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

December 1, 2006

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 two-pass decision methods like those for 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 two-pass proof calculi based on temporal logics, or the the existing solutions in [AJ02], [vD02], we give a single-pass, finitary and cut-free decision procedure for LCK

2 Syntax and Semantics

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

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

We define $[E]$ and $\langle E\rangle$ and $[E]^k$ with $k \geq 1$ as:

$$[E]\varphi := \bigwedge_{a\in A} [a]\varphi \qquad \langle E\rangle\varphi := \bigvee_{a\in A} \langle a\rangle\varphi \qquad [E]^0\varphi := \varphi \qquad [E]^k\varphi := [E][E]^{k-1}\varphi$$

Thus the common knowledge operator is $[C]$ and has a dual $\langle C\rangle$. 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. 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 $R_a \subseteq W\mathtt{x}W$ 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 π is a sequence $w_0, w_1, \ldots$ of worlds from W such that $w_i R_A w_{i+1}$ for all $i \geq 0$. We sometimes write π as $w_0 R_A w_1 R_A \ldots$ write π_i for w_i, and write $\Pi(w)$ as the set of all R-paths starting at w.

Definition 2.3. Let $M = \langle W, R, V\rangle$ 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 $w \in V(p)$
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 $\exists v \in W, wR_a v$ and $M, v \vDash \varphi$
5. $M, w \vDash [a]\varphi$ iff $\forall v \in W, wR_a v$ implies $M, v \vDash \varphi$
6. $M, w \vDash \langle C\rangle\varphi$ iff there exists an R-path $wR_A w_1 R_A \cdots R_A w_n \in \Pi(w)$ with $n \geq 1$ such that $\pi_n \vDash \varphi$ and $\forall i$, $1 \leq i < n \Rightarrow \pi_i \vDash \neg\varphi$.
7. $M, w \vDash [C]\varphi$ iff $\forall$ R-paths $\pi \in \Pi(w)$, and $\forall i \geq 1$, $\pi_i \vDash [C]\varphi$ 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 $M, w \vDash \varphi$. A formula φ is valid iff $\neg\varphi$ is not LCK-satisfiable.

3 Tableau Rules for LCK_n

In this section we give a tableau calculus to determine LCK-satisfiability using traditional tableau extended with histories to pass extra information from parents to children and variables to pass information from children to parents: see [Gor99] for an overview, and also [HSZ96, Sch98] for similar calculi.

As is usual, the rules are categorised into static rules and transitional rules [Gor99]. We assume a strategy for rule applications that applies all the static rules until they are no longer applicable, giving a (saturated) state, and then applies the transitional rules to obtain (the cores of the) new pre-states.

Each rule is composed of a numerator and a list of denominators where the numerator and each denominator 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 the transitional rules,

Br is a list of pairs, of the form $(Core, Fev)$ where the first component of the pair is a set representing the (pre-state) 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 a variable *uev* which is a list of pairs $(\langle C\rangle\varphi, n)$ where n is an integer. *uev* tracks iterated eventualities unfulfilled in this branch, but which may be fulfilled by sibling branches as we pop the recursion stack.

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 (which has no denominator). We say that the tableau is *expanded* if each leaf node is obtained by an application of a terminal rule.

In the following we present a tableau calculus for the logic of common knowledge whenLCK_2 with two agents $A = \{1, 2\}$ to keep things simple. We use $\langle A\rangle\Gamma$ to stand for a set $\langle 1\rangle\Gamma_1 \cup \langle 2\rangle\Gamma_2$ and $[A]\Delta$ to stand for an arbitrary set $[1]\Delta_1 \cup [2]\Delta_2$. To minimise the number of rules, we assume that all formulae are in negation normal form. A literal is an atom or a negated atom. We use Λ to stand for a set of literals.

Static Linear Rules.

$$(\wedge)\ \frac{\varphi \wedge \psi\ ;\ \Gamma :: \cdots}{\varphi\ ;\ \psi\ ;\ \Gamma :: \cdots} \qquad ([C])\ \frac{[C]\varphi\ ;\ \Gamma :: \cdots}{[E]\varphi\ ;\ [E][C]\varphi\ ;\ \Gamma :: \cdots} \qquad ([E])\ \frac{[E]\varphi\ ;\ \Gamma :: \cdots}{[1]\varphi\ ;\ [2]\varphi\ ;\ \Gamma :: \cdots}$$

Static Universal Branching Rules.

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

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

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

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

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

Transitional Existential Branching Rule. For each $a \in A = \{1, 2\}$, a rule:

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

with $n \geq 1$ and for $1 \leq i \leq n$:

$\ddagger(i)$ 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 $1 \leq i \leq n$

$l = len(Br) - 1$

$UEV = \bigcup_{j=1}^{n+1} uev_j$

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

Static Terminal Rules.

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

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

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

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

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

Proposition 3.1 (Termination). *Every LCK-tableau for a node $\varphi :: \emptyset, [] :: uev$ is finite.*

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

Theorem 3.3 (Completeness). *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. Our calculus is easy to extend 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] P Abate and R Goré. System description: The tableaux workbench. In *TABLEAUX*, LNCS, Springer, 2003.

[AJ02] L Alberucci and G Jäeger. About cut elimination for logics of common knowledge. TR, University of Berne, Switzerland, 2002.

[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, 1999.

[HSZ96] A Heuerding, M Seyfried, and H Zimmermann. Efficient loop-check for backward proof search in some non-classical propositional logics. In TABLEAUX96, LNCS pages 210–225, 1996.

[Sch98] S Schwendimann. A new one-pass tableau calculus for PLTL. In *TABLEAUX*, LNCS 1397:277–291, Springer, 1998.

[vD02] Hans van Ditmarsch and Roy Dyckhoff. Sequent calculi for logics with common knowledge. manuscript, June 2002.