ISCAS-LCS-10-16 August, 2010

中国科学院软件研究所
计算机科学实验室报告

Bounded Semantics of CTL

by

Wenhui Zhang

State key Laboratory of Computer Science
Institute of Software
Chinese Academy of Sciences
Beijing 100190. China

Bounded Semantics of CTL

Wenhui Zhang
State Key Laboratory of Computer Science
Institute of Software, Chinese Academy of Sciences
P.O.Box 8718, Beijing 100190, China

18 August 2010

1 Introduction

Bounded model checking has been proposed as a complementary approach to BDD based symbolic model checking for combating the state explosion problem [1]. The idea was first applied to checking LTL properties, essentially for efficient error detection. It has then been applied to checking ACTL properties [4], mostly also for error detection. Both works were based on bounded semantics for existentially interpreted logics, i.e., LTL with existential interpretation and ECTL (the existential fragment of CTL). For verification of a property specified as a universal one that is valid, such semantics is not very useful. It is therefore important to develop bounded semantics of logics capable of specifying universal properties. This document presents the bounded semantics of CTL [5] and related issues, including a QBF-based characterization of CTL formulas and an bounded verification algorithm based on the characterization (called bounded semantics model checking). The rest of this paper is organized as follows. In Section 2, the logic CTL is presented. In Section 3, the bounded semantics of CTL is presented. In Section 4, a QBF-based characterization of CTL and a QBF-based bounded semantics model checking algorithm based on the characterization are presented. In Section 5, a SAT-based characterization of ACTL/ECTL and a SAT-based bounded semantics model checking algorithm based on the characterization are presented. Concluding remarks are presented in Section 6 with a short summary of an experimental evaluation of bounded semantics model checking of CTL properties.

2 Computation Tree Logic

Computation Tree Logic (CTL) is a propositional branching-time temporal logic [3] introduced by Emerson and Clarke as a specification language for finite state systems.

Syntax Let AP be a set of propositional symbols and p range over AP. The set of CTL formulas Φ over AP is defined as follows:

$$\begin{aligned}\Phi ::= & \, p \mid \neg\Phi \mid \Phi \wedge \Phi \mid \Phi \vee \Phi \mid \\ & AX\,\Phi \mid AF\,\Phi \mid AG\,\Phi \mid A(\Phi\, U\, \Phi) \mid A(\Phi\, R\, \Phi) \mid \\ & EX\,\Phi \mid EF\,\Phi \mid EG\,\Phi \mid E(\Phi\, U\, \Phi) \mid E(\Phi\, R\, \Phi)\end{aligned}$$

The property of a finite state system may be specified by such a formula, and conversely, the truth of such a formula may be evaluated in a finite state system.

Models A finite state system may be represented by a Kripke structure which is a quadruple $M = \langle S, T, I, L\rangle$ where S is a set of states, $T \subseteq S \times S$ is a transition relation which is total, $I \subseteq S$ is a set of initial states and $L : S \to 2^{AP}$ is a labeling function that maps each state to a subset of propositions of AP. A Kripke structure is also called a model.

Paths A computation of M is then represented by a path of M which is an infinite sequence of states $\pi = \pi_0\pi_1 \cdots$ such that $(\pi_i, \pi_{i+1}) \in T$ for all $i \geq 0$. Given a path $\pi = \pi_0\pi_1 \cdots$, we use π^i to denote the subpath of π starting at π_i, use $\pi(s)$ to denote a path π with $\pi_0 = s$. Then $\exists\pi(s).\varphi$ means that there is a path π with $\pi_0 = s$ such that φ holds, and $\forall\pi(s).\varphi$ means that for every path π with $\pi_0 = s$, φ holds.

Let $\Psi = \{\psi \mid E\psi \in \Phi\}$ be a set of auxiliary formulas.

Definition 1. *(Semantics of CTL) Let s be a state of M, p a propositional symbol, $\varphi, \varphi_0, \varphi_1$ CTL formulas. The relation that φ holds on s in M is denoted $M, s \models \varphi$. The relation $\models$ is defined as follows (where $M, \pi \models \psi$ is an auxiliary relation for $\psi \in \Psi$).*

$M, s \models p$ *iff* $p \in L(s)$.
$M, s \models \neg\varphi_0$ *iff* $M, s \not\models \varphi_0$
$M, s \models \varphi_0 \wedge \varphi_1$ *iff* $M, s \models \varphi_0$ *and* $M, s \models \varphi_1$
$M, s \models \varphi_0 \vee \varphi_1$ *iff* $M, s \models \varphi_0$ *or* $M, s \models \varphi_1$
$M, s \models A\psi_0$ *iff* $\forall\pi(s).(M, \pi \models \psi_0)$
$M, s \models E\psi_0$ *iff* $\exists\pi(s).(M, \pi \models \psi_0)$
$M, \pi \models X\varphi_0$ *iff* $M, \pi_1 \models \varphi_0$
$M, \pi \models F\varphi_0$ *iff* $\exists k \geq 0.M, \pi_k \models \varphi_0$
$M, \pi \models G\varphi_0$ *iff* $\forall k \geq 0.M, \pi_k \models \varphi_0$
$M, \pi \models \varphi_0 U \varphi_1$ *iff* $\exists k \geq 0.(M, \pi_k \models \varphi_1 \wedge \forall j < k.(M, \pi_j \models \varphi_0))$
$M, \pi \models \varphi_0 R \varphi_1$ *iff* $\forall k \geq 0.(M, \pi_k \models \varphi_1 \vee \exists j < k.(M, \pi_j \models \varphi_0))$

Let $M = \langle S, T, I, L\rangle$ denote the model used in the rest of this paper.

Definition 2. $M \models \varphi$ *iff* $M, s \models \varphi$ *for all* $s \in I$.

A CTL formula is in the negation normal form (NNF), if the negation $\neg$ is applied only to propositional symbols. Every CTL formula can be transformed into an equivalent formula in NNF. Without loss of generality, we only consider formulas in NNF. Formulas not in NNF are considered as an abbreviation of the equivalent one in NNF.

3 Bounded Semantics

Finite Paths A finite path π of M is a finite prefix of an infinite path of M.

k-Paths Let $k \geq 0$. A k-path of M is a finite path of M with length $k+1$. π is a k-path, if $\pi = \pi_0 \cdots \pi_k$ such that $\pi_i \in S$ for $i = 0, ..., k$ and $(\pi_i, \pi_{i+1}) \in T$ for $i = 0, ..., k-1$. For the idea of a k-path, the reader is referred to [1].

Bounded Models The k-model of M is a quadruple $M_k = \langle S, Ph_k, I, L \rangle$ where Ph_k is the set of all k-paths of M. M_k can be considered as an approximation of M. For the idea of a bounded model, the reader is referred to [4].

Loops A loop is a k-path π such that $\pi_i = \pi_k$ for some $0 \leq i < k$. This is similar to the one defined in [1], which is a finite path such that the last element has a successor to some element in the path.

Internal-Loops An internal-loop is a k-path that contains some sub-path which is a loop. Let $ilp(\pi)$ denote that π is an inward-loop. An important property of such a loop is that if π is a prefix of π', then $ilp(\pi) \rightarrow ilp(\pi')$.

Definition 3 (Bounded Semantics of CTL). *Let s be a state of M, p a propositional symbol, $\varphi, \varphi_0, \varphi_1$ CTL formulas. The relation that φ holds on s in M_k is denoted $M_k, s \models \varphi$. Let π denote a k-path of Ph_k. The relation $\models$ is defined as follows (where $M_k, \pi \models \psi$ is an auxiliary relation).*

$M_k, s \models p$ *iff* $p \in L(s)$.
$M_k, s \models \neg p$ *iff* $p \notin L(s)$
$M_k, s \models \varphi_0 \wedge \varphi_1$ *iff* $(M_k, s \models \varphi_0)$ *and* $(M_k, s \models \varphi_1)$
$M_k, s \models \varphi_0 \vee \varphi_1$ *iff* $(M_k, s \models \varphi_0)$ *or* $(M_k, s \models \varphi_1)$
$M_k, s \models A\psi$ *iff* $\forall \pi(s).(M_k, \pi \models \psi)$
$M_k, s \models E\psi$ *iff* $\exists \pi(s).(M_k, \pi \models \psi)$
$M_k, \pi \models X\varphi_0$ *iff* $k \geq 1 \wedge (M_k, \pi_1 \models \varphi_0)$
$M_k, \pi \models F\varphi_0$ *iff* $\exists i \leq k.(M_k, \pi_i \models \varphi_0)$
$M_k, \pi \models G\varphi_0$ *iff* $ilp(\pi) \wedge (\forall i \leq k.(M_k, \pi_i \models \varphi_0))$
$M_k, \pi \models \varphi_0 U \varphi_1$ *iff* $\forall \pi(s).(\exists i \leq k.(M_k, \pi_i \models \varphi_1 \wedge \forall j < i.(M_k, \pi_j \models \varphi_0)))$
$M_k, \pi \models \varphi_0 R \varphi_1$ *iff* $\forall i \leq k.(M_k, \pi_i \models \varphi_1 \vee \exists j < i.(M_k, \pi_j \models \varphi_0)) \wedge (\exists j \leq k.(M_k, \pi_j \models \varphi_0) \vee ilp(\pi))$

Proposition 1. *$M, s \models \varphi$ iff $M_k, s \models \varphi$ for some $k \geq 0$.*

Bounded Semantics Model Checking Principle Let s be a state of M, φ a CTL formula. The bounded semantics model checking principle for the verification of $M, s \models \varphi$ may be formulated as follows.

Let $k = 0$; .
If $M_k, s \models \varphi$ holds, report that φ holds;
If $M_k, s \models \neg\varphi$ holds, report that φ does not hold;
Increase k, go to the first "if"-test;

The correctness and the termination are guaranteed by Proposition 1.

4 QBF-based Characterization of CTL Formulas

From the bounded semantics, a QBF-based characterization of CTL formulas can be developed as follow. Let $k \geq 0$. Let $u_0, ..., u_k$ be a finite sequence of state variables. The sequence $u_0, ..., u_k$ (denoted by $\vec{u}$) is intended to be used as a representation of a path of M_k. This is captured by the following definition of $P_k(\vec{u})$.

Definition 4.

$$P_k(\vec{u}) := \bigwedge_{j=0}^{k-1} T(u_j, u_{j+1})$$

Every assignment to the set of state variables $\{u_0, ..., u_k\}$ satisfying $P_k(\vec{u})$ represents a valid k-path of M. Let $e_k(\vec{u})$ denote that the k-path represented by $\vec{u}$ is a loop. Formally, we have the following definition of $e_k(\vec{u})$.

Definition 5.

$$e_k(\vec{u}) := \bigvee_{x=0}^{k-1} \bigvee_{y=x+1}^{k} u_x = u_y.$$

Let $p \in AP$ be a proposition symbol and $p(v)$ be the propositional formula such that $p(v)$ is true whenever v is assigned the truth value representing a state s in which p holds.

Definition 6 (Translation of CTL Formulas). *Let $k \geq 0$. Let v be a state variable and φ be a CTL formula. The encoding $[[\varphi, v]]_k$ is defined as follows.*

$[[p, v]]_k$	$= p(v)$
$[[\neg p, v]]_k$	$= \neg p(v)$
$[[\varphi \vee \psi, v]]_k$	$= [[\varphi, v]]_k \vee [[\psi, v]]_k$
$[[\varphi \wedge \psi, v]]_k$	$= [[\varphi, v]]_k \wedge [[\psi, v]]_k$
$[[A\varphi, v]]_k$	$= \forall \vec{u}.(P(\vec{u}) \wedge v = u_0 \rightarrow [[\varphi, \vec{u}]]_k)$
$[[E\varphi, v]]_k$	$= \exists \vec{u}.(P(\vec{u}) \wedge v = u_0 \wedge [[\varphi, \vec{u}]]_k)$
$[[X\varphi, \vec{u}]]_k$	$= k \geq 1 \wedge [[\varphi, u_1]]_k$
$[[F\psi, \vec{u}]]_k$	$= \bigvee_{j=0}^{k} [[\psi, u_j]]_k$
$[[G\psi, \vec{u}]]_k$	$= \bigwedge_{j=0}^{k} [[\psi, u_j]]_k \wedge e_k(\vec{u}))$
$[[\varphi U \psi, \vec{u}]]_k$	$= \bigvee_{j=0}^{k} ([[\psi, u_j]]_k \wedge \bigwedge_{t=0}^{j-1} [[\varphi, u_t]]_k)$
$[[\varphi R \psi, \vec{u}]]_k$	$= \bigwedge_{j=0}^{k} ([[\psi, u_j]]_k \vee \bigvee_{t=0}^{j-1} [[\varphi, u_t]]_k) \wedge (\bigvee_{t=0}^{k} [[\varphi, u_t]]_k \vee e_k(\vec{u}))$

Let $v(s)$ denote that the state variable v has been assigned a value corresponding to the state s.

Proposition 2. *Let φ be a CTL formula. $M_k, s \models \varphi$ iff $[[\varphi, v(s)]]_k$ holds.*

Let $I(v)$ denote the propositional formula that restricts potential values of v to the initial states of M.

Corollary 1. *Let φ be a CTL formula. $M \models \varphi$ iff there is a $k \geq 0$ such that $\forall v.(I(v) \rightarrow [[\varphi, v]]_k)$ and there is no k such that $\exists v.(I(v) \wedge [[\neg\varphi, v]]_k)$.*

Bounded Semantics Model Checking Algorithm Let φ be a CTL formula. The corresponding QBF-based bounded semantics model checking algorithm for the verification of $M \models \varphi$ is then as follows.

Let $k = 0$; .
If $\forall v.(I(v) \rightarrow [[\varphi, v]]_k)$ holds, report that φ holds;
If $\exists v.(I(v) \wedge [[\neg\varphi, v]]_k)$ holds, report that φ does not hold;
Increase k, go to the first "if"-test;

5 SAT-based Characterization of ACTL Formulas

The restriction of CTL to formulas (in NNF) not containing the existential path quantifier E is called ACTL. Similarly, the restriction of CTL to formulas not containing the universal path quantifier A is called ECTL.

Submodels Let $M_k = \langle S, Ph_k, I, L\rangle$ be the k-model of M. $M_k^{'} = \langle S, Ph_k^{'}, I, L\rangle$ is a submodel of M_k, if $Ph_k^{'} \subseteq Ph_k$. $M_k^{'}$ is called a (k, n)-submodel of M_k when $|M_k^{'}| = n$ with $|M_k^{'}|$ denoting the size of $Ph_k^{'}$.

Definition 7. *Let φ, ψ be respectively an ACTL formula and an ECTL formula. $n^a(M_k, \varphi)$ is the least number such that for all s, $M_k, s \models \varphi$ iff $M'_k, s \models \varphi$ for all $(k, n^a(\varphi))$-submodels M'_k of M_k; $n^e(M_k, \psi)$ is the least number such that for all s, $M_k, s \models \psi$ iff $M'_k, s \models \psi$ for some $(k, n^e(\psi))$-submodel M'_k of M_k.*

Proposition 3. *Let φ be an ACTL formula and $m \geq n^a(M_k, \varphi)$. $M, s \models \varphi$ iff for all (k, m)-submodel N of M_k, we have $N, s \models \varphi$ for some $k \geq 0$.*

Proposition 4. *Let ψ be an ECTL formula and $m \geq n^e(M_k, \psi)$. $M, s \models \psi$ iff for there is some (k, m)-submodel N of M_k such that $N, s \models \varphi$ for some $k \geq 0$.*

Definition 8. *Let φ be an ACTL formula. $f_k(\varphi)$ is defined as follows.*

$f_k(p)$	$= 0$ *if* $p \in AP$
$f_k(\neg p)$	$= 0$ *if* $p \in AP$
$f_k(\varphi_0 \wedge \varphi_1)$	$= \max(f_k(\varphi_0), f_k(\varphi_1))$
$f_k(\varphi_0 \vee \varphi_1)$	$= f_k(\varphi_0) + f_k(\varphi_1)$
$f_k(AX\varphi)$	$= f_k(\varphi) + 1$
$f_k(AF\varphi)$	$= (k+1) \cdot f_k(\varphi) + 1$
$f_k(AG\varphi)$	$= f_k(\varphi) + 1$
$f_k(A(\varphi_0 U \varphi_1))$	$= k \cdot \max(f_k(\varphi_0), f_k(\varphi_1)) + f_k(\varphi_0) + f_k(\varphi_1) + 1$
$f_k(A(\varphi_0 R \varphi_1))$	$= k \cdot f_k(\varphi_0) + \max(f_k(\varphi_0), f_k(\varphi_1)) + 1$

Proposition 5. *$f_k(\varphi) \geq n^a(M_k, \varphi)$ and $f_k(\varphi) \geq n^e(M_k, \neg\varphi)$.*

Then a SAT-based characterization of ACTL and ECTL formulas can be developed as follow. Let $k \geq 0$. Let $i \geq 1$ and $u_{i,0}, ..., u_{i,k}$ be a finite sequence of state variables. The sequence $u_{i,0}, ..., u_{i,k}$ (denoted by $\vec{u_i}$) is intended to be used as a representation of a path of M_k. Every assignment to the set of state variables $\{u_{i,0}, ..., u_{i,k}\}$ satisfying $P_k(\vec{u_i})$ represents a valid k-path of M.

Definition 9. *Let $k \geq 0, b \geq 1$.*

$$[[M]]_k^b := \bigwedge_{i=1}^{b} P_k(\overrightarrow{u_i})$$

This is a collection of $P_k(l)$ for $l = 1, ..., b$, and is intended to represent the set of the k-paths in a (k, b)-submodel of M_k.

Definition 10 (Translation of ACTL and ECTL formulas). *Let $k \geq 0$. Let u be a state variable and φ be an ACTL formula. The encoding $[[\varphi, u]]_k^b$ is defined as follows.*

$[[p, u]]_k^b$	$= p(u)$
$[[\neg p, u]]_k^b$	$= \neg p(u)$
$[[\varphi \vee \psi, u]]_k^b$	$= [[\varphi, u]]_k^b \vee [[\psi, u]]_k^b$
$[[\varphi \wedge \psi, u]]_k^b$	$= [[\varphi, u]]_k^b \wedge [[\psi, u]]_k^b$
$[[A\varphi, u]]_k^b$	$= \bigwedge_{i=1}^{b}(u = u_{i,0} \rightarrow [[\varphi, \overrightarrow{u_i}]]_k^b)$
$[[E\varphi, u]]_k^b$	$= \bigwedge_{i=1}^{b}(u = u_{i,0} \wedge [[\varphi, \overrightarrow{u_i}]]_k^b)$
$[[X\varphi, \overrightarrow{u_i}]]_k^b$	$= k \geq 1 \wedge [[\varphi, u_{i,1}]]_k^b$
$[[F\psi, \overrightarrow{u_i}]]_k^b$	$= \bigvee_{j=0}^{k} [[\psi, u_{i,j}]]_k^b$
$[[G\psi, \overrightarrow{u_i}]]_k^b$	$= \bigwedge_{j=0}^{k} [[\psi, u_{i,j}]]_k^b \wedge e_k(\overrightarrow{u_i})$
$[[\varphi U \psi, \overrightarrow{u_i}]]_k^b$	$= \bigvee_{j=0}^{k}([[\psi, u_{i,j}]]_k^b \wedge \bigwedge_{t=0}^{j-1} [[\varphi, u_{i,t}]]_k^b)$
$[[\varphi R \psi, \overrightarrow{u_i}]]_k^b$	$= \bigwedge_{j=0}^{k}([[\psi, u_{i,j}]]_k^b \vee \bigvee_{t=0}^{j-1} [[\varphi, u_{i,t}]]_k^b) \wedge (\bigvee_{t=0}^{k} [[\varphi, u_{i,t}]]_k^b \vee e_k(\overrightarrow{u_i}))$

Proposition 6. *Let φ be an ACTL formula and $[[M, \varphi, u]]_k^b = [[M]]_k^b \rightarrow [[\varphi, u]]_k^b$. $[[M, \varphi, u(s)]]_k^b$ is valid iff $M'_k, s \models \varphi$ for all (k, b)-submodel M'_k.*

Proposition 7. *Let φ be an ECTL formula and $[[M, \psi, u]]_k^b := [[M]]_k^b \wedge [[\psi, u]]_k^b$. $[[M, \varphi, u(s)]]_k^b$ is satisfiable iff $M'_k, s \models \varphi$ for some (k, b)-submodel M'_k.*

Proposition 8. *Let φ be an ACTL formula. $M, s \models \varphi$ iff there is a k such that $[[M, \varphi, u(s)]]_k^{f_k(\varphi)}$ is valid and there is no k such that $[[M, \neg\varphi, u(s)]]_k^{f_k(\varphi)}$ is satisfiable.*

Corollary 2. *Let φ be an ACTL formula. $M \models \varphi$ iff there is a k such that $[[M, \varphi]]_k^b := I(u) \rightarrow [[M, \varphi, u]]_k^b$ is valid and there is no k such that $I(u) \wedge [[M, \neg\varphi, u]]_k^{f_k(\varphi)}$ is satisfiable.*

Bounded Semantics Model Checking Algorithm Let φ be an ACTL formula. The corresponding SAT-based bounded semantics model checking algorithm for the verification of $M \models \varphi$ is then as follows.

Let $k = 0$; .
Let $b = f_k(\varphi)$;
If $I(u) \wedge \neg[[M, \varphi, u]]_k^b$ is unsatisfiable, report that φ holds;
If $I(u) \wedge [[M, \neg\varphi, u]]_k^b$ is satisfiable, report that φ does not hold;
Increase k, go to the second "let"-statement;

6 Evaluation and Concluding Remarks

Bounded semantics of CTL and QBF-based characterization of CTL based on such a semantics are presented. Bounded semantics model checking algorithm based on solving QBF-formulas has then been established.

Evaluation Experimental evaluation[1] of the efficiency of QBF-based bounded semantics model checking of CTL formulas has been carried out. The evaluation is based on comparing an implementation of the bounded semantics model checking algorithm in *verds* version 1.30 with an implementation of boolean diagram model checking also in *verds* version 1.30. The evaluation was based on two types of random boolean programs and a set of 24 CTL formulas which includes formulas with nested CTL operators. Based on the test cases, the experimental evaluation shows that the bounded semantics model checking does not have advantage in verifying any of the properties that start with AG. On the other hand, the bounded semantics model checking has advantage in various degrees with respect to the other verification and falsification problems (including falsification of AG properties). In summary, the bounded semantics model checking has advantage in more than 50 percent of the test cases, which are well distributed among verification and falsification of universal properties (of the form $A\varphi$). In this sense, bounded semantics model checking and boolean diagram model checking may be considered complementary with their own advantages.

Note The evaluation of the bounded semantics model checking uses *verds* for comparison, instead of the well known symbolic model checker $NuSMV$ [2], since the boolean diagram model checking in *verds* is generally more efficient than NuSMV with respect to the test cases [6]. The efficiency of bounded semantics model checking also depends very much on the QBF-solving techniques. External QBF-solvers may be used to increase the efficiency of the verification. For ACTL formulas, special considerations are possible, and the use of SAT-solving techniques may be more efficient for this kind of problems.

References

1. A. Biere, A. Cimmatti, E. Clarke, and Y. Zhu. Symbolic Model Checking without BDDs. LNCS 1579:193-207. TACAS 99.
2. A. Cimatti, E. M. Clarke, F. Giunchiglia, M. Roveri. NUSMV: A New Symbolic Model Verifier. CAV 1999: 495-499.
3. E. Allen Emerson and E. M. Clarke. Using Branching-time Temporal Logics to Synthesize Synchronization Skeletons. Sci. of Comp. Prog. 2(3):241-266. 1982.
4. W. Penczek, B. Wozna, and A. Zbrzezny. Bounded Model Checking for the Universal Fragment of CTL. Fundamenta Informaticae 51:135-156. 2002.
5. W. Zhang. Bounded Semantics of CTL and SAT-based Verification. Lecture Notes in Computer Science 5885 (ICFEM 2009):286-305. Springer-Verlag. 2009.
6. W. Zhang. Ternary Boolean Diagrams. Technical Report ISCAS-LCS-10-15, Institute of Software, Chinese Academy of Sciences. 2010.

[1] Details are available at http://lcs.ios.ac.cn/∼zwh/verds/