Towards Agents-Based Model Checking

Norihiro Kamide

Waseda Institute for Advanced Study, Waseda University,
1-6-1 Nishi Waseda, Shinjuku-ku, Tokyo 169-8050, JAPAN.
logician-kamide@aoni.waseda.jp

Abstract. A new logic, agents-indexed computation tree logic (ACTL), is obtained from the standard computation tree logic CTL by adding some agent operators. ACTL is intended to appropriately formalize reasoning about agents-based (or distributed) concurrent systems within an executable temporal logic by model checking. The model-checking, validity and satisfiability problems of ACTL are shown to be decidable.

1 Introduction

Verifying agents-based (or distributed) concurrent systems is growing importance in Computer Science and Artificial Intelligence, since computer systems are generally used by or composed of multi-agents in parallel. It is known that *computation tree logic* (CTL) [2] is one of the most useful temporal logics for verifying concurrent systems by *model checking* [3]. In this paper, an extension of CTL, called an *agents-indexed computation tree logic* (ACTL), is introduced by adding some agent operators $\heartsuit_i$ (agent i has information) and $\heartsuit_c$ (common information) to CTL.

ACTL is intended to appropriately formalize reasoning about agents-based concurrent systems within an executable temporal logic by model checking. ACTL has some useful descriptions concerned with such agents-based reasoning. An example of such descriptions is: $\mathrm{AG}(\heartsuit_c\ \mathit{password}{\to}\heartsuit_i\mathrm{AF}\ \mathit{login})$ which means: "If the login password of a computer is regarded as common information in the group $A := \{1, 2, ..., n\}$ of agents, then an agent i in A will eventually be able to login the computer."

In this paper, a theorem for embedding ACTL into CTL is proved, and by using this embedding theorem, the model-checking, validity and satisfiability problems of ACTL are shown to be decidable. The embedding and decidability results indicate that we can reuse the existing CTL-based algorithms for model-checking, validity and satisfiability. This compatibility with CTL is regarded as a merit of ACTL.

The proposed agent operators in ACTL differ from the standard knowledge operators. In the following, we explain the proposed agent operators $\heartsuit_i$ and $\heartsuit_c$. The symbol ω is used to represent the set of natural numbers, and the symbol N is used to represent a fixed set $\{1, 2, ..., n\}$ of agents. The symbol K is used to represent the set $\{\heartsuit_i \mid i \in N\}$ of agent operators, and the symbol K^* is used to represent the set of all words of finite length of the alphabet K. For example, $\{\iota\alpha \mid \iota \in K^*\}$ denotes the set $\{\heartsuit_{i_1} \cdots \heartsuit_{i_k}\alpha \mid i_1, ..., i_k \in N, k \in \omega\}$. Greek lower-case letters ι and κ are used to represent any members of K^*. The symbol K^m is used to represent the set of all words of at most "m-length" of the alphabet K. Note that K^m is a subset of K^*, and is also finite. This finiteness condition on K^m is critical for obtaining an embedding theorem into CTL. Then, the characteristic axiom scheme for $\heartsuit_i$ and $\heartsuit_c$ is: $\heartsuit_c\alpha \leftrightarrow \bigwedge\{\iota\alpha \mid \iota \in K^m\}$. This axiom scheme corresponds to a "m-bounded" version of the so-called *iterative interpretation of common knowledge*: $\heartsuit_c\alpha \leftrightarrow \bigwedge\{\iota\alpha \mid \iota \in K^*\}$, which is obtained from the m-bounded version by replacing K^m with K^*. If we read $\heartsuit_i\alpha$ as "agent i has information α," then we can understand $\heartsuit_c\alpha$ as "α is finitely approximated common (or group) information of agents." In order to formalize these operators, we need to introduce an *agents-indexed Kripke structure*, which has agents-indexed satisfaction relations $\models^\iota$ $(\iota \in K^*)$.

Some related works are reviewed below. There are some agents-based or knowledge-based approaches to model checking. Some agents-based model checkers have successfully been developed [8, 4, 6, 5]. For example, an approach to model checking for the *modal logic of knowledge and linear-time in distributed systems with perfect recall* was established by van der Meyden and Shilov [8]. They showed that some model checking problems with or without a common knowledge operator are undecidable or PSPACE-complete. There are some approaches to cooperate CTL with knowledge or multi-agent operators. A multi-agent extension ALT of CTL was introduced by Alur et al. [1], and an epistemic extension ATEL of ALT was studied by van der Hoek and Wooldridge [7]. Some epistemic extensions of CTL were studied by van der Meyden and Wong [9]. In particular, the model checking complexity for the logic CKB_m, which has a similar bounded setting to the common knowledge operator in ACTL, was shown to be EXPTIME-complete at least for systems without perfect recall.

2 Agents-Indexed Computation Tree Logic

Let n be a fixed positive integer. Then, the symbol A is used to represent the set $\{1, 2, ..., n\}$ of agents. *Formulas* of ACTL are constructed from countable atomic formulas, $\rightarrow$ (implication) $\wedge$ (conjunction), $\vee$ (disjunction), $\neg$ (negation), $\heartsuit_i$ $(i \in A)$ (information or knowledge, "agent i knows"), $\heartsuit_c$ (bounded-depth common information or knowledge), X (next), G (globally), F (eventually), U (until), A (all computation paths) and E (some computation path). The symbols X, G, F and U are called *temporal operators*, and the symbols A and E are called *path quantifiers*. The symbol ATOM is used to denote the set of atomic formulas. An expression $A \equiv B$ is used to denote the syntactical identity between A and B.

Definition 1 *Formulas α are defined by the following grammar, assuming $p \in$ ATOM and $i \in A$:*

$$\alpha ::= p \mid \alpha{\rightarrow}\alpha \mid \alpha{\wedge}\alpha \mid \alpha{\vee}\alpha \mid \neg\alpha \mid \heartsuit_i\alpha \mid \heartsuit_c\alpha \mid \mathrm{AX}\alpha \mid \mathrm{EX}\alpha \mid \mathrm{AG}\alpha \mid \mathrm{EG}\alpha \mid \mathrm{AF}\alpha \mid \mathrm{EF}\alpha \mid \mathrm{A}(\alpha\mathrm{U}\alpha) \mid \mathrm{E}(\alpha\mathrm{U}\alpha).$$

Note that pairs of symbols like AG and EU are indivisible, and that the symbols $\mathrm{X}, \mathrm{G}, \mathrm{F}$ and U cannot occur without being preceded by an A or an E. Similarly, every A or E must have one of X, G, F and U to accompany it. Some operators are redundant as those in CTL, because some operators can be obtained by the other operators (e.g., $\mathrm{AG}\alpha := \neg\mathrm{EF}\neg\alpha$).

The symbol K is used to represent the set $\{\heartsuit_i \mid i \in A\}$, and the symbol K^* is used to represent the set of all words of finite length of the alphabet K. For example, $\{\iota\alpha \mid \iota \in K^*\}$ denotes the set $\{\heartsuit_{i_1} \cdots \heartsuit_{i_k}\alpha \mid i_1, ..., i_k \in A, k \in \omega\}$. Remark that K^* includes $\emptyset$ and hence $\{\iota\alpha \mid \iota \in K^*\}$ includes α. Greek lower-case letters ι and κ are used to denote any members of K^*. The symbol K^m is used to represent the set of all words of at most *m-length* of the alphabet K. Note that K^m is finite. In the following discussion, the number m of K^m is fixed as a certain positive integer.

Definition 2 *A structure $\langle S, S_0, R, \{L^\iota\}_{\iota\in K^*}\rangle$ is called an* agents-indexed Kripke structure *if:*

1. *S is the set of states,*
2. *S_0 is a set of initial states and $S_0 \subseteq S$,*
3. *R is a binary relation on S which satisfies the condition: $\forall s \in S\ \exists s' \in S\ [(s, s') \in R]$,*
4. *L^ι $(\iota \in K^*)$ are functions from S to the power set of a nonempty subset* AT *of* ATOM.

A path *in an agents-indexed Kripke structure is an infinite sequence of states, $\pi = s_0, s_1, s_2, ...$ such that $\forall i \geq 0\ [(s_i, s_{i+1}) \in R]$.*

The logic ACTL is then defined as an agents-indexed Kripke structure with satisfaction relations $\models^\iota$ $(\iota \in K^*)$.

Definition 3 *Let* AT *be a nonempty subset of* ATOM. Satisfaction relations $\models^\iota$ *$(\iota \in K^*)$ on an agents-indexed Kripke structure $M = \langle S, S_0, R, \{L^\iota\}_{\iota\in K^*}\rangle$ are defined as follows (s represents a state in S):*

1. *for any $p \in \mathrm{AT}$, $M, s \models^\iota p$ iff $p \in L^\iota(s)$,*
2. *$M, s \models^\iota \alpha_1 \rightarrow \alpha_2$ iff $M, s \models^\iota \alpha_1$ implies $M, s \models^\iota \alpha_2$,*
3. *$M, s \models^\iota \alpha_1 \wedge \alpha_2$ iff $M, s \models^\iota \alpha_1$ and $M, s \models^\iota \alpha_2$,*
4. *$M, s \models^\iota \alpha_1 \vee \alpha_2$ iff $M, s \models^\iota \alpha_1$ or $M, s \models^\iota \alpha_2$,*
5. *$M, s \models^\iota \neg\alpha_1$ iff not-$[M, s \models^\iota \alpha_1]$,*
6. *for any $i \in A$, $M, s \models^\iota \heartsuit_i\alpha$ iff $M, s \models^{\iota\heartsuit_i} \alpha$,*
7. *$M, s \models^\iota \heartsuit_c\alpha$ iff $M, s \models^{\iota\kappa} \alpha$ for all $\kappa \in K^m$,*
8. *$M, s \models^\iota \mathrm{AX}\alpha$ iff $\forall s_1 \in S\ \ [(s, s_1) \in R$ implies $M, s_1 \models^\iota \alpha]$,*
9. *$M, s \models^\iota \mathrm{EX}\alpha$ iff $\exists s_1 \in S\ \ [(s, s_1) \in R$ and $M, s_1 \models^\iota \alpha]$,*
10. *$M, s \models^\iota \mathrm{AG}\alpha$ iff for all paths $\pi \equiv s_0, s_1, s_2, ...$, where $s \equiv s_0$, and all states s_i along π, we have $M, s_i \models^\iota \alpha$,*
11. *$M, s \models^\iota \mathrm{EG}\alpha$ iff there is a path $\pi \equiv s_0, s_1, s_2, ...$, where $s \equiv s_0$, and for all states s_i along π, we have $M, s_i \models^\iota \alpha$,*

12. *$M, s \models^{\iota}$ AFα iff for all paths $\pi \equiv s_0, s_1, s_2, ...$, where $s \equiv s_0$, there is a state s_i along π such that $M, s_i \models^{\iota} \alpha$,*
13. *$M, s \models^{\iota}$ EFα iff there is a path $\pi \equiv s_0, s_1, s_2, ...$, where $s \equiv s_0$, and for some state s_i along π, we have $M, s_i \models^{\iota} \alpha$,*
14. *$M, s \models^{\iota}$ A$(\alpha_1 \mathrm{U} \alpha_2)$ iff for all paths $\pi \equiv s_0, s_1, s_2, ...$, where $s \equiv s_0$, there is a state s_k along π such that $[(M, s_k \models^{\iota} \alpha_2)$ and $\forall j$ $(0 \leq j < k$ implies $M, s_j \models^{\iota} \alpha_1)]$,*
15. *$M, s \models^{\iota}$ E$(\alpha_1 \mathrm{U} \alpha_2)$ iff there is a path $\pi \equiv s_0, s_1, s_2, ...$, where $s \equiv s_0$, and for some state s_k along π, we have $[(M, s_k \models^{\iota} \alpha_2)$ and $\forall j$ $(0 \leq j < k$ implies $M, s_j \models^{\iota} \alpha_1)]$.*

Definition 4 *A formula α is* valid *(*satisfiable*) in* ACTL *if and only if $M, s \models^{\emptyset} \alpha$ holds for any (some) agents-indexed Kripke structure $M = \langle S, S_0, R, \{L^{\iota}\}_{\iota \in K^*} \rangle$, any (some) $s \in S$, and any (some) satisfaction relations $\models^{\iota}$ $(\iota \in K^*)$ on M.*

Definition 5 *Let M be an agents-indexed Kripke structure $\langle S, S_0, R, \{L^{\iota}\}_{\iota \in K^*} \rangle$ for* ACTL*, and $\models^{\iota}$ $(\iota \in K^*)$ be satisfaction relations on M. Then, the* model checking problem *of* ACTL *is defined by: for any formula α, find the set $\{s \in S \mid M, s \models^{\emptyset} \alpha\}$.*

Definition 6 (CTL) *A* Kripke structure *for* CTL *is a structure $\langle S, S_0, R, L \rangle$ such that*

1. *S, S_0 and R have the same conditions as in Definition 2*
2. *L is a function from S to the power set of a nonempty subset* AT *of* ATOM.

A satisfaction relation *$\models$ on a Kripke structure $M = \langle S, S_0, R, L \rangle$ for* CTL *is defined by the same conditions 1–5 and 8–15 as in Definition 3 by deleting the superscript ι. The validity, satisfiability and model-checking problems for* CTL *are defined as usual.*

3 Embedding and Decidability

Definition 7 *Let* AT *be a non-empty subset of* ATOM*, and* AT^{ι} *$(\iota \in K^*)$ be the sets $\{p^{\iota} \mid p \in \mathrm{AT}^{\iota}\}$ of atomic formulas where $p^{\emptyset} := p$ (i.e., $\mathrm{AT}^{\emptyset} := \mathrm{AT}$). The language $\mathcal{L}^A$ (the set of formulas) of* ACTL *is defined using* AT*, $\heartsuit_i$ $(i \in A)$, $\heartsuit_c$, $\neg, \rightarrow, \wedge, \vee$, X, F, G, U, A and E. The language $\mathcal{L}$ of* CTL *is obtained from $\mathcal{L}^A$ by adding $\bigcup_{\iota \in K^*} \mathrm{AT}^{\iota}$ and deleting $\{\heartsuit_i, \heartsuit_c\}$. A mapping f from $\mathcal{L}^A$ to $\mathcal{L}$ is defined by:*

1. *for any $p \in \mathrm{AT}$, $f(\iota p) := p^{\iota} \in \mathrm{AT}^{\iota}$, esp., $f(p) := p$,*
2. *$f(\iota(\alpha \circ \beta)) := f(\iota\alpha) \circ f(\iota\beta)$ where $\circ \in \{\wedge, \vee, \rightarrow\}$,*
3. *$f(\iota\dagger\alpha) := \dagger f(\iota\alpha)$ where $\dagger \in \{\neg, \mathrm{AX}, \mathrm{EX}, \mathrm{AG}, \mathrm{EG}, \mathrm{AF}, \mathrm{EF}\}$,*
4. *$f(\iota\dagger(\alpha \mathrm{U} \beta))) := \dagger(f(\iota\alpha) \mathrm{U} f(\iota\beta))$ where $\dagger \in \{\mathrm{A}, \mathrm{E}\}$,*
5. *$f(\iota\heartsuit_c\alpha) := \bigwedge\{f(\iota\kappa\alpha) \mid \kappa \in K^m\}$.*

Lemma 8 *Let f be the mapping defined in Definition 7. For any agents-indexed Kripke structure $M := \langle S, S_0, R, \{L^\iota\}_{\iota \in K^*}\rangle$ for* ACTL*, and any satisfaction relations $\models^\iota$ ($\iota \in K^*$) on M, we can construct a Kripke structure $N := \langle S, S_0, R, L\rangle$ for* CTL *and a satisfaction relation $\models$ on N such that for any formula α in $\mathcal{L}^A$ and any state s in S, $M, s \models^\iota \alpha$ iff $N, s \models f(\iota\alpha)$.*

Proof. Let AT be a nonempty subset of ATOM, and AT$^\iota$ be the sets $\{p^\iota \mid p \in \mathrm{AT}\}$ of atomic formulas. Suppose that M is an agents-indexed Kripke structure $\langle S, S_0, R, \{L^\iota\}_{\iota \in K^*}\rangle$ such that L^ι ($\iota \in K^*$) are functions from S to the power set of AT. Suppose that N is a Kripke structure $\langle S, S_0, R, L\rangle$ such that L is a function from S to the power set of $\bigcup_{\iota \in K^*} \mathrm{AT}^\iota$. Suppose moreover that for any $s \in S$ and any $p \in \mathrm{AT}$, $p \in L^\iota(s)$ iff $p^\iota \in L(s)$. Then, the claim is then proved by induction on the complexity of α.

- Base step:

Case $\alpha \equiv p \in \mathrm{AT}$: We obtain: $M, s \models^\iota p$ iff $p \in L^\iota(s)$ iff $p^\iota \in L(s)$ iff $N, s \models p^\iota$ iff $N, s \models f(\iota p)$ (by the definition of f).

- Induction step: We show some cases.

Case $\alpha \equiv \beta{\rightarrow}\gamma$: We obtain: $M, s \models^\iota \beta{\rightarrow}\gamma$ iff $M, s \models^\iota \beta$ implies $M, s \models^\iota \gamma$ iff $N, s \models f(\iota\beta)$ implies $N, s \models f(\iota\gamma)$ (by induction hypothesis) iff $N, s \models f(\iota\beta){\rightarrow}f(\iota\gamma)$ iff $N, s \models f(\iota(\beta{\rightarrow}\gamma))$ (by the definition of f).

Case $\alpha \equiv \heartsuit_i\beta$: We obtain: $M, s \models^\iota \heartsuit_i\beta$ iff $M, s \models^{\iota\heartsuit_i} \beta$ iff $N, s \models f(\iota\heartsuit_i\beta)$ (by induction hypothesis).

Case $\alpha \equiv \heartsuit_c\beta$: We obtain: $M, s \models^\iota \heartsuit_c\beta$ iff $M, s \models^{\iota\kappa} \beta$ for any $\kappa \in K^m$ iff $N, s \models f(\iota\kappa\beta)$ for any $\kappa \in K^m$ (by induction hypothesis) iff $N, s \models \bigwedge\{f(\iota\kappa\beta) \mid \kappa \in K^m\}$ iff $N, s \models f(\iota\heartsuit_c\beta)$ (by the definition of f).

Case $\alpha \equiv \mathrm{AX}\beta$: We obtain: $M, s \models^\iota \mathrm{AX}\beta$ iff $\forall s_1 \in S$ $[(s, s_1) \in R$ implies $M, s_1 \models^\iota \beta]$ iff $\forall s_1 \in S$ $[(s, s_1) \in R$ implies $N, s_1 \models f(\iota\beta)]$ (by induction hypothesis) iff $N, s \models \mathrm{AX}f(\iota\beta)$ iff $N, s \models f(\iota\mathrm{AX}\beta)$ (by the definition of f).

Case $\alpha \equiv \mathrm{A}(\beta\mathrm{U}\gamma)$: We obtain:

$M, s \models^\iota \mathrm{A}(\beta\mathrm{U}\gamma)$

iff for all paths $\pi \equiv s_0, s_1, s_2, ...$, where $s \equiv s_0$, there is a state s_k along π such that $[M, s_k \models^\iota \gamma$ and $\forall j[i \leq j < k$ implies $M, s_j \models^\iota \beta]$

iff for all paths $\pi \equiv s_0, s_1, s_2, ...$, where $s \equiv s_0$, there is a state s_k along π such that $[N, s_k \models f(\iota\gamma)$ and $\forall j[i \leq j < k$ implies $N, s_j \models f(\iota\beta)]$ (by induction hypothesis)

iff $N, s \models \mathrm{A}(f(\iota\beta)\mathrm{U}f(\iota\gamma))$

iff $N, s \models f(\iota\mathrm{A}(\beta\mathrm{U}\gamma))$ (by the definition of f).

Q.E.D.

Lemma 9 *Let f be the mapping defined in Definition 7. For any Kripke structure $N := \langle S, S_0, R, L\rangle$ for* CTL*, and any satisfaction relation $\models$ on N, we can construct an agents-indexed Kripke structure $M := \langle S, S_0, R, \{L^\iota\}_{\iota \in K^*}\rangle$ for* ACTL *and satisfaction relations $\models^\iota$ ($\iota \in K^*$) on M such that for any formula α in $\mathcal{L}^A$ and any state s in S, $N, s \models f(\iota\alpha)$ iff $M, s \models^\iota \alpha$.*

Proof. Similar to the proof of Lemma 8. **Q.E.D.**

Theorem 10 (Embedding) *Let f be the mapping defined in Definition 7. For any formula α in $\mathcal{L}^A$, α is valid in* ACTL *iff $f(\alpha)$ is valid in* CTL.

Proof. By Lemmas 8 and 9. **Q.E.D.**

Theorem 11 (Decidability) *The model-checking, validity and satisfiability problems of* ACTL *are decidable.*

Proof. By the mapping f defined in Definition 7, a formula α of ACTL can finitely be transformed into the corresponding formula $f(\alpha)$ of CTL. By Lemmas 8 and 9 and Theorem 10, the model-checking, validity and satisfiability problems for ACTL can be transformed into those of CTL. Since the model checking, validity and satisfiability problems for CTL are decidable, the problems for ACTL are also decidable. **Q.E.D.**

Acknowledgments. This research was partially supported by the Japanese Ministry of Education, Culture, Sports, Science and Technology, Grant-in-Aid for Young Scientists (B) 20700015.

References

1. R. Alur, T.A. Henzinger and O. Kupferman, Alternating-time temporal logic, Journal of the ACM 49 (5), pp. 672-713, 2002.
2. E.M. Clarke and E.A. Emerson, Design and synthesis of synchronization skeletons using branching time temporal logic, LNCS 131, pp. 52–71, 1981.
3. E.M. Clarke, O. Grumberg, and D.A. Peled, Model checking, The MIT Press, 1999.
4. P. Gammie and R. van der Meyden, MCK: Model checking the logic of knowledge, Proceedings of the 16th International Conference on Computer Aided Verification (CAV 2004), LNCS 3114, pp. 479–483, 2004.
5. M. Kacprzak, W. Nabialek, A. Niewiadomski, W. Penczek, A. Polroa, M. Szreter, B. Wozawa and A. Zbrzezny, VerICS 2007: A model checker for real-time and multi-agent systems, Fundamenta Informaticae 85 (1–4), pp. 313–328, 2008.
6. A. Lomuscio and F. Raimondi, A model checker for multi-agent systems, Proceedings of the 12th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2006), LNCS 3920, pp. 450–454, 2006.
7. W. van der Hoek and M. Wooldridge, Cooperation, knowledge, and time: Alternating-time temporal epistemic logic and its applications, Studia Logica 75 (1), pp. 125-157, 2003.
8. R. van der Meyden and N.V. Shilov, Model checking knowledge and time in systems with perfect recall (extended abstract), Proceedings of the 19th Conference, Foundations of Software Technology and Theoretical Computer Science, LNCS 1738, pp. 432–445, 1999.
9. R. van der Meyden and Ka-shu Wong, Complete axiomatizations for reasoning about knowledge and branching time, Studia Logica 75 (1), pp. 93-123, 2003.