The Complexity of Propositional Linear Temporal Logics in Simple Cases (Extended Abstract)

S. Demri[1] and Ph. Schnoebelen[2]

[1] Leibniz-IMAG, Univ. Grenoble & CNRS UMR 5522,
46, av. Félix Viallet, 38031 Grenoble Cedex, France
email: demri@imag.fr

[2] Lab. Specification and Verification, ENS de Cachan & CNRS URA 2236,
61, av. Pdt. Wilson, 94235 Cachan Cedex, France
email: phs@lsv.ens-cachan.fr

Abstract. It is well-known that model-checking and satisfiability for PLTL are **PSPACE**-complete. By contrast, very little is known about whether there exist some interesting fragments of PLTL with a lower worst-case complexity. Such results would help understand why PLTL model-checkers are successfully used in practice.
In this paper we investigate this issue and consider model-checking and satisfiability for all fragments of PLTL one obtains when restrictions are put on (1) the temporal connectives allowed, (2) the number of atomic propositions, and (3) the temporal height.

1 Introduction

Background. PLTL is the standard linear-time propositional temporal logic used in the specification and automated verification of reactive systems [MP92,Eme90]. It is well-known that model-checking and satisfiability for PLTL are **PSPACE**-complete [SC85,HR83]. However, many research groups were not deterred from implementing PLTL model-checkers. They often comment about the **PSPACE** complexity by emphasizing that, in practice, PLTL specifications are not very complex, have a low temporal height (number of nested temporal connectives) and are mainly boolean combinations of simple eventuality, safety, responsiveness, fairness, ... properties. Actually a systematic theoretical study deserves to be made in order to understand whether some natural classes of PLTL formulas have lower complexity. We know of no such systematic study of this kind in the literature. This is all the more surprising because PLTL is extensively used in the specification and automated verification of reactive systems.

Our objectives. In this paper, our goal is to revisit the complexity questions from [SC85] when there is a bound on the number of propositions and/or on the temporal height of formulas. The first aim is to obtain a better understanding of where does the complexity come from. For instance [SC85] notes that satisfiability for $\mathsf{L}(\mathsf{F})$, the fragment of PLTL using only the F (sometimes) operator, is **NP**-hard because already satisfiability is **NP**-hard for the propositional calculus. This is not very enlightening. It does not say anything about how much added complexity is brought by introducing F into the propositional calculus. For the propositional calculus, and even for many modal logics [Hal95], satisfiability becomes linear-time when at most n propositions can be used. What about $\mathtt{L}(\mathsf{F})$?

Another aim is to see whether there is a formal way of stating that "practical applications only use *simple* PLTL formulas". For example, in practical applications the temporal height often turns out to be at most 3 (when fairness is involved) even when the specification is quite large and combines a large number of temporal constraints. Could such a bounded height be used to argue about reduced complexity ?

Our contribution. Our contribution is twofold. On one hand we provide a number of polynomial-time reductions allowing us to answer all the questions we put forward (only a few remaining ones are solved with ad-hoc methods). As a matter of fact, we show that (1) when the number of propositions is fixed, satisfiability can be transformed in polynomial-time into model-checking, (2) n propositional variables can be encoded into only one if F (sometimes) and X (next) are allowed, (3) into only two if U (until) is allowed. When arbitrarily many propositions are allowed, (4) temporal height can be reduced to 2 if F is allowed, and (5) model-checking for logics with X can be transformed into model-checking without X. Besides, when the formula φ has temporal height at most 1, (6) knowing whether $S \models \varphi$ only depends on a $O(|\varphi|)$ number of places in S.
On the other hand, we give new proofs showing that satisfiability and model-checking for $\mathtt{L}(\mathsf{U})$ and $\mathtt{L}(\mathsf{F},\mathsf{X})$ are **PSPACE**-hard. These proofs are "simpler" than the construction in [SC85] (e.g. we directly transform any QBF problem into a linear-sized $\mathtt{L}(\mathsf{U})$ formula of temporal height 2) and they directly apply to restricted fragments of PLTL. Also, these proofs transform QBF into a model-checking problem, so that they are new "master reductions" for PLTL, interesting in their own right.

Related work. It is common to find papers considering *extensions* of earlier temporal logics. The search for *fragments* with lower complexity is less common. [EES90] investigates (very restricted) fragments of CTL (a branching-time logic) where satisfiability is polynomial-time. [Hal95] investigates, in a systematic way, the complexity of satisfiability (*not model-checking*) for various multimodal logics when the modal height or the number of atomic propositions is restricted. We found that PLTL behaves differently. As far as PLTL is concerned, some complexity results for some particular restricted fragments of PLTL can be found in [EL87,CL93,Spa93,DFR97] but these are not a systematic study sharing our objectives. [Har85] has a simple proof, based on a general reduction from tiling problems into modal logics, that *satisfiability* for $\mathtt{L}(\mathsf{F},\mathsf{X})$ is **PSPACE**-hard. In fact, his proof shows that **PSPACE**-hardness is already obtained with temporal height 2 but bounding temporal height is not a concern in this paper.

Plan of the paper. Section 2 recalls various definitions we need throughout the paper. Sections 3 and 4 study the complexity of PLTL fragments when the number of atomic propositions is bounded. Polynomial-time transformations from QBF into model-checking problems can be found in Section 5. Section 6 studies the complexity of PLTL fragments when the temporal height is bounded. Section 7 contains concluding remarks and provides a table summarizing the complete picture we have established about complexity for PLTL fragments.

2 Basic definitions and results

Regarding complexity, we assume that the reader understands what is meant by classes such as **P**, **NP** and **PSPACE**, see e.g. [Joh90]. As usual, given two problems $\mathcal{P}_1$ and $\mathcal{P}_2$, we write $\mathcal{P}_1 \leq_p \mathcal{P}_2$ when there exists a polynomial-time transformation ("many-one reduction") from $\mathcal{P}_1$ into $\mathcal{P}_2$.

Regarding temporal logic, we follow notations and definitions from [Eme90]. Some of them are recalled below.

Syntax. PLTL is a propositional linear-time temporal logic based on a countably infinite set $\mathcal{P} = \{A_1, A_2, \ldots, P_1, P_2, \ldots\}$ of propositional variables, the classical connectives $\neg$ and $\wedge$ (negation and conjunction), and the temporal operators X (next), U (until), F (sometimes). The set $\{\varphi, \ldots\}$ of formulas is defined in the standard way. We use the connectives $\vee$, $\Rightarrow$, $\Leftrightarrow$ and G (always) as abbreviations with their standard meaning. We write $\mathcal{P}(\varphi)$ (resp. $sub(\varphi)$) for the set of propositions occurring in (resp. the set of subformulas of) φ. The temporal height of formula φ, written $th(\varphi)$, is the maximum number of nested temporal operators (among $\mathsf{X}, \mathsf{U}, \mathsf{F}$) in φ. We write $|\varphi|$ to denote the *length* (or *size*) of φ, assuming a reasonably succinct encoding. Following the usual notations (see e.g. [SC85,Eme90]), we let $\mathsf{L}(\mathsf{H}_1, \mathsf{H}_2, \ldots)$ denote the fragment of PLTL for which only the temporal operators $\mathsf{H}_1, \mathsf{H}_2, \ldots$ are allowed. For instance $\mathsf{L}(\mathsf{U})$ is "PLTL without X". We write $\mathsf{L}^k_n(\mathsf{H}, \ldots)$ to denote the fragment of $\mathsf{L}(\mathsf{H}, \ldots)$ where at most n propositions are used, and at most temporal height k is allowed. We write nothing for n and/or k, or we use ω, when no bound is required: $\mathsf{L}(\mathsf{H}, \ldots) = \mathsf{L}^\omega_\omega(\mathsf{H}, \ldots)$.

Semantics. A *linear-time structure* (also called a *model*) is a pair (S, ϵ) of an ω-sequence $S = s_0, s_1, \ldots$ of *states*, with a mapping $\epsilon : \{s_0, s_1, \ldots\} \to 2^{\mathcal{P}}$ labeling each state s_i with the set of propositions that hold in s_i. We often only write S for a structure, and we often use the fact that a structure S can be viewed as an infinite string of subsets of $\mathcal{P}$. Let S be a structure, $i \in \mathbb{N}$ and a PLTL formula φ, the satisfiability relation $\models$ is inductively defined as follows (we omit the usual conditions for the propositional connectives):

- $S, i \models A \stackrel{\text{def}}{\Leftrightarrow} A \in \epsilon(s_i)$ (when $A \in \mathcal{P}$) ;
- $S, i \models \mathsf{X}\varphi \stackrel{\text{def}}{\Leftrightarrow} S, i+1 \models \varphi$;
- $S, i \models \mathsf{F}\varphi \stackrel{\text{def}}{\Leftrightarrow}$ for some $j \geq i$, $S, j \models \varphi$;
- $S, i \models \varphi \mathsf{U} \psi \stackrel{\text{def}}{\Leftrightarrow}$ there is a $j \geq i$ s.t. $S, j \models \psi$ and for all $i \leq j' < j$, $S, j' \models \varphi$.

φ is *satisfiable* iff $S, 0 \models \varphi$ (also written $S \models \varphi$ or $S, s_0 \models \varphi$) for some S. The *satisfiability problem* for a fragment $\mathtt{L}(\ldots)$, written $SAT(\mathtt{L}(\ldots))$, is the set of all satisfiable formulas in $\mathtt{L}(\ldots)$.
Two models are *equivalent modulo stuttering*, written $S \approx S'$, if they display the same sequence of subsets of $\mathcal{P}$ when repeated (consecutive) elements are seen as one element only. Lamport [Lam83] argued that one should not distinguish between stutter-equivalent models and he advocated prohibiting X in high-level specifications. Indeed $S \approx S'$ iff S and S' satisfy the same $\mathtt{L}(\mathsf{U})$ formulas.

Model-checking. A *Kripke structure* $T = (N, R, \epsilon)$ is a triple such that N is a non-empty set of *states*, $R \subseteq N \times N$ is a total *next-state relation*, and $\epsilon : N \to 2^{\mathcal{P}}$ labels each state s with the set of propositions that hold in s. A *path* (or an *execution*) in T is an ω-sequence $S = s_0, s_1, \ldots$ of states of N such that $s_i R s_{i+1}$ for all $i \in \mathbb{N}$. (A path in T is a linear-time structure and a linear-time structure is an infinite Kripke structure.) We follow [Eme90,SC85] and write $T, s \models \varphi$ when there *exists* a path S starting from s s.t. $S \models \varphi$. This existential formulation is what we need for our complexity study. It is the dual of the more common "all paths from s satisfy φ" used in verification. All complexity results can be translated, modulo duality, between the two formulations. The *model-checking problem* for a fragment $\mathtt{L}(\ldots)$, written $MC(\mathtt{L}(\ldots))$, is the set of all $\langle T, s, \varphi \rangle$ s.t. $T, s \models \varphi$ where T is finite and φ is in $\mathtt{L}(\ldots)$.

As far as computational complexity is concerned we make a substantial use of the already known upper bounds: $SAT(\mathtt{L}(\mathsf{F}))$ and $MC(\mathtt{L}(\mathsf{F}))$ are **NP**-complete. $SAT(\mathtt{L}(\mathsf{F}, \mathsf{X}))$, $MC(\mathtt{L}(\mathsf{F}, \mathsf{X}))$, $SAT(\mathtt{L}(\mathsf{U}))$ and $MC(\mathtt{L}(\mathsf{U}))$ are **PSPACE**-complete. As a consequence, most of our proofs establish lower bounds.

3 Bounding the number of atomic propositions

In this section we evaluate the complexity of satisfiability and model-checking when the number of propositions is bounded. As a consequence, we show that there exist instances of a (linear temporal) logic for which satisfiability is **NP**-complete (resp. **PSPACE**-complete) but whose restriction to the formulas with at most n atomic propositions for some fixed $n \geq 2$ (resp. with exactly one proposition) is still **NP**-complete (resp. is in **P**). This is in contrast with the results obtained with the standard modal logics S5, KD45 (resp. the modal logic S4) in [Hal95].

We start by observing that, when the number of propositions is bounded, satisfiability can be polynomial-time reduced to model-checking.

Proposition 1. *For any $n \in \mathbb{N}$ and $\overline{\mathsf{H}} \subseteq \{\mathsf{F}, \mathsf{X}, \mathsf{U}\}$, $SAT(\mathtt{L}^{\omega}_{n}(\overline{\mathsf{H}})) \leq_p MC(\mathtt{L}^{\omega}_{n}(\overline{\mathsf{H}}))$.*

Proof. Take $\varphi \in \mathtt{L}^{\omega}_{n}(\overline{\mathsf{H}})$ such that $\mathcal{P}(\varphi) \subseteq \{A_1, \ldots, A_n\}$. Let $T = (N, R, \epsilon)$ be the Kripke structure such that, $N \stackrel{\text{def}}{=} 2^{\{A_1, \ldots, A_n\}}$ is the set of all 2^n valuations, $R \stackrel{\text{def}}{=} N \times N$ relates any two states and for all $s \in N$, s is its own valuation: $\epsilon(s) \stackrel{\text{def}}{=} s$. One can see that φ is satisfiable iff for some $s \in N$, $T, s \models \varphi$. The reduction is in polynomial-time since n and then $|T|$ are constants. Then the polynomial-time transformation can be easily defined.

Proposition 1 is used extensively in the rest of the paper. It only holds when n is bounded and should not be confused with the reductions from model-checking into satisfiability one can find in the literature (e.g. [SC85,Eme90]).

We show that n propositional variables can be encoded into one if F and X are allowed and into only two if U is allowed.

Proposition 2. *For* $\mathsf{H}_1, \ldots$ *a set of temporal operators, (1)* $MC(\mathsf{L}_\omega(\mathsf{H}_1, \ldots)) \leq_p MC(\mathsf{L}_2(\mathsf{U}, \mathsf{H}_1, \ldots))$*, and (2)* $MC(\mathsf{L}_\omega(\mathsf{H}_1, \ldots)) \leq_p MC(\mathsf{L}_1(\mathsf{F}, \mathsf{X}, \mathsf{H}_1, \ldots))$.

Proof. We show (1) here. (2) can be found in [DS97]. To a Kripke structure $T = (N, R, \epsilon)$ on $\mathcal{P} = \{P_1, \ldots, P_n\}$ we associate a Kripke structure $D_n(T) \stackrel{\text{def}}{=} (N', R', \epsilon')$ over $\mathcal{P}' = \{A, B\}$ given by

$$N' \stackrel{\text{def}}{=} \{\langle s, i\rangle \mid s \in N, 1 \leq i \leq 2n+2\}$$

$$\langle s, i\rangle R' \langle s', i'\rangle \stackrel{\text{def}}{\Leftrightarrow} \begin{cases} s = s' \text{ and } i' = i+1, \text{ or} \\ sRs' \text{ and } i = 2n+2 \text{ and } i' = 1, \end{cases}$$

$$\epsilon'(\langle s, 1\rangle) \stackrel{\text{def}}{=} \{A, B\}, \qquad \epsilon'(\langle s, 2j+1\rangle) \stackrel{\text{def}}{=} \{A\},$$

$$\epsilon'(\langle s, 2\rangle) \stackrel{\text{def}}{=} \{\}, \qquad \epsilon'(\langle s, 2j+2\rangle) \stackrel{\text{def}}{=} \begin{cases} \{B\} \text{ if } P_j \in \epsilon(s), \\ \{\} \text{ otherwise.} \end{cases}$$

where $j = 1, \ldots, n$. Fig. 1 displays an example.

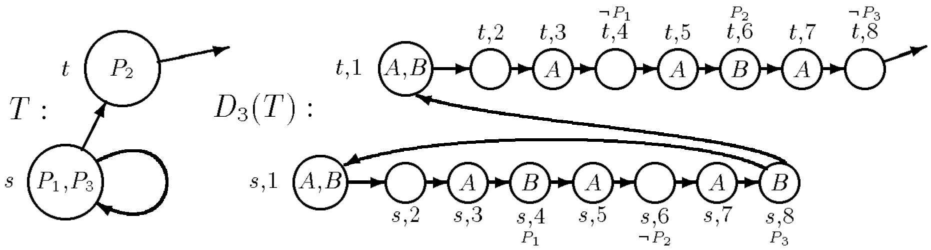

Fig. 1. T and $D_3(T)$

Here alternations between A and $\neg A$ mark slots in $D_n(T)$. B in the i-th encodes that P_i holds. Define $At_D \stackrel{\text{def}}{=} A \wedge B$, $Alt_n^0 \stackrel{\text{def}}{=} At_D$ and $Alt_n^{k+1} \stackrel{\text{def}}{=} A \wedge \neg B \wedge (A \wedge \neg B)\mathsf{U}\Big(\neg A \wedge \big(\neg A \mathsf{U} Alt_n^k\big)\Big)$ for $k \in \{1, \ldots, n-1\}$. Clearly, At_D is satisfied in $D_n(T)$ at all $\langle s, j\rangle$ with $j = 1$ and only there. Alt_n^k expresses the fact that there remains k "A–$\neg A$" alternations before the next state satisfying At_D. We now translate formulas over T into formulas over $D_n(T)$ via the following inductive definition:

$$D_n(P_i) \stackrel{\text{def}}{=} A\mathsf{U}\Big(\neg At_D \;\wedge\; \neg At_D \mathsf{U}\big(Alt_n^{n+1-i} \wedge A\mathsf{U}B\big)\Big);$$

D_n is homomorphic for the boolean connectives;
$D_n(\varphi \mathsf{U} \varphi') \stackrel{\text{def}}{=} (At_D \Rightarrow D_n(\varphi)) \mathsf{U} (At_D \wedge D_n(\varphi'))$

We have for $s \in N$, $T, s \models \varphi$ iff $D_n(T), \langle s, 1\rangle \models D_n(\varphi)$.

Proposition 3. *1.* $SAT(\mathtt{L}(\ldots)) \leq_p SAT(\mathtt{L}_2(\mathsf{U}, \ldots))$ *and 2.* $SAT(\mathtt{L}(\ldots)) \leq_p SAT(\mathtt{L}_1(\mathsf{F}, \mathsf{X} \ldots))$.

Proof. By way of example let us show 1. (see [DS97] for the full details). Let ψ'_n be the formula

$$\psi'_n \stackrel{\text{def}}{=} At_D \wedge \mathsf{G}\big(\neg A \Rightarrow (B \Rightarrow B\mathsf{U}A) \wedge (\neg B \Rightarrow \neg B \mathsf{U} A)\big)$$
$$\wedge \mathsf{G}\Big[At_D \Rightarrow At_D \mathsf{U}\big(\neg A \wedge \neg B \wedge ((\neg A \wedge \neg B)\mathsf{U} Alt^n_n)\big)\Big]$$

One can show that for any model S over $\mathcal{P} = \{P_1, \ldots, P_n\}$, $D_n(S) \models \psi'_n$ and for any S' over $\{A, B\}$, if $S' \models \psi'_n$ then there exists a (unique) S such that $S' \approx D_n(S)$. Then φ, an $\mathtt{L}_n(\ldots)$ formula, is satisfiable iff $\psi'_n \wedge D_n(\varphi)$, an $\mathtt{L}_2(\mathsf{U}, \ldots)$ formula, is satisfiable.

In the full version [DS97], we also show that $MC(\mathtt{L}_2(\mathsf{F}))$ and $SAT(\mathtt{L}_2(\mathsf{F}))$ are **NP**-hard using

Proposition 4. $SAT(\mathtt{L}(\mathsf{F})) \leq_p SAT(\mathtt{L}_2(\mathsf{F}))$.

We also provide in [DS97], a polynomial-time transformation from boolean SAT into $MC(\mathtt{L}^\omega_2(\mathsf{F}))$.

4 One proposition and U is in P

In this section, we give a linear-time algorithm for $\mathtt{L}^\varphi_1(\mathsf{U})$ that relies on linear-sized Büchi automata. Recall that the standard method for PLTL satisfiability and model-checking is to compute, for a given PLTL formula φ, a Büchi automaton [1] $\mathcal{A}_\varphi$ recognizing exactly the models of φ and then checking whether a Kripke structure T satisfies φ by computing a synchronous product of T and $\mathcal{A}_{\neg\varphi}$ and checking whether the resulting system (a larger Büchi automaton) recognizes an empty language or not. This method was first presented in [WVS83], where a first algorithm for computing $\mathcal{A}_\varphi$ was given. **PSPACE**-completeness comes from the fact that $\mathcal{A}_\varphi$ can have exponential size. Indeed, once we have $\mathcal{A}_\varphi$ the rest is easy:

Lemma 5. *It is possible, given a Büchi automaton $\mathcal{A}$ recognizing the models of formula φ, and a Kripke structure T, to say in time $O(|T| \,.\, |\mathcal{A}|)$ whether there is a computation in T which satisfies φ.*

[1] or a Muller automaton, or an alternating Büchi automaton, or ...

We consider a single proposition: $\mathcal{P} = \{A\}$. Any linear model is equivalent, modulo stuttering, to one of the following:

$$\begin{array}{lll} S_1^n \stackrel{\text{def}}{=} (A.\neg A)^n.A^\omega & S_2^n \stackrel{\text{def}}{=} \neg A.(A.\neg A)^n A^\omega & S_3^n \stackrel{\text{def}}{=} (A.\neg A)^\omega \\ S_4^n \stackrel{\text{def}}{=} (\neg A.A)^n.\neg A^\omega & S_5^n \stackrel{\text{def}}{=} A.(\neg A.A)^n \neg A^\omega & S_6^n \stackrel{\text{def}}{=} (\neg A.A)^\omega \end{array}$$

For $1 \leq i \leq 6$, the size of a Büchi automaton recognizing S_i^n (modulo stuttering) is in $O(n)$.

Lemma 6. *For any $i = 1, \ldots, 6$, $\varphi \in \mathsf{L}_1^\omega(\mathsf{U}, \mathsf{X})$ and $n \geq th(\varphi)$, we have $S_i^{n+1} \models \varphi$ iff $S_i^n \models \varphi$.*

Proof. By structural induction on φ and using the fact that the first suffix of a S_i^n is a $S_j^{n'}$ with $n - 1 \leq n' \leq n$, e.g. $(S_1^n)'$ is S_2^{n-1} (for $n > 0$) and $(S_2^n)'$ is S_1^n.

Lemma 7. *1. There exists an algorithm which, given T, s_0, $n \in \mathbb{N}$ and $1 \leq i \leq 6$, checks whether, starting from s_0, T has a path $S \approx S_i^n$ in time $O(n. |T|)$.*
2. There exists an algorithm which, given T, s_0, $n \in \mathbb{N}$ and $1 \leq i \leq 6$, checks whether there is a $m \geq n$ s.t., starting from s_0, T has a path $S \approx S_i^m$ in time $O(n. |T|)$.

Proof. Given $1 \leq i \leq 6$ and $n \in \mathbb{N}$, it is easy to build a Büchi automaton with size $O(n)$ recognizing all models S s.t. $S \approx S_i^n$ (resp. s.t. $S \approx S_i^m$ for some $m \geq n$). Then Lemma 5 concludes the proof.

Theorem 8. *Model-checking for $\mathsf{L}_1^\omega(\mathsf{U})$ is in* **P**.

Proof. We consider a Kripke structure $T = (N, R, \epsilon)$ and some state $s_0 \in N$. If there is a path S from s_0 satisfying $\varphi \in \mathsf{L}_1^\omega(\mathsf{U})$ then $S \approx S_i^n$ for some $n \in \mathbb{N}$ and some $i = 1, \ldots, 6$ and $S_i^n \models \varphi$. Conversely, if $S_i^n \models \varphi$ and there is a path $S \approx S_i^n$ starting from s_0, then $T, s_0 \models \varphi$.

It is possible to check whether T contains such a path in polynomial-time: We consider all S_i^k for $k < th(\varphi)$. When $S_i^k \models \varphi$, seen in time $O(k. \mid\varphi\mid)$, we check in time $O(k. \mid T\mid)$, whether, from s_0, T admits a path $S \approx S_i^k$. We also consider all S_i^k for $k = th(\varphi)$. When $S_i^k \models \varphi$ we know that $S_i^{k+m} \models \varphi$ for all m (Lemma 6) so that it is correct to check whether there is a m s.t. T admits a path $S \approx S_i^{k+m}$. Thanks to Lemma 7, this can be done in polynomial-time. Because $k \leq\mid\varphi\mid$, the complete algorithm only needs $O(\mid T\mid . \mid\varphi\mid^2)$.

With Proposition 1, we get $SAT(\mathsf{L}_1^\omega(\mathsf{U}))$ is in **P**. In order to be exhaustive one can show that $SAT(\mathsf{L}_1^\omega(\mathsf{X}))$ and $MC(\mathsf{L}_1^\omega(\mathsf{X}))$ are **NP**-complete [DS97]. Moreover since there are only a finite number of essentially distinct formulas in a given $\mathsf{L}_n^k(\mathsf{U}, \mathsf{X})$ (for any fixed $k, n < \omega$), $SAT(\mathsf{L}_n^k(\mathsf{X}))$ and $MC(\mathsf{L}_n^k(\mathsf{X}))$ can be proved in **P** (see [DS97]). This concludes the study of all fragments with a bounded number of propositions. In the remaining of the paper, this bound is removed.

5 From QBF into $MC(\mathsf{L}(\mathsf{U}))$

In this section, we offer a polynomial-time transformation from validity of Quantified Boolean Formulas (QBF) into model-checking for $\mathsf{L}(\mathsf{U})$ that involves rather simple constructions of models and formulas. This reduction can be adapted to various fragments and, apart from the fact that it offers a simple means to get **PSPACE**-hardness, we obtain a new master reduction from a well-known logical problem. As a side-effect, we establish that $MC(\mathsf{L}^2_\omega(\mathsf{U}))$ is **PSPACE**-hard, which is not subsumed by any reduction from the literature. In the full version we show reduction from QBF into model-checking for $\mathsf{L}(\mathsf{F},\mathsf{X})$ and give a construction showing that $MC(\mathsf{L}^\omega_2(\mathsf{F}))$ is **NP**-hard.

Consider an instance of QBF. It has the form $P \equiv Q_1 x_1 \ \ldots \ Q_n x_n \overbrace{\wedge_{i=1}^{m} \vee_{j=1}^{k_i} l_{i,j}}^{P_0}$ where every Q_r $(1 \leq r \leq n)$ is a universal, $\forall$, or existential, $\exists$, quantifier. P_0 is a propositional formula without any quantifier. Here we consider w.l.o.g. that P_0 is a conjunction of clauses, i.e. every $l_{i,j}$ is a propositional variable $x_{r(i,j)}$ or the negation $\neg x_{r(i,j)}$ of a propositional variable from $X = \{x_1, \ldots, x_n\}$. The question is to decide whether P is valid or not. Recall that

Lemma 9. *P is valid iff there exists a non-empty set $\mathcal{V} \subseteq \{\top, \bot\}^X$ of valuations (truth-value assignments) s.t.*
correctness: *$\forall v \in \mathcal{V}, v \models P_0$, and*
closure: *for all $v \in \mathcal{V}$, for all r s.t. $Q_r = \forall$, there is a $v' \in \mathcal{V}$ s.t. $v'[x_r] = not(v[x_r])$ and for all $r' < r$, $v'[x_{r'}] = v[x_{r'}]$.*

To P we associate the Kripke structure T_P as given in Figure 2, using labels from $\mathcal{P} = \{A_0, A_1, \ldots, x_1^T, \ldots, L_1^1, \ldots\}$.

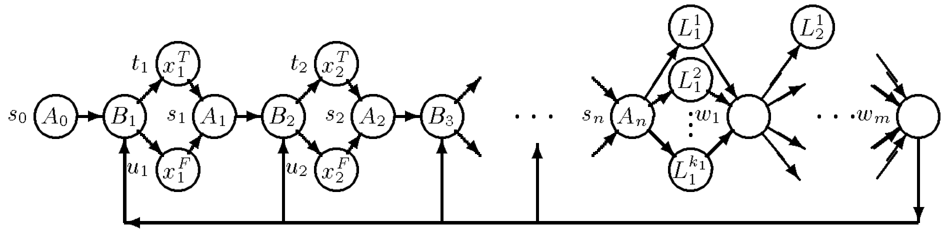

Fig. 2. The structure T_P associated to $P \equiv Q_1 x_1 \ \ldots \ Q_n x_n \wedge_{i=1}^{m} \vee_{j=1}^{k_i} l_{i,j}$

Assume S is an infinite path starting from s_0. Between s_0 and s_n, it picks a boolean valuation for all variables in X, then reaches w_m and goes back to some B_r-labeled state $(1 \leq r \leq n)$ where (possibly distinct) valuations for $x_r, x_{r+1}, \ldots, x_n$ are picked.

In S, at any position lying between a s_n and the next w_m, we have a notion of *current valuation* which associates $\top$ or $\bot$ to any x_r depending on *the latest*

u_r *or* t_r *node* we visited. To S we associate the set $\mathcal{V}(S)$ of all valuations that are current at positions where S visits s_n (there are infinitely many such positions).

Now consider some r with $Q_r = \forall$ and assume that whenever S visits s_{r-1} then it visits both t_r and u_r before any further visit to s_{r-1}. In $\mathsf{L}(\mathsf{U})$, this can be written $S \models \psi_r$ with ψ_r given by

$$\psi_r \stackrel{\text{def}}{=} \mathsf{G}\Big(A_{r-1} \;\Rightarrow\; (\neg B_{r-1} \mathsf{U} x_r^T) \wedge (\neg B_{r-1} \mathsf{U} x_r^F)\Big).$$

Let $\psi_{\text{clo}} \stackrel{\text{def}}{=} \bigwedge\{\psi_r \mid Q_r = \forall\}$: if S satisfies ψ_{clo} then $\mathcal{V}(S)$ is closed in the sense of Lemma 9.

Now, whenever S visits a L_i^j-state, we say it agrees with the current valuation v if $v \models l_{i,j}$. This too can be written in $\mathsf{L}(\mathsf{U})$, using the fact that the current valuation for x_r cannot be changed without first visiting the B_r-state. For $i = 1, \ldots, m$, for $j = 1, \ldots, k_i$, let

$$\psi_{i,j} \stackrel{\text{def}}{=} \begin{cases} \mathsf{G}[x_r^F \;\Rightarrow\; \mathsf{G}\neg L_i^j \vee \neg L_i^j \mathsf{U} B_r] & \text{if } l_{i,j} = x_r, \\[1ex] \mathsf{G}[x_r^T \;\Rightarrow\; \mathsf{G}\neg L_i^j \vee \neg L_i^j \mathsf{U} B_r] & \text{if } l_{i,j} = \neg x_r. \end{cases}$$

Lemma 10. *Let* $\varphi_P \stackrel{\text{def}}{=} \psi_{\text{clo}} \wedge \Big(\bigwedge_{i=1}^m \bigwedge_{j=1}^{k_i} \psi_{i,j}\Big)$. *Then* $T_P, s_0 \models \varphi_P$ *iff* P *is valid.*

Proof. If $S \models \varphi_P$ then $\mathcal{V}(S)$ is closed and correct for P so that P is valid. Conversely, if P is valid, there exists a validating $\mathcal{V}$ (Lemma 9). We can build an infinite path S starting from s_0 s.t. $\mathcal{V}(S) = \mathcal{V}$ and $S \models \varphi_P$: from a lexicographical enumeration of $\mathcal{V}$, S is easily constructed so that $S \models \psi_{\text{clo}}$. Then, to ensure $S \models \varphi_P$, between any visit to s_n and to the next w_m we choose to visit L_i^j-states validated by the current valuation v, which is possible because $v \models P_0$.

Now, because $|T_P|$ and $|\varphi_P|$ are in $O(|P|)$, and because $th(\varphi_P) \leq 2$ (and using Proposition 2), we get

Corollary 11. QBF $\leq_p MC(\mathsf{L}^2_\omega(\mathsf{U})) \leq_p MC(\mathsf{L}^\omega_2(\mathsf{U}))$.

Corollary 12. $MC(\mathsf{L}^2_\omega(\mathsf{U}))$ *and* $MC(\mathsf{L}^\omega_2(\mathsf{U}))$ *are* **PSPACE**-*hard.*

6 Bounding the temporal height

In this section we investigate the complexity of satisfiability and model-checking when the temporal height is bounded. From Section 5, we already know that $MC(\mathsf{L}^2_\omega(\mathsf{U}))$ is **PSPACE**-hard.

Elimination of X We show how problems for $\mathsf{L}(\mathsf{X}, \ldots)$ can be transformed into problems for $\mathsf{L}(\ldots)$. Say a formula φ has *inner-nexts* if the only occurrences of X are in subformulas of the form $\mathsf{X}\mathsf{X}\ldots\mathsf{X}A$ (where A is a propositional variable). Assume φ has inner-nexts, with at most k nested X and that T is a Kripke structure. It is possible to partially unfold T into a Kripke structure T^k where a state $\overline{s}$ (in T^k) codes for a state s_0 in T with the k next states $s_1, \ldots, s_k$ already chosen. We can now replace all $\mathsf{X}^i A$ in φ by new propositions A^i and label T^k so that $A^i \in \epsilon(\overline{s})$ iff $A \in \epsilon(s_i)$. Finally, $T, s \models \varphi$ iff $T^k, \overline{s} \models \varphi^k$ for some $\overline{s}$ starting with s. Because the size of T^k is in $O(|T|^k)$ and $|\varphi^k|$ is in $O(|\varphi|)$, we have

Proposition 13. [DS97] *$MC(\mathsf{L}^k_\omega(\mathsf{X},\ldots)) \leq_p MC(\mathsf{L}^k_\omega(\ldots))$ for any fixed $k \geq 0$.*

As a corollary, $MC(\mathsf{L}^k_\omega(\mathsf{X}))$ is in **P** and $MC(\mathsf{L}^k_\omega(\mathsf{F},\mathsf{X}))$ is in **NP** for any fixed $k \geq 0$. $MC(\mathsf{L}^1_\omega(\mathsf{F}))$ is **NP**-hard as can be seen from the proof for $\mathsf{L}^\omega_\omega(\mathsf{F})$ in [SC85]. Hence for $k \geq 1$, $MC(\mathsf{L}^k_\omega(\mathsf{F},\mathsf{X}))$ is **NP**-complete. Elimination of X can also be performed for satisfiability. If φ is satisfiable, then φ^k is. Now if φ^k is satisfiable, it is perhaps satisfiable in a model that is not a S^k for some S. But we can express the fact that a given model is a S^k with an $\mathsf{L}^2_\omega(\mathsf{F},\mathsf{X})$ formula. Then φ is satisfiable iff $\varphi^k \wedge \mathsf{G}(\bigwedge_{j=1}^n \bigwedge_{i=1}^k A^i_j \Leftrightarrow \mathsf{X}A^{i-1}_j)$ is. Actually, by using systematically the standard renaming techniques (the operator G propagates the constraints of renaming), we can show that $SAT(\mathsf{L}(\overline{\mathsf{H}})) \leq_p SAT(\mathsf{L}^2_\omega(\overline{\mathsf{H}}))$ for $\overline{\mathsf{H}} \in \{\{\mathsf{F}\},\{\mathsf{F},\mathsf{X}\},\{\mathsf{U}\},\{\mathsf{U},\mathsf{X}\}\}$. As a corollary, $SAT(\mathsf{L}^2_\omega(\mathsf{F},\mathsf{X}))$ and $SAT(\mathsf{L}^2_\omega(\mathsf{U}))$ are **PSPACE**-hard. It is worth observing that [Spa93,DFR97] have another proof that $SAT(\mathsf{L}^2_\omega(\mathsf{F},\mathsf{X}))$ is **PSPACE**-hard.

Temporal height less or equal to 1: upper bounds in **NP** Below temporal height 2, the upper bounds can be improved. For any $\varphi \in \mathsf{L}^1_\omega(\mathsf{U},\mathsf{X})$, one can show that φ is satisfiable iff φ is satisfied in a model $S = s_0, s_1, \ldots$ such that for any $i, j \geq 1+ |\varphi|$, for $A \in \mathcal{P}(\varphi)$, $A \in \epsilon(s_i)$ iff $A \in \epsilon(s_j)$. As a corollary, for $\overline{\mathsf{H}} \in \{\{\mathsf{F}\},\{\mathsf{F},\mathsf{X}\},\{\mathsf{U}\},\{\mathsf{U},\mathsf{X}\}\}$, $SAT(\mathsf{L}^1_\omega(\overline{\mathsf{H}}))$ is in **NP**. Since those fragments contain the propositional calculus, **NP**-hardness is immediate. Now let us turn to model-checking when the temporal height is at most 1. We already know that $MC(\mathsf{L}^1_\omega(\mathsf{F}))$ is **NP**-hard [SC85]. We can also show that $MC(\mathsf{L}^1_\omega(\mathsf{U},\mathsf{X}))$ is in **NP** [DS97]. As a corollary, for $\overline{\mathsf{H}} \in \{\{\mathsf{F}\},\{\mathsf{F},\mathsf{X}\},\{\mathsf{U}\},\{\mathsf{U},\mathsf{X}\}\}$, $MC(\mathsf{L}^1_\omega(\overline{\mathsf{H}}))$ is **NP**-complete.

7 Concluding remarks

In the paper we have investigated the complexity of satisfiability and model-checking for all fragments of PLTL obtained (1) by bounding the number of atomic propositions, (2) the temporal height, and (3) restricting the temporal operators one allows.

Our results take advantage of a few general techniques that might be reused to tackle similar problems for other temporal logics. Most of the time, these techniques are used to strengthen earlier hardness results so that they also apply to specific fragments. In some cases we develop specific arguments showing that the complexity really decreases under the identified threshold values.

The table at the end of this section contains the results of the full paper and some general conclusions can be read. In most cases no reduction in complexity occurs when two propositions are allowed, or when temporal height two is allowed. Moreover in most cases, for equal fragments, satisfiability and model-checking belong to the same complexity class. See the table for some exceptions.

	$n, k < \omega$	Model-Checking	Satisfiability
$\mathtt{L}(\dots)$	$\mathtt{L}^k_n(\dots)$	P	P
	$\mathtt{L}^0_\omega(\dots)$	P	NP-complete
$\mathtt{L}(\mathsf{F})$	$\mathtt{L}(\mathsf{F})$	NP-complete [SC85]	NP-complete [NO80]
	$\mathtt{L}^1_\omega(\mathsf{F})$	NP-complete	NP-complete
	$\mathtt{L}^\omega_2(\mathsf{F})$	NP-complete	NP-complete
	$\mathtt{L}^\omega_1(\mathsf{F})$	P	P
$\mathtt{L}(\mathsf{U})$	$\mathtt{L}(\mathsf{U})$	PSPACE-complete [SC85]	PSPACE-complete [SC85,HR83]
	$\mathtt{L}^2_\omega(\mathsf{U})$	PSPACE-complete	PSPACE-complete
	$\mathtt{L}^1_\omega(\mathsf{U})$	NP-complete	NP-complete
	$\mathtt{L}^\omega_2(\mathsf{U})$	PSPACE-complete	PSPACE-complete
	$\mathtt{L}^\omega_1(\mathsf{U})$	P	P
$\mathtt{L}(\mathsf{X})$	$\mathtt{L}(\mathsf{X})$	NP-complete	NP-complete
	$\mathtt{L}^k_\omega(\mathsf{X})$	P	NP-complete
	$\mathtt{L}^\omega_1(\mathsf{X})$	NP-complete	NP-complete
$\mathtt{L}(\mathsf{F},\mathsf{X})$	$\mathtt{L}(\mathsf{F},\mathsf{X})$	PSPACE-complete [SC85]	PSPACE-complete [SC85,HR83]
	$\mathtt{L}^{2+k}_\omega(\mathsf{F},\mathsf{X})$	NP-complete	PSPACE-complete [Har85,Spa93]
	$\mathtt{L}^1_\omega(\mathsf{F},\mathsf{X})$	NP-complete	NP-complete
	$\mathtt{L}^\omega_1(\mathsf{F},\mathsf{X})$	PSPACE-complete	PSPACE-complete
$\mathtt{L}(\mathsf{U},\mathsf{X})$	$\mathtt{L}(\mathsf{U},\mathsf{X})$	PSPACE-complete [SC85]	PSPACE-complete [SC85,HR83]
	$\mathtt{L}^2_\omega(\mathsf{U},\mathsf{X})$	PSPACE-complete	PSPACE-complete [Har85,Spa93]
	$\mathtt{L}^1_\omega(\mathsf{U},\mathsf{X})$	NP-complete	NP-complete
	$\mathtt{L}^\omega_1(\mathsf{U},\mathsf{X})$	PSPACE-complete	PSPACE-complete

The frequent preservation of lower bounds when fragments are taken into account does not explain the alleged simplicity of "simple practical applications". The fact that $\mathtt{L}^\omega_1(\mathsf{U})$ (only one proposition) is in **P** can be used inside a PLTL verifier as an efficient method *for a few special cases*, but it is too restricted for practical applications. The fact that $MC(\mathtt{L}^k_\omega(\mathsf{F},\mathsf{X}))$ is in **NP** has more potential explanatory power, but **NP**-hardness is still intractable. The fact that $\mathtt{L}^k_n(\mathsf{U},\mathsf{X})$ is in **P** really means that the complexity depends at least on the number of propositions or the temporal height. This dependence must be scrutinized in more details.

Understanding and taming the complexity of linear temporal logics remains an important issue and the present work can be seen as some steps in this direction. The ground is open for further investigations. We think future work could investigate

- different, finer definitions of fragments (witness [EES90]) that can be inspired by practical examples, or that aim at defeating one of our hardness proofs, e.g. forbidding the renaming technique we use in section 6,
- other problems that are important for verification: module checking, semantic entailment, ... ,
- other complexity measures: e.g. average complexity, or separated complexity measure for models and formulas,

– restrictions on the models rather than the formulas.

Acknowledgments. We thank B. Bérard, E. A. Emerson, F. Laroussinie, M. Reynolds, A. Zanardo, and the anonymous referees for their suggestions.

References

[CL93] C.-C. Chen and I.-P. Lin. The computational complexity of satisfiability of temporal Horn formulas in propositional linear-time temporal logic. *Information Processing Letters*, 45(3):131–136, March 1993.

[DFR97] C. Dixon, M. Fisher, and M. Reynolds. Execution and proof in a Horn-clause temporal logic. In *Proc. 2nd Int. Conf. on Temporal Logic (ICTL'97), Manchester, UK, July 1997*, 1997. to appear.

[DS97] S. Demri and Ph. Schnoebelen. The complexity of propositional linear temporal logics in simple cases. Research Report LSV-97-11, Lab. Specification and Verification, ENS de Cachan, Cachan, France, December 1997. Available at `http://www.lsv.ens-cachan.fr/~phs`.

[EES90] E. A. Emerson, M. Evangelist, and J. Srinivasan. On the limits of efficient temporal decidability. In *Proc. 5th IEEE Symp. Logic in Computer Science (LICS'90), Philadelphia, PA, USA, June 1990*, pages 464–475, 1990.

[EL87] E. A. Emerson and C. Lei. Modalities for model checking: Branching time logic strikes back. *Science of Computer Programming*, 8:275–306, 1987.

[Eme90] E. A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, vol. B*, chapter 16, pages 995–1072. Elsevier Science Publishers, 1990.

[Hal95] J. Y. Halpern. The effect of bounding the number of primitive propositions and the depth of nesting on the complexity of modal logic. *Artificial Intelligence*, 75(2):361–372, 1995.

[Har85] D. Harel. Recurring dominos: Making the highly undecidable highly understandable. *Annals of Discrete Mathematics*, 24:51–72, 1985.

[HR83] J. Y. Halpern and J. H. Reif. The propositional dynamic logic of deterministic, well-structured programs. *Theor. Comp. Sci.*, 27(1–2):127–165, 1983.

[Joh90] D. S. Johnson. A catalog of complexity classes. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, vol. A*, chapter 2, pages 67–161. Elsevier Science Publishers, 1990.

[Lam83] L. Lamport. What good is temporal logic ? In R. E. A. Mason, editor, *Information Processing'83. Proc. IFIP 9th World Computer Congress, Sep. 1983, Paris, France*, pages 657–668. North-Holland, 1983.

[MP92] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems: Specification*. Springer-Verlag, 1992.

[NO80] A. Nakamura and H. Ono. On the size of refutation Kripke models for some linear modal and tense logics. *Studia Logica*, 39(4):325–333, 1980.

[SC85] A. P. Sistla and E. M. Clarke. The complexity of propositional linear temporal logics. *Journal of the ACM*, 32(3):733–749, July 1985.

[Spa93] E. Spaan. *Complexity of Modal Logics*. PhD thesis, ILLC, Amsterdam University, NL, March 1993.

[WVS83] P. Wolper, M. Y. Vardi, and A. P. Sistla. Reasoning about infinite computation paths (extended abstract). In *Proc. 24th IEEE Symp. Found. of Computer Science (FOCS'83), Tucson, USA, Nov. 1983*, pages 185–194, 1983.