Temporal Logic with Past is Exponentially More Succinct

Nicolas Markey

Lab. Informatique Fondamentale d'Orléans
Univ. Orléans & CNRS FRE 2490
Rue Léonard de Vinci - BP 6759
45067 Orléans Cedex 2 - France

markey@lifo.univ-orleans.fr

Abstract

We positively answer the old question whether temporal logic with past is more succinct than pure-future temporal logic. Surprisingly, the proof is quite simple and elementary, although the question has been open for twenty years.

Introduction

Temporal logics with past. Temporal logic have been defined by Arthur Prior in 1957 [Pri57] as a tool for reasoning about temporal informations in a logical framework. This logic uses temporal modalities for referencing to past or future events. It has been introduced in the field of formal verification by Amir Pnueli in 1977 [Pnu77]. Leslie Lamport suggested to classify temporal logics into two general kinds: linear-time temporal logic and branching-time temporal logic [Lam80]. In linear-time temporal logic (LTL), we can express for instance that any request is eventually granted:

$$\mathbf{G}\,(\mathtt{request} \Rightarrow \mathbf{F}\,\mathtt{grant}). \tag{1}$$

With past-time modalities, we can express that a grant should be preceeded by a request:

$$\mathbf{G}\,(\mathtt{grant} \Rightarrow \mathbf{F}^{-1}\,\mathtt{request}). \tag{2}$$

With or without past? In 1980, Gabbay proved that past-time modalities did not add expressive power to pure future linear-time temporal logic [GPSS80]. He also provided an algorithm for translating formulas with past-time modalities into equivalent pure future ones [Gab89]. Another procedure was given in

[MMSKR94] for temporal logic without next. For instance, equation (2) above can be written

$$\neg((\neg\texttt{request})\,\mathbf{U}\,(\texttt{grant}\wedge\neg\texttt{request})). \tag{3}$$

This formula is more intricate than the natural formula of equation (2). Moreover, the size of the formula computed by the algorithm is assumed to be non elementary in the size of the initial one. But Gabbay's theorem has been used as an argument for dropping past-time modalities from linear-time temporal logic. What we show here is that, as regards succinctness, past-time modalities *do* add expressive power (this result has been published in [LMS02]).

Outline. I will first briefly recall some definitions about linear-time temporal logics with past, and the possibility to "remove" past-time modalities. I will then focus on the translation of LTL formulas into Büchi automata, and more precisely on the size of the resulting automaton. I will then use this result for proving the succinctness gap between LTL+Past and pure future LTL.

1 From PLTL to LTL

LTL and PLTL. Let Prop $= \{p, q, \ldots\}$ be a finite set of atomic propositions. Linear-time temporal logic with past-time modalities (which we denote PLTL in the sequel[1]) is defined with the following syntax:

$$\mathsf{PLTL} \ni \phi,\psi ::= \neg\phi \mid \phi\vee\psi \mid \phi\,\mathbf{U}\,\psi \mid \mathbf{X}\,\phi \mid \phi\,\mathbf{S}\,\psi \mid \mathbf{X}^{-1}\,\phi \mid p \mid q \mid \ldots$$

The pure future fragment of PLTL, denoted by LTL, is defined as follows:

$$\mathsf{LTL} \ni \phi,\psi ::= \neg\phi \mid \phi\vee\psi \mid \phi\,\mathbf{U}\,\psi \mid \mathbf{X}\,\phi \mid p \mid q \mid \ldots$$

Formulas are interpreted at some position along linear paths, *i.e.* along infinite sequences (ω-orders) of elements of 2^{Prop}. The semantics for atomic propositions and boolean operators are the classical ones. For modalities, given a path π and a position i, we have:

$$\begin{array}{lll}
\pi,i\models\phi\,\mathbf{U}\,\psi & \text{if, and only if,} & \exists k\geq i.\ (\pi,k\models\psi\ \wedge\ \forall i\leq j<k.\ \pi,j\models\phi)\\
\pi,i\models\mathbf{X}\,\phi & \text{if, and only if,} & \pi,i+1\models\phi\\
\pi,i\models\phi\,\mathbf{S}\,\psi & \text{if, and only if,} & \exists k\leq i.\ (\pi,k\models\psi\ \wedge\ \forall k<j\leq i.\ \pi,j\models\phi)\\
\pi,i\models\mathbf{X}^{-1}\,\phi & \text{if, and only if,} & i\geq 1 \text{ and } \pi,i-1\models\phi
\end{array}$$

The classical abbreviations $\mathbf{F}$ (eventually) and $\mathbf{G}$ (always), and their past-time counterparts, are defined by:

$$\mathbf{F}\,\phi \stackrel{\text{def}}{=} \top\,\mathbf{U}\,\phi \qquad \mathbf{G}\,\phi \stackrel{\text{def}}{=} \neg\mathbf{F}\,\neg\phi \qquad \mathbf{F}^{-1}\,\phi \stackrel{\text{def}}{=} \top\,\mathbf{S}\,\phi \qquad \mathbf{G}^{-1}\,\phi \stackrel{\text{def}}{=} \neg\mathbf{F}^{-1}\,\neg\phi$$

[1] Here, P in PLTL stands for Past.

Two formulas ϕ, ψ of PLTL are said to be equivalent (which we write $\phi \equiv \psi$) if, and only if, they verify the following property:

$$\text{for any path } \pi \text{ and any position } i,\ \ \pi, i \models \phi \ \Leftrightarrow\ \pi, i \models \psi.$$

Initial equivalence is a weaker notion of equivalence: ϕ and ψ are initially equivalent ($\phi \equiv_i \psi$) if, and only if,

$$\text{for any path } \pi,\ \ \pi, 0 \models \phi \ \Leftrightarrow\ \pi, 0 \models \psi.$$

Both notions of equivalence clearly coincide for LTL, but not for PLTL formulas.

Translations to pure-future temporal logic. [GPSS80] proves that LTL is expressive complete, thus as expressive as PLTL [Kam68]. Moreover, [Gab89] presents a syntactic algorithm for translating a PLTL formula into an initially equivalent LTL one. For many computer scientists, this has been one reason for not considering past in temporal logics, by concern of minimality.

However, since 1980, the cost of the translation has not been precisely characterized. The algorithm provided by [Gab89] is assumed to be non elementary.

By a detour through counter-free Büchi automata, it is possible to get a more efficient algorithm: first translate the formula $\phi \in$ PLTL into a Büchi automaton [LPZ85], then translate that automaton into an equivalent deterministic Muller automaton. The resulting automaton can be assumed to be counter free, since otherwise, the language it defines would not be star free. Then [MP90, MP94] provides a translation of counter free Muller automaton into LTL. All of the three steps possibly involves an exponential blowup, and the size of the final formula is at most triply exponential in the size of the initial one.

In the sequel, we prove that at least one exponential is unavoidable, *i.e.* there exists a family of PLTL formulas ϕ_n with size $O(n)$, whose equivalent formulas have size $\Omega(2^n)$.

2 LTL and Büchi automata

Let's get back to the aforementioned translation of LTL formulas into Büchi automata. The important theorem is the following:

Theorem 1 ([WVS83, VW86])
Given an LTL formula ϕ, one can build a Büchi automaton $\mathcal{A}_\phi = (\Sigma, S, \rho, S_0, F)$ where $\Sigma = 2^{\mathrm{Prop}}$ and $|S| = 2^{O(|\phi|)}$, such that $L(\mathcal{A}_\phi)$ is exactly the set of paths satisfying the formula ϕ.

Let n be a nonnegative integer, $\mathrm{Prop} = \{p_0, p_1, \ldots, p_n\}$. We consider the following path property, mentionned in [EVW02]:

Any two positions of the path that agree on propositions p_1, $p_2, \ldots, p_n$ also agree on proposition p_0. (4)

This property can be expressed by the following LTL formula, which tests all the possible valuations for the atomic propositions:

$$\phi_n \stackrel{\text{def}}{=} \bigwedge_{\substack{a_i \in \{\top,\bot\} \\ i\in[1,n]}} \Big[\Big(\mathbf{F}\,(\bigwedge_{i=0}^{n} p_i = a_i)\Big) \Rightarrow \mathbf{G}\,\Big((\bigwedge_{i=1}^{n} p_i = a_i) \Rightarrow (p_0 = a_0)\Big)\Big]$$

This formula has size $2^{O(n)}$.

We now prove that there is no polynomial size LTL formula expressing the same statement. The proof is adapted from [EVW02]. Assume that there exists a polynomial-size LTL formula for the property (4). From the above Theorem 1, there would exist a Büchi automaton of size single exponential whose language is exactly the set of computations verifying ϕ_n.

We show that this is not possible: any automaton recognizing $L(\phi_n) = \{u \in (2^{\text{Prop}})^\omega \mid u \models \phi_n\}$ requires at least 2^{2^n} states. Indeed, let $\mathcal{A}$ be an automaton recognizing exactly $L(\phi_n)$, let $a_0, \ldots, a_{2^n-1}$ be any sequence of the 2^n subsets of $\{p_1, \ldots, p_n\}$, and let K be a subset of $\{0, \ldots, 2^n-1\}$. We define a sequence of letters $b_i \in \Sigma$ as follows:

$$b_i \stackrel{\text{def}}{=} \begin{cases} a_i & \text{if, and only if, } i \notin K \\ a_i \cup \{p_0\} & \text{if, and only if, } i \in K \end{cases}$$

We also define the finite word $w_K \stackrel{\text{def}}{=} b_0 b_1 \cdots b_{2^n-1}$. Clearly, two different choices of K lead to two different sequences of $(b_i)_{i=0\ldots 2^n-1}$, thus to two different words w_K. Therefore, there exists exactly 2^{2^n} such words.

Let K and K' be two distinct subsets of $\{0, \ldots, 2^n - 1\}$. Obviously, the words w_K^ω and $w_{K'}^\omega$ are accepted by the automaton $\mathcal{A}$. There exist two paths π_K and $\pi_{K'}$ in the automaton $\mathcal{A}$ accepting the words w_K^ω and $w_{K'}^\omega$, respectively. Consider the 2^n-th state of each of these paths, which we name q_K and $q_{K'}$. If these states were identical, then the suffix π_K starting from state q_K could be appended to the prefix of $\pi_{K'}$ up to state $q_{K'}(= q_K)$, thus giving an accepting path (since the Büchi acceptance conditions are satisfied along π_K) for the word $w_{K'} \cdot w_K^\omega$. But that word $w_{K'} \cdot w_K^\omega$ should not be accepted by $\mathcal{A}$ since it does not satisfy the formula ϕ_n. Thus any automaton recognizing exactly $L(\phi_n)$ has at least 2^{2^n} states, so that $|\phi_n|$ is in $\Omega(2^n)$.

3 PLTL is exponentially more succinct than LTL

In order to prove the succinctness result, we will use a slightly modified property, namely:

> *Any position of the path that agrees on propositions $p_1, p_2, \ldots,$ p_n with the initial state also agrees on proposition p_0.* (5)

This property can be expressed in PLTL through the following formula:

$$\psi_n \stackrel{\text{def}}{=} \mathbf{G}\,\Big[\Big(\bigwedge_{i=1}^{n}(p_i \Leftrightarrow \mathbf{F}^{-1}\,\mathbf{G}^{-1}\,p_i)\Big) \Rightarrow (p_0 \Leftrightarrow \mathbf{F}^{-1}\,\mathbf{G}^{-1}\,p_0)\Big].$$

This formula clearly expresses property (5), since $\mathbf{F}^{-1}\,\mathbf{G}^{-1}\,p$ means that p holds in the initial state of the path. Moreover, ϕ_n has size $O(n)$.

Since any PLTL formula can be translated into an initially equivalent LTL formula, we let ψ'_n be an LTL formula initially equivalent to ψ_n, and we define $\phi'_n \stackrel{\text{def}}{=} \mathbf{G}\,\psi'_n$. We claim that ϕ_n and ϕ'_n are equivalent. Indeed, that some path π satisfies ϕ'_n exactly means that for all i, $\pi, i \models \psi'_n$. Since $\psi'_n \in$ LTL, this amounts to say that $\pi^i, 0 \models \psi'_n$. Since $\psi'_n \equiv_i \psi_n$, it is equivalent to the fact that, for all i, $\pi^i, 0 \models \psi_n$, that is "for all position i, for all position j, if position j agrees with position i on propositions $p_1, p_2, \ldots, p_n$, then it also agrees with position i on proposition p_0", which we means that π satisfies ϕ_n.

Bringing all the pieces together, we get the following theorem:

Theorem 2
PLTL *can be exponentially more succinct than* LTL.

The proof is quite simple now: consider formula ψ_n, and its LTL equivalent ψ'_n. We know that ψ_n has size linear in n, and we know that $\mathbf{G}\,\psi'_n$ has size $\Omega(2^n)$, since it is an LTL formula expressing property (4). Thus ψ'_n has size at least $\Omega(2^n)$. In fact, this even proves that ψ'_n has at least $\Omega(2^n)$ distinct subformulas, since in the construction of $\mathcal{A}_\phi$, the states are the subsets of the set of subformulas of ϕ (thus ψ'_n cannot even be succinctly represented as a dag sharing common subformulas).

Remark that the proof requires formulas with bounded temporal height, and uses an unbounded number of atomic propositions. It can be adapted in order to use only finitely many atomic propositions, but with an unbounded temporal height (by encoding atomic propositions with only one atomic proposition [DS02]). Moreover, the proof uses neither the **U** nor the **S** modalities. The result thus carries over to $\mathrm{L}(\mathbf{F}, \mathbf{F}^{-1})$. It can be shown however that $\mathrm{L}(\mathbf{F}, \mathbf{F}^{-1})$ is strictly more expressive (as regards to the classical notion of expressivity) than $\mathrm{L}(\mathbf{F})$.

Conclusion

This simple proof only partially answers the question on the size of the optimal translation from PLTL to LTL: we proved that the gap is at least single exponential, but the best known upper bound is triply exponential.

References

[DS02] Stéphane Demri and Philippe Schnoebelen. The Complexity of Propositional Linear Temporal Logics in Simple Cases. *Information and Computation*, 174(1), pages 84–103, Academic Press, April 2002.

[EVW02] Kousha Etessami, Moshe Y. Vardi, and Thomas Wilke. First-Order Logic with Two Variables and Unary Temporal Logic. *Infor-*

mation and Computation, 179(2), pages 279–295, Academic Press, December 2002.

[Gab89] Dov M. Gabbay. The Declarative Past and Imperative Future: Executable Temporal Logic for Interactive Systems. In Behnam Banieqbal, Howard Barringer, and Amir Pnueli, editors, *Proceedings of the 1st Conference on Temporal Logic in Specification*, April 1987, volume 398 of *Lecture Notes in Computer Science*, pages 409–448. Springer-Verlag, 1989.

[GPSS80] Dov M. Gabbay, Amir Pnueli, Saharon Shelah, and Jonathan Stavi. On the Temporal Analysis of Fairness. In *Conference Record of the 7th ACM Symposium on Principles of Programming Languages (POPL'80)*, January 1980, pages 163–173. ACM Press, January 1980.

[Kam68] Hans W. Kamp. *Tense Logic and the Theory of Linear Order*. PhD thesis, UCLA, Los Angeles, California, USA, 1968.

[Lam80] Leslie Lamport. "Sometimes" is sometimes "Not Never". In *Conference Record of the 7th ACM Symposium on Principles of Programming Languages (POPL'80)*, January 1980, pages 174–185. ACM Press, January 1980.

[LMS02] François Laroussinie, Nicolas Markey, and Philippe Schnoebelen. Temporal Logic with Forgettable Past. In *Proceedings of the 17th Annual Symposium on Logic in Computer Science (LICS 2002)*, July 2002, pages 383–392. IEEE Comp. Soc. Press, July 2002.

[LPZ85] Orna Lichtenstein, Amir Pnueli, and Lenore D. Zuck. The Glory of the Past. In Rohit Parikh, editor, *Proceedings of the Conference on Logics of Programs*, June 1985, volume 193 of *Lecture Notes in Computer Science*, pages 413–424. Springer-Verlag, June 1985.

[MMSKR94] Louise E. Moser, P. Michael Melliar-Smith, George Kutty, and Y. Srinivas Ramakrishna. Completeness and Soundness of Axiomatizations for Temporal Logics without Next. *Fundamenta Informaticae*, 21(4), pages 257–305, IOS Press, October 1994.

[MP90] Oded Maler and Amir Pnueli. Tight Bounds on the Complexity of Cascaded Decomposition of Automata. In *Proceedings of the 31st Annual Symposium on Foundations of Computer Science (FOCS'90)*, October 1990, pages 672–682. IEEE Comp. Soc. Press, October 1990.

[MP94] Oded Maler and Amir Pnueli. On the Cascade Decomposition of Automata, its Complexity and its Application to Logic. Available at: `<http://www-verimag.imag.fr/PEOPLE/Oded.Maler/Papers/decomp.ps>`, 1994.

[Pnu77] Amir Pnueli. The Temporal Logic of Programs. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science (FOCS'77)*, October-November 1977, pages 46–57. IEEE Comp. Soc. Press, October 1977.

[Pri57] Arthur N. Prior. *Time and Modality*. Clarendon Press, 1957.

[VW86] Moshe Y. Vardi and Pierre Wolper. An Automata-Theoretic Approach to Automatic Program Verification. In *Proceedings of the 1st Annual Symposium on Logic in Computer Science (LICS'86)*, June 1986, pages 332–344. IEEE Comp. Soc. Press, June 1986.

[WVS83] Pierre Wolper, Moshe Y. Vardi, and A. Prasad Sistla. Reasoning about Infinite Computation Paths. In *Proceedings of the 24th Annual Symposium on Foundations of Computer Science (FOCS'83)*, November 1983, pages 185–194. IEEE Comp. Soc. Press, November 1983.