A Cut-free Tableau Calculus for the Logic of Common Knowledge

Pietro Abate and Rajeev Goré
Australian National University, Canberra, Australia,
[Pietro.Abate | Rajeev.gore]@anu.edu.au

February 19, 2007

Abstract

In this paper we focus on the development of a cut-free finitary tableau calculus with histories for n-agent modal logics with common knowledge (LCK). Thus, we get a proof system where proof-search becomes feasible and we lay the basis for developing a uniform framework for the treatment of the family of logics of common knowledge. Unlike proof calculi based on temporal logics, our calculus gives a single-pass decision procedure which is space-optimal.

1 Introduction

Reasoning about knowledge, in particular about the knowledge of agents who reason about the world and each other's knowledge, plays an important role in several areas of computer science, philosophy, game theory and many other fields. The notion of Common Knowledge (CK), where everybody knows, everyone knows that everyone knows, etc has also been proved fundamental in fields which deal with the analysis of interacting groups of agents (see [FHMV95] for an extensive overview).

In this paper we focus on the development of a cut-free finitary tableau calculus with histories for n-agent modal logics with common knowledge (LCK). Thus, we get a proof system where proof-search becomes feasible and we lay the basis for developing a uniform framework for the treatment of the family of logics of common knowledge. Unlike proof calculi based on temporal logics, our calculus gives a single-pass decision procedure which is space-optimal.

2 Syntax and Semantics

We consider countably many propositional variables AP, a finite non-empty set of agents A and the connectives $\neg, \wedge, [a]$ with $a \in A$ with $AP \cap A = \emptyset$ and the constants $\bot$ and $\top$. The formulae of LCK are then defined using the BNF grammar below:

$$\begin{array}{lll} p & ::= & p_0 \mid p_1 \mid p_2 \mid \cdots \\ \varphi & ::= & p \mid \top \mid \bot \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \\ & & \mid [a]\varphi \mid \langle a \rangle\varphi \\ & & \mid [C]\varphi \mid \langle C \rangle\varphi \\ & & \mid [E]\varphi \mid \langle E \rangle\varphi \end{array}$$

The abbreviations for $\langle a \rangle$ and $\vee$ are as usual. We define $[E]$ and $\langle E \rangle$ as:

$$[E]\varphi = \bigwedge_{a \in A} [a]\varphi \quad \langle E \rangle\varphi = \bigvee_{a \in A} \langle a \rangle\varphi$$

Letting $[E]^0\varphi = \varphi$, let $[E]^k\varphi$ be $[E][E]^{k-1}\varphi$ for some $k \geq 1$. Finally, we define the common knowledge operator C as $[C]$ and its dual as $\langle C \rangle$. We use the term "iterated eventuality" for formulae of the form $\langle C \rangle\varphi$ and we use the term "eventualities" for formulae of the form $\langle i \rangle\varphi$.

Definition 2.1. Given a finite non-empty set A of agents and a set P of atoms. An LCK model is a triple $\langle W, R, V \rangle$ where W is a non-empty set of worlds, R is a function that assigns to each agent $a \in A$ an accessibility relation $R_a \subseteq W \texttt{x} W$ and V a valuation that assigns to each atom $p \in P$ a set $V(p) \subseteq \mathcal{P}(W)$ where $\mathcal{P}(W)$ is the subset of all subset of W.

Definition 2.2. Let $R_A := (\bigcup_{a \in A} R_a)^+$. Given an LCK model $\langle W, R, V \rangle$, an R-path $\pi = w_0, w_1, \ldots$ is a sequence of worlds such that $w_i R_A w_{i+1}$ for all $i \geq 0$. We write π_i for w_i.

Definition 2.3. Let $M =< W, R, V >$ be an LCK model and $w \in W$ then the value of $M, w \vDash \varphi$ of a formula φ in a world w is inductively defined as follows:

1. $M, w \vDash p$ iff $w \in V(\alpha)$
2. $M, w \vDash \varphi \vee \psi$ iff $M, w \vDash \varphi$ or $M, w \vDash \psi$
3. $M, w \vDash \varphi \wedge \psi$ iff $M, w \vDash \varphi$ and $M, w \vDash \psi$
4. $M, w \vDash \langle a \rangle\varphi$ iff $\exists w' \in W, wR_a w'$ and $M, w' \vDash \varphi$
5. $M, w \vDash [a]\varphi$ iff $\forall w' \in W, wR_a w'$ implies $M, w' \vDash \varphi$
6. $M, w_0 \vDash \langle C \rangle\varphi$ iff there exists an R-path $w_0 R_A w_1 R_A w_n$ with $n \geq 1$ such that $w_n \vDash \varphi$ and $\forall i$, $1 \leq i < n \Rightarrow w_i \vDash \neg\varphi$.
7. $M, w_0 \vDash [C]\varphi$ iff for all R-paths π with $\pi_0 = w_0$, and for all $i \geq 1$, $\pi_i \vDash [C]\varphi$ and $\pi_i \vDash \varphi$

Definition 2.4. A formula ϕ is LCK satisfiable iff there exists an LCK model $M = \langle W, R, V \rangle$ and a world $w \in W$ such that $M, w \vDash \phi$. A formula ϕ is valid iff $\neg\varphi$ is LCK un-satisfiable.

3 Tableau Rules for LCK_n

In this section we give a calculus to reason about LCK using implicit tableau (see [Gor99] for an overview) and histories [HSZ96, Sch98]. Each rule is composed of a numerator and a list of denominators when the numerator and each denominators is of the form $\Gamma \;::\; H \;::\; V$ where:

Γ is a set of formulae as usual,

H consists of two histories called Fev and Br where

- Fev is a set of (eventuality) formulae fulfilled between two consecutive applications of transitional rules ($\langle i \rangle$),
- Br is a list of pairs, of the form $(Core, Fev)$ where the first component of the pair is a set representing the core that led to a state, and the second component is a set of the iterated eventualities that were fulfilled by this branch in creating that state out of the $Core$,

V consists of one variable called uev where uev is a list of pairs (φ, n) where φ is a $\langle C \rangle$-formula and n is an integer. This variable tracks iterated eventualities that might be fulfilled in this branch.

To focus only on locally relevant parts, we use "…" for the not important parts. If "…"appears at the same position in the numerator and denominator(s) of a rule, then we mean that the corresponding parts are the same.

A tableau for a set Γ is a tree of nodes with root $\Gamma :: H :: V$ where the children of a node are obtained by instantiating a rule: we say that a rule is associated with the parent node and conversely, a node is associated with the numerator of a rule. A node is terminal if it is associated with an application of a terminal rule.

We say that the tableau is *expanded* if each leaf node is obtained by an application of a terminal rule. A node is saturated if it is composed only of positive and negative atoms. In the following we present a tableau calculus for the logic of common knowledge with two agents LCK_2.

Linear Rules.

$$(\wedge)\quad \frac{\varphi \wedge \psi \;;\; \Gamma :: \cdots}{\varphi \;;\; \psi \;;\; \Gamma :: \cdots}$$

$$([C])\quad \frac{[C]\varphi \;;\; \Gamma :: ... :: ...}{[E]\varphi \;;\; [E][C]\varphi ; \Gamma :: ... :: ...} \qquad ([E])\quad \frac{[E]\varphi \;;\; \Gamma :: ... :: ...}{[1]\varphi \;;\; [2]\varphi ; \Gamma :: ... :: ...}$$

Universal Branching Rules.

$$(\vee)\quad \frac{\varphi \vee \psi \;;\; \Gamma :: \cdots :: uev}{\varphi \;;\; \Gamma :: \cdots :: uev_1 \mid \psi \;;\; \Gamma :: \cdots :: uev_2}$$

$$(\langle C \rangle) \quad \frac{\langle C \rangle \varphi \ ; \ \Gamma :: Fev, Br :: uev}{\langle E \rangle \varphi \ ; \ \Gamma :: \{\langle C \rangle \varphi\} \cup Fev, Br :: uev_1 \mid \langle E \rangle \langle C \rangle \varphi \ ; \ \Gamma :: Fev, Br :: uev_2}$$

$$(\langle E \rangle) \quad \frac{\langle E \rangle \varphi \ ; \ \Gamma :: Fev, Br :: uev_1}{\langle 1 \rangle \varphi \ ; \ \Gamma :: Fev, Br :: uev \mid \langle 2 \rangle \varphi \ ; \ \Gamma :: Fev, Br :: uev_2}$$

where in the $(\langle C \rangle)$, $(\langle E \rangle)$ and $(\vee)$ rules :

$$\begin{array}{rcl} f(x,y) & = & \{j \mid (x,j) \in y\} \\ UC & = & \{(\langle C \rangle \psi, n) \mid \\ & & \quad f(\langle C \rangle \psi, uev_1) \neq \emptyset \\ & & \quad \& \ f(\langle C \rangle \psi, uev_2) \neq \emptyset \\ & & \quad \& \ n = min\ (f(\langle C \rangle \psi, uev_1) \ \cup \ f(\langle C \rangle \psi, uev_2)) \ \} \\ uev & := & \begin{cases} uev_1 & \text{if } uev_2 = \{(\texttt{false}, \_)\} \\ uev_2 & \text{if } uev_1 = \{(\texttt{false}, \_)\} \\ UC & \text{otherwise} \end{cases} \end{array}$$

Existential Branching Rule.

$$\langle a \rangle \quad \frac{\langle a \rangle \varphi_1 \ ; \ldots ; \langle a \rangle \varphi_n \ ; \ \langle a \rangle \Gamma \ ; \ [a]\Delta \ ; \ \Lambda :: Fev, Br :: uev}{\begin{array}{c} \varphi_1 \ ; \ \Delta \\ :: \emptyset, Br_1 :: uev_1 \\ \ddagger(1) \end{array} \ || \cdots || \ \begin{array}{c} \varphi_n \ ; \ \Delta \\ :: \emptyset, Br_n :: uev_n \\ \ddagger(n) \end{array} \ || \ \begin{array}{c} \langle a \rangle \Gamma \ ; \ [a]\Delta \ ; \ \Lambda \\ :: Fev, Br :: uev_{n+1} \\ \dagger \end{array}}$$

with :

$n \geq 1$
$\ddagger(i)$ is the condition $\forall j \ . \ 0 \leq j \leq len(Br) \Rightarrow \{\varphi_i\} \cup \Delta \neq Br[j].core$
$\dagger$ is the (blocking) condition $\forall \psi \in \Gamma \ . \ \exists j \geq 0$ s.t. $\{\psi\} \cup \Delta = Br[j].core$

and where

$Br_i = (\{\varphi_i\} \cup \Delta, Fev).Br$ for $1 \leq i \leq n$
$l = len(Br) - 1$
$UEV = \bigcup_{j=1}^{n+1} uev_j$

$$uev := \begin{cases} UEV & \text{if } (\texttt{false}, \_) \notin UEV \,\&\, \forall(\_, n) \in UEV \ . \ n \leq l \\ \{(\texttt{false}, l)\} & \text{otherwise} \end{cases}$$

Terminal Rules.

$$(id) \quad \frac{\langle - \rangle \Gamma \ ; \ [-]\Delta \ ; \ \Lambda :: Fev, Br :: uev}{Stop} \ \{\neg p; p\} \subseteq \Lambda$$

where $uev := \{(\texttt{false}, len(Br))\}$

$$(block) \quad \frac{\langle - \rangle \Gamma \ ; \ [-]\Delta \ ; \ \Lambda :: Fev, Br :: uev}{Stop} \ \{\neg p; p\} \not\subseteq \Lambda \text{ and } \dagger$$

where † is the condition $\forall\psi \in \Gamma \,.\, \exists j \geq 0 \,.\, \{\psi\} \cup \Delta = Br[j].core$, and

$$\begin{aligned} uev \;\; := \;\; & \{\; (\langle C\rangle\varphi, i) \mid \exists\, i \text{ such that } \{\langle C\rangle\varphi\} \cup \Delta = Br[i].core \\ & \quad \& \; \forall j \,.\, i \leq j \leq len(Br) \Rightarrow \langle C\rangle\varphi \in Br[j].core \\ & \quad \& \; \langle C\rangle\varphi \notin Br[j].fev \;\} \end{aligned}$$

Proposition 3.1 (Termination)**.** *A LCK-tableau for a node $\varphi :: \emptyset, [] :: uev$ always terminates.*

Theorem 3.2. *If $\mathcal{T}$ is an expanded tableau for $\varphi :: \emptyset, [] :: uev$, with $uev = \emptyset$ then φ is LCK-satisfiable.*

Theorem 3.3. *If a formula φ is LCK-satisfiable then there exists an expanded tableau for $\varphi :: \emptyset, [] :: uev$ where $uev = \emptyset$.*

4 Conclusions

We presented a cut-free tableau calculus based on histories to reason about LCK. The modularity of the system makes it easy to extend it to the logic of common knowledge based upon T, $S4$ and $S5$. An implementation of LCK with two agents using the tableaux workbench (`TWB`) [AG03] is available at `http:\\twb.rsise.anu.edu.au`.

References

[AG03] Pietro Abate and Rajeev Goré. System description: The tableaux workbench (TWB). In *TABLEAUX*, Lecture Notes in Artificial Intelligence. Springer, 2003.

[FHMV95] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning about Knowledge*. The MIT Press, 1995.

[Gor99] R Goré. Chapter 6: Tableau methods for modal and temporal logics. In *Handbook of Tableau Methods*, pages 297–396. Kluwer Academic Publishers, 1999.

[HSZ96] Alain Heuerding, Michael Seyfried, and Heinrich Zimmermann. Efficient loop-check for backward proof search in some non-classical propositional logics. In *Analytic Tableaux and Related Methods*, pages 210–225, 1996.

[Sch98] S Schwendimann. A new one-pass tableau calculus for PLTL. In *TABLEAUX*, volume 1397 of *Lecture Notes in Artificial Intelligence*, pages 277–291. Springer, 1998.