{mo}$ and the *one-variable* fragment $T_{\mathcal{U}}QL^1$, showing that the complexity remains EXPSPACE-complete on finite traces, but lowers down to NEXPTIME if we restrict to traces with a bound k (given in binary) on the number of instants. Moreover, we show that these fragments enjoy two kinds of *bounded model properties*: the *bounded trace property*, which limits the finite traces satisfying a formula to be at most double exponential in the size of the input formula, and the *bounded domain property*, which limits the number of domain elements in a k -bounded trace (finite trace with at most k instants) to be at most double exponential both in the size of the input formula and in the (binary representation) of k . In the context of temporal DLs, we show that complexity results similar to the infinite trace case, as well as bounded model properties, also hold for the temporal DL $T_{\mathcal{U}}\mathcal{ALC}$ when interpreted on finite traces. We further show a more challenging result, i.e., that the complexity for $T_{\mathcal{U}}\mathcal{ALC}$ further reduces to EXPTIME if only global concept inclusions (also known as *terminological axioms*, or *TBox axioms*) together with a temporal dataset (also known as *assertional axioms*, or *ABox axioms*) are interpreted on k -bounded traces.

Finally, in Section 6, we investigate connections with the planning and verification literature. Concerning the former scenario, we study in our setting the notion of *insensitivity to infiniteness* [47], a property applying to formulas that, once satisfiable on finite traces, remain satisfiable on infinite traces verifying an *end event* forever and falsifying all other atomic formulas. Concerning the verification aspect, we establish connections between the finite and infinite trace characterisations, as introduced in Section 4, and the notions of *safety* [24, 95], as well as other related notions from the literature on runtime verification [29]. Section 7 concludes the article.

2 RELATED WORK

In this section, we first highlight related works on finite traces and on temporal DLs, which can be seen as decidable fragments of first-order temporal logic. We then discuss in more detail the differences between our previous results presented in [11, 12, 14] and the current journal version.

2.1 Finite Traces in Temporal Logics

Finite traces [44, 53, 92] have regained momentum in formalisms for AI applications. Together with (propositional) *linear temporal logic (LTL)* [91], also the more expressive *linear dynamic logic* [74], *alternating time logic* [3], and *mu-calculus* [26, 80] have been investigated on semantics based on finite traces [32, 33, 59, 63, 83]. To deal with uncertainty in dynamic systems, a *probabilistic* version of LTL over finite traces has been proposed as well [85], while a recent article

addresses problems in declarative process mining by introducing *metric temporal logic* on finite traces [58]. Significant areas of applications for LTL on finite traces are indeed in the planning domain [25, 38, 41–43, 47, 57], in (declarative) business process modelling, as well as in runtime verification and monitoring [27, 29, 46, 60, 92]. In addition, LTL on finite traces has found applications in the context of synthesis [39, 40, 49, 60, 61, 101], multi-agent systems [65, 66, 78], temporal databases [93], and answer-set programming [35–37]. The problem of establishing connections between finite and infinite traces semantics is also not new to the literature. Several approaches have been proposed to show when satisfiability of formulas is preserved from the finite to the infinite case, so to reuse on infinite traces algorithms developed for the finite case [52, 82, 94]. In this work, we determine conditions that preserve satisfiability in the other direction as well, from infinite to finite traces, in the direction of applying efficient infinite traces reasoners to the finite case [28, 47].

2.2 Fragments of First-order Temporal Logics and Temporal DLs

As already mentioned, a significant body of research has been devoted to the study of fragments of first-order temporal logics, both on linear time [54, 73, 75, 76] and on branching time structures [67, 68, 70, 77], as well as similar fragments of first-order modal, epistemic or dynamic logics [54, 72, 96, 99], or first-order combinations involving both temporal and epistemic modalities [30, 31]. Given the connections of temporal DLs with first-order temporal logic (over linear time flows) and their relevance to the present article, we separately discuss related work on temporal DLs. For a general overview, we refer to the already mentioned surveys [4, 8, 20, 84, 98]. In the linear time case, a wide body of research has focused on temporal DLs with semantics based on the natural numbers [19, 89, 90] or the integers [9], possibly by extending the language with metric temporal operators as well [16, 17, 69]. In applications, temporalised DLs have been considered in the context of runtime verification [15, 21] and business process modelling [1, 10]. However, such proposals are based on the usual infinite trace semantics or are limited in expressivity. To the best of our knowledge, little has been done in order to combine finite traces and temporal DLs. Recent work in this direction can be found in [11, 12]. The complexity landscape of temporal DLs on finite traces semantics has been further enriched by preliminary results on temporal *DL-Lite* logics [13]. These results, which match the corresponding ones on infinite traces semantics [9], are obtained by considering axioms interpreted either globally or locally, and by syntactically restricting the application of the temporal operators (allowing for \mathcal{U} , or only for \square and \circ), or of the DL constructors (focussing on the so-called *bool*, *horn*, *krom*, and *core* fragments).

2.3 Comparison with Our Previous Works

This article extends our previous work presented in [11, 12, 14] by including full proofs, more detailed discussions, and new results. The material from Section 4.1 appeared, without proof details, in [12]. In Section 4.2, which investigates the distinction between finite and infinite traces, we include results for the “one directional” version of the properties introduced in [12] for the semantic characterisation. The syntactic characterisation and the preservation of formula satisfiability parts, still in Section 4.2, extends our previous work [12] with new results, by including formulas built with the until and the release operators. The main result in Section 4.2.3 appeared in [14]. In Section 5, we provide an extended and unified presentation of complexity results both for decidable fragments of first-order temporal logic [14] and for the classical description logic \mathcal{ALC} [11, 12]. In particular, we cover fragments of first-order temporal logic not considered in our previous works, namely, $T_{\mathcal{U}}\mathcal{QL}_{\dot{q}}^{1,mo}$, $T_{\mathcal{U}}\mathcal{QL}^1$, and $T_{\mathcal{U}}\mathcal{QL}_{\square}^{mo}$. Finally, Section 6 is considerably more detailed than the corresponding “Applications” Section in our previous work [14], and all the proofs in Section 6.2 are new.

3 FIRST-ORDER TEMPORAL LOGICS

The *first-order temporal language* $T_{\mathcal{U}}\mathcal{QL}$ [54], which we present in the following, is obtained by extending the usual first-order language with the temporal operator *until* \mathcal{U} interpreted over linear structures, called *traces*.

3.1 Syntax

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

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

where $P \in N_P$, $\bar{\tau} = (\tau_1, \dots, \tau_{\text{ar}(P)})$ is a tuple of *terms*, i.e., constants or variables, and $x \in \text{Var}$. 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 exactly x_1, \dots, x_m . We write $\varphi\{x/y\}$ for the result of uniformly substituting the free occurrences of y in φ by x . 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. The (propositional) language *LTL* is $T_{\mathcal{U}}\mathcal{QL}^0$ with formulas constructed without the existential quantifier. The *monodic fragment* of $T_{\mathcal{U}}\mathcal{QL}$, denoted by $T_{\mathcal{U}}\mathcal{QL}_{\square}^{\text{mon}}$ consists of formulas such that all subformulas of the form $\varphi \mathcal{U} \psi$ have *at most one* free variable. The *monodic fragment* $T_{\mathcal{U}}\mathcal{QL}^{\text{mo}}$ is the fragment of $T_{\mathcal{U}}\mathcal{QL}$ with formulas not containing predicates of arity ≥ 2 . Finally, the *constant-free one-variable monodic fragment*, $T_{\mathcal{U}}\mathcal{QL}_{\square}^{1,\text{mo}}$, is obtained from the one-variable monodic fragment by disallowing constants.

3.2 Semantics

A *first-order temporal interpretation* (or *trace*) is a pair $\mathfrak{M} = (\Delta^{\mathfrak{M}}, (\mathcal{M}_n)_{n \in \mathfrak{T}})$, where \mathfrak{T} is a sub-order of $(\mathbb{N}, <)$ of the form $[0, \infty)$ or $[0, l]$, with $l \in \mathbb{N}$, and each \mathcal{M}_n is a classical first-order interpretation with a non empty domain $\Delta^{\mathfrak{M}}$ (or simply Δ): we have $P^{\mathcal{M}_n} \subseteq \Delta^{\text{ar}(P)}$, for each $P \in N_P$, and $a^{\mathcal{M}_i} = a^{\mathcal{M}_j} \in \Delta$ for all $a \in N_I$ and $i, j \in \mathbb{N}$, i.e., constants are *rigid designators* (with fixed interpretation, denoted simply by $a^{\mathcal{M}}$). 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 [54]. An *assignment* in \mathfrak{M} (or simply an *assignment*, when \mathfrak{M} is clear from the context) is a function α from Var to Δ , and the *value of a term* τ in \mathfrak{M} under α is defined as: $\alpha(\tau) = \alpha(x)$, if $\tau = x$, and $\alpha(\tau) = a^{\mathcal{M}}$, if $\tau = a \in N_I$. Given a tuple of m terms $\bar{\tau} = (\tau_1, \dots, \tau_m)$, we set $\alpha(\bar{\tau}) = (\alpha(\tau_1), \dots, \alpha(\tau_m))$. Given a formula φ , the *satisfaction of φ in \mathfrak{M} at time point $n \in \mathfrak{T}$ under an assignment α* , written $\mathfrak{M}, n \models^{\alpha} \varphi$, is inductively defined as

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

We say that φ is *satisfied in \mathfrak{M} under α* , writing $\mathfrak{M} \models^{\alpha} \varphi$, if $\mathfrak{M}, 0 \models^{\alpha} \varphi$, and that φ is *satisfied in \mathfrak{M}* (or that \mathfrak{M} is a *model* of φ), denoted by $\mathfrak{M} \models \varphi$, if $\mathfrak{M} \models^{\alpha} \varphi$, for some α . Moreover, φ is said to be *satisfiable* if it is satisfied in some \mathfrak{M} . A formula φ *logically implies* a formula ψ if, for every interpretation \mathfrak{M} and every assignment α , $\mathfrak{M} \models^{\alpha} \varphi$ implies $\mathfrak{M} \models^{\alpha} \psi$, and we write $\varphi \models \psi$. We say that φ and ψ are *equivalent*, writing $\varphi \equiv \psi$, if $\varphi \models \psi$ and $\psi \models \varphi$. Since the satisfaction of a formula

$\varphi(x_1, \dots, x_n)$ under an assignment α depends only on the values of its free variables under α , we may write $\mathfrak{M}, n \models \varphi[d_1, \dots, d_n]$ in place of $\mathfrak{M}, n \models^{\alpha} \varphi(x_1, \dots, x_n)$, where $\alpha(x_1) = d_1, \dots, \alpha(x_n) = d_n$. Also, given an assignment α and an element d in an interpretation's domain Δ we denote by $\alpha[x \mapsto d]$ the assignment obtained by modifying α so that x maps to d .

In the following, we call *finite trace* a trace 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 $\mathfrak{S} = (\Delta^{\mathfrak{S}}, (\mathcal{I}_n)_{n \in [0, \infty)})$. We say that a $T_{\mathcal{U}}\mathcal{QL}$ formula φ is *satisfiable on infinite, finite, or k -bounded traces*, respectively, 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. Moreover, given $T_{\mathcal{U}}\mathcal{QL}$ formulas φ and ψ , we write $\varphi \models_i \psi$ (respectively, $\varphi \models_f \psi$) iff φ logically implies ψ on infinite (resp., finite) traces. Similarly, we write $\varphi \equiv_i \psi$ if φ and ψ are equivalent on infinite traces, and $\varphi \equiv_f \psi$ if they are equivalent on finite traces.

In addition to the standard conventions on parenthesis and Boolean equivalences, we will use the following abbreviations for formulas: $\perp := P \wedge \neg P$ (*bottom*, for an arbitrary but fixed 0-ary predicate P); $\top := \neg \perp$ (*top*); $\varphi \mathcal{U}^+ \psi := \psi \vee (\varphi \wedge \varphi \mathcal{U} \psi)$; $\varphi \mathcal{R} \psi := \neg(\neg \varphi \mathcal{U} \neg \psi)$ (*releases*); $\varphi \mathcal{R}^+ \psi := \psi \wedge (\varphi \vee \varphi \mathcal{R} \psi)$; $\diamond \varphi := \top \mathcal{U} \varphi$ (*diamond*); $\diamond^+ \varphi := \top \mathcal{U}^+ \varphi$; $\square \varphi := \perp \mathcal{R} \varphi$ (*box*); $\square^+ \varphi := \perp \mathcal{R}^+ \varphi$; $\circ \varphi := \perp \mathcal{U} \varphi$ (*strong next*); and $\bullet \varphi := \top \mathcal{R} \varphi$ (*weak next*). To refer to the last time point in a finite trace, we use the formula *last* := $\square \perp$, which is indeed satisfied at an instant of a trace iff that instant does not have a successor, i.e., it holds that *last* $\equiv_i \neg \circ \top$. Observe that *last* $\equiv_i \circ \perp \equiv_i \perp$, whereas *last* $\not\equiv_f \perp$. Moreover, we have that $\bullet \varphi \equiv_i \circ \varphi \equiv_i \neg \circ \neg \varphi$, while none of the previous equivalences hold on finite traces. Indeed, we have that $\bullet \varphi \equiv \text{last} \vee \circ \varphi$, meaning that $\bullet \varphi$ is satisfied at a time point of a trace iff either it is the last instant of the trace, or at the next time point φ holds.

We now introduce the notation used in the rest of the article. Given a trace $\mathfrak{M} = (\Delta^{\mathfrak{M}}, (\mathcal{M}_n)_{n \in \mathfrak{T}})$, the *suffix of \mathfrak{M} starting at $i \in \mathfrak{T}$* is the trace $\mathfrak{M}^i = (\Delta^{\mathfrak{M}}, (\mathcal{M}_n^i)_{n \in \mathfrak{T}'})$, where: if $\mathfrak{T} = [0, l]$, we set $\mathfrak{T}' = [0, l - i]$ and $\mathcal{M}_n^i = \mathcal{M}_{i+n}$, for every $n \in \mathfrak{T}'$; whereas, if $\mathfrak{T} = [0, \infty)$, we set $\mathfrak{T}' = [0, \infty)$ and $\mathcal{M}_n^i = \mathcal{M}_{i+n}$, for every $n \in \mathfrak{T}'$. The *prefix of \mathfrak{M} ending at $i \in \mathfrak{T}$* is the trace $\mathfrak{M}_i = (\Delta^{\mathfrak{M}}, (\mathcal{M}_n)_{n \in \mathfrak{T}'})$, where $\mathfrak{T}' = [0, i]$. Clearly, since $T_{\mathcal{U}}\mathcal{QL}$ does not contain past temporal operators, given a $T_{\mathcal{U}}\mathcal{QL}$ formula φ , a trace $\mathfrak{M} = (\Delta^{\mathfrak{M}}, (\mathcal{M}_n)_{n \in \mathfrak{T}})$, an assignment α in \mathfrak{M} , and an instant $n \in \mathfrak{T}$, we have that $\mathfrak{M}, n \models^{\alpha} \varphi$ iff $\mathfrak{M}^n \models^{\alpha} \varphi$.

Let $\mathfrak{F} = (\Delta^{\mathfrak{F}}, (\mathcal{F}_n)_{n \in [0, l]})$ and $\mathfrak{M} = (\Delta^{\mathfrak{M}}, (\mathcal{M}_n)_{n \in \mathfrak{T}})$ be, respectively, a finite trace and a (finite or infinite) trace such that $\Delta^{\mathfrak{F}} = \Delta^{\mathfrak{M}}$ (writing Δ) and $a^{\mathcal{F}} = a^{\mathcal{M}}$, for all $a \in N_{\mathcal{I}}$. We denote by $\mathfrak{F} \cdot \mathfrak{M} = (\Delta^{\mathfrak{F} \cdot \mathfrak{M}}, (\mathcal{F} \cdot \mathcal{M}_n)_{n \in \mathfrak{T}'})$ the *concatenation of \mathfrak{F} with \mathfrak{M}* , defined as the trace with: $\Delta^{\mathfrak{F} \cdot \mathfrak{M}} = \Delta$; $a^{\mathcal{F} \cdot \mathcal{M}} = a^{\mathcal{F}}$, for all $a \in N_{\mathcal{I}}$; $\mathfrak{T}' = [0, \infty)$, if $\mathfrak{T} = [0, \infty)$, and $\mathfrak{T}' = [0, (l + l') + 1]$, if $\mathfrak{T} = [0, l']$; and for $P \in N_{\mathcal{P}}, n \in \mathfrak{T}'$:

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

We define the set of *extensions* of a finite trace \mathfrak{F} as the set of infinite traces $\text{Ext}(\mathfrak{F}) = \{\mathfrak{S} \mid \mathfrak{S} = \mathfrak{F} \cdot \mathfrak{S}', \text{ for some infinite trace } \mathfrak{S}'\}$. Instead, given a trace \mathfrak{M} , the set of *prefixes* of \mathfrak{M} is the set $\text{Pre}(\mathfrak{M}) = \{\mathfrak{F} \mid \mathfrak{M} = \mathfrak{F} \cdot \mathfrak{M}', \text{ for some trace } \mathfrak{M}'\}$. Moreover, we call the *frozen extension* of $\mathfrak{F} = (\Delta, (\mathcal{F}_n)_{n \in [0, l]})$, denoted by \mathfrak{F}^{ω} , the concatenation of \mathfrak{F} with the infinite trace $\mathfrak{S} = (\Delta, (\mathcal{I}_n)_{n \in [0, \infty)})$ such that $\mathcal{I}_n = \mathcal{F}_l$, for every $n \in [0, \infty)$. That is, \mathfrak{F}^{ω} is the infinite trace obtained from \mathfrak{F} by repeating its last time point infinitely often [28].

4 FINITE VS. INFINITE TRACES

In this section, we compare finite and infinite traces semantics. First, in Section 4.1, we lift to the first-order temporal logic setting a well-known reduction of propositional linear temporal logic

formula satisfiability from finite traces to infinite ones. Then, in Section 4.2, we establish model-theoretic conditions under which it is guaranteed that formulas equivalent on finite (respectively, infinite) traces are also equivalent on infinite (respectively, finite) traces. In addition, we syntactically define classes of formulas that are shown to satisfy such model-theoretic conditions, and for which the corresponding results on preservation of formula equivalences are thus inherited. We finally restrict ourselves to the problem of preserving satisfiability of a formula, from finite to infinite traces. For this case as well, we define a class of formulas for which it holds that satisfiability on finite traces implies satisfiability on infinite traces.

4.1 Reduction to Satisfiability on Infinite Traces

In the following, we show how to reduce the formula satisfiability problem on finite traces to the same problem on infinite traces. Similar to the encoding proposed in [48] for (propositional) *LTL*, to capture the finiteness of the temporal dimension, we introduce a fresh unary predicate E , standing for the *end of time*, with the following properties: (i) there is at 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 is permanent. We axiomatise these properties as follows (the propositional case of [48] can be easily recovered as a particular case):

$$\begin{aligned}\psi_f^1 &= \forall x \neg E(x) && \text{(Point (i)),} \\ \psi_f^2 &= \forall x \neg E(x) \mathcal{U} \forall x E(x) && \text{(Points (ii), (iii)),} \\ \psi_f^3 &= \Box \forall x (E(x) \rightarrow \bigcirc E(x)) && \text{(Point (iv)).}\end{aligned}$$

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 $a \in \mathbb{N}_l$. We denote by $\mathfrak{F} \cdot_E \mathfrak{I}$ the *end extension of \mathfrak{F} with \mathfrak{I}* , defined as the concatenation of \mathfrak{F} with \mathfrak{I} such that:

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

Clearly, end extensions characterise the satisfiability of ψ_f . We formalise this in the next lemma.

LEMMA 4.1. *For every infinite trace \mathfrak{I} , $\mathfrak{I} \models \psi_f$ iff $\mathfrak{I} = \mathfrak{F} \cdot_E \mathfrak{I}'$, for some finite trace \mathfrak{F} and some infinite trace \mathfrak{I}' .*

PROOF. ψ_f is satisfied in \mathfrak{I} iff there is $k > 0$ such that, for all $d \in \Delta$, it holds that: $d \notin E^{T_j}$, for all $j \in [0, k)$; and $d \in E^{T_i}$, for all $i \in [k, \infty)$. That is, $\mathfrak{I} = \mathfrak{F} \cdot_E \mathfrak{I}'$, for some finite trace \mathfrak{F} and some infinite trace \mathfrak{I}' . \square

We now introduce a translation \cdot^\dagger for $T_{\mathcal{U}}\mathcal{QL}$ formulas, used together with the end of time formula, ψ_f , to capture satisfiability on finite traces. More formally, a $T_{\mathcal{U}}\mathcal{QL}$ 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

$$\begin{aligned}(P(\bar{\tau}))^\dagger &= P(\bar{\tau}), \\ (\neg\psi)^\dagger &= \neg\psi^\dagger, \\ (\psi \wedge \chi)^\dagger &= \psi^\dagger \wedge \chi^\dagger, \\ (\exists x\psi)^\dagger &= \exists x\psi^\dagger, \\ (\psi \mathcal{U} \chi)^\dagger &= \psi^\dagger \mathcal{U} (\chi^\dagger \wedge \psi_f^1).\end{aligned}$$

Before showing the correctness of the translation, the following lemma shows the relevance of end extensions when interpreting translated formulas.

LEMMA 4.2. *Let $\mathfrak{F} \cdot_E \mathfrak{S}$ be an end extension of a finite trace \mathfrak{F} . For every $T_{\mathcal{U}}\mathcal{QL}$ formula φ and every assignment α , $\mathfrak{F} \models^\alpha \varphi$ iff $\mathfrak{F} \cdot_E \mathfrak{S} \models^\alpha \varphi^\dagger$.*

PROOF. Let $\mathfrak{F} = (\Delta^{\mathfrak{F}}, (\mathcal{F}_n)_{n \in [0, l]})$ be a finite trace and let $\mathfrak{F} \cdot_E \mathfrak{S} = (\Delta^{\mathfrak{F} \cdot \mathfrak{S}}, (\mathcal{F} \cdot \mathcal{I}_n)_{n \in [0, \infty)})$ be an end extension of \mathfrak{F} . We prove by structural induction the following more general statement. For all $n \in [0, l]$ and all assignments α :

$$\mathfrak{F}, n \models^\alpha \varphi \text{ iff } \mathfrak{F} \cdot_E \mathfrak{S}, n \models^\alpha \varphi^\dagger.$$

For the base case $\varphi = P(\bar{\tau})$, the statement follows from the definitions of $\mathfrak{F} \cdot_E \mathfrak{S}$ and \cdot^\dagger , while the proof of the inductive cases $\varphi = \neg\psi$, $\varphi = (\psi \wedge \chi)$, and $\varphi = \exists x\psi$ is straightforward.

We show the inductive case $\varphi = (\psi \mathcal{U} \chi)$. We have that $\mathfrak{F}, n \models^\alpha \psi \mathcal{U} \chi$ iff there is $m \in (n, l]$ such that $\mathfrak{F}, m \models^\alpha \chi$ and for all $i \in (n, m)$, $\mathfrak{F}, i \models^\alpha \psi$. By the inductive hypothesis, this happens iff there is $m \in (n, l]$ such that $\mathfrak{F} \cdot_E \mathfrak{S}, m \models^\alpha \chi^\dagger$ and for all $i \in (n, m)$, $\mathfrak{F} \cdot_E \mathfrak{S}, i \models^\alpha \psi^\dagger$. Since $E^{\mathcal{F} \cdot_E \mathcal{I}_j} = \emptyset$ for all $j \in [0, l]$, this means that $\mathfrak{F} \cdot_E \mathfrak{S}, n \models^\alpha \psi^\dagger \mathcal{U} (\chi^\dagger \wedge \forall x \neg E(x))$. That is, $\mathfrak{F} \cdot_E \mathfrak{S}, n \models^\alpha (\psi \mathcal{U} \chi)^\dagger$. \square

Using the previous lemmas, we can show the correctness of the reduction of the $T_{\mathcal{U}}\mathcal{QL}$ satisfiability problem on finite traces to the same problem for $T_{\mathcal{U}}\mathcal{QL}$ on infinite traces.

THEOREM 4.3. *A $T_{\mathcal{U}}\mathcal{QL}$ formula φ is satisfiable on finite traces iff $\varphi^\dagger \wedge \psi_f$ is satisfiable on infinite traces.*

PROOF. If φ is satisfied in some finite trace \mathfrak{F} , then (by Lemmas 4.1 and 4.2) any end extension $\mathfrak{F} \cdot_E \mathfrak{S}$ satisfies $\varphi^\dagger \wedge \psi_f$. Conversely, suppose that $\varphi^\dagger \wedge \psi_f$ is satisfied in some infinite trace \mathfrak{S} under an assignment α . By Lemma 4.1, $\mathfrak{S} = \mathfrak{F} \cdot_E \mathfrak{S}'$, for some finite trace \mathfrak{F} and some infinite trace \mathfrak{S}' . Since $\mathfrak{F} \cdot_E \mathfrak{S}' \models^\alpha \varphi^\dagger$, by Lemma 4.2, we have that $\mathfrak{F} \models^\alpha \varphi$. \square

4.2 Blurring the Distinction Between Finite and Infinite Traces

While certain formulas, such as $\square\top$, are satisfiable both on finite and infinite traces, others, e.g., $\diamond last$ and $\square^+\circ\top$, are only satisfiable on finite traces and on infinite traces, respectively. It is thus of interest to understand in which cases satisfiability on finite and infinite traces coincide, so that solving the problem in one case answers to the other as well. 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 [28]. Moreover, $\square^+\diamond^+\varphi$ and $\diamond^+\square^+\varphi$ are not equivalent on infinite traces, whereas on finite traces they are both equivalent to $\diamond^+(last \wedge \varphi)$ [48]. 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. We first propose semantic properties under which it is guaranteed that formula equivalences are preserved from finite to infinite traces, or vice versa, thus allowing to blur the distinction between these semantics. Then, we syntactically define classes of formulas satisfying some of these semantic properties, so to provide a sufficient criterion for the preservation of equivalences from finite to infinite traces, or vice versa. Finally, we focus on preserving satisfiability from the finite to the infinite case, devising a wider class of formulas for which this preservation holds.

4.2.1 Finite vs. Infinite Traces: Semantic Characterisation. For a $T_{\mathcal{U}}\mathcal{QL}$ formula φ and a quantifier $Q \in \{\exists, \forall\}$, we say that φ satisfies F_Q (or that φ is F_Q) if, for all finite traces \mathfrak{F} and all assignments α , it satisfies the *finite trace property*:

$$\mathfrak{F} \models^\alpha \varphi \Leftrightarrow Q\mathfrak{S} \in Ext(\mathfrak{F}). \mathfrak{S} \models^\alpha \varphi,$$

and, similarly, that φ satisfies l_Q (or that φ is l_Q) if, for all infinite traces \mathfrak{S} and all assignments α , it satisfies the *infinite trace property*:

$$\mathfrak{S} \models^\alpha \varphi \Leftrightarrow Q\mathfrak{S} \in \text{Pre}(\mathfrak{S}).\mathfrak{S} \models^\alpha \varphi.$$

To show intuitive examples, let us consider the case where φ is a Boolean combination of atomic formulas. Examples of formulas satisfying F_V and l_\exists are formulas of the form $\diamond^+ \varphi$. Formulas of the form $\diamond \varphi$ are also l_\exists , but in general not F_V , as witnessed, for instance, by $\diamond \top$. On the other hand, the properties F_\exists and l_V capture for example formulas of the form $\square^+ \varphi$. Formulas of the form $\square \varphi$ are also l_V , but not necessarily F_\exists , because of, e.g., $\square \perp$. These observations, that can easily be checked, are also immediate consequences of Lemmas 4.11 and 4.12 below.

We 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. Finally, given a property P , we denote by $T_{\mathcal{U}QL}(P)$ the set of $T_{\mathcal{U}QL}$ formulas satisfying P .

The semantic properties F_Q and l_Q capture different classes of $T_{\mathcal{U}QL}$ formulas, as illustrated by the following example.

Example 4.4. The following formulas satisfy exactly one of the corresponding finite or infinite trace properties.

F_\exists	$\diamond^+ last \vee \diamond P$
F_V	$\forall x \diamond^+ Q(x)$
l_\exists	$\square \circ \top \vee last$
l_V	$\square^+ P \vee \diamond^+(P \wedge last)$

Indeed, by using the formulas from Example 4.4, we can prove the following.

PROPOSITION 4.5. *The sets $T_{\mathcal{U}QL}(F_\exists)$, $T_{\mathcal{U}QL}(F_V)$, $T_{\mathcal{U}QL}(l_\exists)$, and $T_{\mathcal{U}QL}(l_V)$ are mutually incomparable with respect to inclusion.*

PROOF. For every $X, Y \in \{T_{\mathcal{U}QL}(F_\exists), T_{\mathcal{U}QL}(F_V), T_{\mathcal{U}QL}(l_\exists), T_{\mathcal{U}QL}(l_V)\}$, we use the formulas from Example 4.4 to show that $X \not\subseteq Y$.

- $T_{\mathcal{U}QL}(F_\exists) \not\subseteq Y$, with $Y \in \{T_{\mathcal{U}QL}(F_V), T_{\mathcal{U}QL}(l_\exists), T_{\mathcal{U}QL}(l_V)\}$. It can be seen that the formula $\diamond^+ last \vee \diamond P$ is F_\exists . However,
 - it is not F_V : (under any assignment) the formula is satisfied in a finite trace $\mathfrak{F} = (\Delta, (\mathcal{F}_0))$, with 0 as its only time point and such that $P^{\mathcal{F}_0} = \emptyset$, but an extension $\mathfrak{S} \in \text{Ext}(\mathfrak{F})$ such that $\mathfrak{S} = (\Delta, (\mathcal{I}_n)_{n \in [0, \infty)})$, with $\mathcal{F}_0 = \mathcal{I}_0$ and $P^{\mathcal{I}_n} = \emptyset$, for every $n \in [0, \infty)$, does not satisfy it;
 - it is not l_\exists : (under any assignment) an infinite trace $\mathfrak{S} = (\Delta, (\mathcal{I}_n)_{n \in [0, \infty)})$ such that $P^{\mathcal{I}_n} = \emptyset$, for every $n \in [0, \infty)$ does not satisfy the formula, whereas any (and thus some) prefix $\mathfrak{F} \in \text{Pre}(\mathfrak{S})$ satisfies it;
 - it is not l_V : shown as in the previous case.
- $T_{\mathcal{U}QL}(F_V) \not\subseteq Y$, with $Y \in \{T_{\mathcal{U}QL}(F_\exists), T_{\mathcal{U}QL}(l_\exists), T_{\mathcal{U}QL}(l_V)\}$. It can be seen that the formula $\forall x \diamond^+ Q(x)$ is F_V . However,
 - it is not F_\exists : (under any assignment) the finite trace $\mathfrak{F} = (\Delta, (\mathcal{F}_0))$, with 0 as its only time point and such that $Q^{\mathcal{F}_0} = \emptyset$, does not satisfy the formula, whereas an extension $\mathfrak{S} \in \text{Ext}(\mathfrak{F})$ such that $\mathfrak{S} = (\Delta, (\mathcal{I}_n)_{n \in [0, \infty)})$, with $Q^{\mathcal{I}_1} = \Delta$, satisfies it;
 - it is not l_\exists : (under any assignment) consider an infinite trace \mathfrak{S} with a (countably) infinite domain $\Delta = \{d_1, d_2, \dots, d_n, \dots\}$, where the n -th domain element is in the extension of Q

- exactly at time point $n \in \mathbb{N}$, i.e., $\mathfrak{S}, n \models Q(d_n)$ and $\mathfrak{S}, i \not\models Q(d_n)$, for any $i \neq n$. It can be seen that there is no finite prefix of this infinite trace where $\forall x \diamond^+ Q(x)$ holds.
- it is not l_\forall : (under any assignment) an infinite trace $\mathfrak{S} = (\Delta, (\mathcal{I}_n)_{n \in [0, \infty)})$ such that $Q^{\mathcal{I}_0} = \emptyset$ and $Q^{\mathcal{I}_1} = \Delta$ satisfies the formula, but the prefix $\mathfrak{F} \in \text{Pre}(\mathfrak{S})$ such that $\mathfrak{F} = (\Delta, (\mathcal{F}_0))$, where $\mathcal{F}_0 = \mathcal{I}_0$, does not satisfy it.
 - $T_{\mathcal{U}}\mathcal{QL}(\text{l}_\exists) \not\subseteq Y$, with $Y \in \{T_{\mathcal{U}}\mathcal{QL}(\text{F}_\exists), T_{\mathcal{U}}\mathcal{QL}(\text{F}_\forall), T_{\mathcal{U}}\mathcal{QL}(\text{l}_\forall)\}$. It can be seen that the formula $\Box \circ \top \vee \text{last}$ is l_\exists . However,
 - it is not F_\exists : (under any assignment) a finite trace $\mathfrak{F} = (\Delta, (\mathcal{F}_0, \mathcal{F}_1))$, with 0, 1 as its only time points, does not satisfy the formula (at time point 0), whereas any (and thus some) extension $\mathfrak{S} \in \text{Ext}(\mathfrak{F})$ satisfies it;
 - it is not F_\forall : shown as in the previous case;
 - it is not l_\forall : (under any assignment) an infinite trace $\mathfrak{S} = (\Delta, (\mathcal{I}_n)_{n \in [0, \infty)})$ satisfies the formula, but a prefix $\mathfrak{F} \in \text{Pre}(\mathfrak{S})$ such that $\mathfrak{F} = (\Delta, (\mathcal{F}_0, \mathcal{F}_1))$, with $\mathcal{F}_i = \mathcal{I}_i$, for $i \in \{0, 1\}$, does not.
 - $T_{\mathcal{U}}\mathcal{QL}(\text{l}_\forall) \not\subseteq Y$, with $Y \in \{T_{\mathcal{U}}\mathcal{QL}(\text{F}_\exists), T_{\mathcal{U}}\mathcal{QL}(\text{F}_\forall), T_{\mathcal{U}}\mathcal{QL}(\text{l}_\exists)\}$. It can be seen that the formula $\Box^+ P \vee \diamond^+(P \wedge \text{last})$ is l_\forall . However,
 - it is not F_\exists : (under any assignment) the formula is satisfied in a finite trace $\mathfrak{F} = (\Delta, (\mathcal{F}_0, \mathcal{F}_1))$, with 0, 1 as its only time points and such that $P^{\mathcal{F}_0} = \emptyset$ and $P^{\mathcal{F}_1} = \Delta$, whereas no extension $\mathfrak{S} \in \text{Ext}(\mathfrak{F})$ satisfies it;
 - it is not F_\forall : shown as in the previous case;
 - it is not l_\exists : (under any assignment) an infinite trace $\mathfrak{S} = (\Delta, (\mathcal{I}_n)_{n \in [0, \infty)})$ such that $P^{\mathcal{I}_0} = \Delta$ and $P^{\mathcal{I}_i} = \emptyset$, for $i > 0$, does not satisfy the formula, whereas the prefix $\mathfrak{F} \in \text{Ext}(\mathfrak{S})$ such that $\mathfrak{F} = (\Delta, (\mathcal{F}_0))$, with $\mathcal{F}_0 = \mathcal{I}_0$, satisfies it. \square

Observe that the F_\forall formula $\forall x \diamond^+ Q(x)$ from Example 4.4 violates $\text{l}_{\Rightarrow \exists}$, and hence does not satisfy l_\exists , due to an interplay between the universal quantifier and the diamond operator. Such behaviour is not shared by the propositional formula $\diamond^+ P$, which is both F_\forall and l_\exists . The remaining cases from Example 4.4 are all *LTL* formulas.

On the relationships between the one directional properties, we have the following.

PROPOSITION 4.6. *The following statements hold.*

- (1) Given $Q, Q' \in \{\exists, \forall\}$, with $Q \neq Q'$, we have: $\varphi \in T_{\mathcal{U}}\mathcal{QL}(\text{F}_{\Rightarrow Q})$ iff $\neg\varphi \in T_{\mathcal{U}}\mathcal{QL}(\text{F}_{\Leftarrow Q'})$, and $\varphi \in T_{\mathcal{U}}\mathcal{QL}(\text{l}_{\Rightarrow Q})$ iff $\neg\varphi \in T_{\mathcal{U}}\mathcal{QL}(\text{l}_{\Leftarrow Q'})$.
- (2) Given $P \in \{\text{F}, \text{l}\}$, we have $T_{\mathcal{U}}\mathcal{QL}(\text{P}_{\Rightarrow \forall}) \subseteq T_{\mathcal{U}}\mathcal{QL}(\text{P}_{\Rightarrow \exists})$ and $T_{\mathcal{U}}\mathcal{QL}(\text{P}_{\Leftarrow \exists}) \subseteq T_{\mathcal{U}}\mathcal{QL}(\text{P}_{\Leftarrow \forall})$.
- (3) $T_{\mathcal{U}}\mathcal{QL}(\text{F}_{\Leftarrow \exists}) = T_{\mathcal{U}}\mathcal{QL}(\text{l}_{\Rightarrow \forall})$ and $T_{\mathcal{U}}\mathcal{QL}(\text{l}_{\Leftarrow \exists}) = T_{\mathcal{U}}\mathcal{QL}(\text{F}_{\Rightarrow \forall})$.
- (4) $T_{\mathcal{U}}\mathcal{QL}(\text{F}_{\Rightarrow \forall})$ and $T_{\mathcal{U}}\mathcal{QL}(\text{l}_{\Rightarrow \exists})$, as well as $T_{\mathcal{U}}\mathcal{QL}(\text{l}_{\Rightarrow \forall})$ and $T_{\mathcal{U}}\mathcal{QL}(\text{F}_{\Rightarrow \exists})$, are incomparable with respect to inclusion.

PROOF. (1) Let $Q, Q' \in \{\exists, \forall\}$, with $Q \neq Q'$, and suppose that $\varphi \in T_{\mathcal{U}}\mathcal{QL}(\text{F}_{\Rightarrow Q})$. This means that, for all \mathfrak{F} and all assignments α , we have: $\mathfrak{F} \models^\alpha \varphi \Rightarrow Q\mathfrak{S} \models^\alpha \varphi$. By contraposition, the previous step means, for all \mathfrak{F} and all assignments α : $Q'\mathfrak{S} \not\models^\alpha \varphi \Rightarrow \mathfrak{F} \not\models^\alpha \varphi$. That is, for all \mathfrak{F} and all assignments α : $\mathfrak{F} \models^\alpha \neg\varphi \Leftarrow Q'\mathfrak{S} \models^\alpha \neg\varphi$. Thus, $\neg\varphi \in T_{\mathcal{U}}\mathcal{QL}(\text{F}_{\Leftarrow Q'})$. Similarly, it can be seen that $\varphi \in T_{\mathcal{U}}\mathcal{QL}(\text{l}_{\Rightarrow Q})$ iff $\neg\varphi \in T_{\mathcal{U}}\mathcal{QL}(\text{l}_{\Leftarrow Q'})$, for $Q, Q' \in \{\exists, \forall\}$, $Q \neq Q'$.

(2) Straightforward from the definitions.

(3) We first show that $T_{\mathcal{U}}\mathcal{QL}(\text{F}_{\Leftarrow \exists}) \subseteq T_{\mathcal{U}}\mathcal{QL}(\text{l}_{\Rightarrow \forall})$. Given a formula φ , suppose that $\varphi \in T_{\mathcal{U}}\mathcal{QL}(\text{F}_{\Leftarrow \exists})$, i.e., for every finite trace \mathfrak{F} and every assignment α , if there exists $\mathfrak{S} \in \text{Ext}(\mathfrak{F})$ such that $\mathfrak{S} \models^\alpha \varphi$, then $\mathfrak{F} \models^\alpha \varphi$. Now, given an infinite trace \mathfrak{S}' and an assignment α' , suppose that $\mathfrak{S}' \models^{\alpha'} \varphi$, and consider an arbitrary $\mathfrak{F}' \in \text{Pre}(\mathfrak{S}')$. Since $\mathfrak{S}' \in \text{Ext}(\mathfrak{F}')$, by the former assumptions we have that $\mathfrak{F}' \models^{\alpha'} \varphi$. Hence, $\varphi \in T_{\mathcal{U}}\mathcal{QL}(\text{l}_{\Rightarrow \forall})$.

To see that $T_{\mathcal{U}}\mathcal{QL}(I_{\Rightarrow\forall}) \subseteq T_{\mathcal{U}}\mathcal{QL}(F_{\Leftarrow\exists})$, suppose that $\varphi \in T_{\mathcal{U}}\mathcal{QL}(I_{\Rightarrow\forall})$, that is: for every infinite trace \mathfrak{S} and assignment α , $\mathfrak{S} \models^{\alpha} \varphi$ implies that every $\mathfrak{F} \in \text{Pre}(\mathfrak{S})$ is such that $\mathfrak{F} \models^{\alpha} \varphi$. Now, for any finite trace \mathfrak{F}' and assignment α' , suppose that there exists $\mathfrak{S}' \in \text{Ext}(\mathfrak{F}')$ such that $\mathfrak{S}' \models^{\alpha'} \varphi$. Since $\mathfrak{F}' \in \text{Pre}(\mathfrak{S}')$, by the assumption above we obtain that $\mathfrak{F}' \models^{\alpha'} \varphi$. Thus, $\varphi \in T_{\mathcal{U}}\mathcal{QL}(F_{\Leftarrow\exists})$.

We now show that $T_{\mathcal{U}}\mathcal{QL}(I_{\Leftarrow\exists}) \subseteq T_{\mathcal{U}}\mathcal{QL}(F_{\Rightarrow\forall})$. Let $\varphi \in T_{\mathcal{U}}\mathcal{QL}(I_{\Leftarrow\exists})$, meaning that: for every infinite trace \mathfrak{S} and every assignment α , if there exists $\mathfrak{F} \in \text{Pre}(\mathfrak{S})$ such that $\mathfrak{F} \models^{\alpha} \varphi$, then $\mathfrak{S} \models^{\alpha} \varphi$. Moreover, given a finite trace \mathfrak{F}' and an assignment α' , assume that $\mathfrak{F}' \models \varphi$. Considering an arbitrary infinite extension $\mathfrak{S}' \in \text{Ext}(\mathfrak{F}')$, we obtain, from the fact that $\mathfrak{F}' \in \text{Pre}(\mathfrak{S}')$ and the assumptions above, that $\mathfrak{S}' \models^{\alpha'} \varphi$. This implies that $\varphi \in T_{\mathcal{U}}\mathcal{QL}(F_{\Rightarrow\forall})$.

Finally, we prove that $T_{\mathcal{U}}\mathcal{QL}(F_{\Rightarrow\forall}) \subseteq T_{\mathcal{U}}\mathcal{QL}(I_{\Leftarrow\exists})$. Suppose that $\varphi \in T_{\mathcal{U}}\mathcal{QL}(F_{\Rightarrow\forall})$, meaning: for every finite trace \mathfrak{F} and assignment α , $\mathfrak{F} \models^{\alpha} \varphi$ implies that every $\mathfrak{S} \in \text{Ext}(\mathfrak{F})$ is such that $\mathfrak{S} \models^{\alpha} \varphi$. Given an arbitrary infinite trace \mathfrak{S}' and an assignment α' such that $\mathfrak{S}' \models^{\alpha'} \varphi$, suppose that, for some $\mathfrak{F}' \in \text{Pre}(\mathfrak{S}')$, we have $\mathfrak{F}' \models^{\alpha'} \varphi$. Since $\mathfrak{S}' \in \text{Ext}(\mathfrak{F}')$, by the assumptions above we have that $\mathfrak{S}' \models^{\alpha'} \varphi$. Therefore, $\varphi \in T_{\mathcal{U}}\mathcal{QL}(I_{\Leftarrow\exists})$.

(4) We have, e.g., that $\Box^+ \circ \top$ always holds on infinite traces, but it is unsatisfiable on finite traces. Hence, $\Box^+ \circ \top \in T_{\mathcal{U}}\mathcal{QL}(F_{\Rightarrow\forall})$, whereas $\Box^+ \circ \top \notin T_{\mathcal{U}}\mathcal{QL}(I_{\Rightarrow\exists})$. By Point (2), this implies also that $\Box^+ \circ \top \in T_{\mathcal{U}}\mathcal{QL}(F_{\Rightarrow\exists})$ and $\Box^+ \circ \top \notin T_{\mathcal{U}}\mathcal{QL}(I_{\Rightarrow\forall})$. On the other hand, $\diamond \text{last}$ always holds on finite traces, while it is unsatisfiable on infinite ones. Hence, $\diamond \text{last} \in T_{\mathcal{U}}\mathcal{QL}(I_{\Rightarrow\forall})$, but $\diamond \text{last} \notin T_{\mathcal{U}}\mathcal{QL}(F_{\Rightarrow\exists})$. By Point (2), we obtain also that $\diamond \text{last} \in T_{\mathcal{U}}\mathcal{QL}(I_{\Rightarrow\exists})$, and $\diamond \text{last} \notin T_{\mathcal{U}}\mathcal{QL}(F_{\Rightarrow\forall})$. \square

We now consider the problem of formula equivalence, by showing under which semantic properties equivalence between formulas can be blurred. The following theorem provides sufficient conditions to preserve formula equivalence from the infinite to the finite case (cf. the notion of *LTL compliance* in [29]).

THEOREM 4.7. *Given $X \in \{T_{\mathcal{U}}\mathcal{QL}(F_{\exists}), T_{\mathcal{U}}\mathcal{QL}(F_{\forall})\}$ and $\varphi, \psi \in X$, it holds that $\varphi \equiv_i \psi$ implies $\varphi \equiv_f \psi$.*

PROOF. First, assume $\varphi, \psi \in T_{\mathcal{U}}\mathcal{QL}(F_{\exists})$. Let \mathfrak{F} be a finite trace and α an assignment such that $\mathfrak{F} \models^{\alpha} \varphi$. By F_{\exists} , there is an infinite trace $\mathfrak{S} \in \text{Ext}(\mathfrak{F})$ such that $\mathfrak{S} \models^{\alpha} \varphi$. Since $\varphi \equiv_i \psi$, we have that $\mathfrak{S} \models^{\alpha} \psi$. By F_{\exists} , $\mathfrak{F} \models^{\alpha} \psi$. The converse direction can be obtained similarly, by swapping φ and ψ . The proof for $\varphi, \psi \in T_{\mathcal{U}}\mathcal{QL}(F_{\forall})$ is analogous. \square

Theorem 4.7 does not hold for formulas that satisfy only I_{\exists} or I_{\forall} . Consider the formulas $\Box \circ \top \vee \text{last}$, from Example 4.4, and $\Box \circ \top \vee \circ \text{last}$, which are both I_{\exists} . These formulas are equivalent only on infinite traces. Also, $\Box^+ P \vee \diamond^+(P \wedge \text{last})$, from Example 4.4, and $\Box^+ P \vee \diamond^+(P \wedge \circ \text{last})$ are I_{\forall} , and equivalent on infinite but not on finite traces. The last example also shows that the condition $F_{\Leftarrow\exists}$ alone, which coincides with $I_{\Rightarrow\forall}$ by Proposition 4.6, Point (3), is not sufficient for Theorem 4.7. Moreover, $F_{\Rightarrow\exists}$ alone is also not sufficient. To see this, consider, e.g., $\Box^+ \circ \top \vee (P \wedge \diamond \text{last})$ and $\Box^+ \circ \top \vee \diamond \text{last}$, which are $F_{\Rightarrow\exists}$ but equivalent only on infinite traces.

We now present sufficient conditions to preserve equivalences from the finite to the infinite case.

THEOREM 4.8. *Given $X \in \{T_{\mathcal{U}}\mathcal{QL}(I_{\exists}), T_{\mathcal{U}}\mathcal{QL}(I_{\forall})\}$ and $\varphi, \psi \in X$, it holds that $\varphi \equiv_f \psi$ implies $\varphi \equiv_i \psi$.*

PROOF. First, suppose that $\varphi, \psi \in T_{\mathcal{U}}\mathcal{QL}(I_{\exists})$. Let \mathfrak{S} be an infinite trace and α an assignment such that $\mathfrak{S} \models^{\alpha} \varphi$. By I_{\exists} , there is $\mathfrak{F} \in \text{Pre}(\mathfrak{S})$ such that $\mathfrak{F} \models^{\alpha} \varphi$. As $\varphi \equiv_f \psi$, this means that $\mathfrak{F} \models^{\alpha} \psi$. Since ψ is I_{\exists} , we have $\mathfrak{S} \models^{\alpha} \psi$. The converse direction is obtained similarly, by swapping φ and ψ . The proof for $\varphi, \psi \in T_{\mathcal{U}}\mathcal{QL}(I_{\forall})$ is analogous. \square

The properties F_{\exists} or F_{\forall} alone 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 $\diamond^+ last \vee \diamond Q(x)$ and $\diamond^+ last \vee \diamond Q(x) \vee \vartheta_i$, where Formula ϑ_i (used also in Section 5.3) is as follows:

$$\vartheta_i = Q(a) \wedge \square^+ \forall x \left(Q(x) \rightarrow \square^+ (\neg Q(x) \wedge \exists y (R(x, y) \wedge Q(y))) \right).$$

These formulas are F_{\exists} , however, they are only equivalent on finite traces. Moreover, if we take $\forall x \diamond^+ Q(x)$, from Example 4.4, and $\forall x \diamond^+ Q(x) \vee \vartheta_i$, we have that they are both F_{\forall} , though equivalent only on finite traces. The last example also shows that the condition $I_{\Leftarrow \exists}$ alone, which corresponds to $F_{\Rightarrow \forall}$ by Proposition 4.6, Point (3), is not sufficient for Theorem 4.8. We now argue that $I_{\Rightarrow \exists}$ alone is also not sufficient. To see this, consider, e.g., $(P \wedge \square^+ \diamond \top) \vee \diamond last$ and $\square^+ \diamond \top \vee \diamond last$, which are $I_{\Rightarrow \exists}$ but are equivalent only on finite traces.

From Theorems 4.7 and 4.8 we have that if $\varphi, \psi \in T_{\mathcal{U}}\mathcal{QL}(F_{\exists})$ or $\varphi, \psi \in T_{\mathcal{U}}\mathcal{QL}(F_{\forall})$, and $\varphi, \psi \in T_{\mathcal{U}}\mathcal{QL}(I_{\exists})$ or $\varphi, \psi \in T_{\mathcal{U}}\mathcal{QL}(I_{\forall})$, then $\varphi \equiv_f \psi$ if and only if $\varphi \equiv_i \psi$. In particular, the above examples show that if, from a given pair of conditions F_Q and I_Q , we remove any of the two properties, then formula equivalences on finite and infinite traces may not coincide.

4.2.2 Preserving Formula Equivalences: Syntactic Characterisation. We now analyse syntactic features of the properties introduced so far, providing classes of formulas that satisfy them. This will in turn allow us to show results on preservation of equivalences, for such formulas, between finite and infinite traces.

First, we make the following observation concerning non-temporal $T_{\mathcal{U}}\mathcal{QL}$ formulas.

PROPOSITION 4.9. *For every non-temporal $T_{\mathcal{U}}\mathcal{QL}$ formula φ , every finite trace $\mathfrak{F} = (\Delta^{\mathfrak{F}}, (\mathcal{F}_n)_{n \in [0,1]})$, every infinite trace $\mathfrak{I} = (\Delta^{\mathfrak{I}}, (\mathcal{I}_n)_{n \in [0,\infty)})$, every $n \geq 0$, and every assignment α in \mathfrak{F} or \mathfrak{I} , respectively, the following hold, where $Q \in \{\exists, \forall\}$:*

$$\begin{aligned} - \mathfrak{F}, n \models^{\alpha} \varphi &\Leftrightarrow Q\mathfrak{I} \in \text{Ext}(\mathfrak{F}).\mathfrak{I}, n \models^{\alpha} \varphi; \\ - \mathfrak{I}, n \models^{\alpha} \varphi &\Leftrightarrow Q\mathfrak{F} \in \text{Pre}(\mathfrak{I}).\mathfrak{F}, n \models^{\alpha} \varphi. \end{aligned}$$

In particular, $\varphi \in T_{\mathcal{U}}\mathcal{QL}(P)$, for every $P \in \{F_{\exists}, F_{\forall}, I_{\exists}, I_{\forall}\}$.

PROOF. Clearly, since φ has no temporal operators, for any finite or infinite trace \mathfrak{M} , \mathfrak{M} satisfies φ at n under α iff any extension or prefix of \mathfrak{M} satisfies φ at n under α , respectively. \square

We now introduce the relevant fragments of $T_{\mathcal{U}}\mathcal{QL}$ that will be analysed in the rest of this section. First, \mathcal{U}^+ -formulas φ, ψ are built according to the grammar (with $P \in N_P$):

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

Moreover, we call \mathcal{U} -formulas the set of formulas generated by allowing $\varphi \mathcal{U} \psi$ in the grammar rule for \mathcal{U}^+ -formulas,¹ and we call $\mathcal{U}^+\forall$ -formulas the result of allowing $\forall x \varphi$ in the grammar rule for \mathcal{U}^+ -formulas.

Next, \mathcal{R}^+ -formulas φ, ψ are built according to the grammar (with $P \in N_P$):

$$P(\bar{\tau}) \mid \neg P(\bar{\tau}) \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \forall x \varphi \mid \varphi \mathcal{R}^+ \psi.$$

We call \mathcal{R} -formulas the set of formulas generated by allowing $\varphi \mathcal{R} \psi$ in the grammar rule for \mathcal{R}^+ -formulas,² and we call $\mathcal{R}^+\exists$ -formulas the result of allowing $\exists x \varphi$ in the grammar rule for \mathcal{R}^+ -formulas.

¹Recall that \mathcal{U}^+ is syntactic sugar in the fragment of \mathcal{U} -formulas, since $\varphi \mathcal{U}^+ \psi := \psi \vee (\varphi \wedge \varphi \mathcal{U} \psi)$.

²Recall that \mathcal{R}^+ is syntactic sugar in the fragment of \mathcal{R} -formulas, since $\varphi \mathcal{R}^+ \psi := \psi \wedge (\varphi \vee \varphi \mathcal{R} \psi)$.

Table 1. Syntactic fragments with corresponding semantic properties and preservation of equivalences

	Properties	Equivalences
\mathcal{U}^{\forall}	F_{\forall} (Lemma 4.11)	$i \Rightarrow f$ (Theorem 4.7)
\mathcal{R}^{\exists}	F_{\exists} (Lemma 4.12)	$i \Rightarrow f$ (Theorem 4.7)
\mathcal{U}	I_{\exists} (Lemma 4.11)	$f \Rightarrow i$ (Theorem 4.8)
\mathcal{R}	I_{\forall} (Lemma 4.12)	$f \Rightarrow i$ (Theorem 4.8)
\mathcal{U}^+	F_{\forall}, I_{\exists} (Lemma 4.11)	$f \Leftrightarrow i$ (Theorems 4.7-4.8)
\mathcal{R}^+	F_{\exists}, I_{\forall} (Lemma 4.12)	$f \Leftrightarrow i$ (Theorems 4.7-4.8)

Having introduced such fragments, the rest of this section will be devoted to the proof of the following theorem, which is a consequence of Theorems 4.7–4.8 above and Lemmas 4.11–4.12 below, as outlined in Table 1.

THEOREM 4.10. *The following hold:*

- (1) for all \mathcal{U}^+ - or \mathcal{R}^+ -formulas φ and ψ , $\varphi \equiv_f \psi$ if and only if $\varphi \equiv_i \psi$;
- (2) for all \mathcal{U}^{\forall} - or \mathcal{R}^{\exists} -formulas φ and ψ , $\varphi \equiv_i \psi$ implies $\varphi \equiv_f \psi$;
- (3) for all \mathcal{U} - or \mathcal{R} -formulas φ and ψ , $\varphi \equiv_f \psi$ implies $\varphi \equiv_i \psi$.

We first show that every \mathcal{U}^{\forall} -formula is F_{\forall} , and every \mathcal{U} -formula is I_{\exists} . As an immediate consequence, we obtain that every \mathcal{U}^+ -formula is both F_{\forall} and I_{\exists} .

LEMMA 4.11. *\mathcal{U}^{\forall} -formulas are F_{\forall} and \mathcal{U} -formulas are I_{\exists} . Thus, \mathcal{U}^+ -formulas are both F_{\forall} and I_{\exists} .*

PROOF. We first show that all \mathcal{U}^{\forall} -formulas are F_{\forall} . In Claim 1, we show that all \mathcal{U}^{\forall} -formulas are $F_{\Rightarrow\forall}$ (in fact, for the $F_{\Rightarrow\forall}$ case, we can also allow $\varphi \mathcal{U} \psi$ in the grammar). Then, in Claim 2, we show that all \mathcal{U}^{\forall} -formulas are $F_{\Leftarrow\forall}$.

CLAIM 1. *\mathcal{U}^{\forall} -formulas are $F_{\Rightarrow\forall}$.*

PROOF OF CLAIM 1. Given a finite trace $\mathfrak{F} = (\Delta, (\mathcal{F}_n)_{n \in [0, l]})$ and an assignment α in \mathfrak{F} , we show that $\mathfrak{F} \models^{\alpha} \varphi$ implies that, for every $\mathfrak{S} \in \text{Ext}(\mathfrak{F})$, $\mathfrak{S} \models^{\alpha} \varphi$. The proof is by structural induction on φ . By Proposition 4.9, the statement holds for the base cases of $\varphi = P(\bar{\tau})$ and $\varphi = \neg P(\bar{\tau})$. We now proceed with the inductive steps.

- $\varphi = \psi \mathcal{U}^+ \chi$. Suppose that $\mathfrak{F} \models^{\alpha} \psi \mathcal{U}^+ \chi$. This means that there exists $n \in [0, l]$ such that $\mathfrak{F}, n \models^{\alpha} \chi$, i.e., $\mathfrak{F}^n \models^{\alpha} \chi$, and, for every $i \in [0, n)$, $\mathfrak{F}, i \models^{\alpha} \psi$, i.e., $\mathfrak{F}^i \models^{\alpha} \psi$. By the inductive hypothesis, we have that there exists $n \in [0, l]$ such that $\mathfrak{S} \models^{\alpha} \chi$, for all $\mathfrak{S} \in \text{Ext}(\mathfrak{F}^n)$, and, for every $i \in [0, n)$, $\mathfrak{F}^i \models^{\alpha} \psi$. Since, for every $\mathfrak{S} \in \text{Ext}(\mathfrak{F})$ and $m \in (0, l]$, we have that $\mathfrak{S} = \mathfrak{F}_{m-1} \cdot \mathfrak{S}'$, for some $\mathfrak{S}' \in \text{Ext}(\mathfrak{F}^m)$, the previous step implies that, for all $\mathfrak{S} \in \text{Ext}(\mathfrak{F})$, there exists $n \in [0, l]$ such that $\mathfrak{S} \models^{\alpha} \chi$, and, for every $i \in [0, n)$, $\mathfrak{S} \models^{\alpha} \psi$. That is, $\mathfrak{S} \models^{\alpha} \psi \mathcal{U}^+ \chi$.
- $\varphi = \forall x \psi$. Suppose that $\mathfrak{F} \models^{\alpha} \forall x \psi$. This means that, for all $d \in \Delta$, $\mathfrak{F} \models^{\alpha[x \mapsto d]} \psi$. By the inductive hypothesis, we have that, for all $d \in \Delta$ and all $\mathfrak{S} \in \text{Ext}(\mathfrak{F})$, $\mathfrak{S} \models^{\alpha[x \mapsto d]} \psi$. Thus, for all $\mathfrak{S} \in \text{Ext}(\mathfrak{F})$, $\mathfrak{S} \models^{\alpha} \forall x \psi$.

- $\varphi = \exists x\psi$. Suppose that $\mathfrak{F} \models^a \exists x\psi$. This means that there is $d \in \Delta$ such that $\mathfrak{F} \models^{a[x \mapsto d]} \psi$. By the inductive hypothesis, we have that, for all $\mathfrak{S} \in \text{Ext}(\mathfrak{F})$, $\mathfrak{S} \models^{a[x \mapsto d]} \psi$. Thus, for all $\mathfrak{S} \in \text{Ext}(\mathfrak{F})$, $\mathfrak{S} \models^a \exists x\psi$.
- The other cases can be proved in a straightforward way using the inductive hypothesis. \square

CLAIM 2. $\mathcal{U}^+\forall$ -formulas are $F_{\Leftarrow\forall}$.

PROOF OF CLAIM 2. We show the following (stronger) claim: for every finite trace $\mathfrak{F} = (\Delta, (\mathcal{F}_n)_{n \in [0, l]})$, and every a in \mathfrak{F} , $\mathfrak{F}^\omega \models^a \varphi$ implies $\mathfrak{F} \models^a \varphi$. The proof is by structural induction on φ . The proof for the base cases $\varphi = P(\bar{\tau})$ and $\varphi = \neg P(\bar{\tau})$ is straightforward. We now proceed with the inductive steps.

- $\varphi = \psi \mathcal{U}^+ \chi$. If $\mathfrak{F}^\omega \models^a \psi \mathcal{U}^+ \chi$, then there is $n \geq 0$ such that $\mathfrak{F}^\omega, n \models^a \chi$ and, for every $i \in [0, n)$, $\mathfrak{F}^\omega, i \models^a \psi$. This means that there is $n \geq 0$ such that $(\mathfrak{F}^\omega)^n \models^a \chi$ and, for every $i \in [0, n)$, $(\mathfrak{F}^\omega)^i \models^a \psi$. If $n > l$, then $(\mathfrak{F}^\omega)^n = (\mathfrak{F}^l)^\omega$. Hence, without loss of generality, we can assume that $n \leq l$, for which it holds by definition that $(\mathfrak{F}^\omega)^n = (\mathfrak{F}^n)^\omega$. Thus, by the inductive hypothesis, we obtain $\mathfrak{F}^n \models^a \chi$, and, for every $i \in [0, n)$, $\mathfrak{F}^i \models^a \psi$, meaning that $\mathfrak{F} \models^a \psi \mathcal{U}^+ \chi$. Hence, $(\mathfrak{F}^l)^\omega \models^a \chi$ and, for every $i \in [0, l)$, $(\mathfrak{F}^l)^\omega \models^a \psi$. By the inductive hypothesis, we obtain that $\mathfrak{F}^l \models^a \chi$ and, for every $i \in [0, l)$, $\mathfrak{F}^i \models^a \psi$, again implying that $\mathfrak{F} \models^a \psi \mathcal{U}^+ \chi$.
- $\varphi = \forall x\psi$. If $\mathfrak{F}^\omega \models^a \forall x\psi$, then for all $d \in \Delta$, $\mathfrak{F}^\omega \models^{a[x \mapsto d]} \psi$. By the inductive hypothesis, for all $d \in \Delta$, $\mathfrak{F} \models^{a[x \mapsto d]} \psi$. So, $\mathfrak{F} \models^a \forall x\psi$.
- $\varphi = \exists x\psi$. If $\mathfrak{F}^\omega \models^a \exists x\psi$, then there is $d \in \Delta$ such that $\mathfrak{F}^\omega \models^{a[x \mapsto d]} \psi$. By the inductive hypothesis, for some $d \in \Delta$, $\mathfrak{F} \models^{a[x \mapsto d]} \psi$. Thus, $\mathfrak{F} \models^a \exists x\psi$.
- The remaining cases follow by a straightforward application of the inductive hypothesis. \square

We now show the second part of Lemma 4.11, i.e., that \mathcal{U} -formulas are I_{\exists} . In Claim 4, we show that \mathcal{U} -formulas are $I_{\Rightarrow\exists}$. Then, in Claim 5, we show that \mathcal{U} -formulas are $I_{\Leftarrow\exists}$. Before proving Claim 4, we show the following statement.

CLAIM 3. Let φ be an \mathcal{U} -formula. For every finite trace $\mathfrak{F} = (\Delta, (\mathcal{F}_n)_{n \in [0, l]})$, every prefix $\mathfrak{F}' = (\Delta, (\mathcal{F}'_n)_{n \in [0, l']}) \in \text{Pre}(\mathfrak{F})$, and every assignment a , $\mathfrak{F}' \models^a \varphi$ implies $\mathfrak{F} \models^a \varphi$.

PROOF OF CLAIM 3. The proof is by structural induction on φ . Clearly, the statement holds for the base cases of $\varphi = P(\bar{\tau})$ and $\varphi = \neg P(\bar{\tau})$. We now proceed with the inductive cases.

- $\varphi = \psi \mathcal{U} \chi$. If $\mathfrak{F}' \models^a \psi \mathcal{U} \chi$, then there is $n \in (0, l']$ such that $\mathfrak{F}', n \models^a \varphi$ and, for every $i \in (0, n)$, $\mathfrak{F}', i \models^a \psi$. That is, for some $n \in (0, l']$, $\mathfrak{F}'^n \models^a \chi$ and, for every $i \in (0, n)$, $\mathfrak{F}'^i \models^a \psi$. As \mathfrak{F}'^m is a prefix of \mathfrak{F}'^n , for every $m \in [0, l']$, and since $l' \leq l$, we have by the induction hypothesis that there exists $n \in (0, l]$ such that $\mathfrak{F}^n \models^a \chi$ and, for every $i \in (0, n)$, $\mathfrak{F}^i \models^a \psi$. Equivalently, for some $n \in (0, l]$, $\mathfrak{F}, n \models^a \chi$ and, for every $i \in (0, n)$, $\mathfrak{F}, i \models^a \psi$. Thus, $\mathfrak{F} \models^a \psi \mathcal{U} \chi$.
- The other cases can be proved by straightforward applications of the inductive hypothesis. \square

We can now proceed with the following claim.

CLAIM 4. \mathcal{U} -formulas are $I_{\Rightarrow\exists}$.

PROOF OF CLAIM 4. Given an infinite trace $\mathfrak{S} = (\Delta, (\mathcal{J}_n)_{n \in [0, \infty)})$ and an assignment a in \mathfrak{S} , we show that $\mathfrak{S} \models^a \varphi$ implies that there exists $\mathfrak{F} \in \text{Pre}(\mathfrak{S})$ such that $\mathfrak{F} \models^a \varphi$. The proof is by structural induction on φ . By Proposition 4.9, the statement holds for the base cases of $\varphi = P(\bar{\tau})$ and $\varphi = \neg P(\bar{\tau})$. We now proceed with the inductive steps.

- $\varphi = \psi \mathcal{U} \chi$. Suppose that $\mathfrak{S} \models^a \psi \mathcal{U} \chi$, meaning that there exists $n > 0$ such that $\mathfrak{S}, n \models^a \chi$ and, for every $i \in (0, n)$, $\mathfrak{S}, i \models^a \chi$. In other words, there exists $n > 0$ such that $\mathfrak{S}^n \models^a \chi$ and, for every $i \in (0, n)$, $\mathfrak{S}^i \models^a \psi$. By the inductive hypothesis, the previous step implies that there exists $n > 0$ such that $\mathfrak{S}_{n_j}^n \models^a \chi$, for some $n_j \geq n$, and, for every $i \in (0, n)$, $\mathfrak{S}_{i_k}^i \models^a \psi$, for some $i_k \geq i$. For such an $n > 0$, let $\bar{n}_j = \min\{n_j \mid \mathfrak{S}_{n_j}^n \models^a \chi\}$ and, for every $i \in (0, n)$, let $\bar{i}_k = \min\{i_k \mid \mathfrak{S}_{i_k}^i \models^a \psi\}$. In addition, let m be the maximum among \bar{n}_j and \bar{i}_k , for $i \in (0, n)$. We have that $\mathfrak{S}_{n_j}^n \in \text{Pre}(\mathfrak{S}_m^n)$, and $\mathfrak{S}_{i_k}^i \in \text{Pre}(\mathfrak{S}_m^i)$, for every $i \in (0, n)$. Since $\mathfrak{S}_{n_j}^n \models^a \chi$ and $\mathfrak{S}_{i_k}^i \models^a \psi$, by Claim 3 we obtain that, for some $n > 0$, $\mathfrak{S}_m^n \models^a \chi$ and $\mathfrak{S}_m^i \models^a \psi$, for every $i \in (0, n)$. In conclusion, there exists $\mathfrak{F} = \mathfrak{S}_m \in \text{Pre}(\mathfrak{S})$ such that $\mathfrak{F} \models^a \psi \mathcal{U} \chi$.
- $\varphi = \exists x\psi$. Suppose that $\mathfrak{S} \models^a \exists x\psi$. This means that there is $d \in \Delta$ such that $\mathfrak{S} \models^{a[x \mapsto d]} \psi$. By the inductive hypothesis, there is $\mathfrak{F} \in \text{Pre}(\mathfrak{S})$ such that $\mathfrak{F} \models^{a[x \mapsto d]} \psi$, for some $d \in \Delta$. So, $\mathfrak{F} \models^a \exists x\psi$.
- $\varphi = \psi \wedge \chi$. Suppose that $\mathfrak{S} \models^a \psi \wedge \chi$. This means that $\mathfrak{S} \models^a \psi$ and $\mathfrak{S} \models^a \chi$. By the inductive hypothesis, there are $\mathfrak{F}, \mathfrak{F}' \in \text{Pre}(\mathfrak{S})$ such that $\mathfrak{F} \models^a \psi$ and $\mathfrak{F}' \models^a \chi$. By definition of \mathfrak{F} and \mathfrak{F}' , either \mathfrak{F}' is a prefix of \mathfrak{F} or vice versa. Assume without loss of generality that \mathfrak{F}' is a prefix of \mathfrak{F} . By Claim 3, if $\mathfrak{F}' \models^a \chi$, then $\mathfrak{F} \models^a \chi$. Then, $\mathfrak{F} \models^a \psi$ and $\mathfrak{F} \models^a \chi$, i.e., $\mathfrak{F} \models^a \psi \wedge \chi$.
- The remaining cases follow by a straightforward application of the inductive hypothesis. \square

We now conclude the proof of Lemma 4.11 by showing the following claim.

CLAIM 5. \mathcal{U} -formulas are $\mathbb{I}_{\Leftarrow \exists}$.

PROOF OF CLAIM 5. Given an infinite trace $\mathfrak{S} = (\Delta, (\mathcal{I}_n)_{n \in [0, \infty)})$ and an assignment a , we show that $\mathfrak{F} \models^a \varphi$, for some $\mathfrak{F} \in \text{Pre}(\mathfrak{S})$, implies $\mathfrak{S} \models^a \varphi$. The proof is by structural induction on φ . By Proposition 4.9, the statement holds for the base cases of $\varphi = P(\bar{\tau})$ and $\varphi = \neg P(\bar{\tau})$. We now proceed with the inductive steps.

- $\varphi = \psi \mathcal{U} \chi$. Suppose that there is $\mathfrak{F} = (\Delta, (\mathcal{F}_n)_{n \in [0, l]}) \in \text{Pre}(\mathfrak{S})$ such that $\mathfrak{F} \models^a \psi \mathcal{U} \chi$. This means that there exists $n \in (0, l]$ such that $\mathfrak{F}, n \models^a \chi$ and, for every $i \in (0, n)$, $\mathfrak{F}, i \models^a \psi$. In other words, there exists $n \in (0, l]$ such that $\mathfrak{F}^n \models^a \chi$ and, for every $i \in (0, n)$, $\mathfrak{F}^i \models^a \psi$. By the inductive hypothesis, the previous step implies that, for some $n \in (0, l]$, $\mathfrak{S}^n \models^a \chi$ and, for every $i \in (0, n)$, $\mathfrak{S}^i \models^a \psi$. Thus, there exists $n > 0$ such that $\mathfrak{S}, n \models^a \chi$ and, for every $i \in (0, n)$, $\mathfrak{S}, i \models^a \psi$, meaning that $\mathfrak{S} \models^a \psi \mathcal{U} \chi$.
- $\varphi = \exists x\psi$. Suppose that there is $\mathfrak{F} \in \text{Pre}(\mathfrak{S})$ such that $\mathfrak{F} \models^a \exists x\psi$. This means that there is $d \in \Delta$ such that $\mathfrak{F} \models^{a[x \mapsto d]} \psi$. By the inductive hypothesis, we obtain $\mathfrak{S} \models^{a[x \mapsto d]} \psi$, for some $d \in \Delta$. Hence, $\mathfrak{S} \models^a \exists x\psi$.
- The remaining cases follow by a straightforward application of the inductive hypothesis. \square

\square

The results of Lemma 4.11 are tight in the sense that we cannot extend the grammar rule for \mathcal{U} -formulas (and not even for \mathcal{U}^+ -formulas) with $\forall x\varphi$ while still satisfying \mathbb{I}_{\exists} , and we cannot extend the grammar rule for $\mathcal{U}^+\forall$ -formulas with $\varphi \mathcal{U} \psi$ and satisfy F_{\forall} . Simple counterexamples are $\forall x \diamond^+ Q(x)$ and $\diamond \top$, which are not \mathbb{I}_{\exists} and F_{\forall} , respectively. To see that $\forall x \diamond^+ Q(x)$ is not $\mathbb{I}_{\Rightarrow \exists}$, and thus not \mathbb{I}_{\exists} , consider the model given in the proof of Proposition 4.5 above: an infinite trace $\mathfrak{S} = (\{d_i\}_{i \in \mathbb{N}}, (\mathcal{I}_n)_{n \in [0, \infty)})$ where $Q^{d_i} = \{d_i\}$, for every $i \in \mathbb{N}$, that satisfies the formula, but with no finite prefix $\mathfrak{F} \in \text{Pre}(\mathfrak{S})$ that satisfies it. On the other hand, $\diamond \top$ holds in any infinite trace, but not on a finite trace with only one time point. Thus, $\diamond \top$ is not $F_{\Leftarrow \forall}$, and hence not F_{\forall} .

We now move to the case of $\mathcal{R}^+\exists$ -, \mathcal{R} -, and \mathcal{R}^+ -formulas, by proving a result similar to Lemma 4.11. This can be shown by exploiting the duality between \mathcal{U} and \mathcal{R} operators, together with the results obtained in Proposition 4.6, Point (1), and Lemma 4.11 above.

LEMMA 4.12. *$\mathcal{R}^+\exists$ -formulas are F_{\exists} and \mathcal{R} -formulas are I_{\forall} . Thus, \mathcal{R}^+ -formulas are both F_{\exists} and I_{\forall} .*

PROOF. Observe that, by definition, any $\mathcal{R}^+\exists$ - and \mathcal{R} -formula is equivalent to the negation of an $\mathcal{U}^+\forall$ - and \mathcal{U} -formula, respectively. Therefore, by Proposition 4.6, Point (1), we obtain that:

- $\mathcal{R}^+\exists$ -formulas are $F_{\Leftarrow\exists}$, by Lemma 4.11, Claim 1;
- $\mathcal{R}^+\exists$ -formulas are $F_{\Rightarrow\exists}$, by Lemma 4.11, Claim 2;

as well as that:

- \mathcal{R} -formulas are $I_{\Leftarrow\forall}$, by Lemma 4.11, Claim 4;
- \mathcal{R} -formulas are $I_{\Rightarrow\forall}$, by Lemma 4.11, Claim 5.

That is, $\mathcal{R}^+\exists$ -formulas are F_{\exists} and \mathcal{R} -formulas are I_{\forall} , hence \mathcal{R}^+ -formulas are both F_{\exists} and I_{\forall} . \square

The results of Lemma 4.12 are also tight in the sense that we cannot extend the grammar rule for \mathcal{R} -formulas (and not even for \mathcal{R}^+ -formulas) with $\exists x\phi$, while still satisfying I_{\forall} , and we cannot extend the grammar rule for $\mathcal{R}^+\exists$ -formulas with $\phi\mathcal{R}\psi$, while satisfying F_{\exists} . Simple counterexamples are $\exists x\Box^+\neg Q(x)$ and *last* := $\Box\perp$, which are not I_{\forall} and F_{\exists} , respectively. To see that $\exists x\Box^+\neg Q(x)$ is not $I_{\Leftarrow\forall}$, and thus not I_{\forall} , consider again the model described above with an infinite (and countable) domain, where each element is in the extension of Q at a specific time point $n \in \mathbb{N}$. The formula $\exists x\Box^+\neg Q(x)$ holds in every finite prefix but it does not hold in this infinite trace. Thus, it is not I_{\forall} . On the other hand, clearly, *last* holds in a finite trace \mathfrak{F} with only one time point but it does not on any extension of \mathfrak{F} . Therefore, it is not $F_{\Rightarrow\exists}$, and thus not F_{\exists} .

Finally, we comment on the results of Theorem 4.10. We observe that $\Diamond\top$ and \top are examples of \mathcal{U} -formulas that are equivalent on infinite, but not on finite, traces. Similarly, $\Box\perp$ and \perp are \mathcal{R} -formulas equivalent on infinite traces only. Thus, the converse of Point (3) of Theorem 4.10 does not hold for such sets of formulas. However, we leave as an open problem to determine whether the converse of Point (2) in Theorem 4.10 holds for $\mathcal{U}^+\forall$ - and $\mathcal{R}^+\exists$ -formulas. We conjecture that, for these fragments, which are in negation normal form and allow for only one kind of reflexive temporal operator (i.e., either \mathcal{U}^+ or \mathcal{R}^+), the set of equivalent formulas on finite and infinite traces coincide. Finally, as stated in Point (1) of Theorem 4.10, we remark that there is no distinction between reasoning on finite and infinite traces whenever a formula is either an \mathcal{U}^+ - or a \mathcal{R}^+ -formula. As already pointed out, however, $\Diamond^+\Box^+P$ and $\Box^+\Diamond^+P$ are only equivalent on finite traces, and so, when considering formula equivalences, the distinction between finite and infinite traces cannot be blurred for the class of formulas that allow both \mathcal{U}^+ and \mathcal{R}^+ .

4.2.3 Preserving Formula Satisfiability: From Finite to Infinite Traces. In this section, we consider the problem of preserving satisfiability of a $T_{\mathcal{U}}\mathcal{QL}$ formula ϕ from finite to infinite traces, i.e., under which conditions, knowing that ϕ is finitely satisfiable, we can conclude that ϕ is also satisfiable on infinite traces. Identifying classes of formulas for which this question can be positively answered is of interest also to develop more efficient automated reasoners. Indeed, under certain conditions which guarantee that satisfiability of a formula on finite traces implies its satisfiability on infinite ones, solvers can simply stop trying to build the lasso of an infinite trace, once a finite trace satisfying the formula is found.

In order to connect this problem with the results obtained in the previous sections, we make the following observations. First, in Theorem 4.3, we have seen that $T_{\mathcal{U}}\mathcal{QL}$ formulas interpreted on finite traces can be translated into equisatisfiable formulas on infinite traces. However, such translation is not always needed, since for some classes of formulas satisfiability is already preserved.

For instance, given $\varphi \in T_{\mathcal{U}}\mathcal{QL}(F_{\Rightarrow\exists})$, we clearly have that, if φ is satisfiable on finite traces, then it is satisfiable on infinite traces. Moreover, the problem of preserving satisfiability from finite to infinite traces can be seen as a special case of the problem of preserving formula equivalences from infinite to finite ones, where we are only interested in determining if a $T_{\mathcal{U}}\mathcal{QL}$ formula φ that is equivalent to \perp on infinite traces (i.e., unsatisfiable on infinite traces) is also unsatisfiable on finite traces. This is not the case in general. For instance, *last*, which is equivalent to \perp on infinite traces but satisfiable on finite traces, is a formula for which satisfiability is not preserved from finite to infinite traces. Instead, from Theorem 4.10, we obtain in particular that, for every $\mathcal{U}^+\forall$ - or $\mathcal{R}^+\exists$ -formulas satisfiability is preserved from finite to infinite traces.

However, the results of the previous section do not allow us to determine classes of formulas that involve both operators \mathcal{U}^+ and \mathcal{R}^+ , and for which satisfiability from finite to infinite traces is preserved. Formulas like $\diamond^+\square^+P$ and $\square^+\diamond^+P$, for instance, are such that their satisfiability is preserved from finite to infinite traces, but they do not fall in any of the fragments identified above. Our aim in the rest of this section is to show that indeed satisfiability from finite to infinite traces is preserved for a larger class of formulas, introduced in the following.

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

$$P(\bar{t}) \mid \neg P(\bar{t}) \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \exists x\varphi \mid \forall x\varphi \mid \varphi \mathcal{U}^+\psi \mid \varphi \mathcal{R}^+\psi.$$

It can be seen that the set of $\mathcal{U}^+\mathcal{R}^+$ -formulas is just a syntactic variant (in negation normal form) of the fragment $T_{\mathcal{U}^+\mathcal{R}^+}\mathcal{QL}$ of $T_{\mathcal{U}}\mathcal{QL}$, i.e., the fragment allowing only for \mathcal{U}^+ and \mathcal{R}^+ as temporal operators. A typical example of an $\mathcal{U}^+\mathcal{R}^+$ -formula, used to express properties in the context of specification and verification of reactive systems, is $\square^+\forall x(P(x) \rightarrow \diamond^+Q(x))$ [54].

We show in the following that the language generated by the grammar rule for $\mathcal{U}^+\mathcal{R}^+$ -formulas contains only formulas whose satisfiability on finite traces implies satisfiability on infinite traces. This result, formalised by the following theorem, is an immediate consequence of Lemma 4.14 below.

THEOREM 4.13. *All $\mathcal{U}^+\mathcal{R}^+$ -formulas satisfiable on finite traces are satisfiable on infinite traces.*

The converse of Theorem 4.13, however, does not hold, as illustrated by the next example. Consider the $\mathcal{U}^+\mathcal{R}^+$ -formula

$$\begin{aligned} & \square^+\forall x((P(x) \wedge \neg Q(x)) \vee (Q(x) \wedge \neg P(x))) \wedge \\ & \square^+\forall x((P(x) \rightarrow \diamond^+Q(x)) \wedge (Q(x) \rightarrow \diamond^+P(x))). \end{aligned} \quad (3)$$

We have that (3) is satisfiable on infinite traces only, since it requires $P(x)$ and $Q(x)$ to alternate infinitely often. Therefore, for $\mathcal{U}^+\mathcal{R}^+$ -formulas, satisfiability on infinite traces does not imply satisfiability on finite traces.

In order to prove Theorem 4.13, we introduce the following preliminary notion. A $T_{\mathcal{U}}\mathcal{QL}$ formula φ is F_ω iff, for all finite traces \mathfrak{F} and all assignments α , it satisfies the *frozen trace property*:

$$\mathfrak{F} \models^\alpha \varphi \Leftrightarrow \mathfrak{F}^\omega \models^\alpha \varphi.$$

We denote by $T_{\mathcal{U}}\mathcal{QL}(F_\omega)$ the set of $T_{\mathcal{U}}\mathcal{QL}$ formulas that are F_ω . Clearly, if $\varphi \in T_{\mathcal{U}}\mathcal{QL}(F_\omega)$ is satisfiable on finite traces, then φ is satisfiable on infinite traces. Thus, Theorem 4.13 above is an immediate consequence of the following lemma.

LEMMA 4.14. *$\mathcal{U}^+\mathcal{R}^+$ -formulas are F_ω .*

PROOF. We write $F_{\Rightarrow\omega}$ and $F_{\Leftarrow\omega}$ for the “one directional” version of F_ω . In Claim 6 we show that all $\mathcal{U}^+\mathcal{R}^+$ -formulas are $F_{\Rightarrow\omega}$. Then, in Claim 7, we show that all $\mathcal{U}^+\mathcal{R}^+$ -formulas are $F_{\Leftarrow\omega}$.

CLAIM 6. *$\mathcal{U}^+\mathcal{R}^+$ -formulas are $F_{\Rightarrow\omega}$.*

PROOF OF CLAIM 6. We show, by structural induction on φ , that $\mathfrak{F} \models^a \varphi$ implies $\mathfrak{F}^\omega \models^a \varphi$, for any finite trace $\mathfrak{F} = (\Delta, (\mathcal{F}_n)_{n \in [0, l]})$ and any assignment a . The proof for the base cases $\varphi = P(\bar{\tau})$ and $\varphi = \neg P(\bar{\tau})$ is straightforward. We now proceed with the inductive cases.

- $\varphi = \psi \mathcal{U}^+ \chi$. Suppose that $\mathfrak{F} \models^a \psi \mathcal{U}^+ \chi$. Then, there exists $n \in [0, l]$ such that $\mathfrak{F}, n \models^a \chi$ and, for every $i \in [0, n)$, $\mathfrak{F}, i \models^a \psi$. In other words, there exists $n \in [0, l]$ such that $\mathfrak{F}^n \models^a \chi$ and, for every $i \in [0, n)$, $\mathfrak{F}^i \models^a \psi$. By the inductive hypothesis, the previous step implies that there exists $n \in [0, l]$ such that $(\mathfrak{F}^n)^\omega \models^a \chi$ and, for every $i \in [0, n)$, $(\mathfrak{F}^i)^\omega \models^a \psi$. Since, for every $m \in [0, l]$, we have that $(\mathfrak{F}^m)^\omega = (\mathfrak{F}^\omega)^m$, the previous step implies that there exists $n \geq 0$ such that $(\mathfrak{F}^\omega)^n \models^a \chi$ and, for every $i \in [0, n)$, $(\mathfrak{F}^\omega)^i \models^a \psi$. In other words, there exists $n \geq 0$ such that $\mathfrak{F}^\omega, n \models^a \chi$ and, for every $i \in [0, n)$, $\mathfrak{F}^\omega, i \models^a \psi$, i.e., $\mathfrak{F}^\omega \models^a \psi \mathcal{U}^+ \chi$.
- $\varphi = \psi \mathcal{R}^+ \chi$. Suppose that $\mathfrak{F} \models^a \psi \mathcal{R}^+ \chi$. This means that, for every $n \in [0, l]$, we have $\mathfrak{F}, n \models^a \chi$, or there exists $i \in [0, n)$ such that $\mathfrak{F}, i \models^a \psi$. That is, for every $n \in [0, l]$, it holds that $\mathfrak{F}^n \models^a \chi$, or there exists $i \in [0, n)$ such that $\mathfrak{F}^i \models^a \psi$. By the inductive hypothesis, this implies that, for every $n \in [0, l]$, we have $(\mathfrak{F}^n)^\omega \models^a \chi$, or there exists $i \in [0, n)$ such that $(\mathfrak{F}^i)^\omega \models^a \psi$. For every $m \in [0, l]$, it holds that $(\mathfrak{F}^m)^\omega = (\mathfrak{F}^\omega)^m$, thus we obtain, for every $n \in [0, l]$, that $(\mathfrak{F}^\omega)^n \models^a \chi$, or there exists $i \in [0, n)$ such that $(\mathfrak{F}^\omega)^i \models^a \psi$. Moreover, since $(\mathfrak{F}^\omega)^i = (\mathfrak{F}^\omega)^m$, for every $m > l$, we have that $(\mathfrak{F}^\omega)^m \models^a \chi$, or there exists $i \in [0, m)$ such that $(\mathfrak{F}^\omega)^i \models^a \psi$. In conclusion, for every $n \geq 0$, $\mathfrak{F}^\omega, n \models^a \chi$, or there exists $i \in [0, n)$ such that $\mathfrak{F}^\omega, i \models^a \psi$. Hence, $\mathfrak{F}^\omega \models^a \psi \mathcal{R}^+ \chi$.
- $\varphi = \forall x \psi$. Suppose that $\mathfrak{F} \models^a \forall x \psi$. This means that, for all $d \in \Delta$, $\mathfrak{F} \models^{a[x \mapsto d]} \psi$. By the inductive hypothesis, for all $d \in \Delta$, $\mathfrak{F}^\omega \models^{a[x \mapsto d]} \psi$. Thus, $\mathfrak{F}^\omega \models^a \forall x \psi$.
- $\varphi = \exists x \psi$. Suppose that $\mathfrak{F} \models^a \exists x \psi$. This means that there is $d \in \Delta$ such that $\mathfrak{F} \models^{a[x \mapsto d]} \psi$. By the inductive hypothesis, we have $\mathfrak{F}^\omega \models^{a[x \mapsto d]} \psi$, for some $d \in \Delta$. That is, $\mathfrak{F}^\omega \models^a \exists x \psi$.
- The other cases can be proved in a straightforward way using the inductive hypothesis. \square

CLAIM 7. $\mathcal{U}^+ \mathcal{R}^+$ -formulas are $F_{\Leftarrow \omega}$.

PROOF OF CLAIM 7. We show, by structural induction on φ , that $\mathfrak{F}^\omega \models^a \varphi$ implies $\mathfrak{F} \models^a \varphi$, for any finite trace $\mathfrak{F} = (\Delta, (\mathcal{F}_n)_{n \in [0, l]})$ and any assignment a . The base cases of $\varphi = P(\bar{\tau})$ and $\varphi = \neg P(\bar{\tau})$, as well as the inductive cases of $\varphi = \psi \wedge \chi$, $\varphi = \psi \vee \chi$, $\varphi = \exists x \psi$, $\varphi = \forall x \psi$, and $\varphi = \psi \mathcal{U}^+ \chi$, can be shown as in the proof of Claim 2. We now show the remaining inductive case.

- $\varphi = \psi \mathcal{R}^+ \chi$. Suppose that $\mathfrak{F}^\omega \models^a \psi \mathcal{R}^+ \chi$, then, for every $n \geq 0$, we have $\mathfrak{F}^\omega, n \models^a \chi$ or there exists $i \in [0, n)$ such that $\mathfrak{F}^\omega, i \models^a \psi$. Thus, in particular, for every $n \in [0, l]$, $(\mathfrak{F}^\omega)^n \models^a \chi$, or there exists $i \in [0, n)$ such that $(\mathfrak{F}^\omega)^i \models^a \psi$. Since, for every $m \in [0, l]$, we have that $(\mathfrak{F}^\omega)^m = (\mathfrak{F}^m)^\omega$, the previous step is equivalent to: for every $n \in [0, l]$, $(\mathfrak{F}^n)^\omega \models^a \chi$, or there exists $i \in [0, n)$ such that $(\mathfrak{F}^i)^\omega \models^a \psi$. By the inductive hypothesis, we obtain that, for every $n \in [0, l]$, $\mathfrak{F}^n \models^a \chi$, or there exists $i \in [0, n)$ such that $\mathfrak{F}^i \models^a \psi$. In other words, for every $n \in [0, l]$, $\mathfrak{F}, n \models^a \chi$, or there exists $i \in [0, n)$ such that $\mathfrak{F}, i \models^a \psi$, i.e., $\mathfrak{F} \models^a \psi \mathcal{R}^+ \chi$. \square

\square

5 COMPLEXITY OF DECIDABLE FRAGMENTS ON FINITE AND k -BOUNDED TRACES

In this section, we study the complexity of the satisfiability problem for formulas taken from well-known decidable fragments of first-order temporal logic, ranging from the constant-free one-variable monadic, to the monadic monodic, and the two-variable monodic, fragments (as introduced in Section 3.1). When referring to the *size* of a formula, $|\varphi|$, we consider the number of symbols used to write the formula.

First, we consider satisfiability on arbitrary finite traces, showing that the complexity does not change compared to the infinite case, i.e., it remains EXPSPACE-complete. Then, we analyse the

case of satisfiability on k -bounded traces, proving that the complexity lowers down to NEXPTIME -complete. Finally, we show that these fragments interpreted on finite traces enjoy both the bounded trace and the bounded domain properties, that is, they are satisfiable on finite traces iff they are satisfied on finite traces with a bounded number of time points, and of elements in the domain, respectively, with a bound that depends on the size of the formula. We conclude the section with an excursus on temporal DLs, by investigating the complexity of the satisfiability problem in the temporal extension of the DL \mathcal{ALC} .

5.1 Complexity Results on Finite Traces

We analyse the complexity of decidable fragments of first-order temporal logic on finite traces. To start with, we show that EXPSpace -hardness holds already for the constant-free one-variable monadic fragment $T_{\mathcal{U}}\mathcal{QL}_{\mathcal{d}}^{1,mo}$. This fragment can be considered as a notational variant of the propositional language of the two-dimensional product $\text{LTL}^f \times \text{S5}$, defined similarly to the product $\text{LTL} \times \text{S5}$ [54], where LTL^f denotes LTL interpreted on finite traces. In particular, the S5 -modality is replaced by the universal quantifier $\forall x$, and propositional letters p are substituted by unary predicates $P(x)$, with free variable x . The lower bound can be proved by applying similar ideas as those used to show hardness of $\text{LTL} \times \text{S5}$ satisfiability.

PROPOSITION 5.1. $T_{\mathcal{U}}\mathcal{QL}_{\mathcal{d}}^{1,mo}$ formula satisfiability on finite traces is EXPSpace -hard.

PROOF. The proof is an adaptation of [54, Theorem 5.43] to the case of $T_{\mathcal{U}}\mathcal{QL}_{\mathcal{d}}^{1,mo}$ on finite traces. A *tile type* is a 4-tuple $t = (\text{up}(t), \text{down}(t), \text{left}(t), \text{right}(t))$ of *colours* (from a set that we assume to include the colour *white*). Let \mathbb{T} be a finite set of tile types, with $t_0, t_1 \in \mathbb{T}$, and let $n \in \mathbb{N}$, given in binary. The $m \times 2^n$ *corridor tiling problem* is the problem of deciding whether there exist $m \in \mathbb{N}$ and a function, called *tiling*, $\tau: m \times 2^n \rightarrow \mathbb{T}$ (where $m \times 2^n$ denotes the set of all pairs $(i, j) \in \mathbb{N} \times \mathbb{N}$ with $0 \leq i < m$ and $0 \leq j < 2^n$) such that:

- $\tau(0, 0) = t_0, \tau(m - 1, 0) = t_1$;
- $\text{up}(\tau(i, j)) = \text{down}(\tau(i, j + 1))$, for $0 \leq i < m, 0 \leq j < 2^n - 1$, and $\text{right}(\tau(i, j)) = \text{left}(\tau(i + 1, j))$, for $0 \leq i < m - 1, 0 \leq j < 2^n$;
- $\text{down}(\tau(i, 0)) = \text{up}(\tau(i, 2^n - 1)) = \text{white}$, for $0 \leq i < m$.

The $m \times 2^n$ corridor tiling problem is known to be EXPSpace -complete [97]. In the following, we will reduce this problem to $T_{\mathcal{U}}\mathcal{QL}_{\mathcal{d}}^{1,mo}$ formula satisfiability on finite traces.

Given a finite set of tile types \mathbb{T} , with $t_0, t_1 \in \mathbb{T}$, and an $n \in \mathbb{N}$, our aim is to construct a $T_{\mathcal{U}}\mathcal{QL}_{\mathcal{d}}^{1,mo}$ formula $\varphi_{n, \mathbb{T}}$ such that: (i) the length of $\varphi_{n, \mathbb{T}}$ is polynomial in n and $|\mathbb{T}|$; (ii) $\varphi_{n, \mathbb{T}}$ is satisfiable on finite traces iff there exist $m \in \mathbb{N}$ and a function $\tau: m \times 2^n \rightarrow \mathbb{T}$ tiling the $m \times 2^n$ corridor (as described by the conditions above).

We start by taking n distinct unary predicates S_0, \dots, S_{n-1} , and let $S_i^0 = \neg S_i$ and $S_i^1 = S_i$, for $0 \leq i \leq n - 1$. Then, we define a binary counter, called σ -counter, up to 2^n by setting

$$\sigma_j(x) = S_{n-1}^{b_{n-1}}(x) \wedge \dots \wedge S_0^{b_0}(x),$$

where b_i is the i th bit in the binary representation of $0 \leq j \leq 2^n - 1$. Moreover, we require

$$\square^+ \bigwedge_{0 \leq i \leq n-1} (\forall x S_i(x) \vee \forall x \neg S_i(x)), \quad (4)$$

so that, at each time point, the σ -counter value will be the same for every element of the domain. The following formula will be used to set the value of the σ -counter to 0 at the first instant of a trace, and to increase its value by one at each subsequent instant (if any). Once the σ -counter

reaches the value of $2^n - 1$, it goes back to 0 at the following time point (if any).

$$\begin{aligned} & \sigma_0(x) \wedge \\ & \square^+ \bigwedge_{0 \leq k < n} \left(\left(\bigwedge_{0 \leq i < k} S_i(x) \wedge \neg S_k(x) \right) \rightarrow \left(\bigwedge_{k < j < n} (S_j(x) \leftrightarrow \circ S_j(x)) \right) \wedge \circ \left(\bigwedge_{0 \leq i < k} \neg S_i(x) \wedge S_k(x) \right) \right) \wedge \\ & \square^+ \left(\bigwedge_{0 \leq i < n} S_i(x) \rightarrow \bullet \left(\bigwedge_{0 \leq i < n} \neg S_i(x) \right) \right). \end{aligned} \quad (5)$$

Now consider n fresh unary predicates P_0, \dots, P_{n-1} . By setting

$$\square^+ \forall x \bigwedge_{0 \leq i < n} (\neg \text{last} \rightarrow (P_i(x) \leftrightarrow \circ P_i(x))), \quad (6)$$

we force their extension to be fixed along the temporal dimension. Then, we set

$$\pi_j(x) = P_{n-1}^{b_{n-1}}(x) \wedge \dots \wedge P_0^{b_0}(x),$$

where again $b_{n-1} \dots b_0$ is the binary representation of $0 \leq j \leq 2^n - 1$, while $P_i^0 = \neg P_i$ and $P_i^1 = P_i$, for $0 \leq i \leq n - 1$.

Moreover, we define the formulas

$$\begin{aligned} \text{equ}(x) &= \bigwedge_{0 \leq i < n} (P_i(x) \leftrightarrow S_i(x)), \\ \text{mark}(x) &= \bigvee_{t \in \mathbb{T}} T(x), \\ \text{tile}(x) &= \text{equ}(x) \wedge \text{mark}(x) \wedge \square \neg \text{mark}(x), \end{aligned}$$

where T is a fresh unary predicate for each $t \in \mathbb{T}$.

Then, consider the formulas

$$\diamond^+ (\text{last} \wedge \sigma_{2^n-1}(x)), \quad (7)$$

$$\text{tile}(x) \wedge \square \exists x \text{tile}(x). \quad (8)$$

Observe that, for Formulas (5) and (7) to be satisfied, a trace has to be finite and so that in its last instant the value of the σ -counter is $2^n - 1$. As it will become clear below, this step differs from the proof of [54, Theorem 5.43], since we exploit the last instant of a finite trace to indicate that the construction of the corridor is completed.

Now define

$$\text{corridor}(x), \quad (9)$$

as the conjunction of (4)-(8). Let $\mathfrak{F} = (\Delta^{\mathfrak{F}}, (\mathcal{F}_n)_{n \in [0, l]})$ be a finite trace satisfying $\text{corridor}(x)$. We have that $\mathfrak{F}, 0 \models \text{corridor}[d_0]$, for some $d_0 \in \Delta^{\mathfrak{F}}$. It can be seen that this implies the existence of $m \cdot 2^n$ distinct elements $d_0, \dots, d_{m \cdot 2^n - 1}$ of $\Delta^{\mathfrak{F}}$, for some $m \in \mathbb{N}$, such that $\mathfrak{F}, i \models \text{tile}[d_i]$, for $0 \leq i \leq m \cdot 2^n - 1$.

As a next step, we set

$$\begin{aligned} \text{up}(x) &= \circ \text{tile}(x), \\ \text{right}(x) &= \text{equ}(x) \wedge (\neg \text{equ}(x) \mathcal{U} \text{tile}(x)). \end{aligned}$$

Given a finite trace \mathfrak{F} satisfying $\text{corridor}(x)$, it can be seen that the following hold:

- for every $0 \leq i < m \cdot 2^n - 1$, $\mathfrak{F}, i \models \text{up}[d_{i+1}]$ and $\mathfrak{F}, i \not\models \text{up}[d_j]$, for every $j \neq i + 1$;
- for every $0 \leq i < (m - 1) \cdot 2^n$, $\mathfrak{F}, i \models \text{right}[d_{i+2^n}]$ and $\mathfrak{F}, i \not\models \text{right}[d_j]$, for every $j \neq i + 2^n$.

The following formula will ensure that each point of the corridor is covered by at most one tile:

$$\Box^+ \forall x \bigwedge_{\substack{t, t' \in \mathbb{T}, \\ t \neq t'}} \neg(T(x) \wedge T'(x)). \quad (10)$$

In addition, we impose that tile t_0 is put onto the point $(0, 0)$ of the corridor and that tile t_1 covers $(m - 1, 0)$ by using the following formulas:

$$T_0(x), \quad (11)$$

$$\Box^+ \forall x (\sigma_0(x) \wedge \text{mark}(x) \wedge \Box \neg \sigma_0(x) \rightarrow T_1(x)). \quad (12)$$

The condition about matching colours on adjacent sides of adjacent tiles is encoded by the formulas:

$$\Box^+ \forall x \left(\neg \sigma_{2^n-1}(x) \rightarrow \bigwedge_{\substack{t, t' \in \mathbb{T}, \\ \text{up}(t) \neq \text{down}(t')}} (T(x) \rightarrow \forall x (\text{up}(x) \rightarrow \Box \neg T'(x))) \right), \quad (13)$$

$$\Box^+ \forall x \left(\bigwedge_{\substack{t, t' \in \mathbb{T}, \\ \text{right}(t) \neq \text{left}(t')}} (T(x) \rightarrow \forall x (\text{right}(x) \rightarrow \Box \neg T'(x))) \right). \quad (14)$$

Finally, we represent as follows that the bottom and the top side of the corridor have to be *white*:

$$\Box^+ \forall x (\sigma_0(x) \wedge \text{mark}(x) \rightarrow \bigvee_{\substack{t \in \mathbb{T}, \\ \text{down}(t) = \text{white}}} T(x)), \quad (15)$$

$$\Box^+ \forall x (\sigma_{2^n-1}(x) \wedge \text{mark}(x) \rightarrow \bigvee_{\substack{t \in \mathbb{T}, \\ \text{up}(t) = \text{white}}} T(x)). \quad (16)$$

We then define the $T_{\mathcal{U}}\mathcal{QL}_{\phi}^{1,mo}$ formula $\varphi_{n, \mathbb{T}}(x)$ as the conjunction of (9)-(16). Clearly, the length of $\varphi_{n, \mathbb{T}}(x)$ is polynomial in n and $|\mathbb{T}|$.

Suppose that $\mathfrak{F}, 0 \models \varphi_{n, \mathbb{T}}[d_0]$, for some $\mathfrak{F} = (\Delta^{\mathfrak{F}}, (\mathcal{F}_n)_{n \in [0, l]})$ and some $d_0 \in \Delta^{\mathfrak{F}}$. It can be seen that there exists $m \in \mathbb{N}$ such that the function $\tau: m \times 2^n \rightarrow \mathbb{T}$ defined so that $\tau(i, j) = t$ iff $\mathfrak{F}, i \cdot 2^n + j \models T[d_{i \cdot 2^n + j}]$, for $0 \leq i < m$ and $0 \leq j < 2^n$, is a tiling of the $m \times 2^n$ corridor (cf. [54, Figure 5.13], as well as the similar Figure 1 below, depicting the tiling of a $2^n \times 2^n$ grid, instead of an $m \times 2^n$ corridor).

Conversely, if there exist $m \in \mathbb{N}$ and a function $\tau: m \times 2^n \rightarrow \mathbb{T}$ tiling the $m \times 2^n$ corridor, then we can construct a finite trace $\mathfrak{F} = (\Delta^{\mathfrak{F}}, (\mathcal{F}_n)_{n \in [0, m \cdot 2^n - 1]})$, such that $\Delta^{\mathfrak{F}} = \{d_0, \dots, d_{m \cdot 2^n - 1}\}$ and $\mathfrak{F}, 0 \models \varphi_{n, \mathbb{T}}[d_0]$ (see also Figure 1). \square

The reduction given in Theorem 4.3 allows us to transfer EXPSPACE upper bounds for the following fragments of first-order temporal logic on infinite traces to the finite traces case (see [75] and [54, Theorem 11.31]): the monadic monodic fragment $T_{\mathcal{U}}\mathcal{QL}_{\Box}^{mo}$, and the two-variable monodic fragment $T_{\mathcal{U}}\mathcal{QL}_{\Box}^2$.

PROPOSITION 5.2. *$T_{\mathcal{U}}\mathcal{QL}_{\Box}^{mo}$ and $T_{\mathcal{U}}\mathcal{QL}_{\Box}^2$ formula satisfiability on finite traces is in EXPSPACE.*

Thanks to the hardness and membership results of, respectively, Propositions 5.1 and 5.2, since $T_{\mathcal{U}}\mathcal{QL}_{\phi}^{1,mo}$ is contained both in $T_{\mathcal{U}}\mathcal{QL}_{\Box}^{mo}$ and $T_{\mathcal{U}}\mathcal{QL}^1$, and since $T_{\mathcal{U}}\mathcal{QL}^1$ is contained in $T_{\mathcal{U}}\mathcal{QL}_{\Box}^2$ we obtain the following result.

THEOREM 5.3. $T_{\mathcal{U}}\mathcal{QL}_{\phi}^{1,mo}$, $T_{\mathcal{U}}\mathcal{QL}^1$, $T_{\mathcal{U}}\mathcal{QL}_{\square}^{mo}$, and $T_{\mathcal{U}}\mathcal{QL}_{\square}^2$ formula satisfiability on finite traces are EXPSPACE-complete problems.

5.2 Complexity Results on k -Bounded Traces

We now study satisfiability of the decidable fragments considered above on traces with at most k time points, where k is given in binary as part of the input. We show that in this case, the complexity of the satisfiability problem in the fragments considered in the previous section decreases from EXPSPACE to NEXPTIME. We start by showing the lower bound for $T_{\mathcal{U}}\mathcal{QL}_{\phi}^{1,mo}$.

PROPOSITION 5.4. $T_{\mathcal{U}}\mathcal{QL}_{\phi}^{1,mo}$ formula satisfiability on k -bounded traces is NEXPTIME-hard.

PROOF. The proof is an adaptation of Proposition 5.1 to the case of $T_{\mathcal{U}}\mathcal{QL}_{\phi}^{1,mo}$ on k -bounded traces. As above, a *tile type* is a 4-tuple $t = (up(t), down(t), left(t), right(t))$ of colours. Let \mathbb{T} be a finite set of tile types, with $t_0 \in \mathbb{T}$. For an $n \in \mathbb{N}$, the $2^n \times 2^n$ *grid tiling problem* is the problem of deciding whether there exists a tiling $\tau: 2^n \times 2^n \rightarrow \mathbb{T}$ such that:

- $\tau(0, 0) = t_0$;
- $up(\tau(i, j)) = down(\tau(i, j + 1))$, for $0 \leq i < 2^n, 0 \leq j < 2^n - 1$,
- $right(\tau(i, j)) = left(\tau(i + 1, j))$, for $0 \leq i < 2^n - 1, 0 \leq j < 2^n$.

The $2^n \times 2^n$ grid tiling problem is known to be NEXPTIME-complete [97]. In the following, we will reduce this problem to $T_{\mathcal{U}}\mathcal{QL}_{\phi}^{1,mo}$ formula satisfiability on k -bounded traces, with $k = 2^{2n}$.

Let \mathbb{T} be a finite set of tile types, with $t_0 \in \mathbb{T}$, and let $n \in \mathbb{N}$. We modify the proof of Proposition 5.1 to construct a $T_{\mathcal{U}}\mathcal{QL}_{\phi}^{1,mo}$ formula $\varphi_{n, \mathbb{T}}$ such that: (i) the length of $\varphi_{n, \mathbb{T}}$ is polynomial in n and $|\mathbb{T}|$; (ii) $\varphi_{n, \mathbb{T}}$ is satisfiable on k -bounded traces, with $k = 2^{2n}$, iff there exists a function $\tau: 2^n \times 2^n \rightarrow \mathbb{T}$ tiling the $2^n \times 2^n$ grid (as described by the conditions above).

Recall the definition of the σ -counter, given in the proof of Proposition 5.1. We introduce other n distinct unary predicates R_0, \dots, R_{n-1} , and let $R_i^0 = \neg R_i$ and $R_i^1 = R_i$, for $0 \leq i \leq n-1$. Then, we define another binary counter, called ρ -counter up to 2^n by setting

$$\rho_j(x) = R_{n-1}^{b_j}(x) \wedge \dots \wedge R_0^{b_0}(x),$$

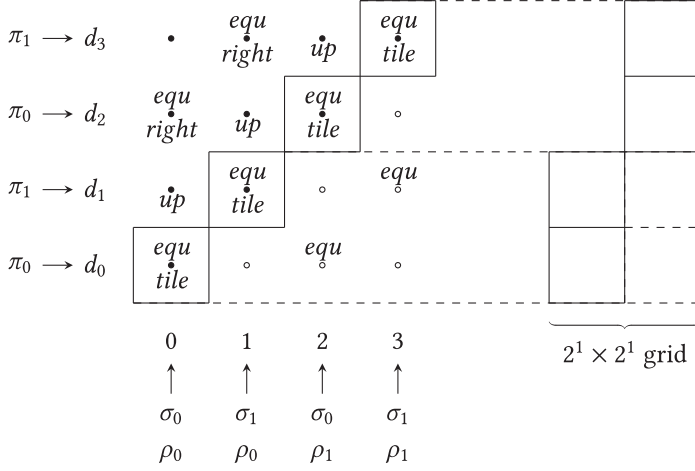
where b_j is the i th bit in the binary representation of $0 \leq j \leq 2^n - 1$. In addition, we require

$$\square^+ \bigwedge_{0 \leq i \leq n-1} (\forall x R_i(x) \vee \forall x \neg R_i(x)), \quad (17)$$

so that, at each time point, the counter value will be the same for every element of the domain. The following formula will set the value of this counter to 0 at the first instant of a trace, and increase its value by one at each future instant where σ_0 holds (if any).

$$\begin{aligned} & \rho_0(x) \wedge \\ & \square^+ \bigwedge_{0 \leq k < n} \left(\bigcirc \neg \sigma_0(x) \rightarrow (R_k(x) \leftrightarrow \bigcirc R_k(x)) \right) \wedge \\ & \square^+ \bigwedge_{0 \leq k < n} \left(\bigcirc \sigma_0(x) \rightarrow \left(\left(\bigwedge_{0 \leq i < k} R_i(x) \wedge \neg R_k(x) \right) \rightarrow \right. \right. \\ & \quad \left. \left. \left(\bigwedge_{k < j < n} (R_j(x) \leftrightarrow \bigcirc R_j(x)) \right) \wedge \bigcirc \left(\bigwedge_{0 \leq i < k} \neg R_i(x) \wedge R_k(x) \right) \right) \right). \quad (18) \end{aligned}$$

Formula (18) implies that, if $\sigma_0(x)$ is satisfied at a given instant, then $\rho_j(x)$ is satisfied as well, for some $0 \leq j < 2^n$.


 Fig. 1. Tiling of the $2^n \times 2^n$ grid, with $n = 1$.

We then modify Formula (7) by imposing instead

$$\diamond^+(last \wedge \sigma_{2^n-1}(x) \wedge \rho_{2^n-1}(x)), \quad (19)$$

so to force a finite trace to be such that in its last instant both $\sigma_{2^n-1}(x)$ and $\rho_{2^n-1}(x)$ hold.

Now define $\varphi_{n,T}$ as the conjunction of (4)-(6), (8), (10), (11), (13), (14), (17)-(19), whose length is polynomial in n and $|\mathbb{T}|$.

We now show the correctness of the encoding, by also relying on Figure 1, where we represent: along the horizontal dimension, the time points of a finite trace, together with the values of σ - and ρ -counter; and on the vertical dimension, the elements of the object domain at each instant, as well as the values of the π -counter, defined in the proof of Proposition 5.1. Suppose that $\mathfrak{F}, 0 \models \varphi_{n,\mathbb{T}}[d_0]$, for some k -bounded trace $\mathfrak{F} = (\Delta^{\mathfrak{F}}, (\mathcal{F}_i)_{i \in [0,l]})$, with $k = 2^{2^n}$ and $l < k$, and some $d_0 \in \Delta^{\mathfrak{F}}$. It can be seen that the function $\tau: 2^n \times 2^n \rightarrow \mathbb{T}$ defined so that $\tau(i, j) = t$ iff $\mathfrak{F}, i \cdot 2^n + j \models T[d_{i \cdot 2^n + j}]$, for $0 \leq i < 2^n$ and $0 \leq j < 2^n$, is a tiling of the $2^n \times 2^n$ grid.

Conversely, if there exists a function $\tau: 2^n \times 2^n \rightarrow \mathbb{T}$ tiling the $2^n \times 2^n$ grid, then we can construct a k -bounded trace $\mathfrak{F} = (\Delta^{\mathfrak{F}}, (\mathcal{F}_i)_{i \in [0,k-1]})$, with $k = 2^{2^n}$, such that $\Delta^{\mathfrak{F}} = \{d_0, \dots, d_{2^{2^n}-1}\}$ and $\mathfrak{F}, 0 \models \varphi_{n,\mathbb{T}}[d_0]$. \square

For the upper bound, we resort to a classical abstraction of models called *quasimodels* [54]. 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.

PROPOSITION 5.5. $T_{\mathcal{U}}\mathcal{QL}_{\square}^{mo}$ and $T_{\mathcal{U}}\mathcal{QL}_{\square}^2$ formula satisfiability on k -bounded traces is in $NEXPTIME$.

PROOF. In the following, with an abuse of notation, $\psi(x)$ denotes a formula ψ with *at most* x as free variable. Let φ be a $T_{\mathcal{U}}\mathcal{QL}_{\square}$ sentence,³ let $\text{Ind}(\varphi)$ be the set of individuals occurring in φ , and

³The restriction to sentences is without loss of generality, since a monodic formula $\varphi(x_1, \dots, x_n)$ is equisatisfiable with the monodic sentence $\exists x_1 \dots \exists x_n \varphi(x_1, \dots, x_n)$.

let $\text{sub}(\varphi)$ be the set of subformulas of φ . For every formula $\psi(y)$ of the form $\psi_1 \mathcal{U} \psi_2$ with one free variable y , we fix a *surrogate* $R_\psi(y)$; and for every sentence ψ of the form $\psi_1 \mathcal{U} \psi_2$, we fix a surrogate P_ψ , where R_ψ and P_ψ are symbols not occurring in φ . Given a $T_{\mathcal{U}}\mathcal{QL}_{\square}^2$ formula φ , we denote by $\bar{\varphi}$ the result of replacing in φ all subformulas of the form $\psi_1 \mathcal{U} \psi_2$ which are not in the scope of any other occurrence of \mathcal{U} by their surrogates. Thus, $\bar{\varphi}$ does not contain occurrences of temporal operators. Let $\text{sub}_0(\varphi)$ be the set of all sentences in $\text{sub}(\varphi)$. Let x be a variable not occurring in φ , and $\text{sub}_x(\varphi)$ be the closure under (single) negation of all formulas $\psi\{x/y\}$ with $\psi(y) \in \text{sub}(\varphi)$. A *type* for φ is a subset t of $\{\bar{\psi} \mid \psi \in \text{sub}_x(\varphi)\} \cup \text{N}_1(\varphi)$ such that:

- $\bar{\psi}_1 \wedge \bar{\psi}_2 \in t$ iff $\bar{\psi}_1 \in t$ and $\bar{\psi}_2 \in t$, for every $\psi_1 \wedge \psi_2 \in \text{sub}_x(\varphi)$;
- $\bar{\neg\psi} \in t$ iff $\bar{\psi} \notin t$, for every $\neg\psi \in \text{sub}_x(\varphi)$; and
- t contains at most one element of $\text{N}_1(\varphi)$.

We omit ‘for φ ’ when there is no risk of confusion. We say that the types t, t' agree on $\text{sub}_0(\varphi)$ if $t \cap \text{sub}_0(\bar{\varphi}) = t' \cap \text{sub}_0(\bar{\varphi})$. Denote with $\text{tp}(\varphi)$ the set of all types for φ . If $a \in t \cap \text{N}_1(\varphi)$, then t “describes” a named element. We write t^a to indicate this and call it a *named type*. A *state candidate* is a subset \mathfrak{C} of $\text{tp}(\varphi)$ with only types that agree on $\text{sub}_0(\varphi)$, containing exactly one t^a for each $a \in \text{N}_1(\varphi)$, and such that $\{t \setminus \{a\} \mid t^a \in \mathfrak{C}\} \subseteq \mathfrak{C}$. Given a classical first-order interpretation I , let $t^I(d) = \{\bar{\psi} \mid \psi \in \text{sub}_x(\varphi), I \models \bar{\psi}[d]\}$, with d in the domain Δ , and let $t^I(a) = \{\bar{\psi} \mid \psi \in \text{sub}_x(\varphi), I \models \bar{\psi}[a^I]\} \cup \{a\}$, with $a \in \text{N}_1(\varphi)$. Clearly, each $t^I(d)$ is a type for φ while each $t^I(a)$ is a named type. An interpretation I realizes a state candidate \mathfrak{C} if $\mathfrak{C} = \{t^I(d) \mid d \in \Delta\} \cup \{t^I(a) \mid a \in \text{N}_1(\varphi)\}$. Vice versa, we say that \mathfrak{C} is (*finitely*) *realisable* if there is a (*finite*) interpretation realizing it.

Given a state candidate \mathfrak{C} , we define the sentence

$$\text{real}_{\mathfrak{C}} = \bigwedge_{t \in \mathfrak{C}} \exists x \bigwedge_{\psi(x) \in t} \psi(x) \wedge \forall x \bigvee_{t \in \mathfrak{C}} \bigwedge_{\psi(x) \in t} \psi(x) \wedge \bigwedge_{t^a \in \mathfrak{C}} \bigwedge_{\psi(x) \in t^a} \psi\{a/x\}.$$

We have that a state candidate \mathfrak{C} is (*finitely*) *realisable* iff the sentence $\text{real}_{\mathfrak{C}}$ is true in some (*finite*) first-order interpretation [54, Lemma 11.6]. Moreover, for $T_{\mathcal{U}}\mathcal{QL}_{\square}^{\text{mo}}$ and $T_{\mathcal{U}}\mathcal{QL}_{\square}^2$, it is known that realisability of state candidates coincides with finite realisability, since the monadic and the 2-variable fragments of first-order logic enjoy the *exponential* (and thus *finite*) *model property* [34, Proposition 6.2.7, Corollary 8.1.5]. Finally, the following holds (for details, see the proof in [54, Theorem 11.31]).

LEMMA 5.6. *Given a state candidate \mathfrak{C} for a $T_{\mathcal{U}}\mathcal{QL}_{\square}^{\text{mo}}$ or a $T_{\mathcal{U}}\mathcal{QL}_{\square}^2$ sentence φ , there is an NEXPTIME algorithm that checks whether $\text{real}_{\mathfrak{C}}$ is satisfiable, and thus whether \mathfrak{C} is realisable.*

Now, consider a language $\mathcal{L} \in \{T_{\mathcal{U}}\mathcal{QL}_{\square}^{\text{mo}}, T_{\mathcal{U}}\mathcal{QL}_{\square}^2\}$. We use standard definitions for *quasimodels* [54, 76], presented here for convenience of the reader. A *quasimodel* for an \mathcal{L} sentence φ is a pair (S, \mathfrak{R}) where S is a finite sequence $(S(0), \dots, S(n))$ of realizable state candidates $S(i)$, denoted in the following as *quasistates*, and \mathfrak{R} is a set of functions r from $\{0, \dots, n\}$ to $\bigcup_{0 \leq i \leq n} S(i)$, called *runs*, mapping each $i \in \{0, \dots, n\}$ to a type in $S(i)$, and satisfying the following conditions:

- (1) for every $\psi_1 \mathcal{U} \psi_2 \in \text{sub}_x(\varphi)$ and every $i \in [0, n]$, we have $\bar{\psi}_1 \mathcal{U} \bar{\psi}_2 \in r(i)$ iff there is $j \in (i, n]$ such that $\bar{\psi}_2 \in r(j)$ and $\bar{\psi}_1 \in r(l)$ for all $l \in (i, j)$;
- (2) for every $a \in \text{N}_1(\varphi)$, every $r \in \mathfrak{R}$ and every $i, j \in [0, n]$, we have $a \in r(i)$ iff $a \in r(j)$;
- (3) $\bar{\varphi} \in t$ for some $t \in S(0)$; and
- (4) for every $i \in [0, n]$ and every $t \in S(i)$ there is a run $r \in \mathfrak{R}$ such that $r(i) = t$.

Every quasimodel for φ describes an interpretation satisfying φ and, conversely, every such interpretation can be abstracted into a quasimodel for φ . We formalise this well-known fact in the next lemma, that follows from an adaptation of [54, Lemma 11.22] to the case of k -bounded traces.

LEMMA 5.7. *Let φ be an $\mathcal{L} \in \{T_{\mathcal{U}}Q\mathcal{L}_{\square}^{mo}, T_{\mathcal{U}}Q\mathcal{L}_{\square}^2\}$ sentence. There is a k -bounded trace satisfying φ iff there is a quasimodel for φ with a sequence of quasistates of length at most k .*

Let us now turn to our main result for \mathcal{L} formulas interpreted on k -bounded traces. Since the length of a trace is bounded from the input, the complexity differs from the satisfiability checking procedures on infinite or finite traces, which are instead in ExpSPACE . In particular, in the following we devise a non-deterministic exponential time algorithm to check satisfiability on k -bounded traces of an \mathcal{L} sentence φ . It follows from the definition of types that the number of distinct types for φ is exponential in $|\varphi|$. Thus, we first guess: a sequence $(S(0), \dots, S(n))$ of sets of types for φ of length $n + 1 \leq k$; and, for each type at position i in this sequence, a sequence of types of length $n + 1$. Denote by \mathfrak{R} the set of such sequences of types. Then, we check: (a) whether each $S(i)$ is a realisable state candidate, which, by Lemma 5.6, can be done in NExpTime in the size of φ ; (b) whether each sequence in \mathfrak{R} satisfies conditions (1) and (2); and (c) whether φ is in a type in $S(0)$, i.e., whether condition (3) holds. Condition (4) is satisfied by definition of \mathfrak{R} . All these conditions can be checked in non-deterministic exponential time with respect to $|\varphi|$ and the binary size of $k, |k|$. The algorithm returns ‘satisfiable’ iff all conditions are satisfied, thus implying that (S, \mathfrak{R}) is a quasimodel for φ . By Lemma 5.7, given an $\mathcal{L} \in \{T_{\mathcal{U}}Q\mathcal{L}_{\square}^{mo}, T_{\mathcal{U}}Q\mathcal{L}_{\square}^2\}$ formula φ , there is a finite trace satisfying φ with at most k time points iff there is a quasimodel for φ with at most k quasistates. Thus, we showed Proposition 5.5 illustrating a NExpTime algorithm for checking the satisfiability of formulas in $\{T_{\mathcal{U}}Q\mathcal{L}_{\square}^{mo}, T_{\mathcal{U}}Q\mathcal{L}_{\square}^2\}$. \square

We can now state the main result of this section. Thanks to the lower and upper bounds shown in Propositions 5.4 and 5.5, since $T_{\mathcal{U}}Q\mathcal{L}_{\square}^{1,mo}$ is contained both in $T_{\mathcal{U}}Q\mathcal{L}_{\square}^{mo}$ and $T_{\mathcal{U}}Q\mathcal{L}_{\square}^1$, and since $T_{\mathcal{U}}Q\mathcal{L}_{\square}^1$ is contained in $T_{\mathcal{U}}Q\mathcal{L}_{\square}^2$, we obtain the following complexity result.

THEOREM 5.8. *$T_{\mathcal{U}}Q\mathcal{L}_{\square}^{1,mo}$, and $T_{\mathcal{U}}Q\mathcal{L}_{\square}^1$, $T_{\mathcal{U}}Q\mathcal{L}_{\square}^{mo}$ and $T_{\mathcal{U}}Q\mathcal{L}_{\square}^2$ formula satisfiability on k -bounded traces are NExpTime -complete problems.*

5.3 Bounded Trace and Domain Properties

In this section, we prove that $\mathcal{L} \in \{T_{\mathcal{U}}Q\mathcal{L}_{\square}^{mo}, T_{\mathcal{U}}Q\mathcal{L}_{\square}^2\}$ on finite traces enjoys two kinds of bounded model properties, one which bounds the domain of elements and one which bounds the number of time points in a trace.

First, we show that an $\mathcal{L} \in \{T_{\mathcal{U}}Q\mathcal{L}_{\square}^{mo}, T_{\mathcal{U}}Q\mathcal{L}_{\square}^2\}$ formula has the *bounded trace property*, i.e., if it is satisfiable on finite traces, then there is a k -bounded trace satisfying it, where k is at most double exponential in $|\varphi|$.

THEOREM 5.9. *Satisfiability of an $\mathcal{L} \in \{T_{\mathcal{U}}Q\mathcal{L}_{\square}^{mo}, T_{\mathcal{U}}Q\mathcal{L}_{\square}^2\}$ formula φ on finite traces implies satisfiability of φ on k -bounded traces, with $k \leq 2^{2^{|\varphi|}}$.*

PROOF. In order to prove the statement, we require some preliminary lemmas. First, we recall the following result [54, Lemma 11.22], applied to the case of finite traces.

LEMMA 5.10. *An \mathcal{L} sentence φ is satisfiable on finite traces iff there is a quasimodel for φ .*

Moreover, we adapt [54, Lemma 11.28] to the case of finite traces, formalising it as follows.

LEMMA 5.11. *For every quasimodel (S, \mathfrak{R}) for φ , there is a quasimodel (S', \mathfrak{R}') for φ such that $|S'| \leq 2^{|\text{tp}(\varphi)|}$.*

PROOF. We first introduce the following notation. Given a sequence $\Sigma = (\sigma_0, \sigma_1, \sigma_2, \dots)$ and $n \in \mathbb{N}$, we denote by Σ_n and Σ^n the prefix ending at n and the suffix starting at n of Σ , respectively.

Moreover, given sequences Σ, Σ' , we denote by $\Sigma \cdot \Sigma'$ the concatenation of Σ with Σ' . Then, we require the following claim, obtained by rephrasing [54, Lemma 11.27] to our terminology.

CLAIM 8. *Given a quasimodel (S, \mathfrak{R}) for φ such that $S(n) = S(m)$, for some $n < m$, we have that $(S_n \cdot S^{m+1}, \mathfrak{R}_n \cdot \mathfrak{R}^{m+1})$, with $\mathfrak{R}_n \cdot \mathfrak{R}^{m+1} = \{r_n \cdot r'^{m+1} \mid r, r' \in \mathfrak{R}, r(n) = r'(m)\}$, is a quasimodel for φ .*

Now, let (S, \mathfrak{R}) be a quasimodel for φ , with $S = (S(0), \dots, S(l))$. We notice that the number of different quasistates is $\leq 2^{|\text{tp}(\varphi)|}$. If, for every $n, m \in [0, l]$, we have $S(n) \neq S(m)$ (meaning that all quasistates in S are distinct), then $|S| \leq 2^{|\text{tp}(\varphi)|}$, and thus (S, \mathfrak{R}) is as required. Otherwise, suppose that there are $n, m \in [0, l]$ such that $S(n) = S(m)$. Without loss of generality, we can assume that $n < m$. Since S is finite, by (repeatedly) applying Claim 8 to (S, \mathfrak{R}) , we obtain a quasimodel (S', \mathfrak{R}') such that S' contains only distinct quasistates, hence $|S'| \leq 2^{|\text{tp}(\varphi)|}$. \square

The theorem now follows from Lemmas 5.10, 5.11, and 5.7, since $|\text{tp}(\varphi)| \leq 2^{|\varphi|}$. \square

We now establish that an $\mathcal{L} \in \{T_{\mathcal{U}}\mathcal{QL}_{\square}^{mo}, T_{\mathcal{U}}\mathcal{QL}_{\square}^2\}$ formula interpreted on k -bounded traces has the *bounded domain property*, i.e., if it is satisfiable on k -bounded traces, then it is satisfied on a trace having finite domain with bounded cardinality.

THEOREM 5.12. *Satisfiability of an $\mathcal{L} \in \{T_{\mathcal{U}}\mathcal{QL}_{\square}^{mo}, T_{\mathcal{U}}\mathcal{QL}_{\square}^2\}$ formula φ on k -bounded traces implies satisfiability of φ on traces having finite domain of cardinality at most $|\text{tp}(\varphi)|^k \cdot 2^{|\text{tp}(\varphi)|}$.*

PROOF. We require the following preliminary definitions and result. A quasimodel (S, \mathfrak{R}) for φ , with $S = (S(0), \dots, S(l))$, is said to be *finitary* if $S(i)$ is finitely realisable, for every $i \in [0, l]$, and \mathfrak{R} is finite. The next lemma is an adaptation of [54, Lemma 11.41] to the case of k -bounded traces.

LEMMA 5.13. *An \mathcal{L} sentence φ is satisfiable on k -bounded traces having finite domain iff there is a finitary quasimodel for φ with a sequence of quasistates of length at most k .*

Now, suppose that φ is satisfied on a k -bounded trace. By Lemma 5.7, there is a quasimodel (S, \mathfrak{R}) for φ , with $S = (S(0), \dots, S(l))$ and $l < k$. It is known that, for $\mathcal{L} \in \{T_{\mathcal{U}}\mathcal{QL}_{\square}^{mo}, T_{\mathcal{U}}\mathcal{QL}_{\square}^2\}$ formulas, a state candidate is realisable iff it is finitely realisable, since monadic and 2-variable first-order formulas enjoy the exponential (hence, finite) model property [34, Proposition 6.2.7, Corollary 8.1.5]. Thus, we have that every $S(i)$, for $i \in [0, l]$, is finitely realisable. Moreover, because S is finite, we have that \mathfrak{R} , which is a set of functions from $\{0, \dots, l\}$ to $\bigcup_{0 \leq i \leq l} S(i)$, is finite as well. Therefore, (S, \mathfrak{R}) is finitary and, by Lemma 5.13, φ is satisfiable on k -bounded traces having finite domain. Finally, having recalled that monadic and 2-variable first-order formulas enjoy the exponential model property, we can assume without loss of generality that a first-order interpretation realising a quasistate in S has domain of cardinality at most $2^{|\text{tp}(\varphi)|}$. Thus, one can adjust the construction in [54, Lemma 11.41] to obtain, from a finitary quasimodel (S, \mathfrak{R}) for φ , with $|S| \leq k$, a k -bounded trace that satisfies φ with domain $\Delta = \{(r, i) \mid r \in \mathfrak{R}, 0 \leq i < m_{\varphi}\}$, for some $m_{\varphi} \leq 2^{|\text{tp}(\varphi)|}$. Since $|\mathfrak{R}| \leq |\text{tp}(\varphi)|^k$, we have that $|\Delta| \leq |\text{tp}(\varphi)|^k \cdot 2^{|\text{tp}(\varphi)|}$. \square

Since the $T_{\mathcal{U}}\mathcal{QL}_{\square}^2$ formula

$$\vartheta_i = Q(a) \wedge \square^+ \forall x \left(Q(x) \rightarrow \square^+ (\neg Q(x) \wedge \exists y (R(x, y) \wedge Q(y))) \right)$$

only admits models with an infinite domain [84], by the previous theorems the formula is unsatisfiable over finite traces.

5.4 Temporal Description Logics

We conclude this section investigating the complexity of the satisfiability problem in temporal DLs. We consider the temporal language $T_{\mathcal{U}}\mathcal{ALC}$ [54] as a temporal extension of the DL \mathcal{ALC} [18]. Let $N_C, N_R \subseteq N_P$ be, respectively, countably infinite and disjoint sets of unary and binary predicates called *concept* and *role names*. A $T_{\mathcal{U}}\mathcal{ALC}$ *concept* is an expression of the form:

$$C, D ::= A \mid \neg C \mid C \sqcap D \mid \exists R.C \mid C \mathcal{U} D,$$

where $A \in N_C$ and $R \in N_R$. A $T_{\mathcal{U}}\mathcal{ALC}$ *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 $T_{\mathcal{U}}\mathcal{ALC}$ concepts, $A \in N_C$, $R \in N_R$, and $a, b \in N_I$. $T_{\mathcal{U}}\mathcal{ALC}$ *formulas* have the form:

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

The semantics of $T_{\mathcal{U}}\mathcal{ALC}$ is given again (with a small abuse of notation) over *finite traces* $\mathfrak{F} = (\Delta, (\mathcal{F}_n)_{n \in [0, l]})$, where $l \in \mathbb{N}$, Δ is a non-empty domain, and, for every $n \in [0, l]$, \mathcal{F}_n is an \mathcal{ALC} interpretation with domain Δ , mapping each concept name $A \in N_C$ to a subset $A^{\mathcal{F}_n}$ of Δ , each role name $R \in N_R$ to a binary relation $R^{\mathcal{F}_n}$ on Δ , and each individual name $a \in N_I$ to a domain element $a^{\mathcal{F}_n}$ in such a way that $a^{\mathcal{F}_i} = a^{\mathcal{F}_j}$, for all $i, j \in [0, l]$ (thus, we just use the notation $a^{\mathcal{F}}$). The interpretation is extended to concepts as usual:

$$\begin{aligned} (\neg C)^{\mathcal{F}_n} &= \Delta \setminus C^{\mathcal{F}_n}, \\ (C \sqcap D)^{\mathcal{F}_n} &= C^{\mathcal{F}_n} \cap D^{\mathcal{F}_n}, \\ (\exists R.C)^{\mathcal{F}_n} &= \{d \in \Delta \mid \text{there is } e \in C^{\mathcal{F}_n} \text{ such that } (d, e) \in R^{\mathcal{F}_n}\}, \\ (C \mathcal{U} D)^{\mathcal{F}_n} &= \{d \in \Delta \mid \text{there is } m \in (n, l] \text{ such that } d \in D^{\mathcal{F}_m} \text{ and } d \in C^{\mathcal{F}_i}, \text{ for all } i \in (n, m)\}. \end{aligned}$$

Given a $T_{\mathcal{U}}\mathcal{ALC}$ formula φ , the *satisfaction* of φ in \mathfrak{F} at time point $n \in [0, l]$, written $\mathfrak{F}, n \models \varphi$, is inductively defined as:

$$\begin{aligned} \mathfrak{F}, n \models C \sqsubseteq D &\quad \text{iff} \quad C^{\mathcal{F}_n} \subseteq D^{\mathcal{F}_n}, \\ \mathfrak{F}, n \models A(a) &\quad \text{iff} \quad a^{\mathcal{F}} \in A^{\mathcal{F}_n}, \\ \mathfrak{F}, n \models R(a, b) &\quad \text{iff} \quad (a^{\mathcal{F}}, b^{\mathcal{F}}) \in R^{\mathcal{F}_n}, \\ \mathfrak{F}, n \models \neg\varphi &\quad \text{iff} \quad \mathfrak{F}, n \not\models \varphi, \\ \mathfrak{F}, n \models \varphi \wedge \psi &\quad \text{iff} \quad \mathfrak{F}, n \models \varphi \text{ and } \mathfrak{F}, n \models \psi, \\ \mathfrak{F}, n \models \varphi \mathcal{U} \psi &\quad \text{iff} \quad \text{there is } m \in (n, l] \text{ such that } \mathfrak{F}, m \models \psi \text{ and } \mathfrak{F}, i \models \varphi, \text{ for all } i \in (n, m). \end{aligned}$$

We say that a $T_{\mathcal{U}}\mathcal{ALC}$ formula φ is *satisfiable on finite traces* if there exists a finite trace \mathfrak{F} such that $\mathfrak{F}, 0 \models \varphi$. If φ is satisfiable on finite traces with at most k instants, with k given in binary, we say that φ is *satisfiable on k -bounded traces*.

Since a $T_{\mathcal{U}}\mathcal{ALC}$ formula can be mapped into an equisatisfiable $T_{\mathcal{U}}\mathcal{QL}_{\square}^2$ formula [54, Theorem 14.12], we can transfer the upper bounds of Propositions 5.2 and 5.5 to $T_{\mathcal{U}}\mathcal{ALC}$ on finite and k -bounded traces, respectively (indeed, the cited standard embedding allows us to see $T_{\mathcal{U}}\mathcal{ALC}$ as a fragment of $T_{\mathcal{U}}\mathcal{QL}_{\square}^2$ also under these finite semantics). The lower bounds can be obtained from Propositions 5.1 and 5.4, since $T_{\mathcal{U}}\mathcal{QL}_{\diamond}^{1, mo}$ can be seen as a fragment of $T_{\mathcal{U}}\mathcal{ALC}$ without role names [54]. Thus, the following holds.

THEOREM 5.14. *$T_{\mathcal{U}}\mathcal{ALC}$ satisfiability is EXPSPACE-complete on finite traces, and NEXPTIME-complete on k -bounded traces.*

Moreover, from Theorems 5.9 and 5.12, we obtain immediately that $T_{\mathcal{U}}\mathcal{ALC}$ on finite traces has both the bounded trace and domain properties.

We also consider the satisfiability problem on k -bounded traces of $T_{\mathcal{U}}\mathcal{ALC}$ restricted to global CIs [5, 84], defined as the fragment of $T_{\mathcal{U}}\mathcal{ALC}$ in which formulas can only be of the form $\Box^+(\mathcal{T}) \wedge \psi$, where \mathcal{T} is a conjunction of CIs and ψ does not contain CIs. The EXP_{TIME} upper bound we provide has a rather challenging proof that uses a form of *type elimination* [54, 69, 84], but in a setting where the number of time points is bounded by a natural number $k > 0$.⁴ The complexity is tight since satisfiability in \mathcal{ALC} is already EXP_{TIME}-hard [18].

To show the following theorem, we rely again on quasimodels [54], which have been used to prove the decidability of the satisfiability problem for various temporal DLs. Our definitions here are similar to those in Section 5.2, now adapted to temporal \mathcal{ALC} .

THEOREM 5.15. *$T_{\mathcal{U}}\mathcal{ALC}$ satisfiability on k -bounded traces restricted to global CIs is EXP_{TIME}-complete.*

PROOF. It is enough to show that satisfiability in $T_{\mathcal{U}}\mathcal{ALC}$ restricted to global CIs on k -bounded traces is in EXP_{TIME}. Let φ be a $T_{\mathcal{U}}\mathcal{ALC}$ formula restricted to global CIs. Assume without loss of generality that φ does not contain abbreviations (i.e., it only contains the logical connectives \neg , \sqcap , \wedge , the existential quantifier \exists , and the temporal operator \mathcal{U}), that \mathcal{T} has the equivalent form $\top \sqsubseteq C_{\mathcal{T}}$ (where $C_{\mathcal{T}}$ is of polynomial size with respect to the size of \mathcal{T}), and that φ contains at least one individual name (we can always add the assertion $A_a(a)$, for a fresh constant a and a fresh concept name A_a). Let $N_1(\varphi)$ be the set of individuals occurring in φ . Following the notation provided by Baader et al. 2017, denote by $\text{cl}^f(\varphi)$ the closure under single negation of the set of all formulas occurring in φ . Similarly, we denote by $\text{cl}^c(\varphi)$ the closure under single negation of the set of all concepts union the concepts $A_a, \exists R.A_a$, for any $a \in N_1(\varphi)$ and R a role occurring in φ , where A_a is fresh. A *concept type* for φ is any subset t of $\text{cl}^c(\varphi) \cup N_1(\varphi)$ such that:

- T1** $\neg C \in t$ iff $C \notin t$, for all $\neg C \in \text{cl}^c(\varphi)$;
- T2** $C \sqcap D \in t$ iff $C, D \in t$, for all $C \sqcap D \in \text{cl}^c(\varphi)$; and
- T3** t contains at most one individual name in $N_1(\varphi)$.

Similarly, we define *formula types* $t \subseteq \text{cl}^f(\varphi)$ for φ with the conditions:

- T1'** $\neg \chi \in t$ iff $\chi \notin t$, for all $\neg \chi \in \text{cl}^f(\varphi)$; and
- T2'** $\chi \wedge \psi \in t$ iff $\chi, \psi \in t$, for all $\chi \wedge \psi \in \text{cl}^f(\varphi)$.

We omit “for φ ” when there is no risk of confusion. A concept type describes one domain element at a single time point, while a formula type expresses assertions or constraints on all domain elements. If $a \in t \cap N_1(\varphi)$, then t describes a named element and is denoted as t^a . We denote with $\text{tp}(\varphi)$ the set of all concept and formula types.

The next notion captures how sets of types need to be constrained so that the DL dimension is respected. We say that a pair of concept types (t, t') is *R-compatible* if $\{\neg F \mid \neg \exists R.F \in t\} \subseteq t'$. A *quasistate* for φ is a set $S \subseteq \text{tp}(\varphi)$ such that:

- Q1** S contains exactly one formula type t_S ;
- Q2** S contains exactly one named type t^a for each $a \in N_1(\varphi)$;
- Q3** $C_{\mathcal{T}} \in t$, for all concept types $t \in S$;
- Q4** for all $A(a) \in \text{cl}^f(\varphi)$, we have $A(a) \in t_S$ iff $A \in t^a$;
- Q5** $t \in S$ and $\exists R.D \in t$ implies that there is $t' \in S$ such that $D \in t'$ and (t, t') is *R-compatible*;
- Q6** for all $R(a, b) \in \text{cl}^f(\varphi)$, we have $R(a, b) \in t_S$ iff (t^a, t^b) is *R-compatible*.

We notice that the critical “iff” condition in **Q6** can be realised using the extra concepts $A_a, \exists R.A_a$, introduced for all $a \in N_1(\varphi)$. In particular, for any quasistate S , with $t_S \in S$, and any $R(a, b) \in \text{cl}^f(\varphi)$,

⁴The main challenge in solving this problem when the number of time points is arbitrarily large, but finite, is mostly due to the presence of *last* sub-formulas (i.e., formulas of the form $\Box \perp$) that can hold just in the last instant of the model.

if $\neg R(a, b) \in t_S$ then the pair of named types (t^a, t^b) can be made not R -compatible by including $\neg \exists R.A_b$ in t^a and A_b in t^b . This trick will be used in the proof of Lemma 5.16.

A (concept/formula) run segment for φ is a finite sequence $\sigma = (\sigma(0), \dots, \sigma(n))$ composed exclusively of concept or formula types, respectively, such that:

R1 for all $a \in N_1(\varphi)$ and all $i \in (0, n]$, we have $a \in \sigma(0)$ iff $a \in \sigma(i)$;

R2 for all $\alpha \mathcal{U} \beta \in \text{cl}^*(\varphi)$ and all $i \in [0, n]$, we have $\alpha \mathcal{U} \beta \in \sigma(i)$ iff there is $j \in (i, n]$ such that $\beta \in \sigma(j)$ and $\alpha \in \sigma(m)$ for all $m \in (i, j)$,

where cl^* is either cl^c or cl^f (as appropriate), and **R1** does not apply to formula run segments. Also notice that condition **R2** disallows any run to contain until concepts/formulas in the last instant n .⁵ Intuitively, a concept run segment describes the temporal dimension of a single domain element, whereas a formula run segment describes constraints on the whole DL interpretation.

Finally, a *quasimodel* for $\varphi = \Box^+(\top \sqsubseteq C_{\mathcal{T}}) \wedge \psi$ is a pair (S, \mathfrak{R}) , with S a finite sequence of quasistates $(S(0), \dots, S(n))$ and \mathfrak{R} a non-empty set of run segments such that:

M1 $\psi \in t_{S_0}$ where t_{S_0} is the formula type in $S(0)$;

M2 for every $\sigma \in \mathfrak{R}$ and every $i \in [0, n]$, $\sigma(i) \in S(i)$; and, conversely, for every $t \in S(i)$, there is $\sigma \in \mathfrak{R}$ with $\sigma(i) = t$.

By **M2** and the definition of a quasistate for φ , \mathfrak{R} always contains exactly one formula run segment and one named run segment for each $a \in N_1(\varphi)$.

Every quasimodel for φ describes an interpretation satisfying φ and, conversely, every such interpretation can be abstracted into a quasimodel for φ . We formalise this notion for finite traces with the following lemma.

LEMMA 5.16. *There is a finite trace satisfying φ with at most k time points iff there is a quasimodel for φ with a sequence of quasistates of length at most k .*

PROOF. (\Rightarrow) Assume there is a finite trace $\mathfrak{F} = (\Delta, (\mathcal{F}_n)_{n \in [0, l]})$, with $l < k$, that satisfies φ , i.e., $\mathfrak{F}, 0 \models \varphi$. Without loss of generality, assume that, for all $a \in N_1(\varphi)$, we have $A_a^{\mathcal{F}_n} = \{a^{\mathcal{F}}\}$ for all $n \in [0, l]$, where A_a are those fresh concept names we used to extend $\text{cl}^c(\varphi)$. We define (S, \mathfrak{R}) in the following way. First, for all $n \in [0, l]$, $d \in \Delta$ and $a \in N_1(\varphi)$, we set:

$$\begin{aligned} \sigma_d(n) &= \{C \in \text{cl}^c(\varphi) \mid d \in C^{\mathcal{F}_n}\}, \\ \sigma_a(n) &= \{C \in \text{cl}^c(\varphi) \mid a^{\mathcal{F}} \in C^{\mathcal{F}_n}\} \cup \{a\}, \end{aligned}$$

and let $\sigma_* = (\sigma_*(0), \dots, \sigma_*(n))$, with $*$ $\in \{d, a\}$. Moreover, we set $\sigma_{\mathfrak{F}}(n) = \{\psi \in \text{cl}^f(\varphi) \mid \mathfrak{F}, n \models \psi\}$, and $\sigma_{\mathfrak{F}} = (\sigma_{\mathfrak{F}}(0), \dots, \sigma_{\mathfrak{F}}(n))$. Finally, define $S(n) = \{\sigma_d(n) \mid d \in \Delta\} \cup \{\sigma_a(n) \mid a \in N_1(\varphi)\} \cup \{\sigma_{\mathfrak{F}}(n)\}$, and $\mathfrak{R} = \{\sigma_d \mid d \in \Delta\} \cup \{\sigma_a \mid a \in N_1(\varphi)\} \cup \{\sigma_{\mathfrak{F}}\}$. We now show that (S, \mathfrak{R}) is a quasimodel for φ . The only critical point is the (\Leftarrow) direction of condition **Q6**. Here we use the concepts $A_a, \exists R.A_a$ introduced for each $a \in N_1(\varphi)$. We show the contrapositive, i.e., if $\neg R(a, b) \in \sigma_{\mathfrak{F}}(n)$, for some $n \in [0, l]$, then, $(\sigma_a(n), \sigma_b(n))$ is not R -compatible. If $\neg R(a, b) \in \sigma_{\mathfrak{F}}(n)$, then, $\mathfrak{F}, n \models \neg R(a, b)$. But then, $\neg \exists R.A_b \in \sigma_a(n)$, and, since $A_b \in \sigma_b(n)$, $(\sigma_a(n), \sigma_b(n))$ is not R -compatible.

⁵A concept/formula of the form $\neg(\alpha \mathcal{U} \beta)$ is not to be considered an until concept/formula. In particular, $\Box \perp$ is allowed in the last time point.

(\Leftarrow) Suppose there is a quasimodel (S, \mathfrak{R}) for φ , with $S = (S(0), \dots, S(l))$ and $l < k$. Define a finite trace $\mathfrak{F} = (\Delta, (\mathcal{F}_n)_{n \in [0, l]})$ as follows:

$$\begin{aligned}\Delta &= \{d_\sigma \mid \sigma \in \mathfrak{R}, \sigma \text{ concept run segment}\}; \\ A^{\mathcal{F}_n} &= \{d_\sigma \mid A \in \sigma(n), \sigma \in \mathfrak{R}\}; \\ R^{\mathcal{F}_n} &= \{(d_\sigma, d_{\sigma'}) \mid (\sigma(n), \sigma'(n)) \text{ are } R\text{-compatible}\}; \\ a^{\mathcal{F}} &= d_\sigma, \text{ for the unique } \sigma \in \mathfrak{R} \text{ with } a \in \sigma(0).\end{aligned}$$

By **Q2**, **R1**, and **M2**, $a^{\mathcal{F}}$ is well-defined. In order to show that \mathfrak{F} satisfies φ , we first show the following claim.

CLAIM 9. For all $\sigma \in \mathfrak{R}$, $C \in \text{cl}^c(\varphi)$ and $n \in [0, l]$, $C \in \sigma(n)$ iff $d_\sigma \in C^{\mathcal{F}_n}$.

PROOF OF CLAIM 9. The proof is by induction on the structure of C . For $C = A$, we have that $A \in \sigma(n)$ iff (by definition of \mathfrak{F}) $d_\sigma \in A^{\mathcal{F}_n}$. The cases $C = \neg C_1$ and $C = C_1 \sqcap C_2$ are straightforward, by using **T1** and **T2**, respectively. It remains to show the following cases.

- $C = \exists R.C_1$. (\Rightarrow) If $\exists R.C_1 \in \sigma(n)$, by **M2** and **Q5**, we have that there is $t' \in S(n)$ such that $C_1 \in t'$ and $(\sigma(n), t')$ is R -compatible. Thus, again by **M2**, there is $\sigma' \in \mathfrak{R}$ such that $C_1 \in \sigma'(n)$ and $(\sigma(n), \sigma'(n))$ is R -compatible. By i.h. and definition of \mathfrak{F} , we have $d_{\sigma'} \in C_1^{\mathcal{F}_n}$ and $(d_\sigma, d_{\sigma'}) \in R^{\mathcal{F}_n}$, i.e., $d_\sigma \in (\exists R.C_1)^{\mathcal{F}_n}$. (\Leftarrow) Suppose that $d_\sigma \in (\exists R.C_1)^{\mathcal{F}_n}$. Then d_σ has an R -successor $d_{\sigma'} \in C_1^{\mathcal{F}_n}$ and, by i.h., $C_1 \in \sigma'(n)$. By definition of $R^{\mathcal{F}_n}$, $(\sigma(n), \sigma'(n))$ are R -compatible. By absurd, assume that $\neg \exists R.C_1 \in \sigma(n)$, then by the R -compatibility, $\neg C_1 \in \sigma'(n)$, which is a contradiction.
- $C = C_1 \mathcal{U} C_2$. $C_1 \mathcal{U} C_2 \in \sigma(n)$ iff (by **R2**) there is $m \in (n, l]$ such that $C_2 \in \sigma(m)$ and for all $i \in (n, m)$, $C_1 \in \sigma(i)$. By i.h., there is $m \in (n, l]$ such that $d_\sigma \in C_2^{\mathcal{F}_m}$ and $d_\sigma \in C_1^{\mathcal{F}_i}$ for all $i \in (n, m)$, i.e., $d_\sigma \in (C_1 \mathcal{U} C_2)^{\mathcal{F}_n}$. \square

We now show that $\mathfrak{F}, 0 \models \varphi$, with $\varphi = \square^+(\top \sqsubseteq C_{\mathcal{T}}) \wedge \psi$. By **Q3**, **M2** and Claim (9), $\mathfrak{F}, 0 \models \square^+(\top \sqsubseteq C_{\mathcal{T}})$. It remains to show that $\mathfrak{F}, 0 \models \psi$, where ψ is a (possibly temporal) Boolean combination of assertions, denoted in the following as *assertion formula*. Let $\sigma_{\mathfrak{F}} \in \mathfrak{R}$ be a formula run segment, which by **Q1** and **M2** exists and is unique. It is enough to show the following claim.

CLAIM 10. For all assertion formulas $\psi \in \text{cl}^f(\varphi)$, and for all $n \in [0, l]$, $\psi \in \sigma_{\mathfrak{F}}(n)$ iff $\mathfrak{F}, n \models \psi$.

PROOF OF CLAIM 10. The proof is by induction on ψ .

- $\psi = A(a)$. By **Q4** and **M2**, $A(a) \in \sigma_{\mathfrak{F}}(n)$ iff $A \in \sigma_a(n)$ iff, by (9), $a^{\mathcal{F}} \in A^{\mathcal{F}_n}$, i.e., $\mathfrak{F}, n \models A(a)$.
- $\psi = R(a, b)$. By **Q6** and **M2**, $R(a, b) \in \sigma_{\mathfrak{F}}(n)$ iff $(\sigma_a(n), \sigma_b(n))$ is R -compatible. By \mathfrak{F} construction, $\mathfrak{F}, n \models R(a, b)$.
- The cases $\neg \chi$, $\chi \wedge \zeta$, $\chi \mathcal{U} \zeta$ are similar to Claim 9 by using **T1'**, **T2'**, and **R2**. \square

Therefore, by Claim 10 and **M1**, we can conclude that $\mathfrak{F}, 0 \models \psi$. This finishes the proof of Lemma 5.16. \square

Before presenting our algorithm we need the following definition. We say that a pair (t, t') of (concept/formula) types is *\mathcal{U} -compatible* if:

- $\alpha \mathcal{U} \beta \in t$ iff either $\beta \in t'$ or $\{\alpha, \alpha \mathcal{U} \beta\} \subseteq t'$, for all $\alpha \mathcal{U} \beta \in \text{cl}^*(\varphi)$,

where cl^* is either cl^c or cl^f (as appropriate).

Our type elimination algorithm iterates over the values in $[1, k - 1]$ to determine in exponential time in $|k|$, with k given in binary, the length of the sequence of quasistates of a quasimodel for

φ , if one exists. We assume that φ has the form $\Box^+(\tau \sqsubseteq C_{\mathcal{T}}) \wedge \psi$. For each $l \in [1, k - 1]$, the l -th iteration starts with sets:

$$S_0, \dots, S_{l-1}, S_l$$

and each S_i is initially set to $\text{tp}(\varphi)$. We start by exhaustively eliminating concept types t from some S_i , with $i \in [0, l]$, if t violates one of the following conditions:

- E1** for all $\exists R.D \in t$, there is $t' \in S_i$ such that $D \in t'$ and (t, t') is R -compatible;
- E2** if $i > 0$, then there is $t' \in S_{i-1}$ such that (t', t) is \mathcal{U} -compatible;
- E3** if $i < l$, then there is $t' \in S_{i+1}$ such that (t, t') is \mathcal{U} -compatible;
- E4** if $i = l$, then there is no $C\mathcal{U}D \in t$;
- E5** $C_{\mathcal{T}} \in t$.

For each $a \in N_1(\varphi)$, if t is a named type t^a then, in **E2** and **E3**, we further require that the mentioned types in a \mathcal{U} -compatible pair contain a . This phase of the algorithm stops when no further concept types can be eliminated. Next, for each formula type t , we say that a function f_t , mapping each $a \in N_1(\varphi)$ to a named type containing a , is *consistent with t* if: (i) for all $A(a) \in \text{cl}^f(\varphi)$, $A(a) \in t$ iff $A \in f_t(a)$; and (ii) for all $R(a, b) \in \text{cl}^f(\varphi)$, $R(a, b) \in t$ iff $(f_t(a), f_t(b))$ is R -compatible. We are going to use these functions to construct our quasimodel as follows. We first add to each S_i all f_t consistent with each formula type $t \in S_i$ such that the image of f_t is contained in S_i . We then exhaustively eliminate such functions f_t from some S_i , with $i \in [0, l]$, if f_t violates one of the following conditions:

- E1'** if $i < l$, then there is $f_{t'} \in S_{i+1}$ such that (t, t') is \mathcal{U} -compatible and, for all $a \in N_1(\varphi)$, $(f_t(a), f_{t'}(a))$ is \mathcal{U} -compatible;
- E2'** if $i = l$, then there is no $\alpha\mathcal{U}\beta \in t$.

It remains to ensure that each S_i contains exactly one formula type t_i and one named type t^a for each $a \in N_1(\varphi)$ (and no functions f_t). For this choose any formula type function f_{t_0} in S_0 such that $\psi \in t_0$ (if one exists) and remove formula types $t'_0 \neq t_0$ from S_0 . Then, for each $i \in [1, l]$, select a formula type function $f_{t_i} \in S_i$ such that (t_{i-1}, t_i) is \mathcal{U} -compatible and for all $a \in N_1(\varphi)$, $(f_{t_{i-1}}(a), f_{t_i}(a))$ is \mathcal{U} -compatible, removing formula types $t'_i \neq t_i$ from S_i , where f_{t_i} is the selected function. The existence of such f_{t_i} is ensured by **E1'**. For each selected function f_{t_i} and each $a \in N_1(\varphi)$, with $i \in [1, l]$, we remove from S_i all named types t^a such that $t^a \neq f_{t_i}(a)$. We now have that each S_i contains exactly one formula type t_i and one named type t^a for each $a \in N_1(\varphi)$. Finally, we proceed removing all functions f_t . We have thus constructed a sequence of quasistates. Until concepts/formulas $\alpha\mathcal{U}\beta$ are satisfied thanks to the \mathcal{U} -compatibility conditions and the fact that there are no expressions of the form $\alpha\mathcal{U}\beta$ in concept/formula types in the last quasistate.

This last step does not affect conditions **E1-E5** (in particular **E1**) for the remaining concept types since for each named type there is an unnamed (concept) type which is the result of removing the individual name from it, and if the named type was not removed during type elimination then the corresponding unnamed type was also not removed. If the algorithm succeeds on these steps with a surviving concept type $t \in S_0$ and a formula type t_{S_0} in S_0 such that $\psi \in t_{S_0}$ then it returns “satisfiable”. Otherwise, it increments l or returns “unsatisfiable” if $l = k - 1$ (i.e., there are no further iterations).

LEMMA 5.17. *The type elimination algorithm returns “satisfiable” iff there is a quasimodel for φ .*

PROOF. For (\Rightarrow) , let $S^* = S_0^*, \dots, S_l^*$ be the result of the type elimination procedure. Define (S^*, \mathfrak{R}) with \mathfrak{R} as the set of sequences σ of (concept/formula) types such that, for all $i \in [0, l]$:

- (1) $\sigma(i) \in S_i^*$, and for every $t \in S_i^*$, there is $\sigma \in \mathfrak{R}$ with $\sigma(i) = t$;
- (2) for all $a \in N_1(\varphi)$, we have $a \in \sigma(0)$ iff $a \in \sigma(i)$;

- (3) for all $\alpha \mathcal{U} \beta \in \text{cl}^*(\varphi)$, we have $\alpha \mathcal{U} \beta \in \sigma(i)$ iff there is $j \in (i, l]$ such that $\beta \in \sigma(j)$ and $\alpha \in \sigma(n)$ for all $n \in (i, j)$.

where cl^* is either cl^c or cl^f (as appropriate). We now argue that (S^*, \mathfrak{R}) is a quasimodel for φ . We first argue that S^* is a sequence of quasistates for φ . **E1** ensures Condition **Q5**, while Condition **Q3** is guaranteed by Condition **E5**. For Conditions **Q4** and **Q6**, we have the fact that named types are taken from functions consistent with the formula types. The last step of our algorithm consists in eliminating formula and named types so that we satisfy Conditions **Q1** and **Q2**. Thus, S^* is a sequence of quasistates for φ . Concerning the construction of \mathfrak{R} , Point (2) can be enforced thanks to our selection procedure for named types, which enforces \mathcal{U} -compatibility, while Point (3) is a consequence of

- Conditions **E2**, **E3** and **E4**, for concept types; and
- Conditions **E1'** and **E2'**, for formula types, together with the selection procedure.

Points (2)–(3) coincide with conditions **R1** and **R2**, so \mathfrak{R} is a set of run segments for φ . Finally, Point (1) ensures that condition **M2** holds, and thus, when the algorithm returns “satisfiable”, also condition **M1** holds. Thus, (S^*, \mathfrak{R}) is a quasimodel for φ .

For the other direction (\Leftarrow), assume there is a quasimodel (S, \mathfrak{R}) for φ . Assume S is of the form $S_0 \dots S_{l-1} S_l$, for some $l \in [1, k-1]$. Let S_0^*, \dots, S_l^* be the result of the type elimination at the l th iteration. Since (S, \mathfrak{R}) is a quasimodel, each concept type satisfies **E1**. Moreover, conditions **E2**–**E5** are consequences of the existence of run segments through each type (by **M2**). Then, for all unnamed (concept) types t , if $t \in S_i$ then $t \in S_i^*$, $i \in [0, l]$. If t is a formula type or a named type then $t \in S_i$ does not necessarily imply that $t \in S_i^*$, $i \in [0, l]$. However, the existence of such types implies that the algorithm should find a sequence of functions f_{t_i} , for $i \in [0, l]$, satisfying **E1'** and **E2'** which is then used to select formula and named types satisfying the quasimodel conditions. In particular, due to **M1**, the selection procedure will select a function f_{t_0} associated with a formula type $t_0 \in S_0^*$ containing ψ . So there is a surviving formula type in S_0^* containing φ and the algorithm returns “satisfiable”.

This finishes the proof of Lemma 5.17. \square

We now argue that our type elimination algorithm runs in exponential time. Since there are polynomially many individuals (with respect to the size of φ) occurring in φ , the number of functions f_t consistent with a formula type is exponential. As the number of (concept/formula) types is exponential the total number of functions and types to consider is exponential. In every step some concept type or function is eliminated (by **E1**–**E5** or by **E1'**–**E2'**, respectively). Conditions **E1**–**E5** and **E1'**–**E2'** can clearly be checked in exponential time. Also, the selection procedure of functions for each S_i , which determine the formula and named types in the result of the algorithm, can also be checked in exponential time, since we can pick any function in S_{i+1} satisfying the \mathcal{U} -compatibility relation, which is a local condition. As this can also be implemented in exponential time, this concludes the proof of Theorem 5.15. \square

We leave the complexity of the satisfiability problem on *finite traces* for $T_{\mathcal{U}}\mathcal{ALC}$ restricted to global CIs as an open problem. It is known that the complexity of the satisfiability problem in this fragment over *infinite* traces is EXPTIME -complete [16, 84]. However, the end of time formula ψ_f is not expressible in this fragment. Thus, we cannot use the same strategy of defining a translation for the semantics based on infinite traces, as we did in Section 4.1. Moreover, the upper bound in [84] is based on type elimination. The main difficulty in devising a type elimination procedure in the case of arbitrary finite traces is that the number of time points is not fixed and the argument in [84], showing that there is a quasimodel iff there is a quasimodel (S, \mathfrak{R}) such that $S(i+1) \subseteq S(i)$, for all $i \geq 0$, is not applicable to finite traces. A type with a concept equivalent to *last* can only

Table 2. Complexity results for formula satisfiability in $T_{\mathcal{U}}\mathcal{QL}$ fragments, both on finite and on bounded traces (with the bound given in binary as part of the input)

	Satisfiability problem	Finite traces	Bounded traces
$T_{\mathcal{U}}\mathcal{QL}_{\neq}^{1,mo}, T_{\mathcal{U}}\mathcal{QL}^1, T_{\mathcal{U}}\mathcal{QL}_{\square}^{mo}, T_{\mathcal{U}}\mathcal{QL}_{\square}^2$	Formula	EXPSpace	NEXPTIME
$T_{\mathcal{U}}\mathcal{ALC}$	Formula	EXPSpace	NEXPTIME
	Formula restricted to global CIs	?	EXPTIME

be in the last quasistate of the quasimodel. Therefore, it is not clear whether one can show that if there is a quasimodel, then there is a quasimodel with an exponential sequence of quasistates, as done in Theorem 5.15. Table 2 summarises the complexity results obtained in this section.

6 FINITE TRACES IN PLANNING AND VERIFICATION

Understanding the connections between finite and infinite traces is of interest to several applications. In the following, we focus on planning and verification. First, we lift to the first-order temporal logic setting the *LTL* notion of insensitivity to infiniteness [47], introduced in the planning domain. Then, we discuss how, in *LTL*, the concepts of safety, as well as impartiality and anticipation [29], can be related to the semantic properties of Section 4.2 for bridging finite and infinite traces.

6.1 Planning

In automated planning, the sequence of states generated by actions is usually finite [28, 42, 47, 48]. To reuse temporal logics based on infinite traces for specifying plan constraints, one approach, developed by De Giacomo et al. 2014a for *LTL* on finite traces, 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 $a \in \mathbb{N}_1$, and for all $P \in \mathbb{N}_P \setminus \{E\}$, $P^{\mathfrak{E}_n} = \emptyset$, for any $n \in [0, \infty)$. The end extension (cf. Section 4.1) 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 α , $\mathfrak{F} \models^{\alpha} \varphi$ iff $\mathfrak{F} \cdot_E \mathfrak{E} \models^{\alpha} \varphi$. Clearly, all insensitive $T_{\mathcal{U}}\mathcal{QL}$ formulas are also $F_{\Rightarrow \exists}$, whereas $\diamond \neg P$ is an example of a (*LTL*) formula that is $F_{\Rightarrow \exists}$ but not insensitive. Moreover, let Σ be a finite subset of \mathbb{N}_P such that $E \in \Sigma$. Assume without loss of generality that the $T_{\mathcal{U}}\mathcal{QL}$ formulas we mention in this subsection have predicates in Σ . Given an infinite trace \mathfrak{S} , the Σ -*reduct* of \mathfrak{S} is the infinite trace $\mathfrak{S}|_{\Sigma}$ coinciding with \mathfrak{S} on Σ and such that $X^{\mathfrak{S}|_{\Sigma}} = \emptyset$, for $X \notin \Sigma$ and $n \in [0, \infty)$. Finally, recalling the definition of ψ_f , we define $\vartheta_f = \psi_f \wedge \chi_f$, with

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

Before we proceed with a formal characterisation of insensitive formulas, we require the following preliminary lemmas.

LEMMA 6.1. *For every infinite trace \mathfrak{S} , $\mathfrak{S} \models \vartheta_f$ iff $\mathfrak{S}|_{\Sigma} = \mathfrak{F} \cdot_E \mathfrak{E}$, for some finite trace \mathfrak{F} .*

PROOF. (\Leftarrow) If $\mathfrak{S}_{\lfloor \Sigma} = \mathfrak{F} \cdot_E \mathfrak{G}$, by Lemma 4.1, $\mathfrak{F} \cdot_E \mathfrak{G} \models \psi_f$. Moreover, where l is the last time point of \mathfrak{F} , we have by definition: for all $n \in [0, l]$, $E^{\mathcal{F} \cdot_E \mathcal{E}^n} = \emptyset$; for all $n \in [l + 1, \infty)$, $E^{\mathcal{F} \cdot_E \mathcal{E}^n} = \Delta$, and $P^{\mathcal{F} \cdot_E \mathcal{E}^n} = \emptyset$, for every $P \in \text{Np} \setminus \{E\}$. Thus, for all $n \in (0, \infty)$, for all objects d and all tuples of objects \vec{d} in Δ , we have: if $d \in E^{\mathcal{F} \cdot_E \mathcal{E}^n}$, then $(d, \vec{d}) \notin P$, for all $P \in \Sigma \setminus \{E\}$. Therefore, $\mathfrak{F} \cdot_E \mathfrak{G} \models \chi_f$, and hence $\mathfrak{F} \cdot_E \mathfrak{G} \models \vartheta_f$.

(\Rightarrow) Suppose $\mathfrak{S} \models \vartheta_f$. Since in particular it satisfies ψ_f , by Lemma 4.1, we have that $\mathfrak{S} = \mathfrak{F} \cdot_E \mathfrak{S}'$, for some finite trace $\mathfrak{F} = (\Delta, (\mathcal{F}_n)_{n \in [0, l]})$ and some infinite trace \mathfrak{S}' . Thus:

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

Since $\mathfrak{S} \models \chi_f$, for all $n \in [l + 1, \infty)$, for all objects d and all tuples of objects \vec{d} in Δ , we have that: if $d \in E^{\mathcal{F} \cdot \mathcal{I}'_n} = \Delta$, then $(d, \vec{d}) \notin P$, for all $P \in \Sigma \setminus \{E\}$. This is equivalent to $P^{\mathcal{I}'_n} = \emptyset$, for all $P \in \Sigma \setminus \{E\}$ and $n \in [l + 1, \infty)$. Therefore, we have that $\mathfrak{S}_{\lfloor \Sigma} = \mathfrak{F} \cdot_E \mathfrak{G}$. \square

LEMMA 6.2. *Let φ be a $T_{\mathcal{U}}\mathcal{QL}$ formula, \mathfrak{F} a finite trace, and α an assignment. We have that $\mathfrak{F} \models^\alpha$ iff $\mathfrak{F} \cdot_E \mathfrak{G} \models^\alpha \varphi^\dagger$.*

PROOF. By definition of $\mathfrak{F} \cdot_E \mathfrak{G}$ and as a consequence of Lemma 4.2. \square

We can now state the following characterisation result for insensitive formulas, which extends [47, Theorem 4] to the first-order language $T_{\mathcal{U}}\mathcal{QL}$.

THEOREM 6.3. *A $T_{\mathcal{U}}\mathcal{QL}$ formula φ is insensitive to infiniteness iff $\vartheta_f \models_i \varphi \leftrightarrow \varphi^\dagger$.*

PROOF. (\Rightarrow) Assume that φ is insensitive. We want to prove that, for every infinite trace \mathfrak{S} and all assignments α , if $\mathfrak{S} \models^\alpha \vartheta_f$, then $\mathfrak{S} \models^\alpha \varphi \leftrightarrow \varphi^\dagger$. Suppose $\mathfrak{S} \models^\alpha \vartheta_f$. By Lemma 6.1, $\mathfrak{S}_{\lfloor \Sigma} = \mathfrak{F} \cdot_E \mathfrak{G}$, for a finite trace \mathfrak{F} . Moreover, thanks to Lemma 6.2, $\mathfrak{F} \models^\alpha \varphi$ iff $\mathfrak{F} \cdot_E \mathfrak{G} \models^\alpha \varphi^\dagger$. Since φ is by hypothesis insensitive, for every finite trace \mathfrak{F} and all assignments α , $\mathfrak{F} \models^\alpha \varphi$ iff $\mathfrak{F} \cdot_E \mathfrak{G} \models^\alpha \varphi$. Thus, $\mathfrak{F} \cdot_E \mathfrak{G} \models^\alpha \varphi$ if and only if $\mathfrak{F} \cdot_E \mathfrak{G} \models^\alpha \varphi^\dagger$. That is, $\mathfrak{F} \cdot_E \mathfrak{G} \models^\alpha \varphi \leftrightarrow \varphi^\dagger$, and therefore $\mathfrak{S} \models^\alpha \varphi \leftrightarrow \varphi^\dagger$ (since all the predicates occurring in φ, φ^\dagger are in Σ).

(\Leftarrow) Assume that $\vartheta_f \models \varphi \leftrightarrow \varphi^\dagger$. By Lemma 6.1, for every infinite trace \mathfrak{S} and every assignment α , $\mathfrak{S} \models^\alpha \vartheta_f$ means that $\mathfrak{S}_{\lfloor \Sigma} = \mathfrak{F} \cdot_E \mathfrak{G}$, for a finite trace \mathfrak{F} . Given our assumption, this implies $\mathfrak{F} \cdot_E \mathfrak{G} \models^\alpha \varphi \leftrightarrow \varphi^\dagger$, that is $\mathfrak{F} \cdot_E \mathfrak{G} \models^\alpha \varphi$ iff $\mathfrak{F} \cdot_E \mathfrak{G} \models^\alpha \varphi^\dagger$, for all assignments α . By Lemma 6.2, $\mathfrak{F} \cdot_E \mathfrak{G} \models^\alpha \varphi^\dagger$ if and only if $\mathfrak{F} \models^\alpha \varphi$. In conclusion, we obtain that, for all assignments α , $\mathfrak{F} \cdot_E \mathfrak{G} \models^\alpha \varphi$ iff $\mathfrak{F} \models^\alpha \varphi$, meaning that φ is insensitive. \square

We now analyse syntactic features of insensitive formulas. Firstly, non-temporal $T_{\mathcal{U}}\mathcal{QL}$ formulas are insensitive. Moreover, this property is preserved under non-temporal operators. We generalise [47, Theorem 5] in our setting as follows.

THEOREM 6.4. *Let φ, ψ be insensitive formulas. Then $\neg\varphi$, $\exists x\varphi$, and $\varphi \wedge \psi$ are insensitive.*

PROOF. Let \mathfrak{F} be a finite trace and α be an assignment. For $\neg\varphi$, we have that $\mathfrak{F} \models^\alpha \neg\varphi$ iff $\mathfrak{F} \not\models^\alpha \varphi$. Since φ is insensitive by hypothesis, this means that $\mathfrak{F} \cdot_E \mathfrak{G} \not\models^\alpha \varphi$. Therefore, $\neg\varphi$ is insensitive as well. For $\exists x\varphi$, we have that $\mathfrak{F} \models^\alpha \exists x\varphi$ iff $\mathfrak{F} \models^{\alpha[x \mapsto d]} \varphi$, for some $d \in \Delta$. Given that φ is insensitive, this is equivalent to $\mathfrak{F} \cdot_E \mathfrak{G} \models^{\alpha[x \mapsto d]} \varphi$, for some $d \in \Delta$. That is, $\mathfrak{F} \cdot_E \mathfrak{G} \models^\alpha \exists x\varphi$, and so $\exists x\varphi$ is insensitive. For $\varphi \wedge \psi$, we have that $\mathfrak{F} \models^\alpha \varphi \wedge \psi$ is equivalent to $\mathfrak{F} \models^\alpha \varphi$ and $\mathfrak{F} \models^\alpha \psi$. Since both φ and ψ are assumed to be insensitive, the previous step is equivalent to: $\mathfrak{F} \cdot_E \mathfrak{G} \models^\alpha \varphi$ and $\mathfrak{F} \cdot_E \mathfrak{G} \models^\alpha \psi$, i.e., $\mathfrak{F} \cdot_E \mathfrak{G} \models^\alpha \varphi \wedge \psi$. \square

Concerning temporal operators, in [47] it is shown how several standard temporal patterns derived from the declarative process modelling language `DECLARE` [2] are insensitive. On the other hand, negation affects the insensitivity of temporal formulas. For instance, given a non temporal $T\mathcal{U}\mathcal{Q}\mathcal{L}$ formula ψ , we have that $\diamond^+\psi$ is insensitive while $\diamond^+\neg\psi$ is not. Dually, $\square^+\neg\psi$ is insensitive, while $\square^+\psi$ is not. Therefore, if a $T\mathcal{U}\mathcal{Q}\mathcal{L}$ formula φ is insensitive, it cannot be concluded that formulas of the form $\diamond^+\varphi$ or $\square^+\varphi$ are insensitive.

Finally, we have that insensitivity is sufficient to ensure that if formulas are equivalent on infinite traces, then they are equivalent on finite traces.

THEOREM 6.5. *For all insensitive formulas φ, ψ , $\varphi \equiv_i \psi$ implies $\varphi \equiv_f \psi$.*

PROOF. Given a finite trace \mathfrak{F} and an assignment α , if $\mathfrak{F} \models^\alpha \varphi$ then, as φ is insensitive, $\mathfrak{F} \cdot_E \mathbb{C} \models^\alpha \varphi$. By assumption, $\varphi \equiv_i \psi$, so $\mathfrak{F} \cdot_E \mathbb{C} \models^\alpha \psi$. As ψ is insensitive, $\mathfrak{F} \cdot_E \mathbb{C} \models^\alpha \psi$ implies $\mathfrak{F} \models^\alpha \psi$. The converse direction is obtained by swapping φ and ψ . \square

Since \perp is insensitive, we obtain the following immediate corollary of the previous result.

COROLLARY 6.6. *All insensitive formulas satisfiable on finite traces are satisfiable on infinite traces.*

However, the converse directions of the above results do not hold, as witnessed, e.g., by formula ϑ_i (cf. Section 5.3), which is trivially insensitive, but satisfiable only on infinite traces. We can obtain the converse directions by using our Theorem 4.8. For instance, $\diamond^+P(x) \vee \diamond^+R(x)$ and $\diamond^+(P(x) \vee R(x))$ are insensitive and \sqcup formulas for which equivalence on finite and infinite traces coincide.

6.2 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. In particular, we establish connections between the finite and infinite trace properties, introduced in Section 4.2, and: (i) the definition of safety in *LTL* on infinite traces [24, 95]; (ii) maxims related to monitoring procedures in runtime verification [29].

6.2.1 Safety. Recall that a safety property intuitively guarantees 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 [24, 95]. In the rest of this section, we will thus restrict ourselves to *LTL*. A typical example of an *LTL* formula used to specify a safety property is represented by $\square\neg P$, where P is an atom standing for an action or task. Dual to safety properties are *co-safety properties*, expressing that “good things” will eventually happen in the execution of a program. The *LTL* formula $\diamond P$ is a standard example of a formula specifying a co-safety property.

To fix notions that will be used in the rest of this section, we start by recalling the definitions of *safety* and *co-safety fragments* of *LTL*. The *LTL safety formulas* [95] are defined as the *LTL* formulas obtained from \perp , \top , and *literals* (i.e., *propositional letters* P , or negated propositional letters $\neg P$), by applying conjunction \wedge , disjunction \vee , strong next \circ , and reflexive release \mathcal{R}^+ operators. The *LTL co-safety formulas* [81] are dually defined as those *LTL* formulas obtained from \perp , \top , and *literals*, by applying conjunction \wedge , disjunction \vee , strong next \circ , and reflexive until \mathcal{U}^+ operators. It is known [40, 81, 95] that every safety formula φ *expresses a safety property*, i.e., for every infinite trace \mathfrak{S} such that $\mathfrak{S} \not\models \varphi$, there exists $\mathfrak{F} \in \text{Pre}(\mathfrak{S})$ so that, for all $\mathfrak{S}' \in \text{Ext}(\mathfrak{F})$, it holds that $\mathfrak{S}' \not\models \varphi$. We call such a finite trace \mathfrak{F} a *bad prefix* for φ , and we define $\text{BadPre}(\varphi)$ as the set of bad prefixes for φ . On the other hand, every co-safety formula φ *expresses a co-safety property*, i.e., for every infinite trace \mathfrak{S} satisfying φ , there is a *good prefix* for φ , that is, a finite prefix \mathfrak{F} of \mathfrak{S} such that every infinite extension \mathfrak{S}' of \mathfrak{F} satisfies φ . Given the equivalence, $\circ\neg\varphi \equiv_i \neg\circ\varphi$, holding on infinite traces, we

have that the *LTL* formulas that are, respectively, in the \mathcal{R} - and \mathcal{U} -fragments defined in Section 4.2 express, respectively, safety and co-safety properties.

In order to establish connections between safety properties and finite traces semantics, we now require, following [24], further definitions and notation. Let N_p^0 be the subset of N_p containing 0-ary predicates, i.e., propositional letters. Given a suborder \mathfrak{T} of $(\mathbb{N}, <)$ of the form $[0, \infty)$ or $[0, l]$, with $l \in \mathbb{N}$, a *trace* is now viewed simply as a sequence $\mathfrak{M} = (M_n)_{n \in \mathfrak{T}}$ with $M_n \in 2^{N_p^0}$. The notion of an *LTL* formula φ being satisfied in a trace \mathfrak{M} , $\mathfrak{M} \models \varphi$, is given similarly as above (cf. Section 3.2), and we let $\text{Trace}_i(\varphi)$, respectively, $\text{Trace}_f(\varphi)$, be the set of infinite, respectively, finite, traces satisfying φ . The sets of prefixes of a trace \mathfrak{M} and of extensions of a finite trace \mathfrak{F} are defined as in Section 3.2. Moreover, given a set of infinite traces, C , and of finite traces, \mathcal{D} , we define

$$\text{Pre}(C) = \bigcup_{\mathfrak{S} \in C} \text{Pre}(\mathfrak{S}), \quad \text{Ext}(\mathcal{D}) = \bigcup_{\mathfrak{F} \in \mathcal{D}} \text{Ext}(\mathfrak{F}).$$

For an *LTL* formula φ , we may write $\text{Pre}(\varphi)$ and $\text{Ext}(\varphi)$ in place of, respectively, $\text{Pre}(\text{Trace}_i(\varphi))$ and $\text{Ext}(\text{Trace}_f(\varphi))$. Moreover, given a set of infinite traces C , we define the *closure* of C as $\text{Clo}(C) = \{\mathfrak{S} \mid \text{Pre}(\mathfrak{S}) \subseteq \text{Pre}(C)\}$. Consider, for instance, the set of infinite traces $C = \bigcup_{i \in [0, \infty)} \{\mathfrak{S}^i\}$, where $\mathfrak{S}^i = (I_n^i)_{n \in [0, \infty)}$ is such that $I_n^i = \{P\}$, if $i = n$, and $I_n^i = \emptyset$, otherwise. We have that the infinite trace $\mathfrak{S} = (I_n)_{n \in [0, \infty)}$ defined so that $I_n = \emptyset$, for every $i \in [0, \infty)$, is such that $\mathfrak{S} \notin C$, whereas $\mathfrak{S} \in \text{Clo}(C)$. The following preliminary lemma adapts to our setting a result presented in [24, Lemma 3.25].

LEMMA 6.7. *Let φ, χ be LTL formulas, with $\varphi \in \text{LTL}(F_{\exists})$ and χ expressing a safety property. It holds that:*

$$\varphi \models_i \chi \quad \text{iff} \quad \text{Trace}_f(\varphi) \cap \text{BadPre}(\chi) = \emptyset.$$

PROOF. (\Rightarrow) By contraposition, assume there exists $\mathfrak{F} \in \text{Trace}_f(\varphi) \cap \text{BadPre}(\chi)$. We have that $\mathfrak{F} \models \varphi$, and since φ is F_{\exists} , there exists $\mathfrak{S} \in \text{Ext}(\mathfrak{F})$ such that $\mathfrak{S} \models \varphi$. Given that $\mathfrak{F} \in \text{BadPre}(\chi)$ and that $\mathfrak{F} \in \text{Pre}(\mathfrak{S})$, we have in particular that $\mathfrak{S} \not\models \chi$. Therefore, $\varphi \not\models_i \chi$.

(\Leftarrow) By contraposition, assume that $\varphi \not\models_i \chi$, i.e., there exists an infinite trace \mathfrak{S} such that $\mathfrak{S} \models \varphi$ and $\mathfrak{S} \not\models \chi$. Since χ expresses a safety property, there exists $\hat{\mathfrak{F}} \in \text{Pre}(\mathfrak{S})$ such that, for all $\mathfrak{S}' \in \text{Ext}(\hat{\mathfrak{F}})$, $\mathfrak{S}' \not\models \chi$. Thus, $\hat{\mathfrak{F}} \in \text{BadPre}(\chi)$. Moreover, since from Proposition 4.6, Point (3), it follows that φ is F_{\exists} iff φ is I_{\forall} , we have that every $\mathfrak{F} \in \text{Pre}(\mathfrak{S})$ is such that $\mathfrak{F} \models \varphi$. Hence, $\hat{\mathfrak{F}} \in \text{Trace}_f(\varphi)$, and so $\text{Trace}_f(\varphi) \cap \text{BadPre}(\chi) \neq \emptyset$. \square

Using the previous result, we can show the following characterisation of two *LTL* formulas being equivalent on finite traces in terms of safety properties, under the assumption that they satisfy the property F_{\exists} , as it is the case for *LTL* \mathcal{R}^+ -formulas (cf. Section 4.2). This characterisation mirrors the one given in [24, Corollary 3.29], but it is presented here in terms of finite traces and *LTL* formulas, as opposed to the one in [24] based on sets of traces and transition systems.

PROPOSITION 6.8. *For every $\varphi, \psi \in \text{LTL}(F_{\exists})$, the following are equivalent:*

- (i) $\varphi \equiv_f \psi$;
- (ii) for every *LTL* formula χ expressing a safety property, $\varphi \models_i \chi$ iff $\psi \models_i \chi$.

PROOF. We are going to show that the following statements are equivalent:

- (i') $\text{Trace}_f(\varphi) \subseteq \text{Trace}_f(\psi)$;
- (ii') for every *LTL* formula χ expressing a safety property, $\psi \models_i \chi$ implies $\varphi \models_i \chi$.

Then, by swapping φ and ψ , the required result will follow.

(i') \Rightarrow (ii') Assume that $\text{Trace}_f(\varphi) \subseteq \text{Trace}_f(\psi)$ and that, for every *LTL* formula χ expressing a safety property, $\psi \models_i \chi$. By Lemma 6.7, this is equivalent to $\text{Trace}_f(\psi) \cap \text{BadPre}(\chi) = \emptyset$. Since

$Trace_f(\varphi) \subseteq Trace_f(\psi)$, we have $Trace_f(\varphi) \cap BadPre(\chi) = \emptyset$, which means, again by Lemma 6.7, that $\varphi \models_i \chi$.

(ii') \Rightarrow (i') Assume that, for every LTL formula χ expressing a safety property, $\psi \models_i \chi$ implies $\varphi \models_i \chi$ and consider the set $Clo(Trace_i(\psi))$. It is known [24] that the closure $Clo(C)$ of a set of infinite traces C corresponds to the *topological closure* of C in the topological space $((2^{\mathbb{N}_p^0})^\omega, \tau)$, where τ is the topology induced by the metric $\delta: (2^{\mathbb{N}_p^0})^\omega \rightarrow \mathbb{R}_0$ (where \mathbb{R}_0 is the set of non-negative real numbers), defined as follows: $\delta(\mathfrak{S}, \mathfrak{S}) = 0$; $\delta(\mathfrak{S}, \mathfrak{S}') = 2^{-n}$, with $n = \min\{m \in \mathbb{N} \mid I_m \neq I'_m\}$, for $\mathfrak{S} \neq \mathfrak{S}'$. Moreover, as shown in [88, Corollary 14], the following holds.

CLAIM 11. For every LTL formula φ , there is an LTL formula φ' such that $Clo(Trace_i(\varphi)) = Trace_i(\varphi')$.

We now require the following claim.

CLAIM 12. For every LTL formula χ , χ expresses a safety property iff $Clo(Trace_i(\chi)) = Trace_i(\chi)$.

PROOF OF CLAIM 12. Following [24], we call *safety property* a set of infinite traces \mathcal{S} such that, if $\mathfrak{S} \notin \mathcal{S}$, then there is $\mathfrak{F} \in Pre(\mathfrak{S})$ such that, for every $\mathfrak{S}' \in Ext(\mathfrak{F})$, it holds that $\mathfrak{S}' \notin \mathcal{S}$. From the definitions, we have that χ expresses a safety property iff $Trace_i(\chi)$ is a safety property. It is known that safety properties coincide with closed sets in the topological space $((2^{\mathbb{N}_p^0})^\omega, \tau)$ defined above [24]. The claim then follows from the fact that a set is closed in a topological space iff it is equal to its topological closure, and the fact that the topological closure of a set \mathcal{S} in $((2^{\mathbb{N}_p^0})^\omega, \tau)$ coincides with $Clo(\mathcal{S})$. \square

We can now finish the proof of the (ii') \Rightarrow (i') direction. By Claim 11, we have that there exists an LTL formula ψ' such that $Clo(Trace_i(\psi)) = Trace_i(\psi')$. Since $Clo(\cdot)$ is a closure operator, we have that

$$Clo(Trace_i(\psi')) = Clo(Clo(Trace_i(\psi))) = Clo(Trace_i(\psi)) = Trace_i(\psi').$$

Thus, by Claim 12, we have that ψ' expresses a safety property. Moreover, the fact that $Clo(\cdot)$ is a closure operator implies also that $Trace_i(\psi) \subseteq Clo(Trace_i(\psi)) = Trace_i(\psi')$, thus $\psi \models_i \psi'$. From (ii'), we obtain that $\varphi \models_i \psi'$. Thus, $Trace_i(\varphi) \subseteq Trace_i(\psi')$ and, by Claim 11, $Trace_i(\varphi) \subseteq Clo(Trace_i(\psi))$. Since $Pre(\cdot)$ is monotonic, i.e., for every set of infinite traces C, C' such that $C \subseteq C'$, we have $Pre(C) \subseteq Pre(C')$, it follows that $Pre(Trace_i(\varphi)) \subseteq Pre(Clo(Trace_i(\psi)))$. Moreover, since φ is $F \Rightarrow \exists$, it holds that $Trace_f(\varphi) \subseteq Pre(Trace_i(\varphi))$, and since by Proposition 4.6, Point (3), we have that ψ is $F \Leftarrow \exists$ iff ψ is $I \Rightarrow \forall$, we obtain $Pre(Trace_i(\psi)) \subseteq Trace_f(\psi)$. Finally, it can be seen that $Pre(Clo(Trace_i(\psi))) = Pre(Trace_i(\psi))$. Hence, we obtain:

$$Trace_f(\varphi) \subseteq Pre(Trace_i(\varphi)) \subseteq Pre(Clo(Trace_i(\psi))) = Pre(Trace_i(\psi)) \subseteq Trace_f(\psi). \quad \square$$

6.2.2 Runtime Verification Maxims. 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 [21, 29, 46]. Here we discuss the relationship between our semantic conditions and the *maxims* for runtime verification in (variants of) LTL introduced by Bauer et al. 2010, which relate finite trace semantics to the infinite case. The authors suggest that any LTL semantics to be used in runtime verification should satisfy, for every LTL formula φ , the maxims of *impartiality* and *anticipation*, defined as follows.

Impartiality For every finite trace \mathfrak{F} ,

$$\mathfrak{F} \models \varphi \Rightarrow \forall \mathfrak{S} \in Ext(\mathfrak{F}). \mathfrak{S} \models \varphi \quad \text{and} \quad \mathfrak{F} \not\models \varphi \Rightarrow \forall \mathfrak{S} \in Ext(\mathfrak{F}). \mathfrak{S} \not\models \varphi.$$

Anticipation For every finite trace \mathfrak{F} ,

$$\mathfrak{F} \models \varphi \Leftarrow \forall \mathfrak{S} \in \text{Ext}(\mathfrak{F}). \mathfrak{S} \models \varphi \quad \text{and} \quad \mathfrak{F} \not\models \varphi \Leftarrow \forall \mathfrak{S} \in \text{Ext}(\mathfrak{F}). \mathfrak{S} \not\models \varphi.$$

It can be seen that impartiality is captured by $LTL(F_{\Rightarrow\forall}) \cap LTL(F_{\Leftarrow\exists})$, while anticipation corresponds to $LTL(F_{\Leftarrow\forall}) \cap LTL(F_{\Rightarrow\exists})$. Therefore, any set of LTL formulas satisfying both impartiality and anticipation is included in the intersection $LTL(F_{\forall}) \cap LTL(F_{\exists})$.

To be able to satisfy these maxims for runtime verification purposes, [Bauer et al. 2010](#) consider non-standard semantics for LTL that allow for more than two truth-values. Indeed, under a two-valued semantics, they show that LTL on finite traces does not satisfy, for every LTL formula φ , impartiality and anticipation. An example of a formula that does not satisfy impartiality is \Box^+P , whereas $\Diamond\top$ violates anticipation. Furthermore, it can be seen that any LTL formula φ satisfying impartiality, i.e., such that $\varphi \in LTL(F_{\Rightarrow\forall}) \cap LTL(F_{\Leftarrow\exists})$, evaluated under a two-valued semantics, essentially expresses a non-temporal property, in the following sense (recall that, for a finite or infinite trace \mathfrak{M} , we denote by \mathfrak{M}_0 the prefix of \mathfrak{M} ending at time point 0).

PROPOSITION 6.9. *Given $\varphi \in LTL$, the following conditions are equivalent:*

- (1) φ satisfies impartiality;
- (2) for any (finite or infinite) trace \mathfrak{M} , $\mathfrak{M} \models \varphi$ iff $\mathfrak{M}_0 \models \varphi$.

PROOF. First, observe that any LTL formula φ satisfying Condition (2) clearly satisfies impartiality, i.e., $\varphi \in LTL(F_{\Rightarrow\forall}) \cap LTL(F_{\Leftarrow\exists})$. Conversely, assume that φ satisfies impartiality, and recall that, by Proposition 4.6, Point (3), we have $LTL(F_{\Leftarrow\exists}) = LTL(I_{\Rightarrow\forall})$. Moreover, let \mathfrak{M} be a (finite or infinite) trace.

From left-to-right, suppose that $\mathfrak{M} \models \varphi$. If $\mathfrak{M} = \mathfrak{S}$, for an infinite trace \mathfrak{S} , by the assumption that $\varphi \in LTL(I_{\Rightarrow\forall})$, we obtain in particular $\mathfrak{S}_0 \models \varphi$. If $\mathfrak{M} = \mathfrak{F}$, for a finite trace \mathfrak{F} , from the assumption that $\varphi \in LTL(F_{\Rightarrow\forall})$, we obtain that $\mathfrak{S}' \models \varphi$, for an arbitrary $\mathfrak{S}' \in \text{Ext}(\mathfrak{F})$. Since $\varphi \in LTL(I_{\Rightarrow\forall})$, this implies that $\mathfrak{F}' \models \varphi$, for every $\mathfrak{F}' \in \text{Pre}(\mathfrak{S}')$. Thus, in particular, we have $\mathfrak{F}'_0 \models \varphi$. Given that $\mathfrak{S}' \in \text{Ext}(\mathfrak{F})$, this means $\mathfrak{F}_0 \models \varphi$. In conclusion, $\mathfrak{M}_0 \models \varphi$.

From right-to-left, suppose that $\mathfrak{M}_0 \models \varphi$. If $\mathfrak{M} = \mathfrak{S}$, with \mathfrak{S} infinite trace, since $\mathfrak{S} \in \text{Ext}(\mathfrak{S}_0)$ we obtain in particular that $\mathfrak{S} \models \varphi$, by $\varphi \in LTL(F_{\Rightarrow\forall})$. If $\mathfrak{M} = \mathfrak{F}$, where \mathfrak{F} is a finite trace, consider its frozen extension \mathfrak{F}^ω . By $\varphi \in LTL(F_{\Rightarrow\forall})$ and the fact that $\mathfrak{F}^\omega \in \text{Ext}(\mathfrak{F}_0)$, we have that $\mathfrak{F}^\omega \models \varphi$. By $\varphi \in LTL(I_{\Rightarrow\forall})$, we then obtain in particular that $\mathfrak{F} \models \varphi$. Hence, in both cases, $\mathfrak{M} \models \varphi$, as required. \square

7 CONCLUSION

We investigated first-order temporal logic on finite traces, by comparing its semantics with the usual one based on infinite traces, and by studying the complexity of formula satisfiability in some of its decidable fragments. 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 $T_{\mathcal{U}}QL$ formulas satisfying some of these conditions have been provided as well. In particular, we have shown that for \mathcal{U}^+ - and \mathcal{R}^+ -formulas, equivalence over finite and infinite traces coincide. Moreover, we have shown that, for the class of $\mathcal{U}^+\mathcal{R}^+$ -formulas, satisfiability is preserved from finite to infinite traces. Concerning the complexity of the satisfiability problem in decidable fragments on finite traces, we have shown that the constant-free one-variable monadic fragment $T_{\mathcal{U}}QL_{\neq}^{1,mo}$, the one-variable fragment $T_{\mathcal{U}}QL^1$, the monadic monodic fragment $T_{\mathcal{U}}QL_{\square}^{mo}$, and the two-variable monodic fragment $T_{\mathcal{U}}QL_{\square}^2$, while being EXPSpace-complete over arbitrary finite traces, lower down to NEXPTIME-complete when interpreted on traces with at most k time points. Similar results have been shown here for $T_{\mathcal{U}}ALC$, a temporal extension of the description logic ALC , interpreted on finite or k -bounded traces. Moreover, we proved that $T_{\mathcal{U}}ALC$ restricted to global

CIs is EXPTIME-complete on traces with at most k time points. Finally, we have lifted results related to the notion of insensitivity to infiniteness [47], introduced in the planning context, to our first-order setting. Moreover, we have analysed the connections between notions from the verification literature (in particular, safety [95], as well as the runtime verification maxims of impartiality and anticipation [29]), and our framework of semantic conditions relating reasoning over finite and infinite traces.

As future work, we are interested in strengthening the results obtained in Section 4, so to obtain semantic and syntactic conditions that are both necessary and sufficient (as opposed to sufficient only) to characterise equivalences on finite and infinite traces. We conjecture also that for $\mathcal{U}^+\mathcal{V}$ - and $\mathcal{R}^+\exists$ -formulas, i.e., $T_{\mathcal{U}}\mathcal{QL}$ formulas in negation normal form involving only one kind of reflexive temporal operator (either \mathcal{U}^+ or \mathcal{R}^+ , respectively), the equivalences on finite traces coincide with the equivalences on infinite traces. Moreover, concerning applications to runtime verification and monitoring, the complexity of various reasoning problems in safety and co-safety fragments of *LTL*, in which the *LTL* \mathcal{U} - and \mathcal{R} -formulas respectively lie, have been recently analysed [7, 45]. We plan to study whether and how these results can be lifted to the first-order temporal setting. Additionally, we intend to apply the semantic conditions introduced in this work to the analysis of monitoring functions for runtime verification [21, 29, 46, 50] and to compare the expressive power of our fragments with that of other first-order temporal formalisms on finite traces proposed in the literature [6, 55, 56]. On the proof-theoretic side, it is known that the set of $T_{\mathcal{U}}\mathcal{QL}$ validities is not recursively enumerable [44]. We plan to investigate the axiomatisability on finite traces of its monodic fragment (similarly to [100] for infinite traces), and to develop dedicated tableaux algorithms (in line with [79] for the infinite case). Finally, it would also be interesting to determine the precise complexity of the satisfiability problem in $T_{\mathcal{U}}\mathcal{ALC}$ on finite traces with just global CIs, as well as in those DLs from the temporal *DL-Lite* family for which this problem remains open [13].

ACKNOWLEDGMENTS

We would like to express our gratitude to the anonymous reviewers for their insightful comments that significantly helped to improve our work.

REFERENCES

- [1] Wil van der Aalst, Alessandro Artale, Marco Montali, and Simone Tritini. 2017. Object-centric behavioral constraints: Integrating data and declarative process modelling. In *Proceedings of the 30th International Workshop on Description Logics*. CEUR-WS.org.
- [2] Wil van der Aalst and Maja Pesic. 2006. DecSerFlow: Towards a truly declarative service flow language. In *Proceedings of the International Workshop on Web Services and Formal Methods*. Springer, 1–23.
- [3] Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. 2002. Alternating-time temporal logic. *Journal of the ACM* 49, 5 (2002), 672–713.
- [4] Alessandro Artale and Enrico Franconi. 2005. Temporal description logics. In *Proceedings of the Handbook of Temporal Reasoning in Artificial Intelligence*. Elsevier, 375–388.
- [5] Alessandro Artale, Enrico Franconi, Frank Wolter, and Michael Zakharyashev. 2002. A temporal description logic for reasoning over conceptual schemas and queries. In *Proceedings of the 8th European Workshop on Logics in Artificial Intelligence on Logics in Artificial Intelligence*. Springer-Verlag, 98–110.
- [6] Alessandro Artale, Luca Geatti, Nicola Gigante, and Andrea Mazzullo. 2023. A landscape of first-order linear temporal logics in infinite-state verification and temporal ontologies. In *Short Paper Proceedings of the 5th Workshop on Artificial Intelligence and Formal Verification, Logic, Automata, and Synthesis*. CEUR-WS.org, 85–92.
- [7] Alessandro Artale, Luca Geatti, Nicola Gigante, Andrea Mazzullo, and Angelo Montanari. 2023. Complexity of safety and coSafety fragments of linear temporal logic. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI'23)*. AAAI Press.
- [8] Alessandro Artale, Roman Kontchakov, Alisa Kovtunova, Vladislav Ryzhikov, Frank Wolter, and Michael Zakharyashev. 2017. Temporal ontology-mediated querying: A survey. In *Proceedings of the 24th Int. Symposium on Temporal Representation and Reasoning*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Mons, Belgium, 1:1–1:37.

- [9] Alessandro Artale, Roman Kontchakov, Vladislav Ryzhikov, and Michael Zakharyashev. 2014. A cookbook for temporal conceptual data modelling with description logics. *ACM Transactions on Computational Logic* 15, 3 (2014), 25:1–25:50.
- [10] Alessandro Artale, Alisa Kovtunova, Marco Montali, and Wil M. P. van der Aalst. 2019. Modeling and reasoning over declarative data-aware processes with object-centric behavioral constraints. In *Business Process Management: 17th International Conference, BPM 2019, Vienna, Austria, September 16, 2019, Proceedings 17*. Springer.
- [11] Alessandro Artale, Andrea Mazzullo, and Ana Ozaki. 2018. Temporal description logics over finite traces. In *Proceedings of the 31st International Workshop on Description Logics*. CEUR-WS.org.
- [12] Alessandro Artale, Andrea Mazzullo, and Ana Ozaki. 2019. Do you need infinite time?. In *Proceedings of the 28th Int. Joint Conf. on Artificial Intelligence*. AAAI Press, 1516–1522.
- [13] Alessandro Artale, Andrea Mazzullo, and Ana Ozaki. 2019. Temporal DL-Lite over finite traces (preliminary results). In *Proceedings of the 32nd International Workshop on Description Logics*. CEUR-WS.org.
- [14] Alessandro Artale, Andrea Mazzullo, and Ana Ozaki. 2020. Finite vs. infinite traces in temporal logics. In *Proceedings of the 2nd Workshop on Artificial Intelligence and Formal Verification, Logic, Automata, and Synthesis*. CEUR-WS.org, 23–27.
- [15] Franz Baader, Andreas Bauer, and Marcel Lippmann. 2009. Runtime verification using a temporal description logic. In *Proceedings of the 7th Int. Symposium on Frontiers on Combining Systems*. Springer, 149–164.
- [16] Franz Baader, Stefan Borgwardt, Patrick Koopmann, Ana Ozaki, and Veronika Thost. 2017. Metric temporal description logics with interval-rigid names. In *Proceedings of the 11th Int. Symposium on Frontiers of Combining Systems*. Springer, 60–76.
- [17] Franz Baader, Stefan Borgwardt, Patrick Koopmann, Ana Ozaki, and Veronika Thost. 2020. Metric temporal description logics with interval-rigid names. *ACM Transactions on Computational Logic* 21, 4 (2020), 30:1–30:46.
- [18] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider (Eds.). 2003. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.
- [19] Franz Baader, Silvio Ghilardi, and Carsten Lutz. 2012. LTL over description logic axioms. *ACM Transactions on Computational Logic* 13, 3 (2012), 21:1–21:32.
- [20] Franz Baader, Ralf Küsters, and Frank Wolter. 2003. Extensions to description logics. In *The Description Logic Handbook*. Cambridge University Press, 219–261.
- [21] Franz Baader and Marcel Lippmann. 2014. Runtime verification using the temporal description logic ALC-LTL revisited. *Journal of Applied Logic* 12, 4 (2014), 584–613.
- [22] Fahiem Bacchus and Froduald Kabanza. 1998. Planning for temporally extended goals. *Annals of Mathematics and Artificial Intelligence* 22, 1-2 (1998), 5–27.
- [23] Fahiem Bacchus and Froduald Kabanza. 2000. Using temporal logics to express search control knowledge for planning. *Artificial Intelligence* 116, 1-2 (2000), 123–191.
- [24] Christel Baier and Joost-Pieter Katoen. 2008. *Principles of Model Checking*. MIT Press.
- [25] Jorge A. Baier and Sheila A. McIlraith. 2006. Planning with first-order temporally extended goals using heuristic search. In *Proceedings of the 21st National Conference on Artificial Intelligence*. AAAI Press, 788–795.
- [26] Howard Barringer, Ruurd Kuiper, and Amir Pnueli. 1986. A really abstract concurrent model and its temporal logic. In *Proceedings of the 13th Annual ACM Symposium on Principles of Programming Languages*. ACM, 173–183.
- [27] Ezio Bartocci, Roderick Bloem, Dejan Nickovic, and Franz Röck. 2018. A counting semantics for monitoring LTL specifications over finite traces. In *Proceedings of the 30th International Conference on Computer Aided Verification*. Springer, 547–564.
- [28] Andreas Bauer and Patrik Haslum. 2010. LTL goal specifications revisited. In *Proceedings of the 19th European Conference on Artificial Intelligence*. IOS Press, 881–886.
- [29] Andreas Bauer, Martin Leucker, and Christian Schallhart. 2010. Comparing LTL semantics for runtime verification. *Journal of Logic and Computation* 20, 3 (2010), 651–674.
- [30] Francesco Belardinelli and Alessio Lomuscio. 2011. First-order linear-time epistemic logic with group knowledge: An axiomatisation of the monodic fragment. *Fundamenta Informaticae* 106, 2-4 (2011), 175–190.
- [31] Francesco Belardinelli and Alessio Lomuscio. 2012. Interactions between knowledge and time in a first-order logic for multi-agent systems: Completeness results. *Journal of Artificial Intelligence Research* 45, 1 (2012), 1–45.
- [32] Francesco Belardinelli, Alessio Lomuscio, Aniello Murano, and Sasha Rubin. 2018. Alternating-time temporal logic on finite traces. In *Proceedings of the 27th Int. Joint Conf. on Artificial Intelligence*. ijcai.org, 77–83.
- [33] Francesco Belardinelli, Alessio Lomuscio, Aniello Murano, and Sasha Rubin. 2019. Imperfect information in alternating-time temporal logic on finite traces. In *PRIMA 2019: Principles and Practice of Multi-Agent Systems: 22nd International Conference, Turin, Italy, October 28-31, 2019, Proceedings 22*. Springer, 469–477.
- [34] Egon Börger, Erich Grädel, and Yuri Gurevich. 1997. *The Classical Decision Problem*. Springer.

- [35] Pedro Cabalar, Martín Diéguez, and Torsten Schaub. 2019. Towards dynamic answer set programming over finite traces. In *Proceedings of the 15th International Conference on Logic Programming and Nonmonotonic Reasoning*. Springer, 148–162.
- [36] Pedro Cabalar, Roland Kaminski, Torsten Schaub, and Anna Schuhmann. 2018. Temporal answer set programming on finite traces. *Theory and Practice of Logic Programming* 18, 3-4 (2018), 406–420.
- [37] Pedro Cabalar and Torsten Schaub. 2019. Dynamic and temporal answer set programming on linear finite traces. In *Proceedings of the 3rd International Workshop on the Resurgence of Datalog in Academia and Industry*. CEUR-WS.org, 3–6.
- [38] Diego Calvanese, Giuseppe De Giacomo, and Moshe Y. Vardi. 2002. Reasoning about actions and planning in LTL action theories. In *Proceedings of the 8th International Conference on Principles of Knowledge Representation and Reasoning*. 593–602.
- [39] Alberto Camacho, Jorge A. Baier, Christian J. Muişe, and Sheila A. McIlraith. 2018. Finite LTL synthesis as planning. In *Proceedings of the 28th International Conference on Automated Planning and Scheduling*. AAAI Press, 29–38.
- [40] Alberto Camacho, Meghyn Bienvenu, and Sheila A. McIlraith. 2018. Finite LTL synthesis with environment assumptions and quality measures. In *Proceedings of the 16th International Conference on Principles of Knowledge Representation and Reasoning*. AAAI Press, 454–463.
- [41] Alberto Camacho, Eleni Triantafillou, Christian J. Muişe, Jorge A. Baier, and Sheila A. McIlraith. 2017. Non-deterministic planning with temporally extended goals: LTL over finite and infinite traces. In *Proceedings of the 31st National Conference on Artificial Intelligence*. AAAI Press, 3716–3724.
- [42] Serenella Cerrito and Marta Cialdea Mayer. 1998. Bounded model search in linear temporal logic and its application to planning. In *Proceedings of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*. Springer, 124–140.
- [43] Serenella Cerrito and Marta Cialdea Mayer. 1998. Using linear temporal logic to model and solve planning problems. In *Proceedings of the 8th International Conference on Artificial Intelligence: Methodology, Systems, and Applications*. Springer, 141–152.
- [44] Serenella Cerrito, Marta Cialdea Mayer, and Sébastien Praud. 1999. First order linear temporal logic over finite time structures. In *Proceedings of the 6th International Conference on Logic Programming and Automated Reasoning*. Springer, 62–76.
- [45] Alessandro Cimatti, Luca Geatti, Nicola Gigante, Angelo Montanari, and Stefano Tonetta. 2023. A first-order logic characterization of safety and co-safety languages. *Log. Methods Comput. Sci.* 19, 3 (2023), 13:1–13:29.
- [46] Giuseppe De Giacomo, Riccardo De Masellis, Marco Grasso, Fabrizio Maria Maggi, and Marco Montali. 2014. Monitoring business metaconstraints based on LTL and LDL for finite traces. In *Proceedings of the 12th International Conference on Business Process Management*. Springer, 1–17.
- [47] Giuseppe De Giacomo, Riccardo De Masellis, and Marco Montali. 2014. Reasoning on LTL on finite traces: Insensitivity to infiniteness. In *Proceedings of the 28th National Conference on Artificial Intelligence*. AAAI Press, 1027–1033.
- [48] Giuseppe De Giacomo and Moshe Y. Vardi. 2013. Linear temporal logic and linear dynamic logic on finite traces. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*. IJCAI/AAAI, 854–860.
- [49] Giuseppe De Giacomo and Moshe Y. Vardi. 2015. Synthesis for LTL and LDL on finite traces. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*. AAAI Press, 1558–1564.
- [50] Paolo Felli, Marco Montali, Fabio Patrizi, and Sarah Winkler. 2023. Monitoring arithmetic temporal properties on finite traces. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence*. Brian Williams, Yiling Chen, and Jennifer Neville (Eds.), AAAI Press, 6346–6354.
- [51] Valeria Fionda and Gianluigi Greco. 2016. The complexity of LTL on finite traces: Hard and easy fragments. In *Proceedings of the 30th National Conference on Artificial Intelligence*. AAAI Press, 971–977.
- [52] Valeria Fionda and Gianluigi Greco. 2018. LTL on finite and process traces: Complexity results and a practical reasoner. *Journal of Artificial Intelligence Research* 63, 1 (2018), 557–623.
- [53] Dov M. Gabbay, Ian Hodkinson, and Mark Reynolds. 1994. *Temporal Logic: Mathematical Foundations and Computational Aspects*. Oxford University Press.
- [54] Dov M. Gabbay, Agi Kurucz, Frank Wolter, and Michael Zakharyashev. 2003. *Many-dimensional Modal Logics: Theory and Applications*. Elsevier.
- [55] Luca Geatti, Alessandro Gianola, and Nicola Gigante. 2022. Linear temporal logic modulo theories over finite traces. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence*. ijcai.org, 2641–2647.
- [56] Luca Geatti, Alessandro Gianola, Nicola Gigante, and Sarah Winkler. 2023. Decidable fragments of LTL_f modulo theories. In *Proceedings of the 26th European Conference on Artificial Intelligence*. IOS Press, 811–818.
- [57] Alfonso Gerevini, Patrik Haslum, Derek Long, Alessandro Saetti, and Yannis Dimopoulos. 2009. Deterministic planning in the fifth international planning competition: PDDL3 and experimental evaluation of the planners. *Artificial Intelligence* 173, 5-6 (2009), 619–668.

- [58] Giuseppe De Giacomo, Aniello Murano, Fabio Patrizi, and Giuseppe Perelli. 2021. Timed trace alignment with metric temporal logic over finite traces. In *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning*. 227–236.
- [59] Giuseppe De Giacomo, Antonio Di Stasio, Francesco Fuggitti, and Sasha Rubin. 2020. Pure-past linear temporal and dynamic logic on finite traces. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*. ijcai.org, 4959–4965.
- [60] Giuseppe De Giacomo, Antonio Di Stasio, Lucas M. Tabajara, Moshe Y. Vardi, and Shufang Zhu. 2021. Finite-trace and generalized-reactivity specifications in temporal synthesis. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*. ijcai.org, 1852–1858.
- [61] Giuseppe De Giacomo, Antonio Di Stasio, Moshe Y. Vardi, and Shufang Zhu. 2020. Two-stage technique for LTL_f synthesis under LTL assumptions. In *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning*. 304–314.
- [62] Dimitra Giannakopoulou and Klaus Havelund. 2001. Automata-based verification of temporal properties on running programs. In *Proceedings of the 16th IEEE International Conference on Automated Software Engineering*. IEEE Computer Society, 412–416.
- [63] Laura Giordano, Alberto Martelli, and Camilla Schwind. 2001. Reasoning about actions in dynamic linear time temporal logic. *Logic Journal of the IGPL* 9, 2 (2001), 273–288.
- [64] Robert Goldblatt. 1987. *Logics of Time and Computation*. Center for the Study of Language and Information.
- [65] Julian Gutierrez, Giuseppe Perelli, and Michael J. Wooldridge. 2017. Iterated games with LDL goals over finite traces. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. ACM, 696–704.
- [66] Julian Gutierrez, Giuseppe Perelli, and Michael J. Wooldridge. 2021. Multi-player games with LDL goals over finite traces. *Information and Computation* 276, 1 (2021), 104555.
- [67] Victor Gutiérrez-Basulto and Jean Christoph Jung. 2017. Combining DL-Lite $_{bool}^N$ with branching time: A gentle marriage. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. ijcai.org, 1074–1080.
- [68] Victor Gutiérrez-Basulto, Jean Christoph Jung, and Carsten Lutz. 2012. Complexity of branching temporal description logics. In *Proceedings of the 20th European Conference on Artificial Intelligence*. IOS Press, 390–395.
- [69] Victor Gutiérrez-Basulto, Jean Christoph Jung, and Ana Ozaki. 2016. On metric temporal description logics. In *Proceedings of the 22nd European Conference on Artificial Intelligence*. IOS Press, 837–845.
- [70] Victor Gutiérrez-Basulto, Jean Christoph Jung, and Thomas Schneider. 2014. Lightweight description logics and branching time: A troublesome marriage. In *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning*. AAAI Press.
- [71] Victor Gutiérrez-Basulto, Jean Christoph Jung, and Thomas Schneider. 2015. Lightweight temporal description logics with rigid roles and restricted TBoxes. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*. AAAI Press, 3015–3021.
- [72] Christopher Hampson. 2016. Decidable first-order modal logics with counting quantifiers. In *Proceedings of the 11th Conference on Advances in Modal Logic*. College Publications, 382–400.
- [73] Christopher Hampson and Agi Kurucz. 2015. Undecidable propositional bimodal logics and one-variable first-order linear temporal logics with counting. *ACM Transactions on Computational Logic* 16, 3 (2015), 27:1–27:36.
- [74] Jesper G. Henriksen and P. S. Thiagarajan. 1999. Dynamic linear time temporal logic. *Annals of Pure and Applied Logic* 96, 1-3 (1999), 187–207.
- [75] Ian M. Hodkinson, Roman Kontchakov, Agi Kurucz, Frank Wolter, and Michael Zakharyashev. 2003. On the computational complexity of decidable fragments of first-order linear temporal logics. In *Proceedings of the 10th International Symposium on Temporal Representation and Reasoning / 4th International Conference on Temporal Logic*. IEEE Computer Society, 91–98.
- [76] Ian M. Hodkinson, Frank Wolter, and Michael Zakharyashev. 2000. Decidable fragments of first-order temporal logics. *Annals of Pure and Applied Logic* 106, 1-3 (2000), 85–134.
- [77] Ian M. Hodkinson, Frank Wolter, and Michael Zakharyashev. 2002. Decidable and undecidable fragments of first-order branching temporal logics. In *Proceedings of the 17th IEEE Symposium on Logic in Computer Science*. IEEE Computer Society, 393–402.
- [78] Jeremy Kong and Alessio Lomuscio. 2018. Model checking multi-agent systems against LDLK specifications on finite traces. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. ACM, 166–174.
- [79] Roman Kontchakov, Carsten Lutz, Frank Wolter, and Michael Zakharyashev. 2004. Temporalising tableaux. *Studia Logica* 76, 1 (2004), 91–134.
- [80] Dexter Kozen. 1983. Results on the propositional mu-Calculus. *Theoretical Computer Science* 27, 3 (1983), 333–354.
- [81] Orna Kupferman and Moshe Y. Vardi. 1999. Model checking of safety properties. In *Proceedings of the 11th International Conference on Computer Aided Verification*. Springer, 172–183.

- [82] Jianwen Li, Lijun Zhang, Geguang Pu, Moshe Y. Vardi, and Jifeng He. 2014. LTL_f satisfiability checking. In *Proceedings of the 21st European Conference on Artificial Intelligence*. IOS Press, 513–518.
- [83] Yao Liu, Zhenhua Duan, Cong Tian, and Bin Cui. 2016. Satisfiability of linear time mu-calculus on finite traces. In *Proceedings of the 22nd International Conference on Computing and Combinatorics*. Springer, 611–622.
- [84] Carsten Lutz, Frank Wolter, and Michael Zakharyashev. 2008. Temporal description logics: A survey. In *Proceedings of the 15th International Symposium on Temporal Representation and Reasoning*. IEEE Computer Society, 3–14.
- [85] Fabrizio Maria Maggi, Marco Montali, and Rafael Peñaloza. 2020. Temporal logics over finite traces with uncertainty. In *Proceedings of the 34th National Conference on Artificial Intelligence*. AAAI Press, 10218–10225.
- [86] Fabrizio Maria Maggi, Marco Montali, Michael Westergaard, and Wil M. P. van der Aalst. 2011. Monitoring business constraints with linear temporal logic: An approach based on colored automata. In *Proceedings of the 9th International Conference on Business Process Management*. Springer, 132–147.
- [87] Zohar Manna and Amir Pnueli. 1995. *Temporal Verification of Reactive Systems - Safety*. Springer.
- [88] Grgur Petric Maretic, Mohammad Torabi Dashti, and David A. Basin. 2014. LTL is closed under topological closure. *Information Processing Letters* 114, 8 (2014), 408–413.
- [89] Ana Ozaki, Markus Krötzsch, and Sebastian Rudolph. 2018. Happy ever after: Temporally attributed description logics. In *Proceedings of the DL*.
- [90] Ana Ozaki, Markus Krötzsch, and Sebastian Rudolph. 2019. Temporally attributed description logics. In *Proceedings of the Description Logic, Theory Combination, and All That - Essays Dedicated to Franz Baader on the Occasion of His 60th Birthday*.
- [91] Amir Pnueli. 1977. The temporal logic of programs. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science*. IEEE Computer Society, 46–57.
- [92] Grigore Rosu. 2018. Finite-trace linear temporal logic: Coinductive completeness. *Formal Methods in System Design* 53, 1 (2018), 138–163.
- [93] Gunter Saake and Udo W. Lipeck. 1988. Using finite-linear temporal logic for specifying database dynamics. In *Proceedings of the 2nd Workshop on Computer Science Logic*. Springer, 288–300.
- [94] Yingying Shi, Shengping Xiao, Jianwen Li, Jian Guo, and Geguang Pu. 2020. SAT-based automata construction for LTL over finite traces. In *Proceedings of the 27th Asia-Pacific Software Engineering Conference*. IEEE, 1–10.
- [95] A. Prasad Sistla. 1994. Safety, liveness and fairness in temporal logic. *Formal Aspects of Computing* 6, 5 (1994), 495–512.
- [96] Holger Sturm, Frank Wolter, and Michael Zakharyashev. 2000. Monodic epistemic predicate logic. In *Proceedings of the European Workshop on Logics in Artificial Intelligence*. Springer, 329–344.
- [97] Peter van Emde Boas. 1997. The convenience of tilings. In *Proceedings of the Complexity, Logic, and Recursion Theory*. CRC Press, 331–363.
- [98] Frank Wolter and Michael Zakharyashev. 1998. Temporalizing description logics. In *Proceedings of the 2nd International Symposium on Frontiers of Combining Systems*. Research Studies Press/Wiley, 104–109.
- [99] Frank Wolter and Michael Zakharyashev. 2001. Decidable fragments of first-order modal logics. *The Journal of Symbolic Logic* 66, 3 (2001), 1415–1438.
- [100] Frank Wolter and Michael Zakharyashev. 2002. Axiomatizing the monodic fragment of first-order temporal logic. *Annals of Pure and Applied Logic* 1-2 (2002), 133–145.
- [101] Shengping Xiao, Jianwen Li, Shufang Zhu, Yingying Shi, Geguang Pu, and Moshe Y. Vardi. 2021. On-the-fly synthesis for LTL over finite traces. In *Proceedings of the 35th National Conference on Artificial Intelligence*. AAAI Press, 6530–6537.

Received 1 February 2022; revised 26 May 2023; accepted 15 February 2024