Lietuvos matematikos rinkinys, Vol. 60, 2019
Proc. of the Lithuanian Mathematical Society, Ser. A
https://doi.org/10.15388/LMR.A.2019.14953

ISSN 0132-2818
eISSN 2335-898X
pages 1–6

A derivation-loop method for temporal logic

Romas Alonderis[a], **Haroldas Giedra**[b]

[a] *Institute of Data Science and Digital Technologies, Vilnius University*
Akademijos st. 4, LT-08412 Vilnius, Lithuania
[b] *Institute of Computer Science, Vilnius University*
Didlaukio st. 47, LT-08303 Vilnius, Lithuania
E-mail: romas.alonderis@mif.vu.lt, haroldas.giedra@mif.vu.lt

Abstract. Various types of calculi (Hilbert, Gentzen sequent, resolution calculi, tableaux) for propositional linear temporal logic (**PLTL**) have been considered in the literature. Cut-free Gentzen-type sequent calculi are convenient tools for backward proof-search search of formulas and sequents.

In this paper we present a cut-free Gentzen type sequent calculus for **PLTL** with the operator "until". We show that the calculus is sound and complete for the considered logic.

Keywords: temporal logics, sequent calculi, derivation loops.

1 Introduction

Propositional linear temporal logic (**PLTL**) is used in computer science for specification and verification of programs [2, 6]. Sequent calculi are used for analysis and effective check of formula validity by performing backward proof-search.

In this paper, we consider the loop-type sequent calculus (**LTSC**) for **PLTL** with temporal operators "next" and "until".

We would like to mention the following sequent calculi for temporal discrete tense temporal logics considered in the literature:

1. Infinitary sequent calculi containing ω-type induction rule. Finitization of the ω-type induction rule is considered in [3].
2. The calculi with invariant-like rule. There are some works in which some constructive methods for finding invariant formulas are considered [13, 14].
3. In 1993, the so called saturation method was proposed in [9]. The saturated calculus contains (instead of induction-like rules) some non-logical axioms indicating the saturation of proof-search process. Saturation intuitively corresponds to a certain type of regularity in proof-search.
4. The loop-type sequent calculi firstly were considered by Wolper in 1985 [16]. The loop-type sequent calculi (as saturated calculi) for temporal, mutual belief and dynamic logics were considered in [10].

5. A cut-free and invariant-free sequent calculus for **PLTL** is presented in [5]. This calculus has the new operator "unless", and do not retain the sub-formula property.

To our knowledge, the loop-type sequent calculus introduced in the present paper has not been considered in the literature before.

The present paper is organized as follows. In Section 2, we recall the syntax and semantics of **PLTL**. The calculus **LTSC** is introduced in Section 3. In this section, we present also the definition of derivation loops and prove some propositions concerning them. The soundness and completeness of **LTSC** with respect to **PLTL** is proved in Section 4. Some concluding remarks are in Section 5.

2 Syntax and semantics

2.1 Syntax

The language of considered **PLTL** contains the constant $\top$ (true); a set $\mathbb{P}$ of propositional symbols $\{p, p_1, p_2, \ldots, q, q_1, q_2, \ldots\}$; the logical operators $\neg, \vee, \wedge, \supset$, temporal operators $\mathcal{U}$ ("until") and $\circ$ ("next"). The language does not contain the temporal operators $\diamond$ ("sometimes") and $\Box$ ("always"), assuming that $\diamond\phi = \top\mathcal{U}\phi$ and $\Box\phi = \neg\diamond\neg\phi$.

Propositional symbols and $\top$ are called atomic formulas. The formulas ϕ of **PLTL** are inductively defined as follows:

$$\phi ::= \top \mid\ p \mid \neg\phi \mid \phi \vee \psi \mid \phi \wedge \psi \mid \phi \supset \psi \mid \circ\phi \mid \phi\mathcal{U}\psi.$$

We use the Greek letters ϕ and ψ, possibly with subscripts, to denote arbitrary formulas.

2.2 Semantics

We assume that time is linear, discrete, and ranges over the set of natural numbers. The formula $\circ\phi$ intuitively means "ϕ is true at the next point of time"; the formula $\phi\mathcal{U}\psi$ intuitively means "either ψ is true now or ϕ is true now and in all future time points until the one at which ψ is true".

An interpretation $\mathcal{M} = (T, I)$ for propositional linear tense logic consists of the set $T = \{t_i : i \geqslant 0\}$, where $t_i < t_j$, if $i < j$, and the function $I : T \mapsto 2^{\mathbb{P}}$, where $2^{\mathbb{P}}$ is the set of subsets of $\mathbb{P}$. The semantics of **PLTL** formulas is provided by the satisfaction relation $\models$:

$\mathcal{M}, t_i \models \top$;

$\mathcal{M}, t_i \models p$, iff $p \in I(t_i)$;

$\mathcal{M}, t_i \models \circ\phi$, iff $\mathcal{M}, t_{i+1} \models \phi$;

$\mathcal{M}, t_i \models \phi\mathcal{U}\psi$, iff there is $m \geq i$ such that $\mathcal{M}, t_m \models \psi$ and for all $i \leq j < m$, $\mathcal{M}, t_j \models \phi$.

($\models$ for the propositional operators is defined in the usual way.)

An interpretation $\mathcal{M}$ is a model for a formula ϕ, iff $\mathcal{M}, t_0 \models \phi$. For an arbitrary sequent $S = (\phi_1, \ldots, \phi_m \Rightarrow \psi_1, \ldots, \psi_n)$, we say that $\mathcal{M}, t_i \models S$, iff there is $\iota \in \{1, 2, \ldots, m\}$ such that $\mathcal{M}, t_i \not\models \phi_\iota$, or there is $\iota \in \{1, 2, \ldots, n\}$ such that $\mathcal{M}, t_i \models \psi_\iota$.

A formula ϕ (sequent S) is called valid, $\models \phi$ ($\models S$) in notation, iff every **PLTL** interpretation is a model for ϕ (S). For example, it is true that $\models (p \wedge \circ q) \supset (p\mathcal{U}q)$.

3 Sequent calculus LTSC for PLTL

The sequent calculus **LTSC** is defined by the following postulates:

1. Logical axioms: $\Gamma, \phi \Rightarrow \Delta, \phi$ and $\Gamma \Rightarrow \Delta, \top$.
2. Propositional rules are the same as of the calculus $\mathbf{LK}_0$ in [1].
3. Temporal rules:

$$\frac{\Gamma \Rightarrow \Delta}{\Sigma, \circ\Gamma \Rightarrow \circ\Delta, \Sigma'}\ (\circ), \qquad \frac{\psi, \Gamma \Rightarrow \Delta \quad \phi, \circ(\phi\mathcal{U}\psi), \Gamma \Rightarrow \Delta}{\phi\mathcal{U}\psi, \Gamma \Rightarrow \Delta}\ (\mathcal{U} \Rightarrow),$$

$$\frac{\Gamma \Rightarrow \Delta, \phi, \psi \quad \Gamma \Rightarrow \Delta, \psi, \circ(\phi\mathcal{U}\psi)}{\Gamma \Rightarrow \Delta, \phi\mathcal{U}\psi}\ (\Rightarrow \mathcal{U}).$$

Here: $\Gamma, \Delta, \Sigma, \Sigma'$ denote finite, possibly empty, multisets of formulas, where $\Sigma \cup \Sigma'$ consists of atomic formulas; the conclusion is not an axiom and $\Gamma \cup \Delta \neq \emptyset$ in $(\circ)$.

Given a sequent S, a **LTSC** proof-search tree with the sequent S at the root is constructed in usual way by subsequently applying backwards the **LTSC** 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 ($S' \succeq S$ in notation), iff S' can be inferred from S by the structural rule of weakening. For example, the sequent $\Gamma, \Pi \Rightarrow \Delta, \Lambda$ subsumes $\Gamma \Rightarrow \Delta$.

Let $\Gamma_1, \Pi_1, \Delta_1, \Lambda_1$ be obtained from $\Gamma, \Pi, \Delta, \Lambda$, respectively, by dropping atomic members. If $\{\Gamma_1\} = \{\Gamma_1, \Pi_1\}$ and $\{\Delta_1\} = \{\Delta_1, \Lambda_1\}$, then we say that the sequent $\Gamma, \Pi \Rightarrow \Delta, \Lambda$ *strongly subsumes* the sequent $\Gamma \Rightarrow \Delta$, denoted by $(\Gamma, \Pi \Rightarrow \Delta, \Lambda) \sqsupseteq (\Gamma \Rightarrow \Delta)$.

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, 2) $S' \succeq S$ ($S' \sqsupseteq S$), and 3) there is no other sequent in p, except S, which (strongly) subsumes 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 $\lambda(S) \leqslant \lambda(S')$.

The expression $(\phi\mathcal{U}\psi \Rightarrow)$ denotes an application of $(\mathcal{U} \Rightarrow)$ with the principal formula $\phi\mathcal{U}\psi$.

Definition 2. A (strong) derivation loop $[S - S']$ is called a (strong) derivation loop with the eventuality formula $\phi\mathcal{U}\psi$, iff: 1) $S = \theta(\phi\mathcal{U}\psi), \Gamma \Rightarrow \Delta$, 2) $S' = \Pi, \theta(\phi\mathcal{U}\psi), \Gamma \Rightarrow \Delta, \Lambda$, where $\theta = \emptyset \mid \circ$, and 3) $[S - S']$ contains the right premise of $(\phi\mathcal{U}\psi \Rightarrow)$, and does not contain the left premise of $(\phi\mathcal{U}\psi \Rightarrow)$.

Proposition 1. *In any derivation loop with an eventuality formula $\phi\mathcal{U}\psi$, there is an application of $(\phi\mathcal{U}\psi \Rightarrow)$ between any two applications of $(\circ)$.*

Proof. The proof follows from the fact that the succedent of the base of the loop has the member $\phi\mathcal{U}\psi$ or $\circ(\phi\mathcal{U}\psi)$, from item 3 of Definition 2, and from the shape of the rule $(\circ)$. □

Definition 3. Any maximal connected graph Υ in a backward proof-search tree V such that each edge of Υ is in some derivation loop with an eventuality formula, is called a connected component.

It is said that a connected component Υ has a common eventuality formula, iff all derivation loops in Υ have a common eventuality formula $\phi\mathcal{U}\psi$. Such a formula is called the eventuality formula of Υ.

A connected component is called strong, iff all derivation loops in it are strong.

Definition 4. Let Υ be a connected component. The path between sequents S_1 and S_2 in Υ is denoted by $p(S_1, S_2)$. Let $[S_1 - S'_1], \ldots, [S_k - S'_k]$ be all the derivation loops with eventuality formulas in Υ. A $(S_i \circ S'_i)$ loop structure of Υ $(i \in \{1, \ldots, k\})$ is obtained by merging Υ into a single path containing all the edges of Υ. It is assumed that any two adjacent $p(S_l, S'_m)$ in the loop structure are connected with an application of an unnamed rule.

Definition 5. A sequent S is called axiomatically derivable in **LTSC**, iff there exists a backward proof-search tree $V(S)$ such that each leaf of $V(S)$ is an axiom.

Definition 6. A sequent S is called derivable in **LTSC** ($\vdash S$ in notation), iff it is axiomatically derivable or there exists a backward proof-search tree $V(S)$ such that:

> 1) each leaf of $V(S)$ is an axiom or a terminal sequent of a derivation loop with some eventuality formula and 2) each connected component in $V(S)$ has a common eventuality formula.

Such a tree $V(S)$ is called a derivation of S or a derivation tree. We use the notation $\vdash^V S$ to say that $V(S)$ is a derivation of S.

A formula ϕ is called derivable in **LTSC**, iff $\vdash \Rightarrow \phi$.

4 Soundness and completeness of LTSC

Lemma 1. *Let*

$$\frac{S_1 \quad (S_2)}{S}\ (r)$$

*be an arbitrary instance of an application of any **LTSC** derivation rule, except $(\circ)$. If $\mathcal{M}, t_i \not\models S$, then there is $j \in \{1, 2\}$ such that $\mathcal{M}, t_i \not\models S_j$.*

Proof. The proof of the lemma is straightforward. □

Lemma 2. *For any instance of an application of $(\circ)$ with the conclusion S and premise S', if $\mathcal{M}, t_i \not\models S$, then $\mathcal{M}, t_{i+1} \not\models S'$.*

Proof. The proof of the lemma is straightforward. □

Theorem 1. *The calculus* ***LTSC*** *is sound: if* $\vdash S$*, then* $\models S$*, where* S *is an arbitrary sequent.*

Proof. The theorem is proved by induction on the number d of the leaves of derivation tree $V(S)$ that are terminal sequents of derivation loops, making use of Lemmas 1, 2, and Proposition 1. □

Theorem 2. *The calculus* ***LTSC*** *is complete: if* $\models S$*, then* $\vdash S$*, where* S *is an arbitrary sequent.*

Proof. We prove that $\nvdash S$ implies $\not\models S$. □

5 Concluding remarks

In the present paper, we have introduced and considered the Gentzen-type sequent calculus **LTSC**. This calculus is sound (Theorem 1) and complete (Theorem 2) for the considered logic **PLTL**. Hence an arbitrary sequent is derivable in **LTSC** if and only if it is valid in **PLTL**. Let us consider, for example, the sequent

$$S:\ p, \top\mathcal{U}\neg p \Rightarrow \top\mathcal{U}\neg(p \supset \circ p).$$

It is derived in **LTSC** as follows:

$$\dfrac{S_1 \qquad \dfrac{S_2 \qquad \dfrac{\dfrac{S_3 \qquad \dfrac{S':\ p, \top\mathcal{U}\neg p \Rightarrow \top\mathcal{U}\neg(p \supset \circ p)}{\circ p, p, \circ(\top\mathcal{U}\neg p) \Rightarrow \circ\big(\top\mathcal{U}\neg(p \supset \circ p)\big)}\,(\circ)}{p \supset \circ p, p, \circ(\top\mathcal{U}\neg p) \Rightarrow \circ\big(\top\mathcal{U}\neg(p \supset \circ p)\big)}\,(\supset\Rightarrow)}{p, \circ(\top\mathcal{U}\neg p) \Rightarrow \neg(p \supset \circ p), \circ\big(\top\mathcal{U}\neg(p \supset \circ p)\big)}\,(\Rightarrow\neg)}{p, \circ(\top\mathcal{U}\neg p) \Rightarrow \top\mathcal{U}\neg(p \supset \circ p)}\,(\Rightarrow\mathcal{U})}{S:\ p, \top\mathcal{U}\neg p \Rightarrow \top\mathcal{U}\neg(p \supset \circ p)}\,(\mathcal{U}\Rightarrow)$$

(Here $S_1 : p, \neg p \Rightarrow \top\mathcal{U}\neg(p \supset \circ p)$ is axiomatically derivable by backward applying $(\neg \Rightarrow)$ to it; $S_2 : p, \circ(\top\mathcal{U}\neg p) \Rightarrow \top, \neg(p \supset \circ p)$ is an axiom; $S_3 : p, \circ(\top\mathcal{U}\neg p) \Rightarrow \circ\big(\top\mathcal{U}\neg(p \supset \circ p)\big), p$ is an axiom; $[S - S']$ is the strong derivation loop with the eventuality formula $\top\mathcal{U}\neg p$.) We obtain that the considered sequent S is valid in **PLTL**.

References

[1] R. Alonderis. sequent calculi for propositional star free likelihood logic. *Lit. Math. J.*, **45**(1):1–15, 2005.

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

[3] K. Brünnler and D. Steiner. *Finitization for Propositional Linear Time Logic.* Unpublished, available on the Web, 2006

[4] M. Fisher, C. Dixon and M. Peim. Clausal temporal resolution. *ACM Trans. Comp. Logic*, **2**(1):12–56, 2001.

[5] J. Gaintzarain, M. Hermo, P. Lucio, M. Navarro and F. Orejas. A cut-free and invariant-free sequent calculus for PLTL. *Lect. Not. Comp. Sci.*, **4646**:481–495, 2007.

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

[7] N. Nide and S. Takata. Deduction systems for BDI logic using sequent calculus. In *Proc. AAMAS02*, pp. 928–935, 2002.

[8] B. Paech. Gentzen-systems for propositional temporal logics. *Lect. Not. Comp. Sci.*, **385**:240–253, 1988.

[9] R. Pliuškevičius. On saturated calculi for linear temporal logic. *Lect. Not. Comp. Sci.*, **711**:640–649, 1993.

[10] R. Pliuškevičius and A. Pliuškevičienė. Decision procedure for a fragment of mutual belief logic with quantified agent variables. *Lect. Not. Art. Int.*, **3900**:112–128, 2006.

[11] S. Schwendimann. A new one-pass tableau calculus for PLTL, *Lect. Not. Comp. Sci.*, **1397/1998**:277–291, 1998.

[12] G. Sundholm. A completeness proof for an infinitary tense-logic. *Theoria*, **43**:47–51, 1977.

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

[14] M.K. Valiev. Decision complexity of variants of propositional dynamic logic. In P. Dembinski (Ed.), *Proc. MFCS'80*, *LNCS*, Vol. 88. Springer-Verlag, Berlin, pp. 656–664, 1980.

[15] P. Wolper. Temporal logic can be more expressive. *Inf. Control*, **56**:72–99, 1983.

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

REZIUMĖ

Įrodymo ciklų metodas laiko logikai

R. Alonderis, H. Giedra

Literatūroje yra nagrinėjamos įvairios propozicinės tiesinės laiko logikos dedukcinės sistemos, tokios kaip: Hilberto tipo skaičiavimai, Gentzeno tipo sekvenciniai skaičiavimai, rezoliucijų ir lentelių metodai. Pjūvio taisyklės neturintys Gentzeno tipo sekvenciniai skaičiavimai leidžia efektyviai atlikti atgalinę formulių ir sekvencijų įrodymo paiešką, siekiant patikrinti jų tapatų teisingumą. Šiame straipsnyje pateikiamas pjūvio taisyklės neturintis Gentzeno tipo sekvencinis skaičiavimas propozicinei tiesinei laiko logikai su operatoriumi "kol". Parodoma, kad šis skaičiavimas yra korektiškas ir pilnas nagrinėjamos logikos atžvilgiu.

Raktiniai žodžiai: laiko logika, sekvenciniai skaičiavimai, įrodymo ciklai.