On the Satisfiability of Indexed Linear Temporal Logics*

Taolue Chen[1], Fu Song[2], and Zhilin Wu[3,4]

1 **Department of Computer Science, Middlesex University London, UK**
2 **Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, China**
3 **State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, China**
4 **LIAFA, Université Paris Diderot, France**

Abstract

Indexed Linear Temporal Logics (ILTL) are an extension of standard Linear Temporal Logics (LTL) with quantifications over index variables which range over a set of process identifiers. ILTL has been widely used in specifying and verifying properties of parameterised systems, e.g., in parameterised model checking of concurrent processes. However there is still a lack of theoretical investigations on properties of ILTL, compared to the well-studied LTL. In this paper, we start to narrow this gap, focusing on the satisfiability problem, i.e., to decide whether a model exists for a given formula. This problem is in general undecidable. Various fragments of ILTL have been considered in the literature typically in parameterised model checking, e.g., ILTL formulae in prenex normal form, or containing only non-nested quantifiers, or admitting limited temporal operators. We carry out a thorough study on the decidability and complexity of the satisfiability problem for these fragments. Namely, for each fragment, we either show that it is undecidable, or otherwise provide tight complexity bounds.

1998 ACM Subject Classification F.4.1 Mathematical Logic

Keywords and phrases Satisfiability, Indexed linear temporal logic, Parameterised systems

1 Introduction

Many concurrent systems are comprised of a finite number, but arbitrary many, of processes running in parallel. They are often referred to as *parameterised systems* [5]. In this context, the parameterised model checking problem, loosely speaking, is to decide whether a given temporal property holds irrespective of the number of participating processes.

Parameterised systems and their verification require specifications of temporal properties. For the verification of standard concurrent processes, temporal logics, typically Linear Temporal Logic (LTL), Computation Tree Logic (CTL), or their combination CTL*, have become the *de facto* specification methods. Specifications of parameterised systems largely follow this paradigm. In the setting, logics are usually extended with quantifiers over a set of process identifiers, giving rise to various *indexed* versions of temporal logics. This dates back to [22] which introduced an extension of LTL with spatial operators ranging over the

* Corresponding authors: Taolue Chen, Fu Song and Zhilin Wu. Taolue Chen is partially supported by an oversea grant from the State Key Laboratory of Novel Software Technology, Nanjing Unviersity. Fu Song is partially supported by Shanghai Pujiang Program (No. 14PJ1403200), NSFC Projects (No. 61402179), Shanghai ChenGuang Program (No. 13CG21). Zhilin Wu is partially supported by the NSFC projects (Grant No. 61100062, 61272135, and 61472474), and the visiting researcher program of China Scholarship Council from 2014.06.10 to 2015.06.09.

processes of a parameterised system. It was shown that the satisfiability problem of this logic is undecidable. After that, there has been a large body of work on indexed temporal logics. To name just a few, indexed $\mathsf{CTL}^*\backslash X$, an extension of CTL^* with quantifiers over process identifiers but excluding the "next" operator X, was introduced by Browne et al. in [5]. They studied the relation between indexed $\mathsf{CTL}^*\backslash X$ and bisimulation among parameterised systems. Emerson et al. investigated the parameterised model checking problem of fragments of indexed $\mathsf{CTL}^*\backslash X$ in prenex normal form over rings [12]. They also studied symmetry properties in model-checking systems against indexed LTL and indexed CTL^* with non-nested index quantifiers and only local atomic propositions [13]. German et al. showed that the parameterised model-checking problem of indexed LTL without global atomic propositions or nested quantifiers is undecidable [16]. Clarke et al. considered the parameterised model checking problem of indexed $\mathsf{LTL}\backslash X$ over token passing systems with respect to general topologies [6]. Very recently, Aminof et al. studied the same problem for indexed $\mathsf{CTL}^*\backslash X$ [1] unifying and extending the results in [12, 6].

Since indexed temporal logics play a fundamental role in specification and verification of parameterised systems, it is of great importance to investigate their basic (meta-)properties, along the same line as LTL, CTL, or CTL^*. In this paper, we focus on one such property, i.e., the satisfiability problem of the indexed LTL (ILTL), which, given an ILTL formula, aims to determine whether there exists a model satisfying the formula. In theory, satisfiability is probably one of the first questions one intends to answer, especially when the computational aspect of the logic is concerned. In practice, decision procedures for satisfiability have potential applications in *synthesis* of concurrent programs from their logical specifications, and play an important role in checking the consistency of specifications in an early stage of system design [27, 28]. However, in spite of its importance, to the best of our knowledge, the satisfiability problem of ILTL has not been studied systematically. The current work aims to fill in such a gap.

One immediate result is that ILTL is undecidable in general (see Proposition 1 in Section 2). We then consider the following two natural fragments of ILTL, i.e.,

- $\mathsf{ILTL}_{\mathsf{pnf}}$, the ILTL formulae in prenex normal form, and
- $\mathsf{NN\text{-}ILTL}$, the ILTL formulae where the quantifiers are non-nested.

We remark these two fragments are largely disjoint (module some trivial cases), and they are two representative classes of properties which are indeed extensively used in the verification of parameterised systems [12, 16]. Furthermore, in most of the work on parameterised model checking, e.g., [1, 11, 12, 14], indexed temporal logics are considered excluding the global atomic propositions, or with only a limited subset of temporal operators (for instance, the "next" operator X is usually disallowed). From this practical point of view, it is of paramount importance to consider these fragments, which are the main objects of the current paper.

In this paper, we mainly focus on the decidability and complexity of the satisfiability problem of $\mathsf{ILTL}_{\mathsf{pnf}}$ and $\mathsf{NN\text{-}ILTL}$. For each of these two fragments, we further study the impact of global atomic propositions and temporal operators on the decidability/complexity. These results implicitly depict that what kind of specifications can be automatically checked for consistency at the early stage of parameterised system design, and how efficiently this can be done.

Contribution. The results obtained in this paper are summarised in Figure 1 which is organised hierarchically in term of syntax inclusion. In particular we show that

- the satisfiability problem of formulae in prenex normal form starting with $\forall\exists$ is undecidable even with only "future" and "global" temporal operators ($\forall\exists - (\mathsf{ILTL}(F, G))_{pnf}$ in Figure

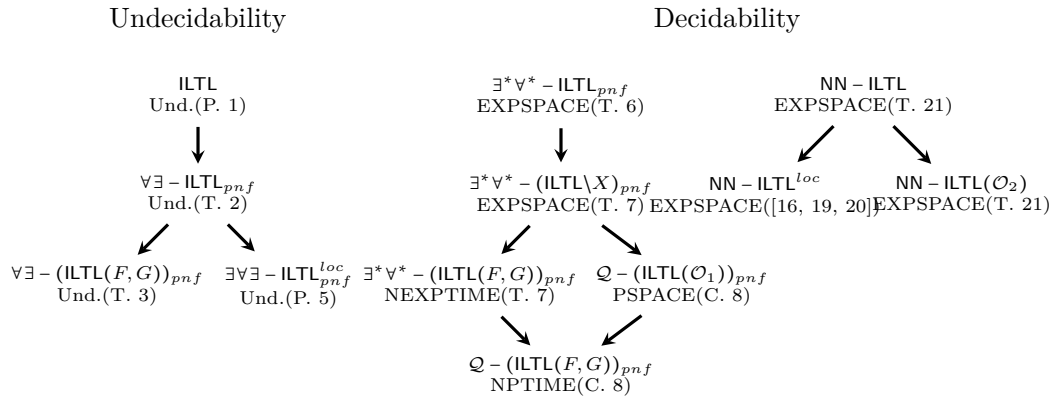

Figure 1 Summary of the results: Und. (Undecidable), $T.$ (Theorem), $C.$ (Corollary), $\mathcal{Q} \in \{\exists\forall^*\} \cup \{\exists^*\forall^k \mid k \in \mathbb{N}\}$, $\mathcal{O}_1$ is $\{X, F, G\}$ or $\{U, R\}$, $\mathcal{O}_2$ is $\{X, F, G\}$ or $\{U, R\}$ or $\{F, G\}$.

1). This extends to formulae starting with $\exists\forall\exists$ with only local atomic propositions ($\exists\forall\exists$ – ILTL$^{loc}_{pnf}$ in Figure 1),

- the satisfiability problem of the formulae in prenex normal form starting with $\exists^*\forall^*$ is decidable with an exponential blow-up in complexity comparing to their counterparts of LTL w.r.t. various combinations of temporal operators ($\exists^*\forall^*$ – ILTL$_{pnf}$, $\exists^*\forall^*$ – (ILTL\X)$_{pnf}$, and $\exists^*\forall^*$ – (ILTL(F,G))$_{pnf}$ in Figure 1),
- the satisfiability problem of the formulae in prenex normal form starting with $\exists\forall^*, \exists^*\forall^k$, where $k \geq 0$ is a fixed number, is decidable with the same complexity as their counterparts of LTL w.r.t. various combinations of temporal operators ($\mathcal{Q}$ – (ILTL$(\mathcal{O}_1)$)$_{pnf}$ and $\mathcal{Q}$ – (ILTL(F,G))$_{pnf}$ in Figure 1),
- the satisfiability problem of the formulae with non-nested quantifiers is EXPSPACE-complete, and this even holds for formulae allowing the "future" and "global" temporal operators only (NN – ILTL$(\mathcal{O}_2)$ in Figure 1).

We outline some techniques we use to obtain the aforementioned results. The upper bound of ILTL$_{\text{pnf}}$ with the quantifier prefixes $\exists^*\forall^*$ is obtained by instantiating the universal quantifiers with all the possible combinations of existentially quantified ones, thus, removing the universal quantifiers (Theorem 6). For the upper bound of NN – ILTL, a concept of potentially Eulerian directed graphs is introduced which plays an essential role in the decision procedure (Theorem 21). Moreover, we exploit the index quantifiers and a property regarding the expressiveness of "future" and "global" temporal operators (cf. Lemma 4) to obtain undecidability and complexity lower bounds (Theorem 3 and Theorem 21).

Related work. We have already discussed various indexed extensions of standard temporal logics and the related results [22, 5, 14, 16]. Since process identifiers can also be seen as a sort of data values, ILTL is also related to temporal logics over data words or words over infinite alphabets.

The most relevant work includes: LTL with freeze quantifiers (i.e. registers) over a singly attributed data word [9, 15]; LTL with navigation mechanisms for a single (or a tuple of) data attribute(s) over multi-attributed data words (i.e., data words where each position carries multiple data values which can be referred to by a fixed set of attributes) [21, 8, 7]; LTL, CTL and CTL* with variable quantifications (called variable LTL/CTL/CTL*), where

the variables range over an infinite data domain [17, 18, 10, 25]. Decidability and complexity issues of these logics and variants thereof were studied.

These logics are interpreted over data words where each position carries only a *fixed number* of data values, whereas ILTL is interpreted over computation traces in parameterised systems. While computation traces can also be seen as data words by treating process identifiers as data values, these data words are significantly different than the traditional ones studied before. Namely, each position of these data words carries an *unbounded* number of data values, and all the data values occur in every position. To our best knowledge, data words of this special structure and their logics have not been considered in the infinite alphabet community.

Structure. The rest of the paper is organised as follows. Section 2 presents the preliminaries. Section 3 is devoted to ILTL formulae in prenex normal form, and Section 4 is for ILTL formulae with non-nested quantifiers. Due to space limitation, most of the proofs are omitted and will appear in the journal version of this paper.

2 Preliminaries

For $k \in \mathbb{N}$, let $[k] = \{0, \ldots, k-1\}$. For a sequence $\alpha = \alpha_0\alpha_1\ldots$ and $j \in \mathbb{N}$, we use $\alpha[j]$ to denote the element of α at position j.

Let $\mathcal{I}$ be an infinite set of process identifiers and $\mathcal{X}$ be a set of index variables which range over $\mathcal{I}$. Let AP be a finite set of *global* atomic propositions and AP' be a finite set of *local* atomic propositions. We assume that $AP \cap AP' = \varnothing$. The intention of AP' is to specify process-specific properties, so each occurrence of AP' in formulae is parameterised with an index variable from $\mathcal{X}$.

The formulae of indexed LTL (ILTL) are defined by the following BNF,

$$\varphi ::= true \mid false \mid p \mid \neg p \mid p'(x) \mid \neg p'(x) \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid X\varphi \mid \varphi\ U\ \varphi \mid \varphi\ R\ \varphi \mid \exists x.\varphi \mid \forall x.\varphi,$$

where $p \in AP, p' \in AP', x \in \mathcal{X}$.

Moreover, standard "future" (F) and "global" (G) temporal operators can be introduced as abbreviations: $F\psi \equiv true\ U\ \psi$, $G\psi \equiv false\ R\ \psi$.

Let $\mathsf{free}(\varphi)$ denote the set of free variables occurring in φ. An ILTL formula containing no free variables is called a *closed* ILTL formula. In addition, the size of φ, denoted by $|\varphi|$, is defined as the number of symbols occurring in φ. For an ILTL formula φ, let $\neg\varphi$ denote its complement (negation), and let $\overline{\varphi}$ denote the *positive normal form* of $\neg\varphi$, that is obtained by pushing the negation inside of operators. For instance, if $\varphi = \exists x.\ Fp'(x)$, then $\overline{\varphi} = \forall x.\ G\neg p'(x)$.

ILTL formulae are typically used to specify and verify parameterised systems. Naturally, ILTL formulae are interpreted over computation traces of parameterised systems. In the present paper, we adopt the definition of the computation traces from [16]. A *computation trace* over $AP \cup AP'$ is a tuple $trc = (\alpha, I, (\beta_i)_{i\in I})$, where $\alpha \in (2^{AP})^\omega$ is an ω-sequence of valuations over the global atomic propositions from AP, $I \subseteq \mathcal{I}$ is a *finite* set of process identifiers, and for each $i \in I$, $\beta_i \in (2^{AP'})^\omega$ is a local computation trace, i.e. an ω-sequence of valuations over the local atomic propositions from AP'.

Let φ be an ILTL formula, $trc = (\alpha, I, (\beta_i)_{i\in I})$ be a computation trace, $\theta : \mathsf{free}(\varphi) \to I$ be an assignment of the process identifiers (from I) to the free variables in φ, and $n \in \mathbb{N}$. Then (trc, θ, n) satisfies φ, denoted by $(trc, \theta, n) \vDash \varphi$, is defined as follows.

- $(trc, \theta, n) \vDash p$ (resp. $\neg p$) if $p \in \alpha[n]$ (resp. $p \notin \alpha[n]$),

- $(trc, \theta, n) \vDash p'(x)$ (resp. $\neg p'(x)$) if $p' \in \beta_{\theta(x)}[n]$ (resp. $p' \notin \beta_{\theta(x)}[n]$),
- $(trc, \theta, n) \vDash \exists x.\varphi_1$ if there is $i \in I$ such that $(trc, \theta[i/x], n) \vDash \varphi_1$, where $\theta[i/x]$ is the same as θ, except for assigning i to x,
- $(trc, \theta, n) \vDash \forall x.\varphi_1$ if for each $i \in I$, $(trc, \theta[i/x], n) \vDash \varphi_1$,
- $(trc, \theta, n) \vDash \varphi_1 \vee \varphi_2$ if $(trc, \theta, n) \vDash \varphi_1$ or $(trc, \theta, n) \vDash \varphi_2$,
- $(trc, \theta, n) \vDash \varphi_1 \wedge \varphi_2$ if $(trc, \theta, n) \vDash \varphi_1$ and $(trc, \theta, n) \vDash \varphi_2$,
- $(trc, \theta, n) \vDash X\varphi$ if $(trc, \theta, n+1) \vDash \varphi$,
- $(trc, \theta, n) \vDash \varphi_1 \ U \ \varphi_2$ if there is $k \geq n$ s.t. $(trc, \theta, k) \vDash \varphi_2$, and for all $j : n \leq j < k$, $(trc, \theta, j) \vDash \varphi_1$,
- $(trc, \theta, n) \vDash \varphi_1 \ R \ \varphi_2$ if for all $k \geq n$, $(trc, \theta, k) \vDash \varphi_2$, or there is $k \geq n$ s.t. $(trc, \theta, k) \vDash \varphi_1$, and for all $j : n \leq j \leq k$, $(trc, \theta, j) \vDash \varphi_2$.

Note that if φ is a closed ILTL formula, then θ has an empty domain and thus is omitted. Namely we simply write $(trc, n) \vDash \varphi$. In addition, for a closed ILTL formula φ, we use $trc \vDash \varphi$ to abbreviate $(trc, 0) \vDash \varphi$. For a closed ILTL formula φ, let $\mathcal{L}(\varphi)$ denote the set of computation traces trc such that $trc \vDash \varphi$. The *satisfiability* problem of ILTL is:

Given a closed ILTL *formula* φ*, decide whether* $\mathcal{L}(\varphi)$ *is empty.*

As a warm-up, we show that the satisfiability problem of ILTL is undecidable in general[1], which is obtained by a reduction from the PCP problem [29]. Recall that PCP problem is, given an instance $(u_j, v_j)_{1 \leq j \leq n}$, where u_j, v_j are finite words over an alphabet Σ, to decide whether there exists a sequence of indices $j_1 \dots j_m$ such that $u_{j_1} \dots u_{j_m} = v_{j_1} \dots v_{j_m}$. The main idea of the reduction is to encode a solution $j_1 \dots j_m$ of the PCP problem as a computation trace $trc = (\alpha, I, (\beta_i)_{i \in I})$ such that $\alpha = w_{j_1} w_{j_2} \dots w_{j_m} atom(\#) \ \overline{w_{j_1}} \dots \overline{w_{j_m}} (atom(\$))^\omega$, where $w_{j_1} w_{j_2} \dots w_{j_m}$ corresponds to $u_{j_1} \dots u_{j_m}$, $\overline{w_{j_1}} \dots \overline{w_{j_m}}$ corresponds to $v_{j_1} \dots v_{j_m}$, and a local atomic proposition p' is used in $(\beta_i)_{i \in I}$ to guarantee the equality of $u_{j_1} \dots u_{j_m}$ and $v_{j_1} \dots v_{j_m}$.

▸ **Proposition 1.** *The satisfiability problem of* ILTL *is undecidable.*

In this paper, we shall consider the following fragments of ILTL with abbreviations:

- ILTL_{pnf} denotes the fragment of ILTL where formulae are in *prenex normal form*, that is $\{\forall, \exists\}$ quantifications appear only at the beginning of the formula. In particular, let $\mathcal{Q} \subseteq \{\exists, \forall\}^*$. Then $\mathcal{Q} - \text{ILTL}_{\text{pnf}}$ denotes the fragment of ILTL_{pnf} where the quantifier prefixes belong to $\mathcal{Q}$.
- NN – ILTL denotes the fragment of ILTL where the quantifiers are *not* nested, that is, for each formula $Q_1 x.\varphi_1$ such that $Q_2 y.\varphi_2$ is a subformula of φ_1, it holds that x is not a free variable of φ_2, where $Q_1, Q_2 \in \{\forall, \exists\}$.
- $\text{ILTL}(\mathcal{O})$ for $\mathcal{O} \subseteq \{X, F, G, U, R\}$ denotes the fragment of ILTL where only temporal operators from $\mathcal{O}$ are used. Moreover, we use $\text{ILTL}\backslash X$ as an abbreviation of $\text{ILTL}(U, R)$, where the X operator is forbidden.
- ILTL^{loc} denotes the fragment of ILTL where there are no global atomic propositions, that is, $AP = \varnothing$.

[1] Proposition 1 is subsumed by Theorem 2 and Theorem 3 in the next section. We choose to present the weaker and easier result first, instead of giving the strongest result (Theorem 3) directly. This might hopefully illustrate the idea of the proof and facilitate readers' understanding.

These notations might be combined to define more (refined) fragments, e.g. $(\mathsf{ILTL}(F,G))_{\mathsf{pnf}}$ denotes the fragment of $\mathsf{ILTL}_{\mathsf{pnf}}$ where only temporal operators F and G are used.

For establishing the complexity lower-bounds, we shall use the *tiling problems*, which have various versions to capture different complexity classes [4]. Among others, the *exponential-size square tiling problem* is specified by a tuple $(n, \Delta, H, V, t_S, t_F)$, where $n \in \mathbb{N}$ is encoded in unary, Δ is a finite set of tiles, $H, V \subseteq \Delta \times \Delta$ are called the horizontal and vertical constraints, $t_S, t_F \in \Delta$ are the initial tile and final tile respectively. The task is to decide whether there is a tiling of the square $[2^n] \times [2^n]$, that is, a function $f : [2^n] \times [2^n] \to \Delta$, satisfying the following conditions,

- horizontal constraint: for every $j_1, j_2 : j_1 \in [2^n], 0 \leq j_2 < 2^n - 1$, $(f(j_1, j_2), f(j_1, j_2+1)) \in H$,
- vertical constraint: for every $j_1, j_2 : 0 \leq j_1 < 2^n - 1, j_2 \in [2^n]$, $(f(j_1, j_2), f(j_1+1, j_2)) \in V$,
- initial and final constraint: $f(0,0) = t_S$, $f(2^n - 1, 2^n - 1) = t_F$.

This problem is known to be NEXPTIME-complete.

Likewise, the *exponential-size corridor tiling problem* is given by a tuple $(n, \Delta, H, V, t_S, t_F)$. The task is to decide whether there is $k \geq 1$ such that the integer plane of size $k \times 2^n$ can be tiled, that is, by a function $f : [k] \times [2^n] \to \Delta$, so that the following conditions hold,

- horizontal constraint: for each $j_1 \in [k], 0 \leq j_2 < 2^n - 1$, $(f(j_1, j_2), f(j_1, j_2 + 1)) \in H$,
- vertical constraint: for each $0 \leq j_1 < k - 1, j_2 \in [2^n]$, $(f(j_1, j_2), f(j_1 + 1, j_2)) \in V$,
- initial and final constraint: $f(0,0) = t_S$, $f(k - 1, 2^n - 1) = t_F$.

This problem is known to be EXPSPACE-complete.

3 Formulae in Prenex Normal Form

In this section, we focus on $\mathsf{ILTL}_{\mathsf{pnf}}$ formulae in prenex normal form, i.e., formulae of the form $Q_1 x_1 \dots Q_k x_k . \psi$, where $Q_j \in \{\forall, \exists\}$ for all $1 \leq j \leq k$, and ψ is quantifier free.

3.1 Undecidability

Our first (negative) result states that even with one alternation of the existential and universal quantifiers, the satisfiability problem of $\mathsf{ILTL}_{\mathsf{pnf}}$ is already undecidable.

▸ **Theorem 2.** *The satisfiability problem of* $\forall\exists - \mathsf{ILTL}_{\mathsf{pnf}}$ *is undecidable.*

The ILTL formulae used in the proof of Proposition 1 are not in $\forall\exists - \mathsf{ILTL}_{\mathsf{pnf}}$. In the proof of Theorem 2, we adapt the reduction in Proposition 1 so that only formulae from $\forall\exists - \mathsf{ILTL}_{\mathsf{pnf}}$ are used in the reduction. We then investigate whether restricting certain temporal operators leads to decidability. Unfortunately, this is not the case. Indeed, the undecidability stands even when only the "future" and "global" temporal operators are present. Clearly, this implies that the satisfiability of indexed temporal logics without "next" temporal operators, which are prevailing in the study of parameterised model checking (e.g. $\mathsf{ICTL}^*\backslash X$ in [1]), is undecidable in general.

▸ **Theorem 3.** *The satisfiability problem of* $\forall\exists - (\mathsf{ILTL}(F,G))_{\mathsf{pnf}}$ *is undecidable.*

To prove this result, we first show the undecidability of $\forall\exists - (\mathsf{ILTL}\backslash X)_{\mathsf{pnf}}$, which is obtained by adapting the proof of Theorem 2 and encoding the "next" operator with the "until" operator. As the next step, we further encode the "until" operator with the "global" and "future" operators, with the help of the index quantifiers and the following Lemma 4 [24]. Lemma 4 shows that F, G are sufficiently strong to express some properties that could

only be defined by the "until" operator at the first sight, although in a more technical and less intuitive way.

▸ **Lemma 4** ([24]). *Let $Z \subseteq \mathcal{X}$ be a finite set of variables and $\Sigma = 2^{AP \cup (AP' \times Z)}$. Suppose $L = A_1^* B_1 A_2^* B_2 \dots A_k^* B_k A_{k+1}^\omega$ such that $A_1, B_1, \dots, A_k, B_k, A_{k+1} \subseteq \Sigma$, and for each $j : 1 \le j \le k$, $B_j \subseteq (A_j \setminus A_{j+1})$. Then L can be defined by an $\mathsf{LTL}(F,G)$ formula φ over the set of atomic propositions $AP \cup (AP' \times Z)$. Moreover, if for each $j : 1 \le j \le k+1$, A_j is defined by a formula ψ_j, and for each $j : 1 \le j \le k$, B_j is defined by a formula ξ_j, then the $\mathsf{LTL}(F,G)$ formula φ can be constructed from these formulae in linear time w.r.t. $\sum_{1 \le j \le k+1} |\psi_j| + \sum_{1 \le j \le k} |\xi_j|$.*

To give an idea how the languages L in Lemma 4 can be expressed in $\mathsf{LTL}(F,G)$, let us look at the following example: Suppose $\Sigma = 2^{\{p_1'(x), p_2'(x)\}}$, and $L = A_1^* B_1 A_2^\omega$, where $A_1 = B_1 = \{\{p_1'(x)\}\}$, and $A_2 = \{\{p_2'(x)\}\}$. Intuitively, L specifies that $p_1'(x) \wedge \neg p_2'(x)$ always holds until $\neg p_1'(x) \wedge p_2'(x)$ holds, and the latter holds forever afterwards. If the "until" operator is allowed, then L can be defined easily by $p_1'(x) \wedge \neg p_2'(x) \wedge (p_1'(x) \wedge \neg p_2'(x)) \; U \; (G(\neg p_1'(x) \wedge p_2'(x)))$. On the other hand, without the "until" operator, L can be defined by

$$\varphi = p_1'(x) \wedge \neg p_2'(x) \wedge G[(p_1'(x) \wedge \neg p_2'(x)) \vee G(\neg p_1'(x) \wedge p_2'(x))] \wedge FG(\neg p_1'(x) \wedge p_2'(x)).$$

As mentioned in the introduction, we are also interested in the influence of the global atomic propositions (which are needed in the above reductions). What happens to Theorem 3 if only local atomic propositions are allowed? We show that in this case the undecidability stands at the cost of a higher level of alternations of quantifiers. Namely, we have

▸ **Proposition 5.** *The satisfiability problem of $\exists\forall\exists - \mathsf{ILTL}^{\mathsf{loc}}_{\mathsf{pnf}}$ is undecidable.*

The proof is obtained by encoding global atomic propositions with local atomic propositions, with the aid of the additional existential quantifier. Similar results also hold when the set of temporal operators is restricted.

3.2 Decidability

From Theorem 2, we know that the satisfiability problem of $\mathsf{ILTL}_{\mathsf{pnf}}$ is undecidable, even with the quantifier prefix $\forall\exists$. In this section, we will show that the undecidability result of Theorem 2 is tight in the sense that the satisfiability problem of $\exists^*\forall^* - \mathsf{ILTL}_{\mathsf{pnf}}$ is decidable (more precisely, EXPSPACE-complete).

▸ **Theorem 6.** *The satisfiability problem of $\exists^*\forall^* - \mathsf{ILTL}_{\mathsf{pnf}}$ is EXPSPACE-complete.*

Proof sketch. The EXPSPACE upper bound is obtained by instantiating the universal quantifiers with all the possible combinations of existentially quantified ones. Let $\varphi = \exists x_1 \dots x_k \forall y_1 \dots y_l.\ \psi$ be an $\exists^*\forall^* - \mathsf{ILTL}_{\mathsf{pnf}}$ formula, where ψ is quantifier-free. We construct an $\exists^* - \mathsf{ILTL}_{\mathsf{pnf}}$ formula φ' as $\exists x_1 \dots x_k.\ \bigwedge_f \psi_f$, where f ranges over all the functions from $\{1, \dots, l\}$ to $\{1, \dots, k\}$, and ψ_f is obtained from ψ by replacing each occurrence of $p'(y_j)$ with $p'(x_{f(j)})$, for every $j : 1 \le j \le l$ and $p' \in AP'$. Note that the size of φ' is exponential in $|\varphi|$. Moreover, it is not hard to see that the satisfiability of $\exists^* - \mathsf{ILTL}_{\mathsf{pnf}}$ can be reduced in linear time to that of LTL. Therefore, as the satisfiability of LTL is PSPACE-complete [23], we get the EXPSPACE upper bound for $\exists^*\forall^* - \mathsf{ILTL}_{\mathsf{pnf}}$.

For the lower bound, we reduce from the exponential-size corridor tiling problem. Let $(n, \Delta, H, V, t_S, t_F)$ be an instance of the exponential-size corridor tiling problem. Suppose $m = \lceil \log(|\Delta|) \rceil$. Let $AP' = \{p', p_1', \dots, p_n', q_1', \dots, q_m'\}$ be the set of local atomic propositions,

where $p'_1, \ldots, p'_n$ are used to encode the addresses of each row (that is, the elements of $[2^n]$) of the tiling problem, $q'_1, \ldots, q'_m$ are used to encode the set of tiles in Δ, and p' is used as a marker. The reduction also uses two existential variables x_1, x_2 such that for each $j : 1 \leq j \leq n$, exactly one of $p'_j(x_1)$ or $p'_j(x_2)$ holds at each position. Intuitively, for each address $\ell \in [2^n]$, $p'_j(x_1)$ (resp. $p'_j(x_2)$) holds iff the j-th bit of the binary encoding of ℓ is 0 (resp. 1). In addition, the universally quantified variables $y_1, \ldots, y_n$ are used to specify the horizontal and universal constraints of the tiling problem. ◂

As before, we now examine whether restricting temporal operators is beneficial to reduce the complexity. An easy observation is that since the lower bound proof of Theorem 6 only uses the operators X, F, G, the satisfiability problem of $\exists^*\forall^* - (\mathsf{ILTL}(X, F, G))_{\mathsf{pnf}}$ is EXPSPACE-complete. For the other restrictions of temporal operators, we obtain the following results.

▸ **Theorem 7.** *The satisfiability problem is*

- *EXPSPACE-complete for* $\exists^*\forall^* - (\mathsf{ILTL}\backslash X)_{\mathsf{pnf}}$,
- *NEXPTIME-complete for* $\exists^*\forall^* - (\mathsf{ILTL}(F, G))_{\mathsf{pnf}}$.

The upper-bounds are obtained by the similar argument of Theorem 6 and the complexity of respective fragments of LTL [23]. The lower bounds are obtained by (refined) reductions from the exponential-size corridor and exponential-size square tiling problems respectively.

Moreover, by a refined analysis of the proof of Theorem 6 and the complexity of various fragments of LTL [23], we obtain the following result.

▸ **Corollary 8.** *For each* $\mathcal{Q} \in \{\exists\forall^*\} \cup \{\exists^*\forall^k \mid k \in \mathbb{N}\}$, *the satisfiability problem is*

- *PSPACE-complete for* $\mathcal{Q} - (\mathsf{ILTL}(X, F, G))_{\mathsf{pnf}}$ *and* $\mathcal{Q} - (\mathsf{ILTL}\backslash X)_{\mathsf{pnf}}$,
- *NP-complete for* $\mathcal{Q} - (\mathsf{ILTL}(F, G))_{\mathsf{pnf}}$.

4 Non-Nested Quantifiers

In this section, we focus on the satisfiability of $\mathsf{NN} - \mathsf{ILTL}$, i.e., the fragment of ILTL where quantifiers are not nested. A "folklore" theorem, concerning $\mathsf{NN} - \mathsf{ILTL}^{\mathsf{loc}}$ (i.e., $\mathsf{NN} - \mathsf{ILTL}$ with *local* atomic propositions solely), states that the satisfiability problem is EXPSPACE-complete. In [16], the authors attributed this result to [19], where the temporal logics with knowledge operators were studied. Among others, the complexity upper bound for the satisfiability of $LKT_{(m)}$ $(m \geq 1)$ over synchronous unbounded memory models[2] was given by introducing a concept called k-trees and reducing to the nonemptiness of Büchi tree automata (cf. Theorem 4.1 in [19]). As pointed out in [16], $\mathsf{NN} - \mathsf{ILTL}^{\mathsf{loc}}$ corresponds to $LKT_{(1)}$ in [19] (note in this case, only 1-trees are needed). Nevertheless, the EXPSPACE result of $\mathsf{NN} - \mathsf{ILTL}^{\mathsf{loc}}$ via [19] is unsatisfactory in that: (1) the construction of [19] is for temporal logics with knowledge operators which have specific syntax and semantics, and is rather technical and only sketched (referred to [26] indeed), hence it is fair to see the result for $\mathsf{NN} - \mathsf{ILTL}^{\mathsf{loc}}$ is not self-contained and difficult to access; (2) more importantly, the correctness proof of the construction in [19] is not available, and based on our efforts in discovering the proof and the results presented in the rest of this section, the correctness of the construction in [19] is not clear, at least to us. We propose a self-contained proof for the complexity results of $\mathsf{NN} - \mathsf{ILTL}$ satisfiability (which also extends the result for $\mathsf{NN} - \mathsf{ILTL}^{\mathsf{loc}}$).

[2] where m is the number of "knowers".

Our construction for the EXPSPACE upper bound is different from that in [19]. Some new concepts, e.g., the potential Eulerian directed graphs (Definition 14), are needed for us. Moreover, we strengthen the EXPSPACE lower bound to $\mathsf{NN}-\mathsf{ILTL}(F,G)$, that is, the fragment of $\mathsf{NN}-\mathsf{ILTL}$ where only "future" and "global" temporal operators are available. This result, together with Theorem 3, shows that with index quantifiers, even very weak temporal operators are powerful enough to exhibit undecidability or complexity lower bounds.

Throughout this section, we assume φ to be an $\mathsf{NN}-\mathsf{ILTL}$ formula. In addition, we assume that only one variable x occurs in φ (i.e., x is reused in distinct quantifiers). Without loss of generality, we assume that AP (resp. AP') is the set of global (resp. local) atomic propositions occurring in φ (otherwise, it is sufficient to consider the restriction of AP and AP' to those occurring in φ). We first introduce some notations.

A *directed multigraph* G is a pair (V,E) where V is a set of vertices, E is a *multiset* of ordered pairs $(v,v') \in V \times V$. The elements of E are called *arcs*. The distinct copies of the same pair (v,v') in E are called *parallel arcs*. For an arc $e \in E$ which is a copy of (v,v'), v and v' are called the *tail* and the *head* of e respectively. An arc-labelled directed multigraph is a tuple (V,E,L), where (V,E) is a directed multigraph and $L : E \to A$ (where A is a finite set) is an arc-labelling function. A (finite) *path* in a directed multigraph $G = (V,E)$ is a sequence $v_0e_1v_1 \dots v_{n-1}e_nv_n$ (where $n \geq 1$) such that for each $j : 1 \leq j \leq n$, e_j is an arc with the tail v_{j-1} and the head v_j. The length of a path is the number of arcs in the path. A *cycle* in G is a path $v_0e_1v_1 \dots v_{n-1}e_nv_n$ such that $v_0 = v_n$. A *directed graph* is a directed multigraph (V,E) without parallel arcs, that is, E is a *set* of pairs $(v,v') \in V \times V$. For a directed graph $G = (V,E)$, since each arc is uniquely identified by its head and its tail, a path can also be seen as a vertex sequence $v_0v_1 \dots v_n$ such that for each $j : 0 \leq j < n$, $(v_j, v_{j+1}) \in E$. In addition, later on, sometimes we also write an arc-labelled directed graph $G = (V,E,L)$ as a pair (V,E') such that $E' = \{(v, L(v,v'), v') \mid (v,v') \in E\}$. A directed multigraph G is said to be *acyclic* if there are no cycles in G, and is said to be *strongly connected* if for every pair of vertices v, v', there is a path from v to v' and vice versa. A directed multigraph G is *connected* if the underlying *undirected* multigraph G, i.e. the multigraph obtained from G by ignoring the directions of arcs, is connected.

We then introduce some definitions related to φ. Let $cl(\varphi)$ denote the *closure* of formulae including the set of subformulae in φ, their complements, as well as $X(\psi_1 U \psi_2)$ and $X(\psi_1 R \psi_2)$ for $\psi_1 U \psi_2, \psi_1 R \psi_2 \in cl(\varphi)$ respectively. It is not difficult to observe that the size of $cl(\varphi)$ (the number of formulae in $cl(\varphi)$), denoted by $|cl(\varphi)|$, satisfies that $|cl(\varphi)| = O(|\varphi|)$.

▸ **Definition 9** (Atom). $\Psi \subseteq cl(\varphi)$ is an *atom* over φ if the following conditions hold:

- For each $\psi \in cl(\varphi)$, $\psi \in \Psi$ iff $\overline{\psi} \notin \Psi$.
- For each $\psi_1 \wedge \psi_2 \in cl(\varphi)$, $\psi_1 \wedge \psi_2 \in \Psi$ iff $\psi_1 \in \Psi$ and $\psi_2 \in \Psi$.
- For each $\psi_1 \vee \psi_2 \in cl(\varphi)$, $\psi_1 \vee \psi_2 \in \Psi$ iff $\psi_1 \in \Psi$ or $\psi_2 \in \Psi$.
- For each $\psi_1 U \psi_2 \in cl(\varphi)$, $\psi_1 U \psi_2 \in \Psi$ iff $\psi_2 \in \Psi$ or $\psi_1, X(\psi_1 U \psi_2) \in \Psi$.
- For each $\psi_1 R \psi_2 \in cl(\varphi)$, $\psi_1 R \psi_2 \in \Psi$ iff $\psi_2, \psi_1 \in \Psi$ or $\psi_2, X(\psi_1 R \psi_2) \in \Psi$.
- For each $\forall x.\psi \in cl(\varphi)$, if $\forall x.\psi \in \Psi$, then $\psi \in \Psi$.
- For each $\exists x.\psi \in cl(\varphi)$, if $\psi \in \Psi$, then $\exists x.\psi \in \Psi$.

▸ Remark. For formulae of the form $\exists x.\psi \in cl(\varphi)$, it is possible that $\exists x.\psi \in \Psi$, but $\psi \notin \Psi$. Let $\mathcal{A}$ denote the set of all atoms. It is not hard to see that $|\mathcal{A}| \leq 2^{|cl(\varphi)|}$.

▸ **Definition 10** (Macro state). A *macro state* S w.r.t. φ is a nonempty set of atoms satisfying the following conditions:

1. for each $p \in AP$ and $\Psi, \Psi' \in S$, $p \in \Psi$ iff $p \in \Psi'$,
2. for each $Qx.\psi \in cl(\varphi)$ and $\Psi, \Psi' \in S$, $Qx.\psi \in \Psi$ iff $Qx.\psi \in \Psi'$, where $Q \in \{\exists, \forall\}$,

3. for each $\exists x.\psi \in cl(\varphi)$ and $\Psi \in S$, $\exists x.\psi \in \Psi$ iff $\psi \in \Psi'$ for some $\Psi' \in S$.

▸ Remark. In the above definition, all atoms Ψ in S agree on the satisfaction of global atomic propositions (item 1) and sentences, i.e., formulae containing no free variables (item 2). Let $\mathcal{S}$ denote the set of all macro states w.r.t. φ.

▸ **Definition 11** (Successor)**.** We have the following definitions:

- Assume two atoms $\Psi, \Psi' \subseteq cl(\varphi)$. Then Ψ' is a *successor* of Ψ, denoted by $\Psi \to \Psi'$, if for each $X\psi \in cl(\varphi)$, $X\psi \in \Psi$ iff $\psi \in \Psi'$.
- Assume two macro states S and S' w.r.t. φ. Then S' is a successor of S if there is a *total and surjective* relation $\eta \subseteq S \times S'$ such that for each $(\Psi, \Psi') \in \eta$, Ψ' is a successor of Ψ. [Recall that $\eta \subseteq S \times S'$ is total (resp. surjective) iff for each $\Psi \in S$ (resp. $\Psi' \in S'$), there is $\Psi' \in S'$ (resp. $\Psi \in S$) such that $(\Psi, \Psi') \in \eta$.]

  For any two macro states S and S', we write $S \overset{\eta}{\rightarrowtail} S'$ to highlight the relation η associated with the transition.

The pairs $(\mathcal{A}, \to)$ and $(\mathcal{S}, \rightarrowtail)$ constitute a directed graph and multigraph respectively. Let's first give some intuition of our decision procedure. One may easily observe: Let $trc = (\alpha, I, (\beta_i)_{i\in I})$ be a computation trace satisfying φ. For each $j \in \mathbb{N}$ and $i \in I$, define $\Phi_{j,i} = \{\psi \in cl(\varphi) \mid (trc, [x \to i], j) \vDash \psi\}$. Then for each $j \in \mathbb{N}$, the tuple of atoms $(\Phi_{j,i})_{i\in I}$ can be *abstracted* into a macro state S_j (which is a set of atoms), and trc is accordingly abstracted into an infinite path $S_0S_1\ldots$ in $(\mathcal{S}, \rightarrowtail)$. From this observation, a natural idea to decide the satisfiability of φ is to search for a path in $(\mathcal{S}, \rightarrowtail)$ which satisfies some constraints (as obviously not all such paths give a valid computation trace). However, it seems nontrivial to specify these constraints. We use the following example to illustrate the difficulties.

▸ **Example 12.** Suppose $\varphi = G(\exists x.(p(x) \wedge XG\neg p(x)))$ (for simplicity, let $\xi = p(x) \wedge XG\neg p(x)$). It is not hard to see that φ is *unsatisfiable* (an obvious "model" of φ requires *infinitely* many process identifiers). The closure of φ is

$$\begin{aligned} cl(\varphi) \quad = \quad & \{p(x), \neg p(x), G\neg p(x), Fp(x), XG\neg p(x), XFp(x),\ \xi,\ \overline{\xi}, \\ & \exists x.\ \xi,\ \forall x.\ \overline{\xi},\ G(\exists x.\ \xi), F(\forall x.\ \overline{\xi}), XG(\exists x.\ \xi), XF(\forall x.\ \overline{\xi})\}. \end{aligned}$$

Let $S = \{\Psi_1, \Psi_2, \Psi_3\}$, where

$$\begin{aligned} \Psi_1 \quad &= \quad \{p(x), Fp(x), XG\neg p(x), \xi,\ \exists x.\ \xi,\ G(\exists x.\ \xi), XG(\exists x.\ \xi)\}, \\ \Psi_2 \quad &= \quad \{p(x), Fp(x), XFp(x),\ \overline{\xi},\ \exists x.\ \xi, G(\exists x.\ \xi), XG(\exists x.\ \xi)\}, \\ \Psi_3 \quad &= \quad \{\neg p(x), G\neg p(x), XG\neg p(x),\ \overline{\xi},\ \exists x.\ \xi, G(\exists x.\ \xi), XG(\exists x.\ \xi)\}. \end{aligned}$$

It is a routine to check that S satisfies all the constraints in Definition 10 and is thus a macro state. In addition, let $\eta = \{(\Psi_1, \Psi_3), (\Psi_2, \Psi_1), (\Psi_2, \Psi_2), (\Psi_3, \Psi_3)\}$. It is easy to verify that each pair of atoms in η satisfies the successor relation between atoms (item 1 of Definition 11), moreover, η is total and surjective, and thus $S \overset{\eta}{\rightarrowtail} S$. Hence $\pi = S \overset{\eta}{\rightarrowtail} S \overset{\eta}{\rightarrowtail} S\ldots$ is an infinite path in $(\mathcal{S}, \rightarrowtail)$. Moreover, since both $Fp(x)$ and $p(x)$ occur in Ψ_1 and Ψ_2, and $Fp(x)$ does not occur in Ψ_3, it is not hard to observe that over every path of atoms in π, that is, every infinite path in Figure 2, all the occurrences of $Fp(x)$ on the path are fulfilled. Therefore, to decide the satisfiability it is far from enough to simply search for a lasso in $(\mathcal{S}, \rightarrowtail)$ where over every path of atoms, all the occurrences of the "until" formulae are fulfilled, as that would lead to the wrong conclusion that φ is satisfiable. ◂

The unsatisfiability of φ in Example 12 is due to the fact that the "satisfaction" of φ requires infinitely many process identifiers (recall that as a model, we require the set

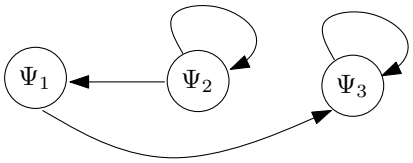

Figure 2 The directed graph (S, η) in Example 12.

of process identifier to be *finite*). To rule out such cases, we introduce a concept called *potentially Eulerian*.

Suppose $G = (V, E)$ is a directed multigraph. An *Eulerian cycle* in G is a cycle in G that traverses each arc in E exactly once. A directed multigraph G is *Eulerian* if it has an Eulerian cycle. For $v \in V$, let $\mathsf{indeg}(v)$ and $\mathsf{outdeg}(v)$ denote respectively the number of incoming arcs of v (i.e. the arcs with v as the head) and the number of outgoing arcs of v (i.e. the arcs with v as the tail) in G.

▸ **Proposition 13** ([3]). *Let $G = (V, E)$ be a directed multigraph. Then G is Eulerian iff G is connected and for each vertex $v \in V$, $\mathsf{indeg}(v) = \mathsf{outdeg}(v)$.*

▸ **Definition 14** (Potentially Eulerian). A directed multigraph $G = (V, E)$ is said to be potentially Eulerian if G can become Eulerian by adding parallel arcs.

Note that in the above definition, only parallel arcs can be added. For instance, let $G = (\{v_1, v_2, v_3\}, \{(v_1, v_2), (v_2, v_1), (v_2, v_3), (v_3, v_1)\})$, then G is not Eulerian, but G is potentially Eulerian since adding a parallel arc (v_1, v_2) makes G Eulerian.

▸ **Proposition 15.** *Let $G = (V, E)$ be a directed multigraph. Then G is potentially Eulerian iff G is strongly connected.*

▸ **Example 16** (Example 12 continued). Let G be the directed graph (S, η) in Example 12, that is, vertices are the atoms in S, and the arc relation is given by η (see Figure 2). Then it is easy to check that G is connected but not strongly connected, that is, G is *not* potentially Eulerian. ◂

Example 16 illustrates that the concept of "potentially Eulerian" may be used to deal with the situation that the satisfaction of φ requires infinitely many process identifiers, which is indeed the case in our construction, as we shall see.

Another difficulty is to formulate a proper constraint to guarantee all occurrences of "until" formulae are fulfilled somehow (which would be much easier for LTL). One natural candidate might be to require that over each path of atoms in the desired lasso of $(\mathcal{S}, \twoheadrightarrow)$, each "until" formula occurring in the path is fulfilled at least once. However, it turns out this would be too restrictive (see Example 19), and indeed we introduce a mechanism to relax this constraint; cf. $L(\cdot)$ function in Definition 17 and Definition 18, item 4.

In the following, we construct an arc-labelled directed graph G_φ from $(\mathcal{S}, \twoheadrightarrow)$ so that searching for a desired lasso in $(\mathcal{S}, \twoheadrightarrow)$ (for the satisfiability of φ) can be reduced to a reachability problem in G_φ.

▸ **Definition 17** (The graph G_φ). The arc-labelled directed graph $G_\varphi = (V_\varphi, E_\varphi)$ is constructed from $(\mathcal{S}, \twoheadrightarrow)$ as follows:

- V_φ is the union of $\mathcal{S}$ and the set of tuples (S, S', G) such that $S, S' \in \mathcal{S}$, and $G = (S \cup S', E, L)$ is an arc-labelled directed graph such that $E \subseteq S \times S'$ and $L : E \to 2^{\mathcal{A} \times \mathcal{A}}$.
- E_φ is the union of

- the set of tuples (S, η, S') such that $S \overset{\eta}{\nrightarrow} S'$,
- the set of tuples $(S, \eta, (S, S', G))$ such that $S \overset{\eta}{\nrightarrow} S'$, and $G = (S \cup S', \eta, L)$, where for each $(\Psi, \Psi') \in \eta$, $L((\Psi, \Psi')) = \{(\Psi, \Psi')\}$,
- the set of tuples $((S, S', G), \eta, (S, S'', G'))$ with
  - $S' \overset{\eta}{\nrightarrow} S''$,
  - let $G = (S \cup S', E, L)$, then $G' = (S \cup S'', E \cdot \eta, L')$, where for each $(\Psi, \Psi'') \in E \cdot \eta$ (note that this implies that there exists some $\Psi' \in S'$ such that $(\Psi, \Psi') \in E$ and $(\Psi', \Psi'') \in \eta$)

$$L'((\Psi, \Psi'')) = \bigcup_{\Psi' \in S', (\Psi, \Psi') \in E, (\Psi', \Psi'') \in \eta} L((\Psi, \Psi')) \cup \{(\Psi', \Psi'')\}.$$

We explain the intuition of the arc labeling function $L(\cdot)$ in G as follows: Suppose π is a path from S to S' in $(\mathcal{S}, \nrightarrow)$, and accordingly the vertex (S, S', G) with $G = (S \cup S', E, L)$ is reached from S in G_φ when going along π, then for each arc $(\Psi, \Psi') \in E$, $L((\Psi, \Psi'))$ is the set of all the possible arcs (Ψ'', Ψ''') on the paths from Ψ to Ψ' in the subgraph over the set of atoms induced by π.

Our goal is to reduce the satisfiability problem of φ to a reachability problem in G_φ, more specifically, to decide whether a vertex (S, S', G) satisfying some proper constraints in G_φ can be reached from a vertex S_0 that contains φ. In order to specify these constraints, we need another notation.

Given $G = (S \cup S', E, L)$ where $E \subseteq S \times S'$ and $L : E \to 2^{\mathcal{A} \times \mathcal{A}}$, and $(\Psi, \Psi') \in E$, the directed graph $G_{L((\Psi, \Psi'))}$ is defined by taking the set of atoms appearing in $L((\Psi, \Psi'))$ as the set of vertices and $L((\Psi, \Psi'))$ as the set of arcs (To put in a different way, $G_{L((\Psi, \Psi'))}$ is the graph corresponding to the relation given by $L((\Psi, \Psi')) \subseteq \mathcal{S} \times \mathcal{S}$). In addition, for a connected component C of G, a directed graph $G_{C,L}$ is defined as the *union* of the directed graphs $G_{L((\Psi, \Psi'))}$, where (Ψ, Ψ') is an arc in C.

▸ **Definition 18** (φ-witnessing path)**.** A path in G_φ is *φ-witnessing* if it is of the form

$$\underbrace{S_0 \overset{\eta_1}{\nrightarrow} S_1 \cdots \overset{\eta_{m-1}}{\nrightarrow} S_{m-1} \overset{\eta_m}{\nrightarrow} S_m}_{\pi_1} \overset{\eta'_1}{\nrightarrow} \underbrace{(S_m, S'_1, G_1) \overset{\eta'_2}{\nrightarrow} \cdots \overset{\eta'_{n-1}}{\nrightarrow} (S_m, S'_{n-1}, G_{n-1}) \overset{\eta'_n}{\nrightarrow} (S_m, S'_n, G_n)}_{\pi_2}$$

and satisfies the following conditions. Let $G_n = (S_m, E_n, L_n)$.

1. There exists some $\Psi \in S_0$ such that $\varphi \in \Psi$,
2. $S_m = S'_n$,
3. each connected component C of G_n is potentially Eulerian (i.e. strongly connected),
4. each connected component C of G_n satisfies that for each $\psi_1 U \psi_2 \in cl(\varphi)$, if $\psi_1 U \psi_2$ occurs in some atom in G_{C,L_n}, then ψ_2 occurs in some atom in G_{C,L_n} as well.

We use the following example to illustrate the concept of φ-witnessing paths.

▸ **Example 19.** Consider the formula $\varphi = G(\exists x.(p'(x) \wedge XF\neg p'(x)))$ (For convenience, let $\xi = p'(x) \wedge XF\neg p'(x)$).

$$\begin{aligned} cl(\varphi) \;=\; \{&p'(x), \neg p'(x), F\neg p'(x), Gp'(x), XF\neg p'(x), XGp'(x), \xi,\ \overline{\xi},\ \exists x.\xi, \forall x.\overline{\xi},\\ &G(\exists x.\ \xi), F(\forall x.\ \overline{\xi}), XG(\exists x.\ \xi), XF(\forall x.\ \overline{\xi})\}. \end{aligned}$$

Let

$$\begin{aligned} \Psi_1 &= \{p'(x), F\neg p'(x), XF\neg p'(x), \xi, \exists x.\xi, G(\exists x.\ \xi), XG(\exists x.\xi)\},\\ \Psi_2 &= \{\neg p'(x), F\neg p'(x), XF\neg p'(x), \overline{\xi}, \exists x.\xi, G(\exists x.\ \xi), XG(\exists x.\xi)\}, \end{aligned}$$

and $S_1 = \{\Psi_1, \Psi_2\}$, $S_2 = \{\Psi_1\}$, $\eta_1 = \{(\Psi_1, \Psi_1), (\Psi_2, \Psi_1)\}$, $\eta_2 = \{(\Psi_1, \Psi_1), (\Psi_1, \Psi_2)\}$. Suppose $\pi = \underline{S_1 \overset{\eta_1}{\mapsto} (S_1, S_2, G_1) \overset{\eta_2}{\mapsto} (S_1, S_1, G_2)}$, where $G_1 = (\{\Psi_1, \Psi_2\}, \eta_1, L_1)$, $G_2 = (\{\Psi_1, \Psi_2\}, \eta_1 \cdot \eta_2, L_2)$,

- $L_1((\Psi_1, \Psi_1)) = \{(\Psi_1, \Psi_1)\}$ and $L_1((\Psi_2, \Psi_1)) = \{(\Psi_2, \Psi_1)\}$,
- $L_2((\Psi_1, \Psi_1)) = \{(\Psi_1, \Psi_1)\}$, $L_2((\Psi_2, \Psi_2)) = \{(\Psi_2, \Psi_1), (\Psi_1, \Psi_2)\}$, moreover, $L_2(e) = \{(\Psi_1, \Psi_1), e\}$ for $e = (\Psi_1, \Psi_2), (\Psi_2, \Psi_1)$.

Then π is a path in G_φ. Moreover, we notice that G_2 satisfies the following conditions: 1) G_2 is strongly connected; 2) let C be the unique connected component of G_2 (that is, G_2 itself), then $F\neg p'(x)$ occurs in G_{C,L_2}, and $\neg p'(x)$ occurs in G_{C,L_2} as well. Therefore, π is a φ-witnessing path in G_φ.

On the other hand, π does *not* satisfy the constraint that over every path of atoms in π, $F\neg p'(x)$ is fulfilled at least once: $\Psi_1\Psi_1\Psi_1$ is a path of atoms in π, $F\neg p'(x)$ occurs everywhere on the path, but $\neg p'(x)$ never appears. This justifies to some extent the use of the labelling function L in our construction, which facilitates a less restrictive constraint than requiring that over every path of atoms, each "until" formula occurring in the path is fulfilled at least once. ◂

The following lemma paves the way to the complexity upper bound and is crucial.

▸ **Lemma 20.** *φ is satisfiable iff there is a φ-witnessing path in G_φ.*

▸ **Theorem 21.** *The satisfiability problem of* $\mathsf{NN-ILTL}$, $\mathsf{NN-ILTL}(X, F, G)$, $\mathsf{NN-ILTL}\backslash X$, *and* $\mathsf{NN-ILTL}(F, G)$ *is EXPSPACE-complete.*

By Lemma 20, the satisfiability of $\mathsf{NN-ILTL}$ can be reduced to the reachability problem in G_φ. Then the EXPSPACE upper bound in Theorem 21 follows from the fact that G_φ is a directed graph of doubly exponential size in $|\varphi|$ and the Savitch's theorem [2]. For the lower bound, we strengthen the lower bound in [16, 20] by providing a reduction from the exponential size corridor tiling problem to the satisfiability problem of $\mathsf{NN-ILTL}(F, G)$, with the help of Lemma 4.

5 Conclusion

In this paper, we have drawn a relatively complete picture on the decidability and complexity of the satisfiability of various fragments of ILTL. To the best of our knowledge, this is the first systematic work on the satisfiability of indexed temporal logics. We believe that these results will deepen the understanding of the meta-properties of this class of logics, and will be instrumental for, e.g., parameterised model checking.

Future work includes investigating satisfiability for indexed branching-time temporal logics like indexed CTL and CTL^*, and other meta-property including expressive power. It is also interesting to see whether some techniques can be applied to the extensions of temporal logics with data variable quantifications.

References

1 B. Aminof, S. Jacobs, A. Khalimov, and S. Rubin. Parameterized model checking of token-passing systems. In *VMCAI'14*, p. 262–281. 2014.

2 S. Arora and B. Barak, Computational Complexity: A Modern Approach. CUP, 2009.

3 J. Bang-Jensen and G. Z. Gutin. *Digraphs: Theory, Algorithms and Applications*. 2008.

4 P. van Emde Boas. The convenience of tilings. In *Complexity, Logic, and Recursion Theory*, p. 331–363, 1997.

5 M. C. Browne, E. M. Clarke, and O. Grumberg. Reasoning about networks with many identical finite state processes. *Information and Computation*, 81(1):13–31, 1989.

6 E. Clarke, M. Talupur, T. Touili, H. Veith. Verification by network decomposition. In *CONCUR 2004*. p. 276–291, 2004.

7 N. Decker, P. Habermehl, M. Leucker, and D. Thoma. Ordered navigation on multi-attributed data words. In *CONCUR'14*, p. 497–511, 2014.

8 S. Demri, D. Figueira, and M. Praveen. Reasoning about data repetitions with counter systems. In *LICS'13*, p. 33–42, 2013.

9 S. Demri and R. Lazic. LTL with the freeze quantifier and register automata. *ACM Transations on Computational Logic*, 10(3), 2009.

10 Orna Grumberg, Orna Kupferman and Sarai Sheinvald. A Game-Theoretic Approach to Simulation of Data-Parameterized Systems. In *ATVA'14*, p. 348–363, 2014.

11 E. A. Emerson and V. Kahlon. Reducing model checking of the many to the few. In *CADE'00*, p. 236–254, 2000.

12 E. A. Emerson and K. S. Namjoshi. On reasoning about rings. *International Journal of Foundations of Computer Science*, 14(4):527–550, 2003.

13 E. A. Emerson and A. P. Sistla. Symmetry and model checking. *Formal Methods in System Design*, 9(1/2):105–131, 1996.

14 E. A. Emerson and J. Srinivasan. A decidable temporal logic to reason about many processes. In *PODC'90*, p. 233–246, 1990.

15 D. Figueira. Alternating register automata on finite words and trees. *Logical Methods in Computer Science*, 8(1), 2012.

16 S. M. German and A. P. Sistla. Reasoning about systems with many processes. *Journal of the ACM*, 39(3):675–735, 1992.

17 O. Grumberg, O. Kupferman, and S. Sheinvald. Model checking systems and specifications with parameterized atomic propositions. In *ATVA'12*, p. 122–136, 2012.

18 O. Grumberg, O. Kupferman, and S. Sheinvald. An automata-theoretic approach to reasoning about parameterized systems and specifications. In *ATVA'13*, p. 397–411, 2013.

19 J. Y. Halpern and M. Y. Vardi. The complexity of reasoning about knowledge and time. In *STOC'86*, p. 304–315, 1986.

20 J. Y. Halpern and M. Y. Vardi. The complexity of reasoning about knowledge and time. I. lower bounds. *Journal Computer and System Sciences*, 38(1):195–237, 1989.

21 A. Kara, T. Schwentick, and T. Zeume. Temporal logics on words with multiple data values. In *FSTTCS'10*, p. 481–492, 2010.

22 J. H. Reif and A. P. Sistla. A multiprocess network logic with temporal and spatial modalities. In *ICALP'83*, p. 629–639, 1983.

23 A. P. Sistla and E. M. Clarke. The complexity of propositional linear temporal logics. *Journal of the ACM*, 32(3):733–749, 1985.

24 A. P. Sistla and L. D. Zuck. Reasoning in a restricted temporal logic. *Inf. Comput.*, 102(2):167–195, 1993.

25 F. Song and Z. Wu. Extending temporal logics with data variable quantifications. In *FSTTCS'14*, p. 253–265, 2014.

26 M. Y. Vardi and P. Wolper. Automata-theoretic techniques for modal logics of programs. *J. Comput. Syst. Sci.*, 32(2):183–221, 1986.

27 K. Y. Rozier and M. Y. Vardi. A Multi-encoding Approach for LTL Symbolic Satisfiability Checking. In*FM'11*, p. 417–431, 2011.

28 J. Li, L. Zhang, G. Pu, M. Y. Vardi and J. He. LTL Satisfiability Checking Revisited. In*TIME'13*, p. 91–98, 2013.

29 J. E. Hopcroft and J. D Ullman. Introduchon to Automata Theory, Languages, and Computation (Second edition). Addison-Wesley, Mass, 2000.