BRANCHING-TIME VERSUS LINEAR-TIME
A Cooperative and Feasible Approach

Norihiro Kamide
Waseda Institute for Advanced Study, 1-6-1 Nishi Waseda, Shinjuku-ku, Tokyo 169-8050, Japan

Keywords: Temporal reasoning, Branching-time formalism, Linear-time formalism, Computation tree logic, Linear-time temporal logic, Model checking.

Abstract: A new temporal logic called linear-time computation tree logic (LCTL) is obtained from computation tree logic (CTL) by adding some modified versions of the temporal operators of linear-time temporal logic (LTL). A theorem for embedding LCTL into CTL is proved. The model-checking, validity and satisfiability problems of LCTL are shown to be deterministic PTIME-complete, EXPTIME-complete and deterministic EXPTIME-complete, respectively.

1 INTRODUCTION

It is known that *computation tree logic* (CTL) (Clarke and Emerson, 1981) and *linear-time temporal logic* (LTL) (Pnueli, 1977) are the most useful temporal logics for verifying concurrent systems by *model checking* (Clarke et al., 1999). CTL has some feasible model checking algorithms, which are deterministic PTIME-complete (Emerson and Clarke, 1982), [1] but CTL cannot express some important temporal properties such as strong fairness. LTL can express almost all important temporal properties, but LTL has no feasible model-checking algorithms. The model-checking problem of LTL is indeed PSPACE-complete (Sistla and Clarke, 1985). Although CTL and LTL have been rivaled each other (Vardi, 2001), cooperating CTL and LTL is considered to be a good choice to obtain a more useful model checking tool. *Full computation-tree logic* (CTL*) (Emerson and Sistla, 1984; Emerson and Halpern, 1986) is known to be a result of cooperating CTL and LTL. However, the model-checking problem of CTL* is PSPACE-complete. This paper tries to obtain a cooperative and feasible approach to the traditional issue of "branching-time versus linear-time". The proposed logic in this paper includes CTL and subsumes some versions of the linear-time temporal operators of LTL (i.e., cooperative). The proposed logic also has the same complexity result as CTL model-checking (i.e., feasible).

The results of this paper are then summarized as follows. A new computation tree logic called *linear-time computation tree logic* (LCTL) is obtained from CTL by adding some bounded versions of the linear-time temporal operators of LTL. A theorem for embedding LCTL into CTL is proved. The model-checking, validity and satisfiability problems of LCTL are shown to be deterministic PTIME-complete, EXPTIME-complete and deterministic EXPTIME-complete, respectively. The embedding and decidability results indicate that we can reuse the existing CTL-based algorithms for model-checking, validity and satisfiability. This fact is regarded as an advantage of LCTL. The proposed bounded linear-time temporal operators, which are regarded as finite approximations of the usual linear-time temporal operators, have the central role for obtaining the complexity results. Although the standard LTL operators have an infinite (unbounded) time domain, i.e., the set ω of natural numbers, the proposed bounded operators have a *bounded time domain* which is restricted by a fixed positive integer l, i.e., the set $\omega_l := \{x \in \omega \mid x \leq l\}$. Despite this restriction, the proposed bounded operators can derive almost all the typical LTL axioms including the *time induction axiom*.

[1] By "feasible", we mean "computable in practice". There is a widespread opinion that PTIME computability is the correct mathematical model of feasible computation.

Kamide N. (2010).
BRANCHING-TIME VERSUS LINEAR-TIME - A Cooperative and Feasible Approach.
In *Proceedings of the 2nd International Conference on Agents and Artificial Intelligence - Artificial Intelligence*, pages 522-526
DOI: 10.5220/0002709205220526
Copyright © SciTePress

2 LINEAR-TIME COMPUTATION TREE LOGIC

Formulas of LCTL are constructed from countably many atomic formulas, $\to$ (implication) $\wedge$ (conjunction), $\vee$ (disjunction), $\neg$ (negation), X (next), G (globally), F (eventually), U (until), X_L (linear next), G_L (linear globally), F_L (linear eventually), A (all computation paths) and E (some computation path) where X_L, G_L and F_L are based on a bounded time domain. The symbols X, G, F, U, X_L, G_L and F_L 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 2.1 *Formulas* α *are defined by the following grammar, assuming* $p \in$ ATOM*:*

$$\begin{aligned}\alpha ::= {} & p \mid \alpha{\to}\alpha \mid \alpha\wedge\alpha \mid \alpha\vee\alpha \mid \neg\alpha \mid \\ & \mathrm{X}_L\alpha \mid \mathrm{G}_L\alpha \mid \mathrm{F}_L\alpha \mid \mathrm{AX}\alpha \mid \mathrm{EX}\alpha \mid \mathrm{AG}\alpha \mid \\ & \mathrm{EG}\alpha \mid \mathrm{AF}\alpha \mid \mathrm{EF}\alpha \mid \mathrm{A}(\alpha\mathrm{U}\alpha) \mid \mathrm{E}(\alpha\mathrm{U}\alpha).\end{aligned}$$

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 ω is used to represent the set of natural numbers. Lower-case letters i, j, k, m and n are sometimes used to denote any natural numbers. An expression $\mathrm{X}_L^m\alpha$ for any $m \in \omega$ is defined inductively by $\mathrm{X}_L^0\alpha \equiv \alpha$ and $\mathrm{X}_L^{n+1}\alpha \equiv \mathrm{X}_L\mathrm{X}_L^n\alpha$. The symbols $\leq$ and $\geq$ are used to represent a linear order on ω. The symbol ω_l is used to represent the set $\{i \in \omega \mid i \leq l\}$. In the following discussion, the number l is fixed as a certain positive integer.

Definition 2.2 *A structure* $\langle S, S_0, R, \{L^m\}_{m\in\omega}\rangle$ *is called a* time-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^m $(m \in \omega)$ are functions from S to the power set of a nonempty subset* AT *of* ATOM*.*

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

The logic LCTL is then defined as a time-indexed Kripke structure with satisfaction relations $\models^m$ $(m \in \omega)$.

Definition 2.3 *Let* AT *be a nonempty subset of* ATOM. Satisfaction relations $\models^m$ $(m \in \omega)$ *on a time-indexed Kripke structure $M = \langle S, S_0, R, \{L^m\}_{m\in\omega}\rangle$ are defined inductively as follows (s represents a state in S):*

1. *for any $p \in$ AT, $M, s \models^m p$ iff $p \in L^m(s)$,*
2. *$M, s \models^m \alpha_1{\to}\alpha_2$ iff $M, s \models^m \alpha_1$ implies $M, s \models^m \alpha_2$,*
3. *$M, s \models^m \alpha_1 \wedge \alpha_2$ iff $M, s \models^m \alpha_1$ and $M, s \models^m \alpha_2$,*
4. *$M, s \models^m \alpha_1 \vee \alpha_2$ iff $M, s \models^m \alpha_1$ or $M, s \models^m \alpha_2$,*
5. *$M, s \models^m \neg\alpha_1$ iff not-$[M, s \models^m \alpha_1]$,*
6. *for any $m \leq l - 1$, $M, s \models^m \mathrm{X}_L\alpha$ iff $M, s \models^{m+1} \alpha$,*
7. *for any $m \geq l$, $M, s \models^m \mathrm{X}_L\alpha$ iff $M, s \models^l \alpha$,*
8. *for any $n \in \omega$, $M, s \models^{l+n} \alpha$ iff $M, s \models^l \alpha$,*
9. *$M, s \models^m \mathrm{G}_L\alpha$ iff $M, s \models^{m+n} \alpha$ for all $n \in \omega_l$,*
10. *$M, s \models^m \mathrm{F}_L\alpha$ iff $M, s \models^{m+n} \alpha$ for some $n \in \omega_l$,*
11. *$M, s \models^m \mathrm{AX}\alpha$ iff $\forall s_1 \in S$ $[(s, s_1) \in R$ implies $M, s_1 \models^m \alpha]$,*
12. *$M, s \models^m \mathrm{EX}\alpha$ iff $\exists s_1 \in S$ $[(s, s_1) \in R$ and $M, s_1 \models^m \alpha]$,*
13. *$M, s \models^m \mathrm{AG}\alpha$ iff for all paths $\pi \equiv s_0, s_1, s_2, \ldots$, where $s \equiv s_0$, and all states s_i along π, we have $M, s_i \models^m \alpha$,*
14. *$M, s \models^m \mathrm{EG}\alpha$ iff there is a path $\pi \equiv s_0, s_1, s_2, \ldots$, where $s \equiv s_0$, and for all states s_i along π, we have $M, s_i \models^m \alpha$,*
15. *$M, s \models^m \mathrm{AF}\alpha$ iff for all paths $\pi \equiv s_0, s_1, s_2, \ldots$, where $s \equiv s_0$, there is a state s_i along π such that $M, s_i \models^m \alpha$,*
16. *$M, s \models^m \mathrm{EF}\alpha$ iff there is a path $\pi \equiv s_0, s_1, s_2, \ldots$, where $s \equiv s_0$, and for some state s_i along π, we have $M, s_i \models^m \alpha$,*
17. *$M, s \models^m \mathrm{A}(\alpha_1\mathrm{U}\alpha_2)$ iff for all paths $\pi \equiv s_0, s_1, s_2, \ldots$, where $s \equiv s_0$, there is a state s_k along π such that $[(M, s_k \models^m \alpha_2)$ and $\forall j\ (0 \leq j < k$ implies $M, s_j \models^m \alpha_1)]$,*
18. *$M, s \models^m \mathrm{E}(\alpha_1\mathrm{U}\alpha_2)$ iff there is a path $\pi \equiv s_0, s_1, s_2, \ldots$, where $s \equiv s_0$, and for some state s_k along π, we have $[(M, s_k \models^m \alpha_2)$ and $\forall j\ (0 \leq j < k$ implies $M, s_j \models^m \alpha_1)]$.*

We can naturally consider the unbounded version LCTL_ω which is obtained from LCTL by deleting the conditions 7 and 8 and replacing the conditions 6, 9 and 10 by the standard conditions:

6′. $M, s \models^m \mathrm{X}_L\alpha$ iff $M, s \models^{m+1} \alpha$,

9′. $M, s \models^m \mathrm{G}_L\alpha$ iff $M, s \models^{m+n} \alpha$ for all $n \in \omega$,

10′. $M, s \models^m \mathrm{F}_L\alpha$ iff $M, s \models^{m+n} \alpha$ for some $n \in \omega$.

However, the decidability of validity, satisfiability and model-checking problems for LCTL$_\omega$ cannot be shown using the proposed embedding-based method. The logic LCTL$_\omega$ is embeddable into the infinitary version CTL$_\omega$ which is obtained from CTL by adding the infinitary conjunction and disjunction connectives $\bigwedge$ and $\bigvee$. But, logics with $\bigwedge$ and $\bigvee$ are known to be undecidable, and hence such an embedding result cannot imply the decidability.

Definition 2.4 *A formula* α *is* valid *(*satisfiable*) in* LCTL *if and only if* $M, s \models^0 \alpha$ *holds for any (some) time-indexed Kripke structure* $M = \langle S, S_0, R, \{L^m\}_{m\in\omega}\rangle$, *any (some)* $s \in S$, *and any (some) satisfaction relations* $\models^m$ $(m \in \omega)$ *on* M.

Definition 2.5 *Let* M *be a time-indexed Kripke structure* $\langle S, S_0, R, \{L^m\}_{m\in\omega}\rangle$ *for* LCTL, *and* $\models^m$ $(m \in \omega)$ *be satisfaction relations on* M. *Then, the* model checking problem *of* LCTL *is defined by: for any formula* α, *find the set* $\{s \in S \mid M, s \models^0 \alpha\}$.

Let C be a finite set of formulas. Then, expressions $\bigwedge C$ and $\bigvee C$ represent the conjunction and disjunction of all elements of C, respectively. An expression $\alpha \leftrightarrow \beta$ is used to represent $(\alpha\rightarrow\beta) \wedge (\beta\rightarrow\alpha)$.

Proposition 2.6 *The following formulas are valid in* LCTL*: for any formulas* α *and* β,

1. $\mathrm{X}_L(\alpha \circ \beta) \leftrightarrow \mathrm{X}_L\alpha \circ \mathrm{X}_L\beta$ *where* $\circ \in \{\rightarrow, \wedge, \vee\}$,
2. $\mathrm{X}_L(\neg\alpha) \leftrightarrow \neg(\mathrm{X}_L\alpha)$,
3. $\mathrm{G}_L\alpha\rightarrow\alpha$,
4. $\mathrm{G}_L\alpha\rightarrow\mathrm{X}_L\alpha$,
5. $\mathrm{G}_L\alpha\rightarrow\mathrm{X}_L\mathrm{G}_L\alpha$,
6. $\mathrm{G}_L\alpha\rightarrow\mathrm{G}_L\mathrm{G}_L\alpha$,
7. $\alpha \wedge \mathrm{G}_L(\alpha\rightarrow\mathrm{X}_L\alpha)\rightarrow\mathrm{G}_L\alpha$ *(time induction),*
8. *for any* $n \in \omega$, $\mathrm{X}_L^{l+n}\alpha \leftrightarrow \mathrm{X}_L^{l}\alpha$,
9. $\mathrm{G}_L\alpha \leftrightarrow \bigwedge\{\mathrm{X}_L^n\alpha \mid n \in \omega_l\}$,
10. $\mathrm{F}_L\alpha \leftrightarrow \bigvee\{\mathrm{X}_L^n\alpha \mid n \in \omega_l\}$.

Note that the formula 8 in in Proposition 2.6 means that the nesting of X is bounded by l. Note also that the formulas 9 and 10 in Proposition 2.6 mean that G_L and F_L are finite approximations of the standard linear-time temporal operators.

Definition 2.7 *A* Kripke structure *for* CTL *is a structure* $\langle S, S_0, R, L\rangle$ *such that*

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 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 9–16 as in Definition 2.3 by deleting the superscript "m". The validity, satisfiability and model-checking problems for* CTL *are defined similarly as those for* LCTL.

Remark that $\models^m$ of LCTL includes $\models$ of CTL, and hence LCTL is an extension of CTL.

3 EMBEDDING AND COMPLEXITY

Definition 3.1 *Let* AT *be a non-empty subset of* ATOM, *and* AT^m $(m \in \omega)$ *be the sets* $\{p^m \mid p \in \mathrm{AT}^m\}$ *of atomic formulas where* $p^0 := p$ *(i.e.,* $\mathrm{AT}^0 := \mathrm{AT}$*). The language* $\mathcal{L}^L$ *(the set of formulas) of* LCTL *is defined using* AT, X_L, G_L, F_L, $\neg, \rightarrow, \wedge, \vee$, X, F, G, U, A *and* E. *The language* $\mathcal{L}$ *of* CTL *is obtained from* $\mathcal{L}^L$ *by adding* $\bigcup_{m\in\omega} \mathrm{AT}^m$ *and deleting* $\{\mathrm{X}_L, \mathrm{G}_L, \mathrm{F}_L\}$.

A mapping f *from* $\mathcal{L}^L$ *to* $\mathcal{L}$ *is defined inductively by:*

1. *for any* $p \in \mathrm{AT}$, $f(\mathrm{X}_L^m p) := p^m \in \mathrm{AT}^m$, *esp.,* $f(p) := p$,
2. $f(\mathrm{X}_L^m(\alpha \sharp \beta)) := f(\mathrm{X}_L^m\alpha) \sharp f(\mathrm{X}_L^m\beta)$ *where* $\sharp \in \{\wedge, \vee, \rightarrow\}$,
3. $f(\mathrm{X}_L^m\sharp\alpha) := \sharp f(\mathrm{X}_L^m\alpha)$ *where* $\sharp \in \{\neg, \mathrm{AX}, \mathrm{EX}, \mathrm{AG}, \mathrm{EG}, \mathrm{AF}, \mathrm{EF}\}$,
4. $f(\mathrm{X}_L^m\sharp(\alpha\mathrm{U}\beta))) := \sharp(f(\mathrm{X}_L^m\alpha)\mathrm{U}f(\mathrm{X}_L^m\beta))$ *where* $\sharp \in \{\mathrm{A}, \mathrm{E}\}$,
5. $f(\mathrm{X}_L^m\mathrm{G}_L\alpha) := \bigwedge\{f(\mathrm{X}_L^{m+n}\alpha) \mid n \in \omega_l\}$,
6. $f(\mathrm{X}_L^m\mathrm{F}_L\alpha) := \bigvee\{f(\mathrm{X}_L^{m+n}\alpha) \mid n \in \omega_l\}$.

Lemma 3.2 *Let* f *be the mapping defined in Definition 3.1. For any time-indexed Kripke structure* $M := \langle S, S_0, R, \{L^m\}_{m\in\omega}\rangle$ *for* LCTL, *and any satisfaction relations* $\models^m$ $(m \in \omega)$ *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* $\mathcal{L}^L$ *and any state* s *in* S,

$$M, s \models^m \alpha \textit{ iff } N, s \models f(\mathrm{X}_L^m\alpha).$$

Proof. Let AT be a nonempty subset of ATOM, and AT^m be the sets $\{p^m \mid p \in \mathrm{AT}\}$ of atomic formulas. Suppose that M is a time-indexed Kripke structure $\langle S, S_0, R, \{L^m\}_{m\in\omega}\rangle$ such that

L^m $(m \in \omega)$ 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 $\bigcup_{m\in\omega} \mathrm{AT}^m$.

Suppose moreover that for any $s \in S$ and any $p \in \mathrm{AT}$,

$p \in L^m(s)$ iff $p^m \in L(s)$.

The lemma is then proved by induction on the complexity of α.

• Base step:

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

• Induction step:

Case $\alpha \equiv \beta \wedge \gamma$: We obtain: $M,s \models^m \beta \wedge \gamma$ iff $M,s \models^m \beta$ and $M,s \models^m \gamma$ iff $N,s \models f(\mathrm{X}_L^m\beta)$ and $N,s \models f(\mathrm{X}_L^m\gamma)$ (by induction hypothesis) iff $N,s \models f(\mathrm{X}_L^m\beta) \wedge f(\mathrm{X}_L^m\gamma)$ iff $N,s \models f(\mathrm{X}_L^m(\beta\wedge\gamma))$ (by the definition of f).

Case $\alpha \equiv \beta \vee \gamma$: Similar to Case $\alpha \equiv \beta \wedge \gamma$.

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

Case $\alpha \equiv \neg\beta$: We obtain: $M,s \models^m \neg\beta$ iff not-$[M,s \models^m \beta]$ iff not-$[N,s \models f(\mathrm{X}_L^m\beta)]$ (by induction hypothesis) iff $N,s \models \neg f(\mathrm{X}_L^m\beta)$ iff $N,s \models f(\mathrm{X}_L^m\neg\beta)$ (by the definition of f).

Case $\alpha \equiv \mathrm{X}_L\beta$:

Subcase $m \leq l-1$: We obtain: $M,s \models^m \mathrm{X}_L\beta$ iff $M,s \models^{m+1} \beta$ iff $N,s \models f(\mathrm{X}_L^{m+1}\beta)$ (by induction hypothesis).

Subcase $m \geq l$: We obtain: $M,s \models^m \mathrm{X}_L\beta$ iff $M,s \models^l \beta$ iff $M,s \models^{m+1} \beta$ iff $N,s \models f(\mathrm{X}_L^{m+1}\beta)$ (by induction hypothesis).

Case $\alpha \equiv \mathrm{G}_L\beta$: We obtain: $M,s \models^m \mathrm{G}_L\beta$ iff $M,s \models^{m+n} \beta$ for any $n \in \omega_l$ iff $N,s \models f(\mathrm{X}_L^{m+n}\beta)$ for any $n \in \omega_l$ (by induction hypothesis) iff $N,s \models \bigwedge\{f(\mathrm{X}_L^{m+n}\beta) \mid n \in \omega_l\}$ iff $N,s \models f(\mathrm{X}_L^m\mathrm{G}_L\beta)$ (by the definition of f).

Case $\alpha \equiv \mathrm{F}_L\beta$: Similar to Case $\alpha \equiv \mathrm{G}_L\beta$.

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

Case $\alpha \equiv \mathrm{EX}\beta$: Similar to Case $\alpha \equiv \mathrm{AX}\beta$.

Case $\alpha \equiv \mathrm{AG}\beta$: We obtain:

$M,s \models^m \mathrm{AG}\beta$

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 $M,s_i \models^m \beta$

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 $N,s_i \models f(\mathrm{X}_L^m\beta)$ (by induction hypothesis)

iff $N,s \models \mathrm{AG}f(\mathrm{X}_L^m\beta)$

iff $N,s \models f(\mathrm{X}_L^m\mathrm{AG}\beta)$ (by the definition of f).

Cases $\alpha \equiv \mathrm{EG}\beta$, $\alpha \equiv \mathrm{AF}\beta$ and $\alpha \equiv \mathrm{EF}\beta$: Similar to Case $\alpha \equiv \mathrm{AG}\beta$.

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

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

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 $[M,s_k \models^m \gamma$ and $\forall j[i \leq j < k$ implies $M,s_j \models^m \beta]$

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 $[N,s_k \models f(\mathrm{X}_L^m\gamma)$ and $\forall j[i \leq j < k$ implies $N,s_j \models f(\mathrm{X}_L^m\beta)]$ (by induction hypothesis)

iff $N,s \models \mathrm{A}(f(\mathrm{X}_L^m\beta)\mathrm{U}f(\mathrm{X}_L^m\gamma))$

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

Case $\alpha \equiv \mathrm{E}(\beta\mathrm{U}\gamma)$: Similar to Case $\alpha \equiv \mathrm{A}(\beta\mathrm{U}\gamma)$. ▮

Lemma 3.3 *Let f be the mapping defined in Definition 3.1. For any Kripke structure $N := \langle S,S_0,R,L\rangle$ for* CTL*, and any satisfaction relation $\models$ on N, we can construct a time-indexed Kripke structure $M := \langle S,S_0,R,\{L^m\}_{m\in\omega}\rangle$ for* LCTL *and satisfaction relations $\models^m$ $(m \in \omega)$ on M such that for any formula α in $\mathcal{L}^L$ and any state s in S,*

$N,s \models f(\mathrm{X}_L^m\alpha)$ *iff* $M,s \models^m \alpha$.

Proof. Similar to the proof of Lemma 3.2. ▮

Theorem 3.4 (Embedding) *Let f be the mapping defined in Definition 3.1. For any formula α, α is valid (satisfiable) in* LCTL *iff $f(\alpha)$ is valid (satisfiable) in* CTL.

Proof. By Lemmas 3.2 and 3.3. ▮

We then obtain the main theorem of this paper.

Theorem 3.5 (Complexity) *The model-checking, validity and satisfiability problems for* LCTL *are deterministic PTIME-complete, EXPTIME-complete and deterministic EXPTIME-complete, respectively.*

Proof. By the mapping f defined in Definition 3.1, a formula α of LCTL can finitely be transformed into the corresponding formula $f(\alpha)$ of CTL. By Lemmas 3.2 and 3.3 and Theorem 3.4, the model checking, validity and satisfiability problems for LCTL can be transformed into those of CTL. Since the model checking, validity and satisfiability problems for CTL are decidable, the problems for LCTL are also decidable. Since the mapping f from LCTL into CTL is a polynomial-time reduction, the complexity results for LCTL become the same results as CTL, i.e., the model-checking, validity and satisfiability problems for LCTL are deterministic PTIME-complete, EXPTIME-complete and deterministic EXPTIME-complete, respectively. ∎

4 CONCLUDING REMARKS

In this paper, a new logic, linear-time computation tree logic (LCTL), was introduced by "cooperating" CTL and LTL, and the deterministic PTIME-completeness (i.e., the existence of "feasible" algorithms) of the LCTL model-checking problem was shown. It was thus shown that there is a cooperative and feasible approach to the traditional issue of "branching-time versus linear-time".

In the following, we give some remarks on the idea of bounding time and on the concept of combining logics.

To restrict the time domain of the LTL operators is not a new idea. Such an idea was discussed in (Biere et al., 2003; Cerrito et al., 1999; Cerrito and Mayer, 1998; Hodkinson et al., 2000). For example, by using and introducing a bounded time domain and the notion of bounded validity in a semantics, *bounded tableaux calculi* (with temporal constraints) for propositional and first-order LTLs were introduced by Cerrito, Mayer and Prand (Cerrito et al., 1999; Cerrito and Mayer, 1998). It is also known that to restrict the time domain is a technique to obtain a decidable or efficient fragment of first-order LTL (Hodkinson et al., 2000). Restricting the time domain implies not only some purely theoretical merits discussed above, but also some practical merits for describing temporal databases and planning specifications (Cerrito et al., 1999; Cerrito and Mayer, 1998), and for implementing an efficient model checking algorithm called *bounded model checking* (Biere et al., 2003). Such practical merits are due to the fact that there are problems in computer science and artificial intelligence where only a finite fragment of the time sequence is of interest (Cerrito et al., 1999).

As mentioned in (Sernadas and Sernadas, 2003), there are some general theories for various combined modal logics (Sernadas and Sernadas, 2003), including the theories of *fusion*, *product* and *fibring*. Various combined modal logics have been studied based on these theories. The proposed logic LCTL may be categorized by a fusion of CTL and a bounded-time version of LTL.

ACKNOWLEDGEMENTS

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

REFERENCES

Biere, A., Cimatti, A., Clarke, E., Strichman, O., and Zhu, Y. (2003). Bounded model checking. *Advances in Computers*, 58:118–149.

Cerrito, S. and Mayer, M. (1998). Bounded model search in linear temporal logic and its application to planning. In *Lecture Notes in Computer Science*, volume 1397, pages 124–140.

Cerrito, S., Mayer, M., and Prand, S. (1999). First order linear temporal logic over finite time structures. In *Lecture Notes in Computer Science*, volume 1705, pages 62–76.

Clarke, E. and Emerson, E. (1981). Design and synthesis of synchronization skeletons using branching time temporal logic. In *Lecture Notes in Computer Science*, volume 131, pages 52–71.

Clarke, E., Grumberg, O., and Peled, D. (1999). *Model checking*. The MIT Press.

Emerson, E. and Clarke, E. (1982). Using branching time temporal logic to synthesize synchronization skeletons. *Science of Computer Programming*, 2:241–266.

Emerson, E. and Halpern, J. (1986). "sometimes" and "not never" revisited: on branching versus linear time temporal logic. *Journal of the ACM*, 33 (1):151–178.

Emerson, E. and Sistla, P. (1984). Deciding full branching time logic. *Information and Control*, 61:175–201.

Hodkinson, I., Wolter, F., and Zakharyaschev, M. (2000). Decidable fragments of first-order temporal logics. *Annals of Pure and Applied Logic*, 106:85–134.

Pnueli, A. (1977). The temporal logic of programs. In *Proceedings of the 18th IEEE Symposium on Foundations of Computer Science*, pages 46–57.

Sernadas, A. and Sernadas, C. (2003). Combining logic systems: why, how, what for? *CIM Bulletin*, 15:9–14.

Sistla, A. and Clarke, E. (1985). The complexity of propositional linear temporal logic. *Journal of the ACM*, 32 (3):733–749.

Vardi, M. (2001). Branching vs. linear time: final showdown. In *Lecture Notes in Computer Science*, volume 2031, pages 1–22.