More efficient proof-search for sequents of temporal logic

Articles

More efficient proof-search for sequents of temporal logic

Efektyvesnė laiko logikos sekvencijų įrodymo paieška

Romas Alonderis romas.alonderis@mif.vu.lt
Vilnius University, Lituania
https://orcid.org/0000-0002-7792-5285

Lietuvos matematikos rinkinys, vol. 63
Ser. A, 2022

Vilniaus Universitetas, Lituania

Recepción: 03 Mayo 2022
Publicación: 10 Diciembre 2022

DOI: https://doi.org/10.15388/LMR.2022.29752

Redalyc: https://www.redalyc.org/articulo.oa?id=692674313001

Abstract: The present paper deals with efficiency improvement of backward proof-search of sequents of propositional linear temporal logic, using a loop-type sequent calculus. The improvement is achieved by syntactic transformation of sequents into equivalent to them simpler ones. It is proved that some formulas can be removed from sequents with no impact on their derivability.

Keywords: temporal logics, backward proof-search, loop-type sequent calculi.

Summary: Šiame straipsnyje pateikiamas dalinis metodas leidžiantis gauti efektyvesnę sekvencijų įrodymo paiešką propozicinei tiesinio laiko logikai, naudojant ciklinį sekvencinį skaičiavimą. Šis metodas yra pagrįstas sintaksine sekvencijų transformacija į joms ekvivalenčias paprastesnes sekvencijas. Straipsnyje taip pat parodoma, kad kai kurios formulės gali būti pašalintos iš sekvencijų niekaip nepaveikiant jų įrodomumo.

Keywords: laiko logika, atgalinė, irodymo paieška, cikliniai sekvenciniai skaičiavimai.

1 Introduction

Propositional linear temporal logic (**PLTL**) is used in computer science for specification and verification of programs [2, 5]. Sequent calculi are convenient tools for check of formula validity by means of proof-search. Various tableaux and sequent deductive systems are considered in the literature: tableaux proof-search systems [9, 13]; infinitary sequent calculi containing ω-type induction rule [10]; sequent calculi with invariant-like rule [7, 11, 12]; saturated sequent calculi [8]; a cut-free and invariant-free sequent calculus [4]; loop-type sequent calculi based on sequent history method [3, 6]; loop-type sequent calculus based on derivation loop check [1]. Backward proof-search using the sequent calculus $\mathbf{G_LT}$ introduced in [1] involves checks of global conditions, so called derivation loops. The checks hinder efficiency of proof-search. The present paper concerns with some partial methods allowing us to make proof-search shorter, reducing the number of the checks and hence making the proof-search more efficient. This is achieved by syntactic transformation of sequents into equivalent to them simpler ones. It has been proved that some formulas can be removed from sequents with no impact of their derivability. The present paper is organized as follows. In Section 2, we recall the syntax and semantics of **PLTL** and the calculus $\mathbf{G_LT}$. The correct sequents are defined in Section 3. Sequent simplification and backward proof-search reduction are considered in Section 4. Some concluding remarks are in Section 5.

2 Syntax, semantics, and sequent calculus G$_L$T

The language of **PLTL** contains a set P of propositional symbols $\{p, p_1, p_2, ..., q, q_1, q_2, ...\}$ the logical operators $\neg, \vee, \wedge, \supset,$, temporal operators $\square$ ("henceforth always") and # ("next"). The language does not contain the temporal operator # ("sometimes"), assuming that $\square\varnothing = \neg\square\neg\ \varnothing$. Propositional symbols are called atomic formulas. The formulas $\varnothing$ of **PLTL** are inductively defined as follows:

$$\varnothing ::= p \mid \neg\ \varnothing \mid \varnothing \vee \psi \mid \varnothing \wedge \psi \mid \varnothing \supset \psi \mid \square\varnothing \mid \square\ \varnothing$$

The Greek letters $\varnothing$ and ψ are used to denote arbitrary formulas. The expression O^n denotes the sequence of *n'o' - s, e.g., $o^2p = oop$..*

An interpretation M= (N, I) consists of the set of natural numbers N and the function $I : \mathrm{N} \mapsto 2^{\mathrm{P}}$, where 2^{P} is the set of subsets of P. The semantics of **PLTL** formulas is provided by the satisfaction relation $\vDash$:

$$M, j \vDash p, iff\ p \in I(j);$$

$$M, i \vDash \neg\ \varnothing, iff M, i \square\ \varnothing;$$

$$M, i \vDash \varnothing\ V\ \psi, iff M, i \vDash \varnothing\ or M, i \vDash \psi;$$

$$M, i \vDash \varnothing \wedge \psi, iff M, i \vDash \varnothing\ and M, i \vDash \psi;$$

$$M, i \vDash\ \supset \psi, iff M, i \square\ \varnothing\ or M, i \vDash \psi;$$

$$M, i \vDash O\varnothing, iff M, i+1 \vDash \varnothing;$$

$$M, i \vDash \square\varnothing, iff M, j \vDash \varnothing\ \$\ for all j \geq i.$$

An interpretation M is a *model* for a formula $\varnothing$, iff M,0|=$\varnothing$. A formula $\varnothing$ is called valid, $\vDash \varnothing$ in notation, iff every interpretation is a model for $\varnothing$

The sequent calculus **G$_L$T** is defined in [1]. We recall here some definitions. The temporal rules:

$$\frac{|T \Rightarrow \Delta\ |}{\Sigma, oT \Rightarrow o\Delta,\ \Sigma'}\left(o\right)\frac{|\quad \varnothing, o\square\quad \varnothing, T \Rightarrow \Delta\ |}{\square\varnothing, T \Rightarrow \Delta}\left(\square \Rightarrow\right),$$

$$\frac{|\,T \Rightarrow \Delta,\ \varnothing\ |\,|\,T \Rightarrow \Delta, o\square\varnothing}{T \Rightarrow \Delta, \square\varnothing}\left(\Rightarrow\square\right)$$

Here: *Γ, Δ,Σ,Σ′* denote finite, possibly empty, multisets of formulas, where $\Sigma \cup \Sigma'$ consists of atomic formulas; the conclusion is not an axiom and $T \cup \Delta \neq \theta$ in (O).

Given a sequent *S*, a GLT proof-search tree with the sequent S at the root is constructed in usual way by subsequently applying backwards the GLT derivation rules to S and the sequents obtained in the course of the tree construction. A proof search tree is denoted by *V*. The expression *V(S)* denotes that *S* is the root of *V*.

We say that a sequent *S#* subsumes *S*($S' \Box S$ in notation), iff *S#* can be inferred from *S* by the structural rule of weakening. If $S' = S$, then we say S' strongly subsumes *S*.

Definition 1. Given a proof-search tree, the upward path *p* from some sequent *S* in the tree to *S#* inclusive is called a (strong) derivation loop, $[S - S']$ in notation, iff: 1) the length of *p* is greater than 0 and 2) $S' \Box S$ $(S' = S)$. The nodes marked with *S* and *S#* are called the base and terminal of $[S - S']$, respectively. The sequents S and S' are called the base and terminal sequents of $[S - S']$, respectively. It is true that $\lambda(S) \Box \lambda(S')$.

Definition 2. A (strong) derivation loop $[S - S']$ is called a (strong) derivation loop with the universality formula $\Box\varnothing$, iff: 1) $S = (\Gamma \Rightarrow \Delta, \theta)$, 2) $S' = \Pi, \Gamma \Rightarrow \Delta, \theta, \Lambda$ where $\theta \in \{\Box\varnothing, \Box\varnothing\}$, and 3) $[S - S']$ contains the right premise of $(\Rightarrow \Box\varnothing)$, and does not contain the left premise of $(\Rightarrow \Box\varnothing)$.

If a derivation loop is not the derivation loop with a universality formula, then the derivation loop is called *α*-void.

The following proposition is proved in [1]:

Proposition 1. *Any derivation loop* $[S - S']$ *has an application of.* (O).

Definition 3. A sequent *S* is called derivable in $\mathbf{G_LT}$ ($\vdash S$ in notation), iff there exists a backward proof-search tree *V(S)* such that each leaf of *V(S)* is an axiom or a terminal sequent of a derivation loop with some universality formula. Such a tree *V(S)* is called a derivation of *S* or a derivation tree.

3 Correct sequents

We write $\psi \equiv \phi$, iff $M, i \vDash \varnothing$ implies $M, i \vDash \varnothing$ and vice versa for any pair *M,i*.

Proposition 2.

$$O(\varnothing\,\theta\,\psi) \equiv O\varnothing\,\theta\varnothing\,\psi,\ O\eta\varnothing \equiv \eta O\varnothing,$$

where $\theta \in \{\vee, \wedge, \supset\}$ and $\eta \in \{\Box, \neg\}$.

Proof. Let us consider, e.g., the case when $\eta = \Box$. If $M, i \vDash O\Box\psi$, then $M, i+1 \vDash \Box\psi$. Hence $M, j+1 \vDash \psi$ for all $j \geq i$. This fact implies $M, j \vDash O\psi$ for all $j \geq i$. We obtain $M, i \vDash \Box O\psi$.

If $M, i \vDash \Box O\psi$, then $M, j \vDash O\psi$ for all $j \geq i$. Hence $M, j+1 \vDash \psi$ for all $j \geq i$.

This fact implies $M, i+1 \vDash \Box\psi$. It follows that $M, i \vDash O\Box\psi$.

The remaining cases are considered similarly, using the semantics of propositional connectives and'*O*'.

Corollary 1. If $\varnothing \equiv \psi$, then $\vDash \varnothing\ iff \vDash \psi$.

Using Proposition 2,wepusheach '*O*' inward in formulas so that it binds only propositional symbols and other '*O*'. For example, the formula

$$\Box O\left(q \supset O\neg\left(p \wedge q_1\right)\right)$$

is transformed into the formula $\Box\left(Oq \supset \neg\left(OOp \wedge OOq_1\right)\right)$. Such formulas are called *correct*. A sequent S is called *correct*, iff each member of S is correct or of the type $O\Box\varnothing$, where $\Box\varnothing$ is correct.

Proposition 3. *If S is correct, then all sequents in any backward proof-search tree V (S) are correct.*

Proof. The proof follows from the obvious fact that if the conclusion of an arbitrary $\mathbf{G_LT}$ derivation rule is a correct sequent, then any premise of this rule is a correct sequent too.

From now on, we consider only correct sequents. The generality is not lost, since any formula $\varnothing$ can be transformed into a correct formula ψ such that $\varnothing \equiv \psi$, using Proposition 2. Hence $\vdash \varnothing \; iff \vdash \psi$, based on Corollary 1 and the fact that $\mathbf{G_LT}$ is sound and complete, according to Theorems 4.4 and 5.4, respectively, in [1].

4 Sequent and backward proof-search simplification

The sequent $\tau(S)$ is obtained from S by substituting q_i for. $O^n p_i$, where $n > 0$, $O^n p_i$ is not a sub-formula of $OO^n p_i$ in S, and all q_i are different and do not occur in S. The obtained sequent has no formulas of the type$O^n p$, where $n > 0$. For example:

$$T\left(o\Box\left(oop \wedge ooop\right) \Rightarrow ooop,\, p,\, q\right) = o\Box\left(q_1 \wedge q_2\right) \Rightarrow q_1,\, p,\, q.$$

A signed formula $\varnothing^{\sigma}$ is defined inductively as follows:

$$\varnothing^{\sigma}\begin{cases} p^{\sigma} & if\, \varnothing = p, \\ \varnothing_1^{\sigma}\theta^{\sigma}\varnothing_2^{\sigma} & if\, \varnothing = \varnothing_0\theta\varnothing_2 \\ \varnothing_1^{\eta}\supset^{\sigma}\varnothing_2^{\sigma} & if\, \varnothing = \varnothing_1 \supset \varnothing_2 \end{cases}, \quad where\theta \in \{\wedge, V\}, \qquad \begin{cases} \Box^{\sigma} & if\, \varnothing = \neg\,\psi \\ \theta^{\sigma}\Psi^{\sigma} & if\, \varnothing = \theta\psi,\, where\theta \in \{O, \Box\} \end{cases}$$

where η, $\sigma \in \{l,r\}$ and $\eta \neq \sigma$ (“l” stands for left side of the sequent (antecedent) and “r” stands for the right side of the sequent (succedent)). If $T = \left(q_1, \dots, q_m\right)$, then $T^{\sigma} = \left(q_1^{\sigma}, \dots, q_m^{\sigma}\right)$. If $S = (\Pi \Rightarrow \Delta)$, then

$$s^{sg} = \Box^{l} \Rightarrow \Delta^{r}$$

S^{sg} is called a signed sequent. A sequent S is called $\Box^{l}$ - $free$, if S^{sg} does not contain $\Box^{l}$. The maximal class of $\Box^{l}$ - $free$ sequents is denoted by $\mathrm{C}\left(\setminus \Box^{\Box l}\right)$.

Lemma 1. *If* $S \in \mathrm{C}\left(\setminus \Box^{\Box l}\right)$, *then any path in V(S) that goes via the conclusion and left premise of* $(\Rightarrow \Box)$ *is not a derivation loop.*

Proof. If

$$\frac{|T \Rightarrow \varnothing\ \Delta\, |T \Rightarrow O\Box\varnothing,\ \Delta|}{T \Rightarrow \Box\varnothing,\ \Delta}\left(\Rightarrow \Box\right)$$

is in *V(S)*, then any sequent in any upward path π_1 starting with the left premise contains at least one occurrence of $\Box\varnothing$ less than any sequent in any upward path π_2 ending by the conclusion. Hence no sequent in π_1 can subsume any sequent in π_2.

Lemma 2. *If* $S \in \mathrm{C}\left(\backslash\Box^{l}\right)$*, then any derivation loop in V S) consists of sequents of the type* $\Rightarrow O\Box\Delta, \Box\Delta$ *and is strong.*

Proof. Let us consider any path π in *V(S)*. Assume that π starts with a sequent of the type $\Rightarrow O\Box\Delta, \Box\Delta$. If π is a derivation loop, then there is no left premise of $(\Rightarrow\Box)$ in π, according to Lemma 1.Hence π consists of sequents of type $\Rightarrow O\Box\Delta', \Box\Lambda'$ and is strong. If the path π starts with a sequent which is not of the type $\Rightarrow O\Box\Delta, \Box\Lambda$, then it cannot subsume any sequent above (O) in π. Hence π is not a derivation loop, based on Proposition 1.

Lemma 3. *If* $S \in \mathrm{C}\left(\backslash\Box^{l}\right)$*, then any connected component in any V(S) consists of one derivation loop.*

Proof. The proof follows from Lemmas 1 and 2.

Corollary 2. *If* $S \in \mathrm{C}\left(\Box^{l}\right)$*, then there is no β-void derivation loop in any V(S).*

Theorem 1. *If* $S \in \mathrm{C}\left(\backslash\Box^{l}\right)$*, then* $\vdash$ *iff* $\vdash T(S)$.

Proof. The Theorem is proved by induction on the derivation height *h*. If $h = 0$, then both *S* and $T(S)$ are axioms. Let $h > 0$. If *S* is not of type $(\Rightarrow O\Box\Delta, \Box\Lambda)$, then neither *S* nor $T(S)$ can be in a derivation loop, according to Lemma 2.Thetheoremisproved traditionally by the inductive hypothesis in this case. Let $S:(\Rightarrow O\Box\varnothing, \Box\Delta, \Box\Lambda)$ be the base sequent of a derivation loop $[S - S_1]$ and the derivation of *S* start at the bottom by

$$\frac{|\Rightarrow \varnothing, \Box\Delta, O\Box\Lambda \| \Rightarrow O\Box\varnothing, \Box\Lambda|}{S: \Rightarrow \Box\varnothing, \Box\Delta, O\Box}\left(\Rightarrow\Box\right)$$

We have $S = S_1$, by Lemma 2.Lemma 1 implies that any sequent S_L that is the left premise of any application of $(\Rightarrow\Box)$ the conclusion of which is in $[S - S_1]$ cannot be in $[S - S_1]$. Based on this fact, we apply the inductive hypothesis to each such S_L in the considered derivation tree. $S = S_1$ implies $T(S) = T(S_1)$. Hence $[T(S) - T(S_1)]$ is a derivation loop. We get $\vdash T(S)$. The direction from $\vdash T(S)$ to $\vdash S$ is considered in the same way.

Let $S:(\Rightarrow O\Box\Lambda)$ be the base sequent of a derivation loop $[S - S_1]$ and the derivation of *S* start at the bottom by

$$\frac{|\Rightarrow \Box\Lambda|}{S: \Rightarrow O\Box\Lambda}\left(O\right)$$

Only $(\Rightarrow\Box)$ can be backward applied to the premise and we consider this case in the same way as the previous one.

Example 1. If $S \in \mathrm{C}\left(\backslash\Box^{l}\right)$, then $\vdash S$ iff $\vdash T(S)$, according to Theorem1. Hence we can use $T(S)$ so that to check if *S* is derivable. The reduction of S to $T(S)$ may substantially reduce backward proof-search because the number of 'O' in $T(S)$ is diminished in comparison with S. For example, let $S = (p, op, oop \Rightarrow \Box p)$. The backward proof-search of S is as follows:

$$\frac{s_1 : \Rightarrow p \qquad \Rightarrow o\Box p}{\Rightarrow \Box p} \qquad \frac{p \Rightarrow p \qquad p \Rightarrow o\Box p}{p \Rightarrow \Box p} \qquad \frac{p, op \Rightarrow p \qquad p, op \Rightarrow o\Box p}{p, op \Rightarrow \Box p} \frac{p, op, oop \Rightarrow p \qquad p, op, oop \Rightarrow o\Box p}{p, op, oop \Rightarrow \Box p}$$

We get $\nvdash S$ because no rule is backward applicable to S_1. Using $t(s) = (p, q, q_1 \Rightarrow \Box p)$ instead of S, the same result is achieved as follows:

$$\frac{S_1 : \Rightarrow p \qquad \Rightarrow o\Box p}{\Rightarrow \Box p} \qquad \frac{p, q, q_1 \Rightarrow p \qquad p, q, q_1 \Rightarrow o\Box p}{p, q, q_1 \Rightarrow \Box p}$$

We obtain $\nvdash T(S)$ because no rule is backward applicable to S_1. Hence $\nvdash S$, according to Theorem 1.Wehave 3 rule applications in the backward proof-search of $T(S)$ versus 8 rule applications in the backward proof-search of S.

Example 2. Let S be any non-axiom sequent of the type $\Xi \Rightarrow \Xi_1$, where each formula in Ξ and Ξ_1 is of the type$O^n p(n \geq 0)$. It follows from Theorem 1 that S is not derivable because $T(S)$ is an atomic non-axiom sequent, i.e., no further backward proof-search is needed.

Sequents of the type $\Xi, \Box T \Rightarrow \Box \ \Delta, \Xi_1$, where only propositional symbols and 'O' occur in (Ξ, Ξ_1), are called canonical. Let $O^n p(n \geq 0)$ be a member of a canonical sequent $\Xi, \Box T \Rightarrow \Box \Delta, \Xi_1$. The formula $O^n p$ is called redundant in the sequent if p does not occur in $(T, \ \Delta)$.

A backward proof-search tree V is called a *one-step reduction tree*, iff 1) there is at most one application of (O) on each branch and 2) each non-atomic and non-axiom leaf of V is a premise of (O). It is easy to see that every leaf of any one-step reduction tree is an axiom or a canonical sequent. We have that proof-search of an arbitrary sequent can be reduced to proof-search of canonical sequents.

Theorem 2. *If S' is obtained from a non-axiom canonical sequent S by dropping redundant formulas, then $\vdash S$ iff $\vdash S'$.*

Proof. If $\vdash S'$, then $\vdash S$, using the rule of weakening and Theorem 6.5 in [1]. Let $\vdash S$. It follows that $\vdash S$ because $\mathbf{G_L T}$ is sound, Theorem 4.4 in [1]. It is easy to see that this fact implies $\vDash S'$. Hence $\vdash S'$ because $\mathbf{G_L T}$ is complete, Theorem 5.4 in [1].

Dropping redundant members allows us to simplify sequents and reduce backward proof-search in some cases. Let us consider, e. g., the sequent $S = (\Box p \Rightarrow oop)$. The formula ooq is redundant in S. We drop it and obtain $\Box p \Rightarrow$. This sequent is equivalent to S by Theorem 2.

5 Concluding remarks

In the present paper, correct sequents have been defined and it has been shown that any sequent S is equivalent to some correct sequent. The sequent $T(S)$ for any correct sequent S and the class of sequents

$C\left(\setminus\Box^{l}\right)$ have been defined. We have proved that if S belongs to this class, then it is derivable if and only if $T(S)$ is derivable, Theorem 1.Thatenablestocheckderivabilityof S by means of a simpler sequent $T(S)$, which substantially reduces backward proof-search in cases when S has many occurrences of 'O'. Also, redundant formulas have been defined and it has been proved that dropping redundant formulas from a canonical sequent has no impact on its derivability, Theorem 2. Theoptimizationsofbackwardproof-searchpresentedin the present paper concern only partial cases. The general case could be a topic for further investigation.

References

[1] R. Alonderis, R. Pliuškevičius, A. Pliuškevičiene˙, H. Giedra. Loop-type sequent calculi for temporal logic. *J. Autom. Reason*, **64**(8):1663–1684, 2020.

[2] C. Baier, J.P. Katoen. *Principles of Model Checking*. The MIT Press Cambridge, Massachusetts, London, England, 2008.

[3] K. Brünnler, M. Lange. Cut-free sequent systems for temporal logic. *J. Log. Algebr. Program.,* **76**(2):216–225, 2008.

[4] J. Gaintzarain, M. Hermo, P. Lucio, M. Navarro, F. Orejas. A cut-free and invariantfree sequent calculus for pltl. In J. Duparc, T.A. Henzinger(Eds.), *Computer Science Logic. CSL 2007*, volume 4646 of *Lect. Notes Comput. Sci.*, pp. 481–495, 2007. https://doi.org/10.1007/978-3-540-74915-8_36.

[5] M. Huth, M. Ryan. *Logic in Computer Science: Modelling and Reasoning about Systems*. Cambridge University Press, 2012.

[6] I. Kokkinis, T. Studer. Cyclic proofs for linear temporal logic. In D. Probst, P. Schuster(Eds.), *Concepts of Proof in Mathematics, Philosophy, and Computer Science*, pp. 171–192. De Gruyter, Berlin, Boston, 2016.

[7] B. Paech. Gentzen-systems for propositional temporal logics. In E. Börger, H.K. Büning, M.M. Richter (Eds.), *CSL '88. CSL 1988*, volume 385 of *Lect. Notes Comput. Sci.*, pp. 240–253. Springer, Berlin, Heidelberg, 1988. https://doi.org/10.1007/BFb0026305.

[8] R. Pliuškevičius. On saturated calculi for linear temporal logic. In A.M. Borzyszkowski, S. Sokołowski(Eds.), *Mathematical Foundations of Computer Science 1993. MFCS 1993*, volume 711 of *Lect. Notes Comput. Sci.*, pp. 640–649. Springer, Berlin, Heidelberg, 1993. https://doi.org/10.1007/3-540-57182-5_55.

[9] S. Schwendimann. A new one-pass tableau calculus for PLTL. In *Automated Reasoning with Analytic Tableaux and Related Methods. TABLEAUX 1998*, volume 1397 of *Lect. Notes Comput. Sci.*, pp. 277–291. Springer, Berlin, Heidelberg, 1998. https://doi.org/10.1007/3-540-69778-0_28.

[10] G. Sundholm. A completeness proof for an infinitary tense-logic. *Theoria*, **43**:47–51, 1977. https://doi.org/10.1111/j.1755-2567.1977.tb00778.x.

[11] M. Valiev. On temporal logic of von Vright. In *Soviet-Finland Colloquim on Logic, Moscow*, pp. 7–11, 1979 (in Russian).

[12] M.K. Valiev. Decision complexity of variants of propositional dynamic logic. In P. Dembinski(Ed.), *Mathematical Foundations of Computer Science 1980. MFCS 1980*, volume 88 of *Lect. Notes Comput. Sci.*, pp. 656–664, Berlin, 1980. Springer-Verlag.

[13] P. Wolper. The tableau method for temporal logic: an overview. *Log. Anal.*, **28**:119–136, 1985.