Liet. mat. rink. LMD darbai, **50**, 2009, 247–252

Restrictions for loop-check in sequent calculus for temporal logic with until operator

Adomas BIRŠTUNAS
Vilnius University, Faculty of Mathematics and Informatics
Naugarduko g. 24, LT-03225 Vilnius, Lithuania
e-mail: adomas.birstunas@mif.vu.lt

Abstract. In this paper, we present sequent calculus for branching-time temporal logic with *until* operator. This sequent calculus uses efficient loop-check techinque. We prove that we can use not all but only several special sequents from the derivation tree for the loop-check. We use indexes to discover these special sequents in the sequent calculus. These restrictions let us to get efficient decision procedure based on introduced sequent calculus.

Keywords: sequent calculus, branchnig-time temporal logic, *until* operator, loop-check.

1. Introduction

Usual sequent calculi with cut rule are practically unusable in automated environment. So, sequent calculi with analytic cut, or infinitary rule, or some kind of the loop-check must be used to get a decision procedure. Unfortunately, such a sequent calculi are inefficient and need additional modifications to get more or less usable decision procedure. One of the possibilities is constructing sequent calculi with an efficient loop-check (as it is done for some modal logics in [3]), or (if it is only possible) loop-check free calculi (as it is done for $KD45$ logic in [1]).

There is known sequent calculus for linear temporal logic with *until* which is cut free and invariant free sequent calculus [2]. Another cut free and invariant free sequent calculus for branching-time temporal logic with *until* operator may be find in [5] (as a special fragment of the presented sequent calculus for BDI logic). Both calculi requires loop-check to get decision procedure. In the first paper, loop leads only to non-derivable sequent. In the second paper, loop leads to non-derivable seuent or to special loop-axiom. We will define restrictions for a loop-check applicable for both cases. In this paper we concentrate on the branching-time temporal logic with modal operators: $\circ$ ('next'), $A(\phi \cup \psi)$ ('in all futures until') and $E(\phi \cup \psi)$ ('in at least on future until'). Restrictions allows us to use only special sequents in a loop-check.

We define formula in a usual way. In details logic semantics is described in [5].

DEFINITION 1. We say that sequent S is an ancestor of the sequent S' in the derivation tree, if there exist sequence of the sequents $S_1 = S, S_2, \ldots, S_n = S'$ that for every $i = 1, 2, \ldots (n-1)$, sequent S_i is a conclusion and sequent S_{i+1} is a premise of some rule application.

DEFINITION 2. We say, that we have a loop $S \rightsquigarrow S'$ in the sequent derivation tree if the following conditions are true:

– sequent S is an ancestor of the sequent S';

– S' may be obtained from S by rule $(Weak)$ application: $\frac{S}{S'}$ $(Weak)$.

DEFINITION 3. We say, that sequent S' is a *loop-axiom* if there exists sequent S satisfying the following conditions:

– $S \rightsquigarrow S'$ is a loop in the derivation tree;

– between sequents S and S', there exists such a rule $(AU - L)$ or $(EU - L)$ application, that its right premise is sequent S' ancestor.

We have sequent calculus for branching-time temporal logic, which uses loop-check [5]. Sequent calculus has logical rules and the following rules:

$$\frac{\psi, \Gamma \to \Delta \quad \phi, \circ\Xi(\phi \cup \psi), \Gamma \to \Delta}{\Xi(\phi \cup \psi), \Gamma \to \Delta}\,\Xi(U\text{-}L) \qquad \circ\Xi = \begin{cases} \circ A, & \text{if } \Xi = A \\ \neg\circ\neg E, & \text{if } \Xi = E \end{cases}$$

$$\frac{\Gamma \to \psi, \phi, \Delta \quad \Gamma \to \psi, \circ\Xi(\phi \cup \psi), \Delta}{\Gamma \to \Xi(\phi \cup \psi), \Delta}\,\Xi(U\text{-}R) \qquad \Xi \in \{A, E\}$$

$$\frac{\Gamma \to \Theta}{\circ\Gamma \to \circ\Theta}\,(\circ) \qquad \frac{\Gamma \to \Delta}{\Gamma, \Gamma' \to \Delta\Delta'}\,(Weak)$$

(Θ is at most one formula).

DEFINITION 4. Sequent calculus for branching-time temporal logic with inference rules $(\vee L)$, $(\vee R)$, $(\& L)$, $(\& R)$, $(\neg L)$, $(\neg R)$, $(AU - L)$, $(EU - L)$, $(AU - R)$, $(EU - R)$, $(\circ)$, $(Weak)$, with a loop-axiom and with an axiom $\Gamma, \phi \to \Delta, \phi$ we call sequent calculus PTL_1.

2. Restrictions for the loop-check

There is known decision procedure (described in [5]) for sequent calculus PTL_1. Unfortunately, this procedure use direct loop-check technique to detect non-derivable sequent, or to detect loop-axiom.

In other words, we have to deal with two types of the loops. One is a loop-axiom (Definition 3) which may lead initial sequent to be derivable. For this type of the loop we use term loop-axiom. Other is a simple loop, which is not a loop-axiom, and leads initial sequent to be non derivable. For this type of the loop we use term 'nonderivable' loop. We use term loop to denote both types of the loop. All restrictions for loop check presented in this paper are applied for both loop types.

DEFINITION 5. Sequent S is *primary* if S has the shape $\Sigma, \circ\Gamma \to \Pi, \circ\Delta$ and formula sets Σ, Π contains only propositional variables, $\Sigma \cap \Pi = \emptyset$.

DEFINITION 6. Sequent calculus with inference rules $(\vee L)$, $(\vee R)$, $(\& L)$, $(\& R)$, $(\neg L)$, $(\neg R)$, $(AU - L)$, $(EU - L)$, $(AU - R)$, $(EU - R)$, $(\circ_p)$, with a loop-axiom and with an axiom $\Gamma, \phi \to \Delta, \phi$ we call sequent calculus PTL_2.

$$\frac{\Gamma \to \Theta_i}{\Sigma, \circ\Gamma \to \Pi, \circ\Theta_1, \ldots, \circ\Theta_n}\,(\circ_p),$$

where Σ, Π contains only propositional variables and $\Sigma \cap \Pi = \emptyset$.

Simple speaking, rule ($\circ_p$) may be applied only for primary sequent.

LEMMA 1. *Sequent S is derivable in sequent calculus PTL_1 if and only if sequent S is derivable in sequent calculus PTL_2.*

Now we introduce sequent calculus PTL_3 wich uses loop-check only for sequents those are some premises of the rule ($\circ_p$) application. This modification reduces the number of the checked sequents in the derivation tree.

DEFINITION 7. Loop $S \rightsquigarrow S'$ is a $\circ$-loop if there exist primary sequents S_1, S_1' in the derivation tree, and S_1 is obtained from the sequent S, and S_1' is obtained from the sequent S' by rule ($\circ_p$) applications (i.e., $\frac{S}{S_1}(\circ_p)$, $\frac{S'}{S_1'}(\circ_p)$).

DEFINITION 8. We say, that sequent S' is a $\circ$-*loop-axiom* if there exists sequent S satisfying the following conditions:
– $S \rightsquigarrow S'$ is a $\circ$-loop in the derivation tree,
– $S \rightsquigarrow S'$ is a loop-axiom.

DEFINITION 9. Sequent calculus with inference rules $(\vee L)$, $(\vee R)$, $(\& L)$, $(\& R)$, $(\neg L)$, $(\neg R)$, $(AU - L)$, $(EU - L)$, $(AU - R)$, $(EU - R)$, ($\circ_p$), with $\circ$-loop-axiom and with an axiom $\Gamma, \phi \to \Delta, \phi$ we call sequent calculus PTL_3.

LEMMA 2. *Sequent S is derivable in sequent calculus PTL_2 if and only if sequent S is derivable in sequent calculus PTL_3.*

Now we prove some lemmas to introduce main restrictions to the loop-check used. First, we define subformulas and prove some features for them.

DEFINITION 10. We define formulas F *extended set* $Ext(F)$ as follows:
– $Ext(A(\phi \cup \psi)) = Ext(\circ A(\phi \cup \psi)) = \{A(\phi \cup \psi), \circ A(\phi \cup \psi)\}$,
– $Ext(E(\phi \cup \psi)) = Ext(\neg E(\phi \cup \psi)) = Ext(\circ\neg E(\phi \cup \psi)) =$ $Ext(\neg\circ\neg E(\phi\cup\psi)) = \{E(\phi \cup \psi), \neg E(\phi \cup \psi), \circ\neg E(\phi \cup \psi), \neg \circ \neg E(\phi \cup \psi)\}$,
– otherwise $Ext(F) = \emptyset$.

DEFINITION 11. We write $F \subseteq_{sf} G$ to define that F is subformula of G.
We use extended term subformula: if $F \in Ext(G)$, then also $F \subseteq_{sf} G$.

It is evident, that if $F \subseteq_{sf} G$ and $G \subseteq_{sf} F$, then $F, G \in Ext(F) = Ext(G)$.

DEFINITION 12. Formula F is *proper subformula* of G (we write $F \subset_{sf} G$), if $F \subseteq_{sf} G$ and formula G length is greater then formula F length.

DEFINITION 13. Formula F in sequent S is *ground* if for ever formula $G \in S$, formula F is not a proper subformula of G ($F \not\subset_{sf} G$).

It is evident, that if formula F is ground in sequent S and $F \subseteq_{sf} G$ for some formula $G \in S$, then $F, G \in Ext(F) = Ext(G)$.

LEMMA 3. *If $F \subset_{sf} G$, and $G \subseteq_{sf} H$ (or $F \subseteq_{sf} G$, and $G \subset_{sf} H$), then $F \subset_{sf} H$ or formulas $F, G, H \in Ext(F) = Ext(G) = Ext(H)$.*

Proof. It is evident, that $F \subseteq_{sf} H$. Therefore, $F \subset_{sf} H$ (satisfies lemma) or $H \subseteq_{sf} F$. In the second case, we get that $F, G, H \in Ext(F) = Ext(G) = Ext(H)$, because $H \subseteq_{sf} F \subset_{sf} G \subseteq_{sf} H$ (or $H \subseteq_{sf} F \subseteq_{sf} G \subset_{sf} H$).

We have to notice that F has one of the shape: $A(\phi \cup \psi), \circ A(\phi \cup \psi)$, $E(\phi \cup \psi), \neg E(\phi \cup \psi), \circ\neg E(\phi \cup \psi), \neg \circ \neg E(\phi \cup \psi)$ if $Ext(F) \neq \emptyset$.

LEMMA 4. *If $S \rightsquigarrow S'$ is a $\circ$-loop and T is any sequent inside this loop, then any ground formula F in sequent T is such, that $Ext(F) \neq \emptyset$.*

Proof. Since $S \rightsquigarrow S'$ is a $\circ$-loop, there exists such a sequent S'_1, that S'_1 is obtained from the sequent S' by rule $(\circ_p)$ application $(\frac{S'}{S'_1}(\circ_p))$.

There exists ground formula $G \in S$, that $F \subseteq_{sf} G$. $G \in S'$ and $\circ G \in S'_1$, because $S \rightsquigarrow S'$ is a $\circ$-loop. There exists ground formula $H \in T$, that $\circ G \subseteq_{sf} H$.

We have, that $F \subseteq_{sf} G$, $G \subset_{sf} \circ G$, $\circ G \subseteq_{sf} H$. So, according to Lemma 3, $F \subset_{sf} H$ (we get a contradiction, because F is ground formula in T), or $F, G \in Ext(F) = Ext(G) \neq \emptyset$ (satisfies lemma).

LEMMA 5. *Suppose, that $S \rightsquigarrow S'$ is a $\circ$-loop in the derivation tree constructed according sequent calculus PTL_3. If F is ground formula in sequent S or in sequent S', then for every sequent T in $\circ$-loop $S \rightsquigarrow S'$ there exist formula $F' \in Ext(F)$, which is ground in T.*

Proof. Every rule premise contains only subformulas of the rule conclusion.

Case 1) Ground formula $F \in S$. Then $F \in S'$, because $S \rightsquigarrow S'$ is a $\circ$-loop.

Suppose, that formula F is not ground on some sequent T inside the $\circ$-loop $S \rightsquigarrow S'$. So, there exist ground formula $G \in T$, that $F \subset_{sf} G$.

There exist ground formula $H \in S$, that $G \subseteq_{sf} H$. According to Lemma 3, $F \subset_{sf} H \in S$ (contradiction for F being ground in S), or $G \in Ext(F)$ and G is ground in T (satisfies lemma). We got that if F is ground in sequent S, then for any sequent T in a $\circ$-loop $S \rightsquigarrow S'$ there exist formula $F' \in Ext(F)$ which is ground in T.

Case 2) Ground formula $F \in S'$. Suppose, that formula F is not ground on some sequent T inside the $\circ$-loop $S \rightsquigarrow S'$. So, there exist ground formula $G \in T$, that $F \subset_{sf} G$.

There exist ground formula $H \in S$, that $G \subseteq_{sf} H$. According to Lemma 3, $F \subset_{sf} H$ or $G \in Ext(F)$ (satisfies lemma). In the case $F \subset_{sf} H$, we have that $H \in S'$, because $S \rightsquigarrow S'$ is a $\circ$-loop. Formula F is ground in S'. Therefore, $F \not\subset_{sf} H$, and we get a contradiction, because $F \subset_{sf} H$.

COROLLARY 1. *If we have derivation tree satisfying the following items:*
– we have rule application with conclusion T and premise T',
– there exist ground formula F in the sequent T,
– there is no ground formula $F' \in Ext(F)$ in sequent T'.
Then Lemma 5 says, that sequent T is not inside any $\circ$-loop $S \rightsquigarrow S'$.

Proof goes straightforward from Lemma 4 and Lemma 5.

Simple speaking, if some ground formula was deleted during some rule application (in bottom-up direction), then we do not need to check any sequent below that rule application in order to catch a loop. The main problem is to identify such a situation, because every time we delete some ground formula, at least one new ground formula appears.

Rules (in calculus PTL_3), those may satify above conditions, is $(AU - L)$, $(AU - R)$, $(EU - L)$, $(EU - R)$, then we take left premise (nonmodal case), or rule $(\circ_p)$. So, we can add special indexes for *until* operators to catch such a situation.

We add different upper-indexes for every *different* subformula $A(\phi \cup \psi)$, $E(\phi \cup \psi)$ in the initial sequent S. The bottom-index will be a set of indexes. The bottom-index is defined according to the following rule:

if we have that $A^i_U(\phi \cup \psi) \subset_{sf} A^j_V(\gamma \cup \kappa)$ (or $A^i_U(\phi \cup \psi) \subset_{sf} E^j_V(\gamma \cup \kappa)$, $E^i_U(\phi \cup \psi) \subset_{sf} A^j_V(\gamma \cup \kappa)$, $E^i_U(\phi \cup \psi) \subset_{sf} E^j_V(\gamma \cup \kappa)$) then $j \in U$ and $V \subset U$.

It means that every ground formula F in any sequent S have *until* operator with empty set as its bottom index: $A^i_\emptyset(\phi \cup \psi)$, $\circ A^i_\emptyset(\phi \cup \psi)$, $E^i_\emptyset(\phi \cup \psi)$, $\neg E^i_\emptyset(\phi \cup \psi)$, $\circ\neg E^i_\emptyset(\phi \cup \psi)$, $\neg \circ \neg E^i_\emptyset(\phi \cup \psi)$.

Indexes are some kind of the histories, because they store information abaut applied rules. Histories are one of the possible instruments used to make efficient loop-check (some calculi with used histories may be found in [3,4]).

DEFINITION 14. Sequent calculus with inference rules $(\vee L)$, $(\vee R)$, $(\& L)$, $(\& R)$, $(\neg L)$, $(\neg R)$, $(AU - L^*)$, $(EU - L^*)$, $(AU - R^*)$, $(EU - R^*)$, $(\circ^*_p)$, with $\circ$-loop-axiom and with an axiom $\Gamma, \phi \to \Delta, \phi$ we call sequent calculus PTL_4.

$$\frac{\psi^*, \Gamma^* \overset{\delta}{\to} \Delta^* \quad \phi, \circ\Xi^j_U(\phi \cup \psi), \Gamma \to \Delta}{\Xi^j_U(\phi \cup \psi), \Gamma \to \Delta}\ \Xi(U\text{-}L^*) \qquad \circ\Xi = \begin{cases} \circ A, & \text{if } \Xi = A \\ \neg \circ \neg E, & \text{if } \Xi = E \end{cases}$$

$$\frac{\Gamma^* \overset{\delta}{\to} \psi^*, \phi^*, \Delta^* \quad \Gamma \to \psi, \circ\Xi^j_U(\phi \cup \psi), \Delta}{\Gamma \to \Xi^j_U(\phi \cup \psi), \Delta}\ \Xi(U\text{-}R^*) \quad \Xi \in \{A, E\}$$

$$\frac{\Gamma^* \overset{\circ\delta}{\to} \Theta^*_i}{\Sigma, \circ\Gamma \to \Pi, \circ\Theta_1, \ldots, \circ\Theta_n}\ (\circ^*_p), \qquad \begin{array}{c}\Sigma, \Pi \text{ contains only propositional variables} \\ \Sigma \cap \Pi = \emptyset.\end{array}$$

Left premise of the rule $(AU - L^*)$, $(EU - L^*)$, $(AU - R^*)$, $(EU - R^*)$, $(\circ^*_p)$ is sequent S' and conclusion is sequent S. If there is some subformula $A^j_U(\phi \cup \psi)$ or $E^j_U(\phi \cup \psi)$ in sequent S and there is no such subformula (with upper-index j) in sequent S', then $\delta = +$ and bottom-index j is removed from sequent S'. Otherwise, $\delta = \emptyset$ (and $\Gamma^* = \Gamma$, $\Delta^* = \Delta$, $\phi^* = \phi$, $\psi^* = \psi$, $\Theta^*_i = \Theta_i$).

Simple speaking, if we got a sequent marked with $+$, we know, that some ground formula was just deleted and loop cannot appear here. If we delete some ground formula (we also delete some index j), we must get some new ground formula. These new ground formulas will be formulas containing modalized subformula with emptyset as their bottom index.

LEMMA 6. *Sequent S is derivable in sequent calculus PTL_3 if and only if sequent S is derivable in sequent calculus PTL_4.*

The proof is straightforward from Corollary 1.

LEMMA 7. *For any sequent S, decision procedure based on sequent calculus PTL_4 terminates in finite time.*

We obtain an efficient decision procedure for branching-time temporal logic if we use sequent calculus PTL_4 with restricted loop-check (for both loop types). Because of the restrictions during loop-check we check only sequents marked by $\circ$ and only till the first sequent marked with $+$ (in top-down direction).

3. Conclusion

In this paper, we prove that some restrictions for loop-check for branching-time temporal logic with *until* operator may be applied without loosing derivability. We prove that any ground formula is modalized and stable in any $\circ$-loop. These restrictions let us to construct sequent calculus with efficient loop-check, because (during loop-check) only several special marked sequents must be checked.

References

1. A. Birstunas. Efficient loop-check for KD45 logic. *Lith. Math. J.*, 46(1):1–12, 2006.
2. J. Gaintzarain, M. Hermo, P. Lucio, M. Navarro, F. Orejas. A cut-free and invariant-free sequent calculus for PLTL. *Lecture Notes in Computer Science*, Vol. 4646. Springer, Berlin/Heidelberg, 2007, pp. 481–495.
3. A. Heuerding, M. Seyfried, H. Zimmermann. Efficient loop-check for backward proof search in some non-classical propositional logics. In: *Tableaux 96*. *LNCS*, Vol. 1071. 1996, pp. 210–225.
4. M. Mouri. Constructing counter-models for modal logic K4 from refutation trees. *Bull. Section of Logic*, 31(2):81–90, 2002.
5. N. Nide, S. Takata. Deducton systems for BDI logic using sequent calculus. In: *Proc. AAMAS'02*, 2002, pp. 928–935.

REZIUMĖ

A. Birštunas. Apribojimai ciklų radimui sekvenciniame laiko logikos skaičiavime su* until *operatoriumi

Darbe pateiktas sekvencinis skaičiavimas skaidaus laiko logikai su *until* operatoriumi, kuris naudoja efektyvų ciklų radimo mechanizmą. Darbe įrodyta, kad ciklų radimui gali būti naudojamos ne visos, o tik tam tikros specialios išvedimo medžio sekvencijos. Mes naudojame specialius indeksus toms specialioms sekvencijoms aptikti. Šie apribojimai leidžia mums sukonstruoti efektyvią išvedimo paieškos procedūrą, paremtą pristatytu sekvenciniu skaičiavimu.

Raktiniai žodžiai: sekvencinis skaičiavimas, skaidaus laiko logika, *until* operatorius, ciklų radimas.