The Relative Completeness of a Version of CTL*

Hidetomo Machi[1] **Kohji Tomita**[2] **Chiharu Hosono**[1]

[1] University of Tsukuba, Japan
Email: {machi, hosono}@ailab.is.tsukuba.ac.jp
[2] National Institute of Advanced Industrial Science and Technology (AIST), Japan
Email: k.tomita@aist.go.jp

Abstract

We show the relative completeness of a version of CTL* over possibly infinite states. It is based on a fair discrete system, which is with strong and weak fairness. Validity of a CTL* formula is reduced to the validity of an assertion by eliminating the temporal operators and path quantifiers inductively. The idea is to represent the predicate describing that "a fair path begins with the current state" in the underlying assertion language using fixpoint operators.

1 Introduction

Temporal logic (Pnueli 1977) has been widely used for specification and verification of programs or reactive systems (Emerson 1990). There are many variants of temporal logics depending on choices in modeling such as linear time/branching time, propositional/predicate, and so on. Among them, a propositional branching time temporal logic (CTL, computational tree logic) (Clarke & Emerson 1981) has been successfully applied to model checking because of its computational efficiency. However, it requires a path quantifier to prefix only one temporal operator, and some temporal properties cannot be represented in CTL.

CTL* (Emerson & Halpern 1986, Clarke, Emerson & Sistla 1986) is a temporal logic combining CTL and LTL (linear time temporal logic). This is an expressive system permitting path quantifiers to prefix any combination of temporal operators. Much study has been done on the propositional case. As is well known, the validity of a propositional CTL* formula is decidable. It was shown that any formula of propositional CTL* can be translated into an equivalent expression in a first-order logic augmented with transitive closure (Immerman & Vardi 1997). Moreover, a complete axiomatization of CTL* was given by (Reynolds 2001), also for the propositional case. We are interested in the predicate case to verify CTL* properties of reactive systems over possibly infinite states.

A deductive proof system for (predicate) CTL* has been presented in (Pnueli & Kesten 2002), by extending a deductive proof system for (predicate) LTL in (Manna & Pnueli 1991). It is based on a fair discrete system, which is with strong and weak fairness, over possibly infinite states. Using sophisticated assertions, they gave a relatively complete deductive proof system for the verification of CTL* on an underlying assertion language which contains the predicate calculus augmented with fixpoint operators and finite sequences of data. In their proof system, however, it was necessary to devise intermediate assertions at one position.

In this paper, we give a proof of the relative completeness of CTL* over possibly infinite states following the approach by (Pnueli & Kesten 2002). They gave a BASIC-PATH rule that reduces the validity of a CTL* formula into the validity of a simpler formula, with less temporal operators by one, on an extended system. In addition to this, we show that the path quantifiers can be eliminated. The idea is to represent the predicate describing that "a fair path begins with the current state" in the underlying assertion language using fixpoint operators. Therefore, the validity of a CTL* formula is reduced to the validity of an assertion. Our proof proceeds algorithmically without requiring any intermediate assertion to be devised.

In the following, we show syntax and semantics in section 2, and prove relative completeness in section 3. A simple example is given in section 4. Section 5 concludes the paper.

2 Syntax and Semantics

In this section, we review necessary definitions from (Pnueli & Kesten 2002).

2.1 Computational Model

We take the fair discrete system (FDS) as a computational model. An FDS $\mathcal{D} : \langle V, \Theta, \rho, \mathcal{J}, \mathcal{C} \rangle$ consists of the following components.

- $V = \{u_1, \ldots, u_n\}$: A finite set of typed state variables over possibly infinite domains. We define a state s to be a type-consistent interpretation of V, assigning to each variable $u \in V$ a value $s[u]$ in its domain. We denote by Σ the set of all states.
- Θ: The initial condition. This is an assertion characterizing all the initial states.
- ρ: A transition relation. This is an assertion $\rho(V, V')$, relating a state $s \in \Sigma$ to its $\mathcal{D}$-successor $s' \in \Sigma$ by referring to both unprimed and primed versions of the state variables.
- $\mathcal{J} = \{J_1, \ldots, J_k\}$: A set of assertions expressing the justice (weak fairness) requirements.
- $\mathcal{C} = \{\langle p_1, q_1 \rangle, \ldots, \langle p_n, q_n \rangle\}$: A set of assertions expressing the compassion (strong fairness) requirements.

Let $\sigma : s_0, s_1, \ldots$ be a sequence of states, φ be an assertion, and $j \geq 0$ be a natural number. We refer

to a state which satisfies p as a p-state. We say that j is a φ-position of σ if s_j is a φ-state, i.e., $\varphi(s_j)$ holds. Let $\mathcal{D}$ be an FDS. We define a run of $\mathcal{D}$ to be a finite or infinite sequence of states $\sigma : s_0, s_1, \ldots$, satisfying the requirement of

- *Consecution*: For each $j = 0, 1, \ldots$, the state s_{j+1} is a $\mathcal{D}$-successor of the state s_j.

and such that it is either infinite, or terminates at a state s_k which has no $\mathcal{D}$-successors. A run is also called a path.

We denote by $runs(\mathcal{D})$ the set of all runs of $\mathcal{D}$. An infinite run of $\mathcal{D}$ is called fair if it satisfies the following:

- *Justice*: For each $J \in \mathcal{J}$, σ contains infinitely many J-positions.
- *Compassion*: For each $\langle p, q\rangle \in \mathcal{C}$, if σ contains infinitely many p-positions, it must also contain infinitely many q-positions.

We say that a fair run $\sigma : s_0, s_1, \ldots$ is a computation of $\mathcal{D}$ if it satisfies

- *Initiality*: s_0 is initial, i.e., $s_0 \models \Theta$.

2.2 Underlying Assertion Language

We assume an underlying assertion language $\mathcal{L}$, which contains the predicate calculus, and interpreted symbols for expressing the standard operations and relations over some concrete domains. We require that one of the domains is that of the integers, or another domain with similar expressive power, so that we can encode records (i.e., lists) of data elements, and lists of records. We are interested in records r of size $|V|$, which indicate states, and sequences π of them, which are finite sequences of states. We use the subscripted expression $\pi[i]$ to refer to the i-th element of π. For instance, a formula

$$\exists\pi(\forall i(0 \le i < |\pi| - 1 \rightarrow \rho(\pi[i], \pi[i+1]))),$$

describes that there exists a finite sequence of states satisfying consecution, and we simply denote it by

$$\exists s_0, \ldots, s_n(\forall i(0 \le i < n \rightarrow \rho(s_i, s_{i+1}))),$$

for readability.

Moreover, as is known, a first-order language is not adequate to express the assertions necessary for completeness, and we include in $\mathcal{L}$ also the fixpoint operators μ and ν following (Manna & Pnueli 1991) and (Pnueli & Kesten 2002). Let $Q(u_1, \ldots, u_r)$ be a predicate symbol of arity r, and $\Phi(Q, u_1, \ldots, u_r)$ be a formula whose free variables are a subset of $\{u_1, \ldots, u_r\}$, and that have only positive occurrences of Q, i.e., occurrences under an even number of negations. Then, the assertion language $\mathcal{L}$ contains formulas of the forms $\mu Q : \Phi$ and $\nu Q : \Phi$ for Q and Φ restricted as above. These are the least and greatest fixpoints of the fixpoint equation

$$Q(u_1, \ldots, u_r) \equiv \Phi(Q, u_1, \ldots, u_r),$$

respectively. These definitions are also written in the following form. Let Y be a function from 2^Σ to 2^Σ:

$$Y = \lambda Q.\lambda s.\, \Phi,$$

where s is a variable representing a state (i.e., a sequence $u_1, \ldots, u_r$). Then, the least and greatest fixpoints of Y are $\mu Q : \Phi$ and $\nu Q : \Phi$, respectively. Here, a predicate (i.e., a function from Σ to $\{true, false\}$) is identified with an element of the complete lattice defined on 2^Σ with inclusion as an ordering. The elements $\bot$ and $\top$ are ϕ and Σ, respectively. For instance, $Y(Q)(s)$ means that $s \in Y(Q)$. In the following, we use this representation of the fixpoint equations.

2.3 The Logic CTL*

A CTL* formula is constructed out of assertions (formulas over $\mathcal{L}$) to which we apply the boolean operators, temporal operators and path quantifiers. The basic temporal operators are $\bigcirc$(Next), $\mathcal{U}$(Until) and $\mathcal{W}$ (waiting for). Additional temporal operators may be defined by: $\Diamond p = \mathrm{T}\,\mathcal{U} p$ and $\Box p = \neg\Diamond\neg p$. The path quantifiers are E, A, E_f and A_f.

In the following, we present the syntax and semantics of the logic which is interpreted over the computation tree generated by an FDS.

Let $\pi : s_0, s_1, \ldots$ be a run of $\mathcal{D}$. Then, we write $\pi[0]$ to denote s_0, the first state in π and, for $j \ge 0$, we write $\pi[j..] = s_j, s_{j+1}, \ldots$ to denote the suffix of π obtained by omitting the first j states. If the path π is finite, we use $|\pi|$ to denote its length.

There are two types of sub-formulas in CTL*: state formulas that are interpreted over states, and path formulas that are interpreted over paths. The syntax of a CTL* formula is defined inductively as follows:
State formulas:

- Every assertion in $\mathcal{L}$ is a state formula.
- If p is a path formula, then Ep, Ap, $E_f p$ and $A_f p$ are state formulas.
- If p and q are state formulas then so are $p \vee q$, $p \wedge q$ and $\neg p$.

Path formulas:

- Every state formula is a path formula.
- If p and q are path formulas, then so are $p \vee q$, $p \wedge q$, $\neg p$, $\bigcirc p$, $p\,\mathcal{U} q$ and $p\,\mathcal{W} q$.

The formulas of CTL* are all the state formulas generated by the above rules.

The semantics of a CTL* formula p is defined with respect to an FDS $\mathcal{D}$ over the vocabulary of p. The semantics is defined inductively as follows.

State formulas are interpreted over states in $\mathcal{D}$. We define the notion of a CTL* formula p holding at a state s in $\mathcal{D}$, denoted $(\mathcal{D}, s) \models p$, as follows:

- For an assertion p,

  $$(\mathcal{D}, s) \models p \Leftrightarrow s \models p.$$

  That is, we evaluate p locally, using the interpretation given by s.

- For state formulas p and q,

  $$\begin{array}{lcl} (\mathcal{D}, s) \models p \vee q & \Leftrightarrow & (\mathcal{D}, s) \models p \text{ or } (\mathcal{D}, s) \models q, \\ (\mathcal{D}, s) \models p \wedge q & \Leftrightarrow & (\mathcal{D}, s) \models p \text{ and } (\mathcal{D}, s) \models q, \\ (\mathcal{D}, s) \models \neg p & \Leftrightarrow & (\mathcal{D}, s) \not\models p. \end{array}$$

- For a path formula p,

  $$\begin{array}{lcl} (\mathcal{D}, s) \models Ep & \Leftrightarrow & (\mathcal{D}, \pi) \models p \\ & & \text{for some path } \pi \in runs(\mathcal{D}) \\ & & \text{satisfying } \pi[0] = s, \\ (\mathcal{D}, s) \models Ap & \Leftrightarrow & (\mathcal{D}, \pi) \models p \\ & & \text{for all paths } \pi \in runs(\mathcal{D}) \\ & & \text{satisfying } \pi[0] = s. \end{array}$$

The semantics of $E_f p$ and $A_f p$ are defined similar to Ep and Ap respectively, replacing *path* (run) by *fair path* (fair runs).

Path formulas are interpreted over runs of $\mathcal{D}$. We define the notion of a CTL* formula p holding at a run $\pi \in runs(\mathcal{D})$, denoted $(\mathcal{D}, \pi) \models p$, as follows:

- For a state formula p,

$$(\mathcal{D}, \pi) \models p \Leftrightarrow (\mathcal{D}, \pi[0]) \models p.$$

- For path formulas p and q (here, $|\pi|$ may be infinite),

$$\begin{array}{lll}
(\mathcal{D}, \pi) \models p \vee q & \Leftrightarrow & (\mathcal{D}, \pi) \models p \text{ or } (\mathcal{D}, \pi) \models q, \\
(\mathcal{D}, \pi) \models p \wedge q & \Leftrightarrow & (\mathcal{D}, \pi) \models p \text{ and } (\mathcal{D}, \pi) \models q, \\
(\mathcal{D}, \pi) \models \neg p & \Leftrightarrow & (\mathcal{D}, \pi) \not\models p, \\
(\mathcal{D}, \pi) \models \bigcirc p & \Leftrightarrow & |\pi| > 1 \text{ and } (\mathcal{D}, \pi[1..]) \models p, \\
(\mathcal{D}, \pi) \models p \mathcal{U} q & \Leftrightarrow & \\
\quad (\mathcal{D}, \pi[k..]) \models q \text{ for some } k \geq 0, \text{ and} & & \\
\quad (\mathcal{D}, \pi[i..]) \models p \text{ for every } i, 0 \leq i < k, & & \\
(\mathcal{D}, \pi) \models p \mathcal{W} q & \Leftrightarrow & (\mathcal{D}, \pi) \models p \mathcal{U} q, \text{ or} \\
\quad (\mathcal{D}, \pi[i..]) \models p \text{ for all } i < |\pi|. & &
\end{array}$$

We say that a CTL* formula p holds on $\mathcal{D}$ (p is $\mathcal{D}$-valid), denoted $\mathcal{D} \models p$, if $(\mathcal{D}, s) \models p$, for every initial state s in $\mathcal{D}$.

Let p and q be CTL* formulas. We introduce the abbreviation

$$p \Leftrightarrow q \quad \text{for} \quad A\Box((p \rightarrow q) \wedge (q \rightarrow p)),$$

where $p \rightarrow q$ is the logical implication equivalent to $\neg p \vee q$. If $\mathcal{D} \models p \Leftrightarrow q$, then, $\mathcal{D} \models f(p) \Leftrightarrow f(q)$ holds.

Without loss of generality, we assume that a formula is given in positive normal form, which means that negation is only applied to assertions, namely, to formulas with no temporal or path operators.

3 Relative Completeness

In this section, we give a proof of the relative completeness of CTL* by eliminating the temporal operators and path quantifiers inductively. In the following, T represents any tautology.

Lemma *Let $\mathcal{D}$ be an FDS. There exists an assertion fpb such that $\mathcal{D} \models fpb \Leftrightarrow E_f \mathrm{T}$, i.e., fpb describes that a fair path begins with the current state.*

Proof Let $\mathcal{J} = \{J_1, \ldots, J_n\}$, $\mathcal{C} = \{\langle a_1, b_1 \rangle, \ldots, \langle a_m, b_m \rangle\}$, and I be the index set of $\mathcal{C}$.

For a subset X of I, we define $\mathcal{J}_X$ and $\mathcal{K}_X$ as follows:

$$\mathcal{J}_X = \mathcal{J} \cup \{a_j, b_j \mid j \in X\},$$
$$\mathcal{K}_X = \{a_j \mid j \in I - X\}.$$

At first, the predicate S_X that describes there exists a path from the current state such that, for each c in $\mathcal{J}_X$, it contains infinitely many c-positions, and for each a in $\mathcal{K}_X$, it contains no a-position, can be defined by the greatest fixpoint of the following Y:

$$\begin{array}{l}
Y = \lambda P.\lambda s.\exists s_0, \ldots, s_n (n \geq 1 \wedge s_0 = s \wedge \\
\qquad \forall i (0 \leq i < n \rightarrow \rho(s_i, s_{i+1})) \wedge \\
\qquad \forall i (0 \leq i < n \rightarrow \bigwedge_{a \in \mathcal{K}_X} \neg a(s_i)) \wedge \\
\qquad \bigwedge_{c \in \mathcal{J}_X} \exists i (0 \leq i < n \wedge c(s_i)) \wedge P(s_n)).
\end{array}$$

It is shown that the above defined S_X has the intended property as follows. At first, let s be a state initiating a path with the property, and show $S_X(s)$. Define a sequence $k_0, k_1, \ldots$ as follows: $k_0 = 0$; k_1 is the first index where all the conditions in $\mathcal{J}_X$ have been satisfied at least once after k_0; k_2 is the next such index after k_1, etc. Since $\top(s_{k_{i+1}})$ holds, $Y(\top)(s_{k_i})$ also holds, for any i. Then, by shifting the index, $Y^2(\top)(s_{k_i})$ holds for any i. Similarly, we have $Y^n(\top)(s_{k_i})$ for any i and n. Thus, $Y^n(\top)(s)$ holds for any n, so that $S_X(s)$ holds. Conversely, we assume $S_X(s)$. The right-hand side of the fixpoint equation implies that in the initial sequence from $s = s_{k_0}$ to s_{k_1}, no condition in $\mathcal{K}_X$ is satisfied and every condition in $\mathcal{J}_X$ is satisfied at least once. Because s_{k_1} satisfies S_X, it can be extended to s_{k_2}. Since this process can be repeated infinitely, $s = s_0$ initiates a path satisfying the intended property.

Next, the predicate F_X that describes there exists a path from the current state such that, for each c in $\mathcal{J}_X$, it contains infinitely many c-positions, and for each a in $\mathcal{K}_X$, it contains only finite a-positions, can be defined by the least fixpoint of the following Y:

$$Y = \lambda P.\lambda s.(S_X(s) \vee \exists t(\rho(s,t) \wedge P(t))).$$

We show that F_X has the intended property. At first, let s be a state that initiates such a path $\pi = s_0, s_1, \ldots$, where $s = s_0$, and show $F_X(s)$. Because every condition in $\mathcal{K}_X$ is satisfied finitely on this path and $\mathcal{K}_X$ is finite, there exists an index k after which no condition in $\mathcal{K}_X$ is satisfied. Every condition in $\mathcal{J}_X$ is satisfied infinitely after s_k. Hence s_k satisfies S_X, and this means $F_X(s_k)$. By taking s_k for t in the right-hand side of the fixpoint equation, we have $F_X(s_{k-1})$. Repeating this process yields $F_X(s)$. Next, we suppose $F_X(s)$ and show s initiates an intended path. It suffices to show that if s satisfies $Y^n(\bot)$ then it initiates an intended path. It is shown by induction on n. If $n = 0$, it is apparent from $S_X(s)$. If $n > 0$, it is clear because the difference is only that an additional sequence precedes it.

Finally, *fpb* is defined by

$$\forall s(fpb(s) \equiv \bigvee_{X \subset I} F_X(s)).$$

This *fpb* satisfies the intended property clearly; one fair path from the current state corresponds to one of F_X for some X, and if $F_X(s)$ holds, the state s initiates a fair path. □

Theorem *For an FDS $\mathcal{D}$ and a CTL* formula φ, we can effectively construct an assertion $p_{\mathcal{D}}$ such that $\mathcal{D} \models \varphi$ is equivalent to $\models p_{\mathcal{D}}$.*

Proof We show this by induction on the number of temporal operators and path quantifiers in φ. In the following, q denotes an assertion.

- If there is no temporal operator or path quantifier, then, $\mathcal{D} \models \varphi$ is equivalent to $\models \Theta \rightarrow \varphi$.
- If there exists a temporal operator whose body is an assertion, then this operator was shown to be eliminated by the BASIC-PATH rule (Pnueli & Kesten 2002). Details of the rule are in the Appendix A, and the sketch is as follows: For this subformula ψ, augment $\mathcal{D}$ by introducing a boolean variable x_ψ and obtain $\mathcal{D}'$ so that $\mathcal{D} \models f(\psi)$ is equivalent to $\mathcal{D}' \models f(x_\psi)$.
- If there exists Aq in φ, then we can eliminate this A, because $Aq \Leftrightarrow q$.
- The case for Eq is similar.
- If there exists $A_f q$, then we can replace it by $fpb \rightarrow q$, because $A_f q \Leftrightarrow fpb \rightarrow q$.
- If there exists $E_f q$, then we can replace it by $fpb \wedge q$, because $E_f q \Leftrightarrow fpb \wedge q$. □

4 Example

In this section, we show an example and prove a property by reducing a CTL* formula into an assertion

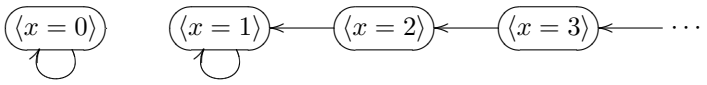

Figure 1: Transition relation of $\mathcal{D}$.

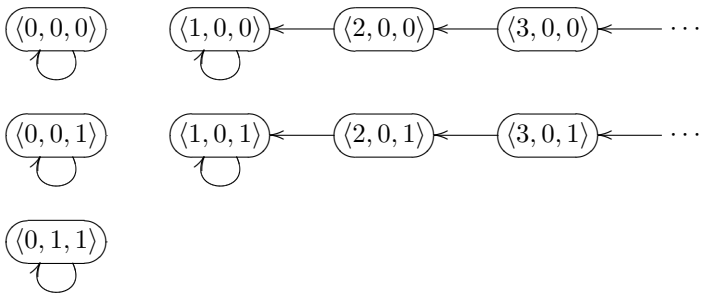

Figure 2: Transition relation of $\mathcal{D}_3$.

using the theorem. Let $\mathcal{D}$ be an FDS $\langle V, \Theta, \rho, \mathcal{J}, \mathcal{C}\rangle$, where

$$\begin{array}{ll} V & : \{x\}, \\ \Theta & : \mathrm{T}, \\ \rho & : (x \le 1 \wedge x' = x) \vee (x \ge 2 \wedge x' = x - 1), \\ \mathcal{J} & : \{x = 0 \vee x \ge 2\}, \\ \mathcal{C} & : \emptyset, \end{array}$$

and the domain is the set of natural numbers. Transition relation is illustrated in Fig. 1. For this system, we prove the property $A_f \Diamond\Box(x = 0)$. By applying BASIC-PATH rule twice, we have the following equivalent steps:

$$\begin{array}{rl} & \mathcal{D} \models A_f \Diamond\Box(x = 0), \\ & \text{A tester for } \Box(x = 0) \text{ with } x_\Box = \Box(x = 0) \\ \mathcal{D}|||T_\Box & \models A_f \Diamond x_\Box, \\ & \text{A tester for } \Diamond x_\Box \text{ with } x_\Diamond = \Diamond x_\Box \\ \mathcal{D}|||T_\Box|||T_\Diamond & \models A_f x_\Diamond, \end{array}$$

where transition relations and justice requirements of $T_\Box$ and $T_\Diamond$ are:

$$\begin{array}{ll} \rho_1 : x_\Box = (x = 0 \wedge x'_\Box), & \rho_2 : x_\Diamond = (x_\Box \vee x'_\Diamond), \\ \mathcal{J}_1 : \{x_\Diamond \vee x \ne 0\}, & \mathcal{J}_2 : \{\neg x_\Diamond \vee x_\Box\}. \end{array}$$

(Note that the symbol $|||$ denotes parallel composition of FDSs and the definition is given in the Appendix A.) Thus, the task of verifying the formula $A_f \Diamond\Box(x = 0)$ over system $\mathcal{D}$ is reduced to the verification of the simpler formula $A_f x_\Diamond$ over the system $\mathcal{D}_3 = \mathcal{D}|||T_\Box|||T_\Diamond : \langle V_3, \Theta_3, \rho_3, \mathcal{J}_3, \mathcal{C}_3\rangle$, where

$$\begin{array}{ll} V_3 & : \{x, x_\Box, x_\Diamond\}, \\ \Theta_3 & : \mathrm{T}, \\ \rho_3 & : \rho \wedge \rho_1 \wedge \rho_2, \\ \mathcal{J}_3 & : \mathcal{J} \cup \mathcal{J}_1 \cup \mathcal{J}_2, \\ \mathcal{C}_3 & : \emptyset. \end{array}$$

In the following, we identify a state with a triple $\langle v_x, v_{x_\Box}, v_{x_\Diamond}\rangle$, where $v_x, v_{x_\Box}$, and $v_{x_\Diamond}$ are the values of $x, x_\Box$, and $x_\Diamond$, respectively. The transition relation of system $\mathcal{D}_3$ is presented in Fig. 2, where the states are represented using this notation.

We now use the predicate *fpb* and reduce the current goal $\mathcal{D}_3 \models A_f x_\Diamond$ into $\models fpb \to x_\Diamond$. From the definition, *fpb* is obtained as follows:

$$\begin{array}{ll} fpb(s) & \equiv F_\emptyset(s) \\ & \equiv \mu Q.(S_\emptyset(s) \vee \exists t(\rho_3(s,t) \wedge Q(t)))(s). \end{array}$$

$S_\emptyset$ is the greatest fixpoint of the following Y:

$$\begin{array}{l} Y = \lambda P.\lambda s.\exists s_0, \ldots, s_n(n \ge 1 \wedge s_0 = s \wedge \\ \qquad \forall i(0 \le i < n \to \rho_3(s_i, s_{i+1})) \wedge \\ \qquad \bigwedge_{c \in J_3} \exists i(0 \le i < n \wedge c(s_i)) \wedge P(s_n)). \end{array}$$

Thus,

$$\begin{array}{rcl} Y^0(\top) & = & \top, \\ Y^1(\top) & = & \{\langle 0,1,1\rangle\} \cup \{\langle i,0,0\rangle | i \ge 2\}, \\ Y^2(\top) & = & \{\langle 0,1,1\rangle\} \cup \{\langle i,0,0\rangle | i \ge 3\}, \\ & \vdots & \end{array}$$

and we have

$$S_\emptyset = \bigcap_{i \ge 0} Y^i(\top) = \{\langle 0,1,1\rangle\}.$$

Hence, *fpb* is the least fixpoint of the following Y:

$$Y = \lambda P.\lambda s.(s = \langle 0,1,1\rangle \vee \exists t(\rho_3(s,t) \wedge P(t))).$$

and is clearly $fpb \equiv x = 0 \wedge x_\Box \wedge x_\Diamond$.

Therefore, $\models fpb \to x_\Diamond$ is apparently satisfied, and the target property is shown.

5 Conclusions

In this paper, we showed the relative completeness of a version of CTL*. In the proof, validity of a CTL* formula was reduced to the validity of an assertion by eliminating the temporal operators and path quantifiers inductively. It proceeds algorithmically without requiring us to devise any intermediate assertion.

We concentrated on reducing CTL* formulas into assertions, and the resulting assertions can be complicated. The assertions, however, are in some forms, which could be helpful in proving them. It would be possible to apply our method to the verification of CTL* properties over possibly infinite state systems, which is our future work.

References

Clarke, E. M. & Emerson, E. A. (1981), Design and synthesis of synchronization skeletons using branching time temporal logic, Proceedings of the Workshop on Logic of Programs, *Lecture Notes in Computer Science*, Vol. 131, pp. 52–71.

Clarke, E. M., Emerson, E. A. & Sistla, A. P. (1986), 'Automatic verification of finite-state concurrent systems using temporal specifications', *ACM Transactions on Programming Languages and Systems* **8**(2), 244–263.

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

Emerson, E. A. (1990), Temporal and modal logic, *in* J. van Leeuwen, ed, 'Handbook of Theoretical Computer Science', Vol. B, Elsevier and MIT Press, pp. 995–1072.

Immerman, N. & Vardi, M. Y. (1997) Model checking and transitive-closure logic, CAV '97, *Lecture Notes in Computer Science*, Vol. 1254, pp. 291–302.

Manna, Z. & Pnueli, A. (1991), 'Completing the temporal picture', *Theoretical Computer Science* **83**(1), 97–130.

Pnueli, A. (1977), The temporal logic of programs, *in* 'Proceedings of the Eighteenth Symposium on Foundations of Computer Science', pp. 46–57.

Pnueli, A. & Kesten, Y. (2002), A Deductive proof system for CTL*, CONCUR 2002, *Lecture Notes in Computer Science*, Vol. 2421 pp. 24–40.

Reynolds, M. (2001), 'An axiomatization of full computational tree logic', *Journal of Symbolic Logic* **66**(3), 1011–1057.

Appendix A

Elimination of temporal operators is based on BASIC-PATH rule (Pnueli & Kesten 2002). Necessary portions are shown here.

Fair discrete systems can be composed in parallel. Let $\mathcal{D}_i = \langle V_i, \Theta_i, \rho_i, \mathcal{J}_i, \mathcal{C}_i\rangle$, $i \in \{1, 2\}$, be two fair discrete systems. We define the synchronous parallel composition of two FDSs to be

$$\mathcal{D} = \langle V, \Theta, \rho, \mathcal{J}, \mathcal{C}\rangle = \langle V_1, \Theta_1, \rho_1, \mathcal{J}_1, \mathcal{C}_1\rangle ||| \langle V_2, \Theta_2, \rho_2, \mathcal{J}_2, \mathcal{C}_2\rangle,$$

where

$$V = V_1 \cup V_2, \Theta = \Theta_1 \wedge \Theta_2, \rho = \rho_1 \wedge \rho_2,$$
$$\mathcal{J} = \mathcal{J}_1 \cup \mathcal{J}_2, \mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2.$$

Temporal operators are eliminated successively from the formula φ until it becomes an assertional basic CTL* formula. Elimination is based on the construction of temporal testers, as follows.

We define a temporal tester for each of the three temporal operators $\bigcirc$, $\mathcal{U}$ and $\mathcal{W}$. For an assertion p, we denote by V_p the set of variables occurring in p. For the basic path formula $\bigcirc p$, we construct the tester $T_{\bigcirc p}$ as follows:

$$\begin{array}{ll} V & : V_p \cup \{x_\bigcirc\}, \\ \Theta & : \mathrm{T}, \\ \rho & : x_\bigcirc = p', \\ \mathcal{J} = \mathcal{C} & : \emptyset. \end{array}$$

For the basic path formula $p\mathcal{U}q$, we construct the tester $T_{p\mathcal{U}q}$ as follows:

$$\begin{array}{ll} V & : V_p \cup V_q \cup \{x_\mathcal{U}\}, \\ \Theta & : \mathrm{T}, \\ \rho & : x_\mathcal{U} = (q \vee p \wedge x'_\mathcal{U}), \\ \mathcal{J} & : \{\neg x_\mathcal{U} \vee q\}, \\ \mathcal{C} & : \emptyset. \end{array}$$

For the basic path formula $p\mathcal{W}q$, we construct the tester $T_{p\mathcal{W}q}$ as follows:

$$\begin{array}{ll} V & : V_p \cup V_q \cup \{x_\mathcal{W}\}, \\ \Theta & : \mathrm{T}, \\ \rho & : x_\mathcal{W} = (q \vee p \wedge x'_\mathcal{W}), \\ \mathcal{J} & : \{x_\mathcal{W} \vee (\neg p \wedge \neg q))\}, \\ \mathcal{C} & : \emptyset. \end{array}$$

Let $f(\varphi)$ be an arbitrary CTL* formula containing one or more occurrences of the basic path formula φ. Following is the rule BASIC-PATH which reduces the proof of $f(\varphi)$, where x_φ is a boolean variable, and $f(x_\varphi)$ is obtained from $f(\varphi)$ by replacing every occurrence of φ by x_φ.

BASIC-PATH
For a CTL* formula $f(\varphi)$,
a basic path formula φ,
and an FDS $\mathcal{D}$,

$$\frac{\mathcal{D}|||T_\varphi \models f(x_\varphi)}{\mathcal{D} \models f(\varphi)}$$