Lietuvos matematikos rinkinys. LMD darbai ISSN 0132-2818
52 tomas, 2011, 231–236 www.mii.lt/LMR/

Cut, invariant rule, and loop-check free sequent calculus for PLTL

Romas Alonderis, Regimantas Pliuškevičius

Institute of Mathematics and Informatics, Vilnius University
Akademijos 4, Vilnius 2600
E-mail: romas.alonderis@mii.vu.lt; regimantas.pliuskevicius@mii.vu.lt

Abstract. In this paper, some loop-check free saturation-like decision procedure is proposed for propositional linear temporal logic (PLTL) with temporal operators "next" and "always". This saturation procedure terminates when special type sequents are obtained. Properties of PLTL allows us to construct backtracking-free decision procedure without histories and loop-check.

Keywords: decision procedure, sequent calculus, loop-check, temporal logics, termination.

1 Introduction

It is well known that check of termination of a decision procedure plays a crucial role in constructing derivations. Along with non-classical logics without induction like tools, there are very important for computer science and artificial intelligence non-classical logics containing induction like tools, e.g., temporal, dynamic, common knowledge logics and their various modifications and combinations. Usually, these induction like tools are realized using loop axioms. Determination of these loop-type axioms involves creating new loops ("good loops" in opposite to "bad loops") and the new loop checking along with ordinary non-induction-type loop checking.

Determination of "good loops" are closely related to finding the so-called invariant formulas. Some method for finding invariant formulas for Horn-like PLTL is proposed in Pliuškevičius [9]. A specified loop-check procedure for the mutually belief logic is proposed in Pliuškevičius et al. [10]. Using histories, some cut-free and invariant-free calculi for PLTL are proposed by Gaintzarain et al. [6] and by Brünnler et al. [2]. Some efficient loop-check methods for various temporal logics was proposed using Fisher's [4] resolution method, see also Fisher et al. [5].

In the present paper, a propositional linear temporal logic with temporal operators $\bigcirc$ ("next") and $\Box$ ("always") is considered. It is known that combination of these temporal operators requires to use induction-like tools. To determine such tools, some simple saturation procedure is proposed. This procedure allows us to eliminate (not using histories) the search of "bad" and "good" loops at all. Instead of these loops, the proposed procedure generates some terminal sequents of special shape. The constructed procedure is loop-check-free and backtracking-free.

The paper is organized as follows. In Section 2, initial proof-search procedures for PLTL are described. In Section 3, a proposed loop-check-free backward proof-search procedure is described. Foundation of the procedure is proved in Section 4.

2 Initial proof procedures for PLTL

The language of considered PLTL contains a set of propositional symbols $P, P_1, P_2, \ldots,$ $Q, Q_1, Q_2, \ldots$; the set of logical connectives $\supset, \wedge, \vee, \neg$; temporal operators $\Box$ ("always") and $\bigcirc$ ("next"). The language does not contain the temporal operator $\diamond$ ("sometimes"), assuming that $\diamond A = \neg\Box\neg A$. We assume that time is linear, discrete, and ranges over the set of natural numbers.

Formulas in the considered calculi are constructed in the traditional way from propositional symbols, using the logical connectives and temporal operators. The formula $\bigcirc A$ means "A is true at the next moment of time"; the formula $\Box A$ means "A is true now and in all moments of time in the future".

We consider sequents, i.e., formal expressions $\Gamma \to \Delta$, where Γ and Δ are finite multisets of formulas.

Formulas and sequents without temporal operators are called logical. A sequent which consists of propositional symbols is called elementary.

As far as sequent (tableaux) calculi for logics with induction-like axioms are concerned, it is known that most theoretical investigations are based on three types of proof-search procedures, namely: (a) procedures containing unrestricted or restricted ω-type rules (b) procedures containing analytical cut-type rules, and (c) procedures containing loop-type axioms (or "good loops").

For the considered PLTL logic there are known proof procedures of the following types:

(1) Proof procedures based on infinitary calculus G_ωPLTL defined by the following postulates:

Axioms: $\Gamma, A \to \Delta, A$.

Logical rules: standard, see. e.g., [1].

Temporal rules: $(\Gamma \to \Delta)/(\Pi, \bigcirc\Gamma \to \Theta, \bigcirc\Delta)(\bigcirc)$; $(A, \bigcirc\Box A, \Gamma \to \Delta)/(\Box A, \Gamma \to \Delta)(\Box \to)$, and $(\Gamma \to \Delta, A; \Gamma \to \Delta, \bigcirc A; \ldots; \Gamma \to \Delta, \overbrace{\bigcirc, \ldots, \bigcirc}^{n} A; \ldots)/(\Gamma \to \Delta, \Box A)(\to \Box_\omega)$.

It follows from Sundholm [12] that G_ωPLTL is sound and complete. There are some interesting works concerning finitization of ω-type rule $(\to \Box_\omega)$ (see, e.g., Brünnler and Steiner [3].

(2) Proof procedures based on calculus $\mathrm{G_I}$PLTL with analytical cut (or invariant-like) rule. The calculus is obtained from G_ωPLTL by replacing the ω-type rule $(\to \Box_\omega)$ by the following analytical cut-type rule: $(\Gamma \to \Delta, I; I \to \bigcirc I; I \to A)/(\Gamma \to \Delta, \Box A)(\to \Box_{\mathrm{I}})$.

(3) Proof procedures containing loop-type axioms (or good loops). The procedure $\mathrm{G_L}$PLTL is obtained from the calculus $\mathrm{G_I}$PLTL by

(a) replacing rule $(\to \Box_{\mathrm{I}})$ by the rule $(\Gamma \to \Delta, A; \Gamma \to \Delta, \bigcirc\Box A)/(\Gamma \to \Delta, \Box A)(\to \Box)$;

(b) adding loop-type axioms as follows: a sequent S is a loop-type axiom iff (1) S is above a sequent S' on a branch of a derivation tree, (2) S' is such that S subsumes S' ($S \succcurlyeq S'$ in notation), i.e., S' can be obtained from S by using structural rules of weakening and contraction only (S and S' coincide in special case), and (3) there is the right premise of $(\to \Box)$ between S and S'. See also Nide and Takata [8].

It follows from Gough [7], Wolper [13], Schwendimann [11] that $\mathrm{G_L}$PLTL is sound and complete.

A sequent S is a primary one iff $S = \Sigma_1, \bigcirc\Gamma_1 \to \bigcirc\Gamma_2, \Sigma_2$, where Σ_i $(i \in \{1,2\})$ is empty or consists of propositional symbols; $\bigcirc\Gamma_i$ $(i \in \{1,2\})$ is empty or consists of formulas of the shape $\bigcirc A$, A arbitrary.

Let $\mathrm{G'_L}$PLTL be the procedure obtained from $\mathrm{G_L}$PLTL by replacing the rule $(\bigcirc)$ by the following one: $(\Pi \to \Delta)/(\Sigma_1, \bigcirc\Pi \to \Sigma_2, \bigcirc\Delta)$ $(\bigcirc')$, where $\Sigma_1 \cap \Sigma_2 = \emptyset$. It is easy to see that rule $(\bigcirc')$ is invertible and $\mathrm{G_L}$PLTL and $\mathrm{G'_L}$PLTL are equivalent.

3 Loop-check-free procedure

In this section, loop-check-free proof procedure for the considered PLTL is constructed.

We introduce operation $*$ which is applied to any formula and defined as follows:

1. $(P)^* = P^*,\ (P^*)^* = P^*,$ 2. $(A \odot B)^* = A^* \odot B^*,\ \odot \in \{\supset, \vee, \wedge\},$

3. $(\sigma A)^* = \sigma A^*,\ \sigma \in \{\neg, \bigcirc\},$ 4. $(\Box A)^* = \begin{cases} \Box A^*, & \text{if } A \neq A^*, \\ \Box^* A^*, & \text{if } A = A^*. \end{cases}$

A marked formula of the shape A^* is called $*$-marked.

We also introduce another operation $+$ which is applied to any $*$-marked formula and defined as follows:

1. $(P^*)^+ = P^+,\ (P^+)^+ = P^+,$ 2. $(A \odot B)^+ = A^+ \odot B^+,\ \odot \in \{\supset, \vee, \wedge\},$

3. $(\sigma A)^+ = \sigma A^+,\ \sigma \in \{\neg, \bigcirc\},$ 4. $(\Box A)^+ = \Box^+ A^+.$

A marked formula of the shape A^+ is called $+$-marked.

A sequent S is δ-reduced $(\delta \in \{*, +\})$ if

$$S = \bigcirc^{k_1}\Pi^\delta_{11}, \ldots, \bigcirc^{k_n}\Pi^\delta_{1n}, \Box^\delta\Gamma^\delta \to \bigcirc^{l_1}\Pi^\delta_{21}, \ldots, \bigcirc^{l_m}\Pi^\delta_{2m}, \Box^\delta\Delta^\delta,$$

where any of $\bigcirc^{k_i}\Pi^\delta_{1i}$ $(1 \leqslant i \leqslant n)$ and $\bigcirc^{l_j}\Pi^\delta_{2j}$ $(1 \leqslant j \leqslant m)$ is empty or consists of δ-marked formulas of the shape $\overbrace{\bigcirc\ldots\bigcirc}^{k} A^\delta$ ($k \geqslant 0$, A is arbitrary if $k > 0$ and A is a propositional symbol if $k = 0$) such that these formulas are sub-formulas of formulas in $\Box^\delta\Gamma^\delta$. The formulas in $\Box^\delta\Gamma^\delta$ are called essential.

A δ-reduced sequent is proper (improper), if $\Box^\delta\Delta^\delta \neq \emptyset$ ($\Box^\delta\Delta^\delta = \emptyset$, correspondingly). An improper sequent which is not a logical axiom is simple final (s-final in short) one.

Let us introduce δ-marked $(\delta \in \{*, +\})$ modal rules, which allow us (along with the rule $(\bigcirc)$ and logical rules) to generate in backward way δ-reduced sequents.

$(A, \bigcirc(\Box A)^*, \Gamma \to \Delta)/(\Box A, \Gamma \to \Delta)(\Box^* \to)$; $(\Gamma \to \Delta, A;\ \Gamma \to \Delta, \bigcirc(\Box A)^*)/(\Gamma \to \Delta, \Box A)(\to \Box^*)$, where conclusions of these $*$-marked rules are not $*$-reduced.

$(A, \bigcirc\Box^+ A^+, \Gamma \to \Delta)/(\Box A, \Gamma \to \Delta)(\Box^+ \to)$; $(\Gamma \to \Delta, A; \Gamma \to \Delta, \bigcirc\Box^+ A^+)/(\Gamma \to \Delta, \Box A)(\to \Box^+)$, where conclusions of these +-marked rules are not +-reduced.

$$\frac{\bigcirc^{k_1-1}\Pi_{11}^+, \bigcirc^{l_1-1}\Pi_{12}^+, \Box^+\Gamma^+ \to \bigcirc^{k_2-1}\Pi_{21}^+, \bigcirc^{l_2-1}\Pi_{22}^+, \Box^+\Delta^+}{\Sigma_1^\delta, \bigcirc^{k_1}\Pi_{11}^+, \bigcirc^{l_1}\Pi_{12}^*, \bigcirc\Box^+\Gamma^+ \to \Sigma_2^\delta, \bigcirc^{k_2}\Pi_{21}^+, \bigcirc^{l_2}\Pi_{22}^*, \bigcirc\Box^+\Delta^+}\ (\bigcirc)^+.$$

Here $k_1, l_1, k_2, l_2 > 0$, $\Delta \neq \emptyset$ consists of arbitrary formulas; $\Pi_{1,i}, \Pi_{2,j}$ $(i, j \in \{1,2\})$ are empty or consist of arbitrary formulas; Σ_i^δ $(i \in \{1,2\})$ is empty or consists of δ-marked propositional symbols $P_j^{\delta_j}$, where $\delta_j \in \{+, *\}$ and $\Sigma_1^\delta \cap \Sigma_2^\delta = \emptyset$, assuming that $P = P^* = P^+$.

The marked calculus G′TL is obtained from the calculus G'_{L}PLTL by (1) replacing the modal rules $(\Box \to)$ and $(\to \Box)$ by the marked rules $(\Box^\delta \to)$, $(\to \Box^\delta)$ $(\delta \in \{+, *\})$, and (2) adding the rule $(\bigcirc)^+$.

Definition 1. Let S be a sequent. A δ-transformation $(\delta \in \{*, +\})$ of S is called reduction of S to the sets $\{S_{11}, \ldots, S_{1n}\}$ $(n > 0)$ (so called improper set) and $\{S_{21}, \ldots, S_{2m}\}$ $(m \geqslant 0)$ (so called proper set), where S_{1i} $(1 \leqslant i \leqslant n)$ is an improper δ-reduced sequent, a logical axiom, or a non-axiom elementary sequent; S_{2j} $(1 \leqslant j \leqslant m)$ is a proper δ-reduced sequent. The sequents $S_{i,j}$ $(i \in \{1,2\})$ are leaves of a G′TL derivation tree with S at the root: starting with S, apply backward applicable marked modal rules $(\Box^\delta \to)$, $(\to \Box^\delta)$, $(\bigcirc)'$, $(\bigcirc)^+$, and logical rules until logical axioms, non-axiom elementary sequents, or δ-reduced sequents are obtained.

A +-transformation of a sequent S in G′TL is successful iff each member of the improper set is a logical axiom.

A proper $*$-reduced sequent S^* is i-final iff its +-transformation is successful. Such a successful +-transformation of S^* is a final one and is denoted by $R_f(S^*)$.

The loop-check-free calculus G*TL is obtained from G′TL by (1) removing loop-type axioms and (2) adding i-final sequents as non-logic axioms.

A G*TL derivation D of a given sequent S is constructed backwards as follows. $*$-transformation of S is performed first. If there is a sequent in the improper set which is not a logical axiom, then D is unsuccessful. Otherwise, D is successful if the proper set is empty. If it is not, then +-transformation is performed for each sequent in the proper set. If every such +-transformation is successful, then D is successful; otherwise, D is unsuccessful.

Lemma 1 [Repeating property of $*$-reduced sequents]. *Let $G^*TL \vdash^V S$ and S^* be any i-final sequent in V. Then there exists +-reduced sequent S^+ in $R_f(S^*)$ such that $S^+ \succeq S^*$ (assuming that formulas coincide if they differ only in marks, i.e., $F = F^* = F^+$).*

Let us consider termination of backward proof-search of a sequent S in procedure G*TL. The complexity of a sequent S (denoted $C(S)$) is defined as an ordered pair $\langle m^\delta(S), n^*(S)\rangle$ $(\delta \in \{+, *\})$, where $m^\delta(S)$ is the number of different occurrences of non-δ-marked $\Box$ in S; $n^*(S)$ is the number of different occurrences of $\Box^*$ in S.

Lemma 2. *Let $G^*TL \vdash^V S$ and S_1, S_2 be any primary sequents in V such that S_2 is above S_1 and there are no primary sequents between them, then $C(S_2) < C(S_1)$.*

Relying on the definition of G*TL, definition of derivability, Lemma 2, and using invertibility of the rules, we get loop-check-free P-SPACE decision procedure without back-tracking. The procedure is defined as follows. First $*$-transformation of a given sequent S is performed separately in each branch l. If the branch ends-up by an s-final sequent or elementary non-axiom sequent, then S is not derivable. If the branch ends-up by a proper sequent S^*, then $+$-transformation of S^* is performed separately in each branch l_1. If l_1 ends-up by an s-final sequent or elementary non-axiom sequent, then S is not derivable; otherwise, next proper sequent S_1^* in the proper set of $*$-transformation of S is considered. If either a logical axiom or i-final sequent is obtained in every branch, then $G^*TL \vdash S$; otherwise $G^*TL \nvdash S$.

Termination of the algorithm follows from Lemma 2.

4 Foundation of G*TL

Let G*′TL be a calculus obtained from G*TL by adding the loop-type axioms and non-marked modal rules $(\Box \to)$, $(\to \Box)$.

Lemma 3. (1) *If* $G^{*\prime}TL \vdash^{V} S$, *then* $G^*TL \vdash^{V^*} S$.
(2) *If* $G^{*\prime}TL \vdash^{V} S$, *then* $G'_L PLTL \vdash^{V^*} S^*$.

Theorem 1. *The calculus* G^*TL *is sound and complete.*

References

[1] R. Alonderis. A coding method for a sequent calculus of propositional logic. *Lith. Math. J.*, **48**(2):123–136, 2008.

[2] K. Brünnler and M. Lange. Cut-free sequent systems for temporal logic. *J. Logic Alg. Progr.*, **76**(2):216–225, 2008.

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

[4] M. Fisher. A resolution method for temporal logic. In *Proceedings of 12-th International Joint Conference on AI* (*IJCAI*), Sydney, 1991.

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

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

[7] G. D. Gough. *Decision procedures for temporal logic.* Master's thesis, Department of Computer Science, Manchester University, Oxford Rd., Manchester M13 9PL, UK, October, 1984

[8] N. Nide and S. Takata. Deduction systems for bdi logic using sequent calculus. In *Proc. AAMAS'02*, pp. 928–935, 2002.

[9] R. Pliuškevičius. The saturation tableaux for linear miniscoped Horn-like temporal logic. *J. Autm. Reas.*, **13**:51–67, 1994.

[10] R. Pliuškevičius and A. Pliuškevičienė. Decision procedure for a fragment of mutual belief logic with quantified agent variables. *Lect. Not. Art. Int.*, **3900**:0112–0128, 2006.

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

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

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

REZIUMĖ

Sekvencinis skaičiavimas be pjūvio, invariantinės taisyklės ir ciklų tikrinimo tiesinio laiko teiginių logikai

R. Alonderis, R. Pliuškevičius

Straipsnyje pateikta prisotinimu pagrįsta išsprendžiamumo procedūra tiesinio laiko teiginių logikai (TLTL) su operatoriais „sekantis" ir „visada". Ši prisotinimo procedūra baigia darbą kai gaunamos tam tikro tipo sekvencijos. TLTL savybės leidžia sukonstruoti polinomonio erdvinio sudėtingumo išsprendžiamumo procedūrą, nenaudojant istorijų, ir ciklų tikrinimo.

Raktiniai žodžiai: išsprendimo procedūra, sekvencinis skaičiavimas, ciklų tikrinimas, laiko logika.