Temporal logic with forgettable past

Nicolas Markey

CNRS – IRISA (Univ. Rennes, France)

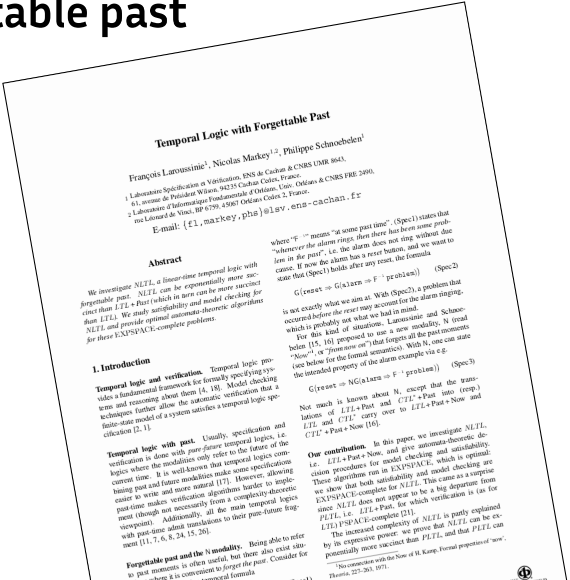
joint work with my former PhD advisors
François Laroussinie and Philippe Schnoebelen

Temporal logic with forgettable past

Nicolas Markey

CNRS – IRISA (Univ. Rennes, France)

joint work with my former PhD advisors
François Laroussinie and Philippe Schnoebelen



Temporal Logic with Forgettable Past

François Laroussinie[1], Nicolas Markey[1,2], Philippe Schnoebelen[1]

1 Laboratoire Spécification et Vérification, ENS de Cachan & CNRS UMR 8643, 61, avenue de Président Wilson, 94235 Cachan Cedex, France.
2 Laboratoire d'Informatique Fondamentale d'Orléans, Univ. Orléans & CNRS FRE 2490, rue Léonard de Vinci, BP 6759, 45067 Orléans Cedex 2, France.

E-mail: {fl,markey,phs}@lsv.ens-cachan.fr

Abstract

We investigate NLTL, a linear-time temporal logic with forgettable past. NLTL can be exponentially more succinct than LTL+Past (which in turn can be more succinct than LTL). We study satisfiability and model checking for NLTL and provide optimal automata-theoretic algorithms for these EXPSPACE*-complete problems.*

1. Introduction

Temporal logic and verification. Temporal logic provides a fundamental framework for formally specifying systems and reasoning about them [4, 18]. Model checking techniques further allow the automatic verification that a finite-state model of a system satisfies a temporal logic specification [2, 1].

Temporal logic with past. Usually, specification and verification is done with *pure-future* temporal logics, i.e. logics where the modalities only refer to the future of the current time. It is well-known that temporal logics combining past and future modalities make some specifications easier to write and more natural [17]. However, allowing past-time makes verification algorithms harder to implement (though not necessarily from a complexity-theoretic viewpoint). Additionally, all the main temporal logics with past-time admit translations to their pure-future fragment [11, 7, 6, 8, 24, 15, 26].

Forgettable past and the N modality. Being able to refer to past moments is often useful, but there also exist situ- … where it is convenient to *forget the past*. Consider for … temporal formula

where "F^{-1}" means "at some past time". (Spec1) states that "*whenever the alarm rings, then there has been some problem in the past*", i.e. the alarm does not ring without due cause. If now the alarm has a *reset* button, and we want to state that (Spec1) holds after any reset, the formula

$$G(\texttt{reset} \Rightarrow G(\texttt{alarm} \Rightarrow F^{-1}\,\texttt{problem})) \qquad \text{(Spec2)}$$

is not exactly what we aim at. With (Spec2), a problem that occurred *before the reset* may account for the alarm ringing, which is probably not what we had in mind.

For this kind of situations, Laroussinie and Schnoebelen [15, 16] proposed to use a new modality, N (read "*Now*"[1], or "*from now on*") that forgets all the past moments (see below for the formal semantics). With N, one can state the intended property of the alarm example via e.g.

$$G(\texttt{reset} \Rightarrow NG(\texttt{alarm} \Rightarrow F^{-1}\,\texttt{problem})) \qquad \text{(Spec3)}$$

Not much is known about N, except that the translations of *LTL*+Past and *CTL**+Past into (resp.) *LTL* and *CTL** carry over to *LTL*+Past+Now and *CTL**+Past+Now [16].

Our contribution. In this paper, we investigate *NLTL*, i.e. *LTL*+Past+Now, and give automata-theoretic decision procedures for model checking and satisfiability. These algorithms run in EXPSPACE, which is optimal: we show that both satisfiability and model checking are EXPSPACE-complete for *NLTL*. This came as a surprise since *NLTL* does not appear to be a big departure from *PLTL*, i.e. *LTL*+Past, for which verification is (as for *LTL*) PSPACE-complete [21].

The increased complexity of *NLTL* is partly explained by its expressive power: we prove that *NLTL* can be exponentially more succinct than *PLTL*, and that *PLTL* can

[1] No connection with the Now of H. Kamp, Formal properties of 'now', *Theoria*, 227–263, 1971.

(Linear-time) temporal logic with forgettable past

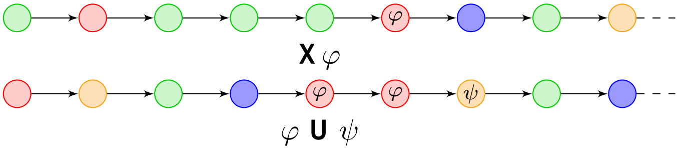
(Linear-time) temporal logic with forgettable past

(Linear-time) temporal logic with forgettable past

- **atomic propositions:** , , ...
- **Boolean combinators:** $\neg\varphi$, $\varphi \vee \psi$, $\varphi \wedge \psi$, ...

(Linear-time) temporal logic with forgettable past

- **atomic propositions:** , , ...
- **Boolean combinators:** $\neg\varphi$, $\varphi \vee \psi$, $\varphi \wedge \psi$, ...
- **temporal modalities:**

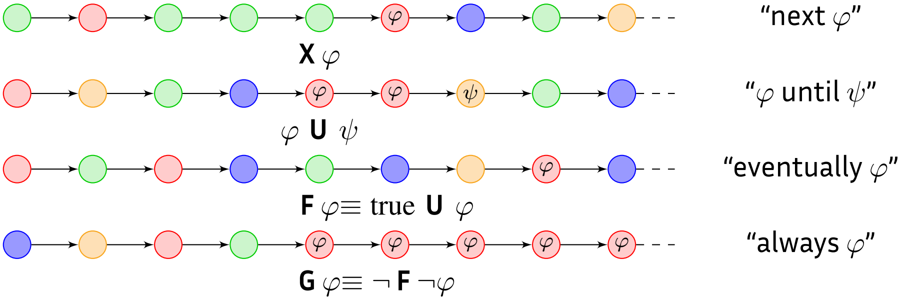


“next φ”

“φ until ψ”

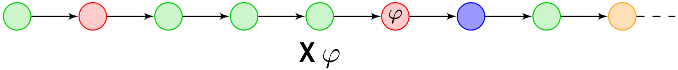
(Linear-time) temporal logic with forgettable past

- **atomic propositions:** (green), (red), ...
- **Boolean combinators:** $\neg\varphi$, $\varphi \vee \psi$, $\varphi \wedge \psi$, ...
- **temporal modalities:**

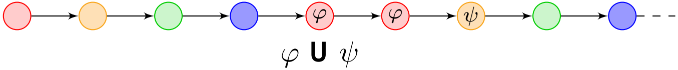


(Linear-time) temporal logic with forgettable past

- **atomic propositions:** , , ...
- **Boolean combinators:** $\neg\varphi, \varphi \vee \psi, \varphi \wedge \psi$, ...
- **temporal modalities:**

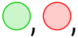


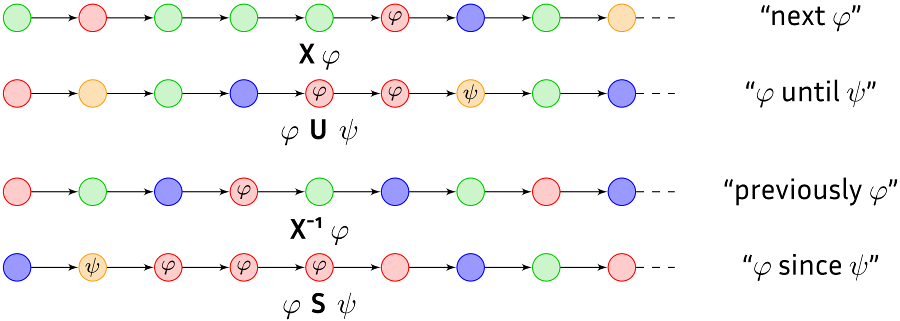
"next φ"



"φ until ψ"

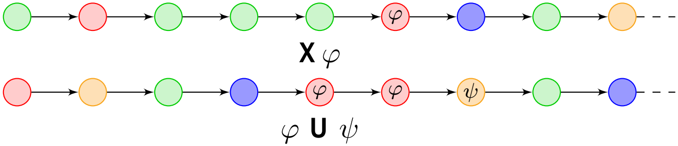
(Linear-time) temporal logic with forgettable past

- **atomic propositions:** , ...
- **Boolean combinators:** $\neg\varphi$, $\varphi \vee \psi$, $\varphi \wedge \psi$, ...
- **temporal modalities:**



(Linear-time) temporal logic with forgettable past

- **atomic propositions:** , , ...
- **Boolean combinators:** $\neg\varphi$, $\varphi \vee \psi$, $\varphi \wedge \psi$, ...
- **temporal modalities:**



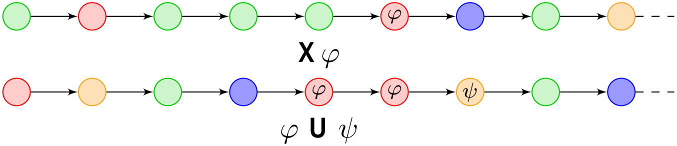
"next φ"

"φ until ψ"

Example

(Linear-time) temporal logic with forgettable past

- **atomic propositions:** , , ...
- **Boolean combinators:** $\neg\varphi$, $\varphi \vee \psi$, $\varphi \wedge \psi$, ...
- **temporal modalities:**



"next φ"

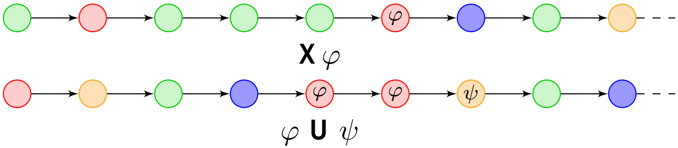
"φ until ψ"

Example

$\mathbf{G}(\neg \textit{submit}) \Rightarrow \mathbf{G}(\neg \textit{accepted})$

(Linear-time) temporal logic with forgettable past

- **atomic propositions:** ⬤, ⬤, ...
- **Boolean combinators:** $\neg\varphi, \varphi \vee \psi, \varphi \wedge \psi$, ...
- **temporal modalities:**



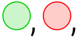
"next φ"

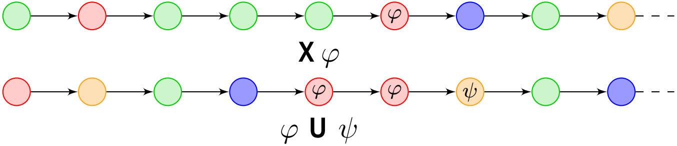
"φ until ψ"

Example

$\neg\,\mathbf{F}(\mathit{accepted} \wedge \mathbf{X}^{-1}\,\mathbf{G}^{-1}\,\neg\mathit{submit})$

(Linear-time) temporal logic with forgettable past

- **atomic propositions:** , ...
- **Boolean combinators:** $\neg\varphi, \varphi \vee \psi, \varphi \wedge \psi$, ...
- **temporal modalities:**



"next φ"

"φ until ψ"

Example

$$\neg\,\mathbf{F}(\textit{accepted} \wedge \mathbf{X}^{-1}\,\mathbf{G}^{-1}\,\neg\textit{submit}) \equiv_i \neg((\neg\textit{submit})\ \mathbf{U}\ \textit{accepted})$$

(Linear-time) temporal logic with forgettable past

Theorem (Sistla, Clarke (1982) + Vardi, Wolper (1986))

Model checking PastLTL and LTL is PSPACE-complete.

PastLTL and LTL formulas can be compiled into equivalent exponential-size Büchi automata.


THE COMPLEXITY OF PROPOSITIONAL LINEAR TEMPORAL LOGICS

A. P. Sistla and E. M. Clarke
Aiken Computation Laboratory
Harvard University
Cambridge, MA 02138

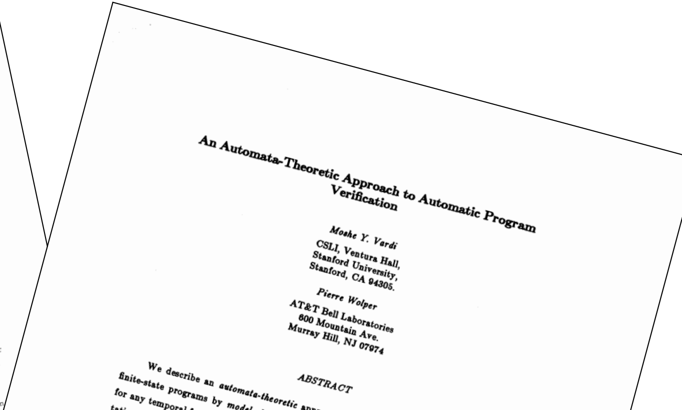
Abstract

We consider the complexity of satisfiability and determination of truth in a particular finite structure for different propositional linear temporal logics. We show that both the above problems are NP-complete for the logic with F operator and are PSPACE-complete for the logics with F,X, with U, with U,S,X, and Wolper's extended logic with regular operators [Wo81].

1. Introduction

ments of the individual processes. An important special case occurs when the program is finite state. In this case, the program axioms and correctness specification can be expressed in the *propositional* version of the logic and provability becomes *decidable*. A number of researchers (e.g., [MW81]) have attempted to use such a decision procedure for constructing correct finite-state programs.

In this paper we examine the inherent complexity of decision procedures for *validity*, *satisfiability*, and *truth in a particular structure* for propositional logics with the temporal operators F (eventually), G (globally), X (next-time), U (until) and S (since). We first consider the logic L(F) in which F is the only temporal


An Automata-Theoretic Approach to Automatic Program Verification

Moshe Y. Vardi
CSLI, Ventura Hall,
Stanford University,
Stanford, CA 94305.

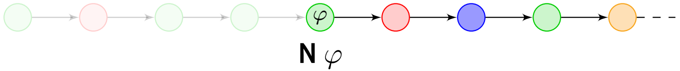
Pierre Wolper
AT&T Bell Laboratories
600 Mountain Ave.
Murray Hill, NJ 07974

ABSTRACT

We describe an *automata-theoretic*

(Linear-time) temporal logic with forgettable past

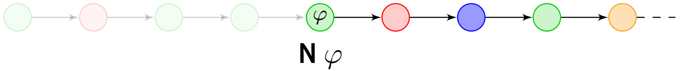
- **operator Now**:



"from now on φ"

(Linear-time) temporal logic with forgettable past

- **operator Now:**



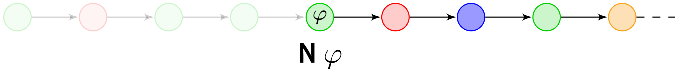
"from now on φ"

Example

$\neg\, \mathbf{N}\, \mathbf{F}(\mathit{accepted} \wedge \mathbf{X}^{-1}\, \mathbf{G}^{-1}\, \neg \mathit{submit})) \equiv \neg((\neg \mathit{submit})\ \mathbf{U}\ \mathit{accepted})$

(Linear-time) temporal logic with forgettable past

- **operator Now**:



"from now on φ"

Example

$\neg\, \mathbf{N}\, \mathbf{F}(\mathit{accepted} \wedge \mathbf{X}^{-1}\, \mathbf{G}^{-1}\, \neg \mathit{submit})) \equiv \neg((\neg \mathit{submit})\ \mathbf{U}\ \mathit{accepted})$

Theorem

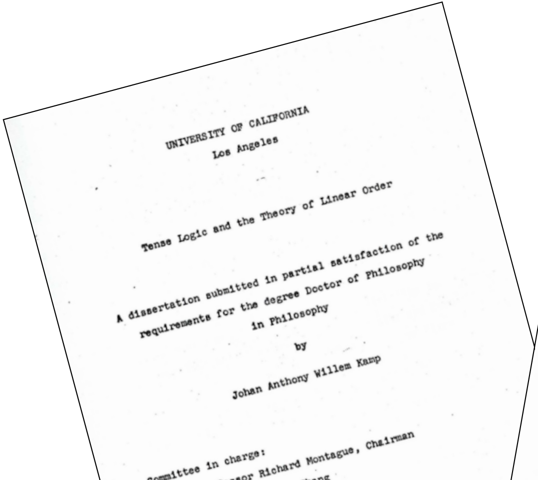
Any formula in PastLTL+Now can be compiled into an equivalent exponential-size alternating Büchi automaton.

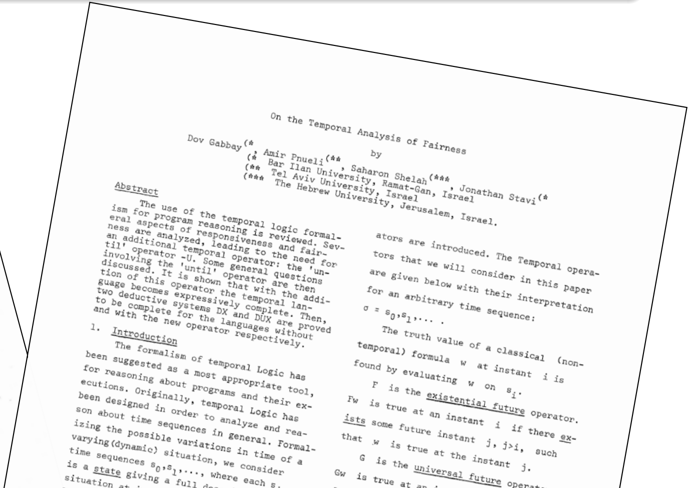
Model checking PastLTL+Now is EXPSPACE-complete.

(Linear-time) temporal logic with forgettable past

Theorem (Kamp (1968) + Gabbay, Pnueli, Shellah, Stavi (1980))

PastLTL and LTL are equally expressive.





(Linear-time) temporal logic with forgettable past

Theorem

PastLTL can be exponentially more succinct than LTL.

(Linear-time) temporal logic with forgettable past

Theorem

PastLTL can be exponentially more succinct than LTL.

Proof

R_n: “any state that agrees with the initial state on propositions p_1 to p_n also agrees on p_0”

(Linear-time) temporal logic with forgettable past

Theorem

PastLTL can be exponentially more succinct than LTL.

Proof

R_n: "any state that agrees with the initial state on propositions p_1 to p_n also agrees on p_0"

- property R_n can be expressed as a "small" PastLTL formula;

(Linear-time) temporal logic with forgettable past

Theorem

PastLTL can be exponentially more succinct than LTL.

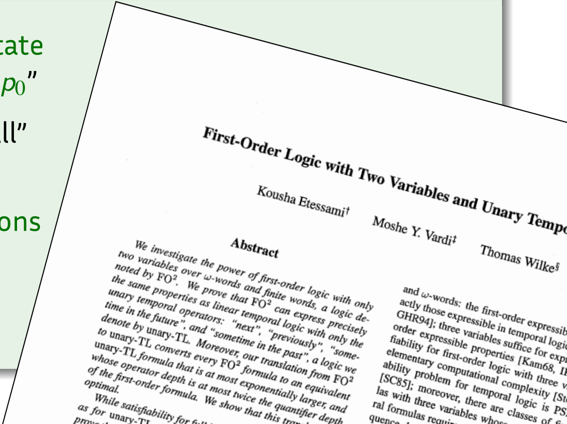
Proof

R_n: "any state that agrees with the initial state on propositions p_1 to p_n also agrees on p_0"

- property R_n can be expressed as a "small" PastLTL formula;

R'_n: "any two states that agree on propositions p_1 to p_n also agree on p_0"

- property R'_n cannot be expressed as a "small" (Past)LTL formula.


First-Order Logic with Two Variables and Unary Tempo

Kousha Etessami† Moshe Y. Vardi‡ Thomas Wilke§

Abstract

We investigate the power of first-order logic with only two variables over ω-words and finite words, a logic denoted by FO². We prove that FO² can express precisely the same properties as linear temporal logic with only the unary temporal operators: "next", "previously", "sometime in the future", and "sometime in the past", a logic we denote by unary-TL. Moreover, our translation from FO² to unary-TL converts every FO² formula to an equivalent unary-TL formula that is at most exponentially larger, and whose operator depth is at most twice the quantifier depth of the first-order formula. We show that this tra[...] optimal.