Liet. mat. rink. LMD darbai, **48/49**, 2008, 278–282

Loop-check elimination for non-transitive distributed knowledge logic

Aida PLIUŠKEVIČIENĖ (MII)
e-mail: aida@ktl.mii.lt

Abstract. A non-transitive distributed knowledge logic $T_n D$, obtained from multi-modal logic T_n by adding distributed knowledge operator, is considered. Sound and complete loop-check-free sequent calculus for this logic is proposed. Termination of derivations in proposed calculus is justified.

Keywords: logic of knowledge, distributed knowledge, sequent calculus, loop-check.

1. Introduction

To consider properties of distributed systems logics of knowledge with distributed knowledge operator were introduced. The decidability (based on finite model property) and the completeness (based on Hilbert-style calculi) of logics with distributed knowledge was proved in [1]. Distributed knowledge, sometimes also called "implicit knowledge", cannot be defined in terms of "everybody knows" or common knowledge [1]. Intuitively, distributed knowledge is the knowledge that can be obtained when the agents (from some agent group) pool their knowledge. If we have only one agent the distributed knowledge reduces to knowledge.

A proof that suitable logical calculus (e.g., sequent or tableaux calculus) allows us to get a decision procedure is crucial but it is not enough. Check of termination of a decision procedure is very important problem and sometimes require serious efforts.

Traditional techniques used to ensure termination of a decision procedure in non-classical (e.g., knowledge-based) sequent (and tableau) calculi is based on *loop-check* [2]. Namely, before applying any rule it is checked if this rule was already applied to "essentially the same" sequent; if this is the case we block the application of the rule. However, loop-check method often leads to an inefficient implementation. Therefore unrestricted loop-check is often considered as useless. In [3] efficient loop-check for modal logics $S4$, tense logic K_t, and a fragment of intuitionistic logic was presented using sequents with two halves and extended by the notion of a history. For modal logic T loop-check-free sequent calculus is presented in [3] using sequents with two halves.

In this paper non-transitive distributed knowledge logic $T_n D$ (obtained from multi-modal logic T_n by adding distributed knowledge operator) is considered. In the paper a loop-check-free sequent calculus for distributed knowledge logic $T_n D$ is constructed. This calculus does not require sequents in a certain normal form and does not use sequents with two halves. To avoid loop-check, applications of reflexivity rules are restricted using marked knowledge and distributed knowledge operators. More efficient loop-check-free specialization of reflexivity rules, different from [4] is presented.

2. Initial Gentzen-style calculus for the logic $T_n D$

The logic $T_n D$ is obtained from the multi-modal logic T_n by adding distributed knowledge operator D.

A *language* of this logic contains: a set of propositional symbols $P, P_1, P_2, \ldots, Q, Q_1, Q_2, \ldots$; a set of agent constants $i, i_1, i_2, \ldots$ $(i, i_l \in \{1, \ldots, n\})$; a set of knowledge operators $\mathbf{K}_1$, $\mathbf{K}_2, \ldots$, $\mathbf{K}_n$; the distributed knowledge operator $\mathbf{D}$; the set of logical symbols $\supset, \wedge, \vee, \neg$.

Formulas are defined in traditional way from propositional symbols using logical symbols, knowledge operators $\mathbf{K}_i$, $i \in \{1, \ldots, n\}$, and distributed knowledge operator $\mathbf{D}$.

The formula $\mathbf{K}_i A$ means "agent i knows A". The formula $\mathbf{D}A$ means "A is distributed knowledge of all set (group) of agents". Distributed knowledge is the knowledge that is implicitly present in a group of agents, and which might become explicit if the agents were able to communicate. For instance, it is possible that no agent knows the assertion Q, while at the same time $\mathbf{D}Q$ may be derived from $\mathbf{K}_1 P \wedge \mathbf{K}_2(P \supset Q)$. We have distributed knowledge of Q if, *putting our knowledge together*, Q *may be deduced, even if none of us individually knows* Q.

Knowledge operators $\mathbf{K}_i$, $i \in \{1, \ldots, n\}$ and distributed knowledge operator $\mathbf{D}$ for the logic $T_n D$ satisfy relations which comply with reflexivity property. The semantics of these operators is defined using Kripke structure (see, e.g., [1]).

In Gentzen-style calculus for the considered logic instead of formulas we consider sequents, i.e., formal expressions $A_1, \ldots, A_k \to B_1, \ldots, B_m$ where $A_1, \ldots, A_k$ $(B_1, \ldots, B_m)$ is a multiset of formulas.

Let us introduce a Gentzen-style calculus $GT_n D$ for the logic $T_n D$. The calculus $GT_n D$ is defined by the following postulates:

Axiom: $\Gamma, P \to \Delta, P$ where P is a propositional symbol.

Logical rules: traditional invertible rules for logical symbols $\supset, \wedge, \vee, \neg$.

Modal rules: rules for knowledge operators $\mathbf{K}_i$ and distributed knowledge operator $\mathbf{D}$

$$\frac{\Pi, A, \mathbf{K}_i A \to \Delta}{\Pi, \mathbf{K}_i A \to \Delta}\ (\mathbf{K}_i \to) \qquad \frac{\Pi, A, \mathbf{D}A \to \Delta}{\Pi, \mathbf{D}A \to \Delta}\ (\mathbf{D} \to),$$

$$\frac{\Gamma \to A}{\Pi, \mathbf{K}_i \Gamma \to \Delta, \mathbf{K}_i A}\ (\mathbf{K}_i) \qquad \frac{\Gamma \to A}{\Pi, \mathbf{D}\Gamma \to \Delta, \mathbf{D}A}\ (\mathbf{D}),$$

$$\frac{\Gamma \to A}{\Pi, \mathbf{K}\Gamma \to \Delta, \mathbf{D}A}\ (I),$$

where in the rules $(\mathbf{K}_i)$, $(\mathbf{D})$ $Q\Gamma$ $(Q \in \{\mathbf{K}_i, \mathbf{D}\})$ means either empty word or multiset of formulas $QA_1, \ldots, QA_m$ $(m \geqslant 1)$; in the rule (I) $\mathbf{K}\Gamma$ means either empty word or multiset of formulas $\mathbf{K}_1\Gamma_1, \ldots, \mathbf{K}_n\Gamma_n$ $(n \geqslant 1)$ where $\mathbf{K}_j\Gamma_j$ means either empty word or multiset of formulas $\mathbf{K}_j B_1, \ldots, \mathbf{K}_j B_{q_j}$ $(q_j \geqslant 1)$. The rules $(\mathbf{K}_i \to)$ and $(\mathbf{D} \to)$

are called reflexivity rules because they correspond to reflexivity axioms; the rules $(\mathbf{K}_i)$ and $(\mathbf{D})$ are called distributivity rules because they correspond to distributivity axioms; the rule (I) is called interaction rule and it corresponds to interaction axiom $\mathbf{K}_i A \supset \mathbf{D}A$, $(i = 1, \ldots, n)$.

Analogously as in [4] we get

THEOREM 1. *The calculus GT_nD is sound and complete.*

3. Loop-check-free sequent calculus for T_nD

With a view to get stopping device different from loop checking let us introduce marked knowledge operators $\mathbf{K}_i^*$ and marked distributed knowledge operator $\mathbf{D}^*$ which allow us to get loop-check-free reflexivity rules.

A sequent S is a *primary* one, if S is of the following shape:

$\Sigma_1,\ \mathbf{K}^\sigma\Gamma,\ \mathbf{D}^\sigma\Pi_1 \to \Sigma_2,\ \mathbf{K}\Delta,\ \mathbf{D}\Pi_2$, where $\sigma \in \{\varnothing, *\}$ and for every i $(i \in \{1, 2\})$

– Σ_i is empty or consists of propositional symbols;

– $\mathbf{K}^\sigma\Gamma$ is empty or consists of formulas of the shape $\mathbf{K}_l^\sigma A$;

– $\mathbf{D}^\sigma\Pi_1$ is empty or consists of formulas of the shape $\mathbf{D}^\sigma B$;

– $\mathbf{K}\Delta$ is empty or consists of formulas of the shape $\mathbf{K}_l M$;

– $\mathbf{D}\Pi_2$ is empty or consists of formulas of the shape $\mathbf{D}N$.

Let G_1T_nD be a calculus obtained from the calculus GT_nD by the following transformations:

• replacing the reflexivity rules $(\mathbf{K}_i \to)$ and $(\mathbf{D} \to)$ by the following ones:

$$\frac{\Sigma_1,\ \mathbf{K}^\sigma\Gamma,\ \mathbf{D}^\sigma\Pi,\ A,\ \mathbf{K}_i^* A \to \Sigma_2}{\Sigma_1,\ \mathbf{K}^\sigma\Gamma,\ \mathbf{D}^\sigma\Pi,\ \mathbf{K}_i A \to \Sigma_2}\ (\mathbf{K}_i^* \to),$$

$$\frac{\Sigma_1,\ \mathbf{K}^\sigma\Gamma,\ \mathbf{D}^\sigma\Pi,\ A,\ \mathbf{D}^* A \to \Sigma_2}{\Sigma_1,\ \mathbf{K}^\sigma\Gamma,\ \mathbf{D}^\sigma\Pi,\ \mathbf{D}A \to \Sigma_2}\ (\mathbf{D}^* \to),$$

where in the conclusion of the rules $(Q^* \to)$ $(Q \in \{\mathbf{K}_i, \mathbf{D}\})$ $\Sigma_1 \cap \Sigma_2$ is empty and the operator Q in the formula QA is not marked;

• replacing the rules $(\mathbf{K}_i)$, $(\mathbf{D})$, and (I)) by the following rules where the conclusion is a primary sequent and $\Sigma_1 \cap \Sigma_2$ is empty:

$$\frac{\Gamma_i^\circ \to\ A}{\Sigma_1,\ \mathbf{K}^\sigma\Gamma,\ \mathbf{D}^\sigma\Pi_1 \to \Sigma_2,\ \mathbf{K}\Delta,\ \mathbf{K}_i A,\ \mathbf{D}\Pi_2}(\mathbf{K}_i^p),$$

where $i \in \{1, \ldots, n\}$; $\mathbf{K}^\sigma\Gamma = \mathbf{K}_1^\sigma\Gamma_1, \ldots, \mathbf{K}_n^\sigma\Gamma_n$ and $\mathbf{K}_j^\sigma\Gamma_j (j \in \{1, \ldots, n\})$ is empty or consists of formulas of the shape $\mathbf{K}_j^\sigma B$; if $\mathbf{K}^\sigma\Gamma$ contains $\mathbf{K}_i^\sigma\Gamma_i$ then Γ_i° is Γ_i otherwise Γ_i° is empty.

$$\frac{\Gamma, \Pi_1 \to A}{\Sigma_1,\ \mathbf{K}^\sigma\Gamma,\ \mathbf{D}^\sigma\Pi_1 \to \Sigma_2,\ \mathbf{K}\Delta,\ \mathbf{D}\Pi_2,\ \mathbf{D}A}(I_c^p).$$

Let us note that in the calculus G_1T_nD applications of the rules $(\mathbf{K}_i^* \to)$ and $(\mathbf{D}^* \to)$ are restricted in such way that it is not possible to apply these rule twice using the same occurrence of a formula as main formula.

From the shape of primary sequent, invertibility of the logical and reflexivity rules we get

LEMMA 1 (reduction to primary sequents). *Every sequent S can be reduced to a set of primary sequents $\{S_1, \ldots, S_m\}$, $m \geqslant 1$, by applying the logical and reflexivity rules of G_1T_nD backwards. Moreover, if $G_1T_nD \vdash^V S$ then for all j $(j \in \{1, \ldots, m\})$ $G_1T_nD \vdash^{V_j} S_j$.*

A primary sequent of the shape $\Sigma_1, \mathbf{K}^*\Gamma, \mathbf{D}^*\Pi \to \Sigma_2$ where $\Sigma_1 \cap \Sigma_2$ is empty and $\mathbf{K}^*\Gamma$ $(\mathbf{D}^*\Pi)$ is empty or consist of formulas of the shape $\mathbf{K}_i^*M$ $(\mathbf{D}^*M$, correspondingly), is a *final* one. It is impossible to apply any rule to final sequent.

A derivation V of a sequent S in the calculus G_1T_nD is a *successful* one, if *each* branch of V ends with an axiom. A derivation V of S in the calculus G_1T_nD is an *unsuccessful* one if V contains a branch ending with a final sequent. A sequent S is *derivable* in the calculus G_1T_nD if and only if *there exists* a successful derivation V of S. Thus, if *all possible* derivations of S in G_1T_nD are unsuccessful, the sequent S is *non-derivable*.

Analogously as in [4] we get

THEOREM 2. *If $GT_nD \vdash S$ then $G_1T_nD \vdash S$.*

4. Termination of derivations in G_1T_nD

Let us describe a loop-check-free algorithm with restricted backtracking for backward proof search in G_1T_nD.

LEMMA 2 (existential invertibility of the rules $(\mathbf{K}_i^p)$ and (I_c^p)). *Let S be a primary sequent $\Sigma_1, \mathbf{K}^\sigma\Gamma, \mathbf{D}^\sigma\Pi_1 \to \Sigma_2, \mathbf{K}\Delta, \mathbf{D}\Pi_2$ such that $\Sigma_1 \cap \Sigma_2$ is empty and $\mathbf{K}\Delta \cup \mathbf{D}\Pi_2$ is not empty. Let $G_1T_nD \vdash S$, then*

– there exists a formula $\mathbf{K}_iA_i$ from $\mathbf{K}\Delta$ such that $G_1T_nD \vdash \Gamma_i \to A_i$;

– or there exists a formula $\mathbf{D}A$ from $\mathbf{D}\Pi_2$ such that $G_1T_nD \vdash \Gamma, \Pi_1 \to A$.

Proof. The proof is carried out by induction on the height of the given derivation of the sequent S.

Relying on the calculus G_1T_nD, definition of derivability in G_1T_nD, Lemmas 1, 2, and using invertibility of the logical rules and reflexivity rules we get that the decision algorithm consists of several levels. Each level contains three main parts:

- the considered sequent S is reduced to a set of primary sequents;
- the obtained set of primary sequents is checked. If the considered primary sequent is an axiom then the considered branch of derivation is finished and a derivation of the next primary sequent is constructed;
- if the considered primary sequent is not an axiom then, according to Lemma 2, rules $(\mathbf{K}_i^p)$ and (I_c^p) are backward applied (in all possible ways). The premise of this application is used to start a new level of algorithm.

Thus a derivation in G_1T_nD consists of repeating reductions to primary sequents and following backward application of one-in-two rules $(\mathbf{K}_i^p)$, (I_c^p) to each received primary sequent. It is obvious that algorithm finishes a search when either in all branches an axiom is obtained or in all possible derivations a final sequent is obtained.

With the aim to prove termination of presented algorithm a *complexity* of derivability of a sequent S in G_1T_nD is considered. Let B be a formula entering in S. A subformula of B is a *modal* one if it has the shape $Q^\mu M$ where $Q \in \{\mathbf{K}_i, \mathbf{D}\}$ and $\mu \in \{\varnothing, *\}$. A modal subformula $Q^\mu M$ may occur both positively and negatively in B. The *complexity* of sequent S (denoted by $C(S)$) is defined as an ordered triple $< k(S), n(S), l(S) >$ where

- $k(S)$ is the number of different modal subformulas of the shape QM ($Q \in \{\mathbf{K}_i, \mathbf{D}\}$) entering in S *positively*;
- $n(S)$ is the number of different modal subformulas of the shape QM (i.e., the outmost operator in QM is not marked) entering in S and such that at least one occurrence of QM enters in S *negatively* and *does not occur within the scope of marked operator* Q^* ($Q \in \{\mathbf{K}_i, \mathbf{D}\}$ (it means that if a considered modal subformula enters in S negatively and occurs *only* within the scope of marked operators then this subformula is not counted);
- $l(S)$ is the length of S defined as $\sum_{i=1}^{k} l(B_i)$, where $l(B_i)$ is the length (defined in a traditional way) of i-th ($1 \leqslant i \leqslant k$) member of a sequent S.

LEMMA 3. *Let $G_1T_nD \vdash^V S^*$, and (j) is a rule of the calculus G_1T_nD. Let a sequent S be a conclusion of an application of the rule (j) in V and S_1 be a premise of the same application of the rule (j). Then $C(S_1) < C(S)$.*

Proof. If (j) is a logical rule then $k(S_1) = k(S)$, $n(S_1) = n(S)$ but $l(S_1) < l(S)$. If $(j) = (Q^* \rightarrow)$ ($Q^* \in \{\mathbf{K}_i^*, \mathbf{D}^*\}$) then $n(S_1) < n(S)$. If $(j) = (Q^p)$ ($Q^p \in \{\mathbf{K}_i^p, I_c^p\}$ then $k(S_1) < k(S)$. Thus, in all cases $C(S_1) < C(S)$.

References

1. R. Fagin, J.Y. Halpern, Y. Moses, M.Y. Vardi, *Reasoning about Knowledge*, MIT Press, Cambridge, Mass (1995).
2. M. Fitting, *Proof Methods for Modal and Intuitionistic Logics*, *Synthese Library*, **169**, D. Reidel, Dordrecht, Holland (1983).
3. A. Heuerding, M. Seyfried, H. Zimmermann, Efficient loop-check for backward proof search in some non-classical propositional logics, *Lecture Notes in Computer Science*, **1071**, 210–225 (1996).
4. R. Pliuškevičius, A. Pliuškevičienė, Termination of derivations in a fragment of transitive distributed knowledge logic, *Informatica*, **19**(4), 597–616 (2008).

REZIUMĖ

A. Pliuškevičienė. Ciklų tikrinimo eliminavimas paskirstyto žinojimo netranzityviai logikai

Sukonstruotas korektiškas ir pilnas beciklis sekvencinis skaičiavimas netranzityviai paskirstyto žinojimo logikai. Ciklų tikrinimo eliminavimui yra siūloma efektyvi refleksyvumo taisyklių specializacija. Pagrįstas išvedimų pateiktame skaičiavime baigtinumas.

Raktiniai žodžiai: žinojimo logika, paskirstytas žinojimas, sekvencinis skaičiavimas, ciklų tikrinimas.