Bounded Semantics of CTL and SAT-based Verification

Wenhui Zhang
Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences
P.O.Box 8718, Beijing 100080, China

April 28, 2009

Abstract

Bounded model checking has been proposed as a complementary approach to BDD based symbolic model checking for combating the state explosion problem, esp. for efficient error detection [3]. This has led to a lot of successful work with respect to error detection in the checking of LTL, ACTL (the universal fragment of CTL) and ACTL properties by satisfiability testing [3, 22, 25]. The use of bounded model checking for verification (in contrast to error detection) of LTL and ACTL properties has later also been studied [28, 30]. This paper studies the potentials and limitations of bounded model checking for the verification of CTL and CTL* formulas. On the theoretical side, we first provide a framework for discussion of bounded semantics, which serves as the basis for bounded model checking, then extend the bounded semantics of ACTL [30] to a bounded semantics of CTL, and discuss the limitation of developing such a bounded semantics for CTL*. On the practical side, a deduction of a SAT-based bounded model checking approach for ACTL properties from the bounded semantics of CTL is demonstrated, and a comparison of such an approach with BDD-based model checking is presented based on experimental results.*

1 Introduction

Bounded semantics of LTL with existential interpretation and that of ECTL (the existential fragment of CTL), and the characterization of these existentially interpreted properties have been studied and used as the theoretical basis for SAT-based bounded model checking [3, 22]. This has led to successful works with respect to error detection in the checking of LTL and ACTL (the universal fragment of CTL) properties by satisfiability testing [2]. It is considered as a complementary technique to BDD-based model checking [5, 4, 20, 7] for combating the state explosion problem [6], esp. for efficient error detection [23]. Bounded semantics of existential LTL and that of ECTL, and the characterization of such properties are consistent with the fact that the witness of the properties can be searched within a fragment of the valid paths. For verification purposes, one need to reach a completeness threshold or some termination criteria [17, 9, 16, 10, 1] in order to show the non-existence of a counter-example. This may not be as efficient. On the other hand, the principle of bounded model checking for verification (called bounded verification for short) should be similar to bounded error detection, such that we start with a small bounded model, if this is not sufficient, we increase the bound, until we have a conclusion or we run out of resources. With this principle in mind, bounded verification should check whether every representative part (a k-path or a set of such paths) of the system satisfies some property, and according to this, conclude whether the system satisfies this property. Bounded verification of general LTL formulas has been considered in [28], which can equivalently be formulated as that a model satisfies an LTL formula φ if there is a k-model such that every k-path starting with some initial state satisfies φ. The paper provides a sufficient condition (not a sufficient and necessary condition) and has discussed that in some special cases such as when dealing with LTL formulas restricted to pUq, the sufficient condition is actually a sufficient and necessary condition (for formulas of the form Fp, a similar characterization is already known [2]). Similar ideas have then been applied to ACTL formulas [29] and a similar result was obtained, and an implementation with experimental results was reported in [27]. These approaches for verification of LTL and ACTL formulas are based on bounded semantics of existential LTL and ECTL with some kinds of weakening, which result in an incomplete bounded characterization (i.e. with only a sufficient condition) of LTL and ACTL formulas. This problem has then been studied in [30] and a bounded semantics for ACTL and a characterization of ACTL properties by propositional formulas were provided. An improvement of the SAT-based encoding of the verification problem was considered in [8].

In this paper, on the theoretical side, we first propose a framework for discussion of bounded semantics, then extend the bounded semantics of ACTL to a sound and complete bounded semantics of CTL, and show that there is no such sound and complete semantics for CTL* in the given framework. On the practical side, we apply the bounded semantics of CTL to derive a SAT-based characterization of ACTL properties, and compare such a characterization with BDD based verification approaches.

2 Bounded Semantics

In this section, we provide a framework for discussion of bounded semantics.

A Kripke structure 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.

An infinite path $\pi = \pi_0\pi_1 \cdots$ of M is an infinite sequence of states such that $(\pi_i, \pi_{i+1}) \in T$ for all $i \geq 0$. A finite path π of M is a finite prefix of an infinite path of M. A k-path is a finite path with length $k+1$ (and k transitions). 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 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.

Semantics of temporal logics is defined with respect to Kripke structures. For brevity, a Kripke structure is called a model. We require a semantic relation to be compositional. We first define what we mean by compositionality with respect to path quantifiers (universal and existential) and with respect to propositional connectives (conjunction and disjunction).

Definition 2.1 (Compositionality w.r.t. Path Quantifiers) *Let M be a model and s be a state. Let $\models_a$ be a relation defined for path formulas and state formulas. The relation $\models_a$ is compositional with respect to path quantifiers, if the following hold:*

- $M, s \models_a A\varphi$ *iff* $M, \pi(s) \models_a \varphi$ *for all* $\pi(s)$ *of* M.
- $M, s \models_a E\varphi$ *iff* $M, \pi(s) \models_a \varphi$ *for some* $\pi(s)$ *of* M.

Let a structure be either a combination of M and a state or of M and a path (either a finite one or an infinite one).

Definition 2.2 (Compositionality w.r.t. Prop. Connectives) *Let $\mathcal{S}$ be a structure. Let $\models_a$ be a relation. The relation $\models_a$ is compositional with respect to propositional connectives, if the following hold:*

- $\mathcal{S} \models_a \varphi \vee \psi$ *iff* $\mathcal{S} \models_a \psi$ *or* $\mathcal{S} \models_a \psi$
- $\mathcal{S} \models_a \varphi \wedge \psi$ *iff* $\mathcal{S} \models_a \psi$ *and* $\mathcal{S} \models_a \psi$

The compositionality property is a formalization of the standard understanding of the path quantifiers and the propositional connectives. In addition, we formulate a consistency property with respect to the labeling at given positions (in a sequence of states).

Definition 2.3 (Consistency w.r.t. Position Labeling) *Let M be a model and s be a state. Let $\models_a$ be a relation defined for path formulas and state formulas. Let X be the next-time operator. Let p be a proposition. The relation $\models_a$ satisfies the consistency property, if the following hold:*

- $M, s \models_a p$ *iff* $p \in L(s)$.
- $M, \pi \models_a X^n p$ *iff* $p \in L(\pi_n)$ *when* π_n *is the* $(n+1)$*-th state of* π.

A simple compositional semantic relation is a relation which is compositional with respect to both propositional connectives and path quantifiers (whenever applicable) and satisfies the consistency property. For brevity, such a relation is called a *simple relation*. A compositional semantic relation is either a simple compositional semantic relation or a propositional combination of such relations. For brevity, such a relation is called a *semantic relation*.

Without loss of generality, a propositional combination of simple relations may be written as a disjunction of conjunctions of such relations, for instance, a semantic relation $\models$ may be written as $\bigvee_{i=1}^{m} \bigwedge_{j=1}^{n} \models_{i,j}$. Then $\mathcal{S} \models \varphi$ iff there is an i such that $\mathcal{S} \models_{i,j} \varphi$ holds for all j.

A bounded semantics is then represented by a family of semantic relations each defined on a bounded structure with a parameter indicating the bound. Let us call such a structure a k-structure.

Definition 2.4 (Soundness and Completeness) *Let $\mathcal{S}_k$ be a k-structure of $\mathcal{S}$. Let $\models_{a,k}$ be a family of semantic relations with respect to a given relation $\models$. The bounded semantics defined by $\models_{a,k}$ is sound and complete, iff the following hold:*

- *(Soundness) If* $\mathcal{S}_k \models_{a,k} \varphi$ *for some* $k \geq 0$*, then* $\mathcal{S} \models \varphi$.
- *(Completeness) If* $\mathcal{S} \models \varphi$*, then there is a* $k \geq 0$ *such that* $\mathcal{S}_k \models_{a,k} \varphi$.

Remark The purpose of this framework is to formalize the usual understanding of good bounded semantics. It excludes some definitions from being considered as semantic definitions, for instance, the following one: $M, s \models A\varphi$ iff $L(\mathcal{A}(M, s)) \subseteq L(\mathcal{A}(\varphi))$ where $L(\mathcal{A}(M, s))$ and $L(\mathcal{A}(\varphi))$ are the languages of the automata constructed from respectively the structure M, s and the LTL formula φ, because

it does not comply with compositionality and lacks good characteristics of a semantic definition of temporal logics. For the definition of the semantics of $M, s \models A\varphi$, it is reasonable to look for other kinds of definitions (with good structure and intuition).

3 On CTL

In this section, we provide a bounded semantics for CTL, and formulate a bounded model checking and verification principle for CTL properties.

Computation tree logic (CTL) is a propositional branching-time temporal logic [13] introduced by Emerson and Clarke as a specification language for finite state systems. Let AP be a set of propositional symbols. The set of CTL formulas is defined as follows:

Every member of AP is a CTL formula.
The logical connectives of CTL are: $\neg$, $\wedge$, and $\vee$.
If φ and ψ are CTL formulas, then so are:
$\neg\varphi$, $\varphi \wedge \psi$, and $\varphi \vee \psi$.
The temporal operators are:
EX, ER, EU, AX, AR, and AU.
If φ and ψ are CTL formulas, then so are:
$EX\ \varphi$, $E(\varphi\ R\ \psi)$, $E(\varphi\ U\ \psi)$, $AX\ \varphi$, $A(\varphi\ R\ \psi)$, and $A(\varphi\ U\ \psi)$.

3.1 Semantics of CTL

Let M be a model, s a state, φ a CTL formula. The relation that φ holds on s in M is denoted $M, s \models \varphi$.

Definition 3.1 (Semantics of CTL) *Let p be a propositional symbol, φ and ψ CTL formulas. Let $\pi = \pi_0\pi_1\cdots$ be an infinite path of M. The relation $M, s \models \varphi$ is defined as follows.*

$M, s \models p$	*iff* $p \in L(s)$.
$M, s \models \neg\varphi$	*iff* $M, s \not\models \varphi$
$M, s \models \varphi \wedge \psi$	*iff* $(M, s \models \varphi)$ *and* $(M, s \models \psi)$
$M, s \models \varphi \vee \psi$	*iff* $(M, s \models \varphi)$ *or* $(M, s \models \psi)$
$M, s \models AX\varphi$	*iff* $\forall\pi(s).(M, \pi_1 \models \varphi)$
$M, s \models AF\psi$	*iff* $\forall\pi(s).(\exists k \geq 0.(M, \pi_k \models \psi)$
$M, s \models AG\psi$	*iff* $\forall\pi(s).(\forall k \geq 0.(M, \pi_k \models \psi)$
$M, s \models A(\varphi U\psi)$	*iff* $\forall\pi(s).(\exists k \geq 0.(M, \pi_k \models \psi \wedge \forall j < k.(M, \pi_j \models \varphi)))$
$M, s \models A(\varphi R\psi)$	*iff* $\forall\pi(s).(\forall k \geq 0.(M, \pi_k \models \psi \vee \exists j < k.(M, \pi_j \models \varphi)))$
$M, s \models EX\varphi$	*iff* $\exists\pi(s).(M, \pi_1 \models \varphi)$
$M, s \models EF\psi$	*iff* $\exists\pi(s).(\exists k \geq 0.(M, \pi_k \models \psi)$
$M, s \models EG\psi$	*iff* $\exists\pi(s).(\forall k \geq 0.(M, \pi_k \models \psi)$
$M, s \models E(\varphi U\psi)$	*iff* $\exists\pi(s).(\exists k \geq 0.(M, \pi_k \models \psi \wedge \forall j < k.(M, \pi_j \models \varphi)))$
$M, s \models E(\varphi R\psi)$	*iff* $\exists\pi(s).(\forall k \geq 0.(M, \pi_k \models \psi \vee \exists j < k.(M, \pi_j \models \varphi)))$

A CTL formula is in negation normal form (NNF), if the symbol $\neg$ is applied only to propositional symbols. Every formula can be transformed into an equivalent formula in NNF.

The sublogic ACTL is the subset of CTL formulas that can be transformed into NNF formulas such that the temporal operators are restricted to $\{AX, AF, AG, AU, AR\}$. The sublogic ECTL is the subset of CTL formulas that can be transformed into NNF formulas such that the temporal operators are restricted to $\{EX, EF, EG, EU, ER\}$.

Definition 3.2 *Let φ be an ACTL formula. φ is true in M, denoted $M \models \varphi$, iff φ is true at all initial states of M.*

Definition 3.3 *Let φ be an ECTL formula. φ is true in M, denoted $M \models \varphi$, iff φ is true at some initial states of M.*

3.2 Bounded Semantics of CTL

Since every CTL formula can be transformed into an equivalent formula in NNF, we only consider formulas in NNF. Therefore, in the following, a formula refers to such a CTL formula unless otherwise stated. For simplicity, we fix the model under consideration to be $M = \langle S, T, I, L\rangle$, and in the sequel, M refers to this model, unless otherwise stated.

k-Path 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 [3].

Bounded Model The k-model of M is a structure $M_k = \langle S, Ph_k, I, L\rangle$ where Ph_k is the set of all different 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 [22].

Loop A loop is a k-path π such that $\pi_i = \pi_j$ for some $0 \leq i < j \leq k$. Let $lp(\pi)$ denote that π is a loop. An important property of a loop is that if π is a prefix of π', then $lp(\pi) \rightarrow lp(\pi')$. Note that this notation of loop is different from the one defined in [3], which is a loop such that the last element has a successor to some element in the loop. Such a loop does not have the property stated above.

Definition 3.4 (Bounded Semantics of CTL) *Let M_k be the k-model of M, s a state, p a propositional symbol, φ and ψ CTL formulas. The relation that φ holds on s in M_k is denoted $M_k, s \models \varphi$. Let $\pi = \pi_0 \cdots \pi_k$ be a k-path of Ph_k. Let $[n]$ denote the set $\{0, ..., n\}$. The relation $\models$ is defined as follows.*

$M_k, s \models p$ *iff* $p \in L(s)$.
$M_k, s \models \neg p$ *iff* $p \notin L(s)$
$M_k, s \models \varphi \wedge \psi$ *iff* $(M_k, s \models \varphi)$ *and* $(M_k, s \models \psi)$
$M_k, s \models \varphi \vee \psi$ *iff* $(M_k, s \models \varphi)$ *or* $(M_k, s \models \psi)$
$M_k, s \models AX\varphi$ *iff* $k \geq 1 \wedge \forall\pi(s).(M_k, \pi_1 \models \varphi)$
$M_k, s \models AF\psi$ *iff* $\forall\pi(s).(\exists i \leq k.(M_k, \pi_i \models \psi))$
$M_k, s \models AG\psi$ *iff* $\forall\pi(s).(lp(\pi) \wedge (\forall i \leq k.(M_k, \pi_i \models \psi)))$
$M_k, s \models A(\varphi U\psi)$ *iff* $\forall\pi(s).(\exists i \leq k.(M_k, \pi_i \models \psi \wedge \forall j < i.(M_k, \pi_j \models \varphi)))$
$M_k, s \models A(\varphi R\psi)$ *iff* $\forall\pi(s).(\forall i \leq k.(M_k, \pi_i \models \psi \vee \exists j < i.(M_k, \pi_j \models \varphi)) \wedge (\exists j \leq k.(M_k, \pi_j \models \varphi) \vee lp(\pi)))$
$M_k, s \models EX\varphi$ *iff* $k \geq 1 \wedge \exists\pi(s).(M_k, \pi_1 \models \varphi)$
$M_k, s \models EF\psi$ *iff* $\exists\pi(s).(\exists i \leq k.(M_k, \pi_i \models \psi))$
$M_k, s \models EG\psi$ *iff* $\exists\pi(s).(lp(\pi) \wedge (\forall i \leq k.(M_k, \pi_i \models \psi)))$
$M_k, s \models E(\varphi U\psi)$ *iff* $\exists\pi(s).(\exists i \leq k.(M_k, \pi_i \models \psi \wedge \forall j < i.(M_k, \pi_j \models \varphi)))$
$M_k, s \models E(\varphi R\psi)$ *iff* $\exists\pi(s).(\forall i \leq k.(M_k, \pi_i \models \psi \vee \exists j < i.(M_k, \pi_j \models \varphi)) \wedge (\exists j \leq k.(M_k, \pi_j \models \varphi) \vee lp(\pi)))$

This semantics of CTL is an extension of the ACTL bounded semantics given in [30]. Note that an extension of the ECTL and ECTL* bounded semantics given in [22] to a bounded semantics of CTL* has been done in [24], however the bounded semantics given in [24] is not regarded as a sound one within our framework[1]. We establish that for CTL, the bounded semantics given above is sound by first presenting some lemmas.

Lemma 3.1 *If* $M_k, s \models \varphi$*, then* $M_{k+1}, s \models \varphi$.

A formal proof is to be based on structural induction. The main arguments are explained as follows. For the first, we observe that every k-path in M_k is a prefix of a path in M_{k+1}, and every $(k+1)$-path in M_{k+1} is an extension of a path in M_k. By looking at the definition, we can be assured that there is no problem in the cases of AX, AF, AU, EX, EF, EU. By recognizing that the semantics of AG and EG can be derived from that of AR and ER (also in this bounded semantics), we only need to look further at the two cases AR and ER. We first consider the case of AR. Suppose that $M_k, s \models A(\varphi R\psi)$ holds and $M_{k+1}, s \models A(\varphi R\psi)$ does not hold. Then there is a $\pi(s)$ such that

$$\forall i \leq k+1.(M_{k+1}, \pi_i \models \psi \vee \exists j < i.(M_{k+1}, \pi_j \models \varphi)) \wedge (\exists j \leq k+1.(M_{k+1}, \pi_j \models \varphi) \vee lp(\pi))$$

(denote hereafter by (*)) does not hold. Let π' be the k-path that is at the same time a prefix of π. If $lp(\pi')$ does not hold, then $\forall i \leq k.(M_k, \pi_i \models \psi \vee \exists j < i.(M_k, \pi_j \models \varphi)) \wedge (\exists j \leq k.(M_k, \pi_j \models \varphi))$ holds. Then by the induction hypothesis, we have $\forall i \leq k.(M_{k+1}, \pi_i \models \psi \vee \exists j < i.(M_{k+1}, \pi_j \models \varphi)) \wedge (\exists j \leq k.(M_{k+1}, \pi_j \models \varphi))$. This contradicts to that (*) does not hold. If $lp(\pi')$ holds, then $\forall i \leq k.(M_k, \pi_i \models \psi \vee \exists j < i.(M_k, \pi_j \models \varphi)) \wedge lp(\pi')$ holds. Similarly, by the induction hypothesis, we have $\forall i \leq k.(M_{k+1}, \pi_i \models \psi \vee \exists j < i.(M_{k+1}, \pi_j \models \varphi))$ and since $lp(\pi')$ implies $lp(\pi)$, the only possible case that may fail (*) is that $(M_{k+1}, \pi_{k+1} \models \psi \vee \exists j < k+1.(M_{k+1}, \pi_j \models \varphi))$ does not hold. Let $\pi = \pi_0 \cdots \pi_k \pi_{k+1}$. Since $lp(\pi')$ holds, we have that $\pi_i = \pi_j$ for some $0 \leq i < j \leq k$. Let $\pi'' = \pi_0 \cdots \pi_i \pi_{j+1} \cdots \pi_k \pi_{k+1}$. Then π'' is a prefix (not necessarily a proper one) of some k-path starting with s. Since $M_k, s \models A(\varphi R\psi)$, $\forall i \leq k.(M_k, \pi''_i \models \psi \vee \exists j < i.(M_k, \pi''_j \models \varphi)) \wedge (\exists j \leq k.(M_k, \pi''_j \models \varphi) \vee lp(\pi''))$ holds. Let the position of π_{k+1} in π be $l+1$ (i.e. $\pi''_l = \pi_{k+1}$). We obtain that $(M_k, \pi''_l \models \psi \vee \exists j < l.(M_k, \pi''_j \models \varphi))$ holds. Again, by the induction hypothesis, $(M_{k+1}, \pi''_l \models \psi \vee \exists j < l.(M_{k+1}, \pi''_j \models \varphi))$ holds. By comparing π and π'', we obtain that $(M_{k+1}, \pi_{k+1} \models \psi \vee \exists j < k+1.(M_{k+1}, \pi_j \models \varphi))$ holds. This contradicts to that (*) does not hold. For the case of ER, the reasoning is similar.

Lemma 3.2 *If* $M_n, s \models \varphi$ *for some* $n \geq 0$*, then* $M, s \models \varphi$.

According to Lemma 3.1, if $M_n, s \models \varphi$ for some n, then $M_k, s \models \varphi$ holds for a large k. Given a model, all properties other than those of the form $AG\psi, A(\varphi R\psi), EG\psi, E(\varphi R\psi)$ can be witnessed by finite paths. Let k be larger than the length of such paths and also larger than the number of reachable states of M. Suppose that a property of the form $AG\psi, A(\varphi R\psi), EG\psi, E(\varphi R\psi)$ such that φ does not hold in any state of π and ψ must hold in all states of π, and therefore a prefix is not sufficient for showing the truth of such a property. Since AG and EG can be considered as subcases of AR and ER, we only consider $A(\varphi R\psi)$ and $E(\varphi R\psi)$. Assume the aforementioned situation occurs and $A(\varphi R\psi)$ holds in the bounded semantics. We want to show that $\varphi R\psi$ also holds on such a path π. For the first, the situation implies that ψ is true on every state of every k-path of which the set of states is a subset of that of π. For the second, the set of states of all these k-paths with the starting state π_0 covers the set of states of π. These two conditions guarantee that ψ is true on every state of π and therefore $\varphi R\psi$ holds on π. For the case of $E(\varphi R\psi)$, since π satisfies $(\varphi R\psi)$ in the bounded semantics such that

[1]The bounded semantics stated that a property is true iff the property is true in a bounded model for a given k (not for some $k \geq 0$ as in our framework), and since the given k is very large, it is not useful as a basis for establishing an efficient bounded model checking approach.

ψ holds on all states of π, an infinite path in which all states satisfying ψ can be constructed, therefore $E(\varphi R \psi)$ holds.

Lemma 3.3 *If $M, s \models \varphi$, then $M_k, s \models \varphi$ for some $k \geq 0$.*

By looking at the definitions, the bounded semantics is similar to the normal semantics, except that the bounded semantics has a few additional constraints. Let k be sufficiently large. Then the two conditions $k \geq 1$ and $lp(\pi)$ in the bounded semantics hold without any problem. By simplifying the bounded semantics based on this fact, the difference between the bounded semantics and the normal semantics is that the paths in the bounded semantics are restricted to k-paths, while the paths in the normal semantics are infinite paths. Therefore if $M, s \models \varphi$ holds, then $M_k, s \models \varphi$ holds for a sufficiently large k (large enough to make $lp(\pi)$ true for all k-paths). In particular, the number of reachable states of M will be such a k.

Theorem 3.1 (Soundness and Completeness) *$M, s \models \varphi$ iff $M_k, s \models \varphi$ for some $k \geq 0$.*

This theorem is a combination of the above lemmas.

Completeness Threshold The completeness threshold of the problem $M, s \models \varphi$ is defined as the least k such that if $M_k, s \models \varphi$ does not hold then $M_{k'}, s \models \varphi$ does not hold for all $k' > k$. Theorem 3.1 guarantees the existence of such a completeness threshold.

Lemma 3.4 *The completeness threshold of the problem $M, s \models \varphi$ exists.*

If the completeness threshold ct of the problem $M, s \models \varphi$ is known, then the problem is almost solved. If $ct = 0$, then we only need to check whether $M_0, s \models \varphi$ holds. If $ct > 0$, then we know that $M_{ct}, s \models \varphi$ holds and therefore $M, s \models \varphi$ also holds. Therefore the complexity of knowing the completeness threshold is the same as solving the problem.

Corollary 3.1 *Let ct_0 be an over-approximation of the completeness threshold of $M, s \models \varphi$. $M, s \models \varphi$ iff $M_k, s \models \varphi$ for some $k \leq ct_0$.*

Let $|M|$ denote the number of reachable states of M. $|M|$ is an over-approximation of the completeness threshold of $M, s \models \varphi$ for any CTL formula φ. For a given triple M, s, φ, we may use a more accurate over-approximation. Similar to the definitions in [17, 9], let the initial recurrence diameter of a state s of M be the number of states in the longest loop-free path between s and any reachable state, and the recurrence diameter of M be the number of states in the longest loop-free path between any two reachable states. Let $ct(M, s, \varphi)$ denote the completeness threshold of $M, s \models \varphi$. Let p, q be propositional formulas. Then the initial recurrence diameter of s of M is an over-approximation of $ct(M, s, A(pUq))$, while the recurrence diameter of M is an over-approximation of $ct(M, s, A(pUA(qUr)))$.

Bounded Model Checking Principle for CTL Let M be a model, s a state and φ a CTL formula. The bounded model checking principle[2] may be formulated as follows.

Let ct_0 be an over approx. of $ct(M, s, \varphi)$;
Let $k = 0$; .
If $M_k, s \models \varphi$ holds, report that φ holds;
If $k = ct_0$, report that φ does not hold;
Increase k, go to the first "if"-test;

Because CTL is closed under negation, Theorem 3.1 also provides a basis for bounded model checking and verification (emphasizing the possibility to check whether a formula is true or the negation of the formula is true without using a completeness threshold or other termination criteria)[3] of CTL properties.

Theorem 3.2 *$M, s \models \varphi$ iff there is a k such that $M_k, s \models \varphi$ and there is no k such that $M_k, s \models \neg\varphi$.*

Note that $\neg\varphi$ represents the NNF formula equivalent to $\neg\varphi$ and $M_k, s \not\models \varphi$ is not equivalent to $M_k, s \models \neg\varphi$.

Bounded Model Checking and Verification Let M be a model, s a state and φ a CTL formula. The bounded model checking and verification principle 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;

This approach is guaranteed to terminate by Theorem 3.2. One of the features of this bounded model checking and verification principle is that we do not have to worry about the completeness threshold which is important in the previous bounded model checking principle.

4 On CTL*

In this section, we discuss the possibility of extending the bounded semantics of CTL to CTL*, and prove that there

[2] We call this a principle, not a model checking approach, in the sense that a direct implementation may not be efficient for general CTL properties. Later we shall develop an implementable approach for bounded model checking and verification of ACTL formulas.

[3] This is not possible with the bounded semantics defined in [3, 22] for model checking, respectively, LTL and ACTL properties

are no such extensions in our framework of bounded semantics, in contrast to that there is a natural extension (within this framework) of the bounded semantics of ECTL to that of ECTL* [25].

We first introduce CTL* . The temporal logic CTL* was proposed in [14] as a unifying framework subsuming both CTL and LTL. This extension of CTL waives the restriction of the use of path quantifiers and path operators such that they can be used separately. Then there are two types of formulas in CTL*. One is state formulas and the other is path formulas. Let AP be a set of propositional symbols. The set of CTL* formulas over AP is defined as follows:

If $p \in AP$, then p is a state formula.
If φ_0 and φ_1 are state formulas,
then $\neg\varphi_0$, $\varphi_0 \wedge \varphi_1$ and $\varphi_0 \vee \varphi_1$ are state formulas.
If ψ is a path formula, then $E\psi$ and $A\psi$ are state formulas.
If φ is a state formula, then φ is a path formula.
If ψ_0 and ψ_1 are path formulas,
then $\neg\psi_0$, $\psi_0 \wedge \psi_1$, $\psi_0 \vee \psi_1$, $X\psi_0$, $F\psi_0$, $G\psi_0$,
$\psi_0 U \psi_1$ and $\psi_0 R \psi_1$ are path formulas.

4.1 Semantics of CTL*

Let M be a model, s a state of M, π a path of M. The relation ψ holds on π in M for a path formula ψ is denoted by $M, \pi \models \psi$, and the relation φ holds on s in M for a state formula φ is denoted by $M, s \models \varphi$.

Definition 4.1 *(Semantics of CTL*) Let φ be a state formula and ψ be a path formula. The relation $M, \pi \models \psi$ and $M, s \models \varphi$ are defined as follows.*

$M, s \models p$ *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* $M, s \models \varphi_1$
$M, s \models E\psi_0$ *iff* $\exists\pi(s).(M, \pi \models \psi_0)$
$M, s \models A\psi_0$ *iff* $\forall\pi(s).(M, \pi \models \psi_0)$
$M, \pi \models \varphi$ *iff* $M, \pi_0 \models \varphi$
$M, \pi \models \neg\psi_0$ *iff* $M, \pi \not\models \psi_0$
$M, \pi \models \psi_0 \wedge \psi_1$ *iff* $M, \pi \models \psi_0$ *and* $M, \pi \models \psi_1$
$M, \pi \models \psi_0 \vee \psi_1$ *iff* $M, \pi \models \psi_0$ *or* $M, \pi \models \psi_1$
$M, \pi \models X\psi_0$ *iff* $M, \pi^1 \models \psi_0$
$M, \pi \models F\psi_0$ *iff* $\exists k \geq 0. M, \pi^k \models \psi_0$
$M, \pi \models G\psi_0$ *iff* $\forall k \geq 0. M, \pi^k \models \psi_0$
$M, \pi \models \psi_0 U \psi_1$ *iff*
$\exists k \geq 0. \forall j < k. (M, \pi^k \models \psi_1 \wedge M, \pi^j \models \psi_0)$
$M, \pi \models \psi_0 R \psi_1$ *iff*
$\forall j \geq 0. (M, \pi^j \models \psi_1) \vee$
$\exists k \geq 0. ((M, \pi^k \models \psi_0) \wedge (\forall j \leq k. (M, \pi^j \models \psi_1))$

The restriction of CTL* to path formulas such that path quantifers (E,A) do not occur in the formulas is LTL. The restriction of CTL* to state formulas such that temporal path operators (X,F,G,U,R) and path quantifers (E,A) occur in pair and each path operator is immediately preceded by a path quantifer is CTL.

A CTL* formula is in NNF, if the negation $\neg$ is applied only to propositional symbols. Every CTL* formula can be transformed into an equivalent formula in NNF. The restriction of CTL* to NNF formulas not containing the existential path quantifier is called ACTL*. The restriction of CTL* to NNF formulas not containing the universal path quantifier is called ECTL*.

Definition 4.2 *Let φ be an ACTL* formula. φ is true in M, denoted $M \models \varphi$, iff φ is true at all initial states of M. Let φ be an ECTL* formula. φ is true in M, also denoted $M \models \varphi$, iff φ is true at some initial state of M.*

4.2 Bounded Semantics of CTL*

Let $\models_{a,k}$ be a family of bounded relations each defined as a propositional combination of simple relations with respect to $\models$ (for state formulas) as follows.

$$\bigvee_{i=1}^{m} (\bigwedge_{j=1}^{n} \models_{a,k}^{i,j})$$

Since $\models_{a,k}^{i,j}$ is a simple relation for state formulas, when evaluating formulas of the form $A\varphi$, it must be related to the corresponding path relation. For clarity, we use a different notation for the corresponding path relation. Let $\models_{a,k,p}^{i,j}$ denote the relation $\models_{a,k}^{i,j}$ for path formulas. Then $M_k, s \models_{a,k}^{i,j} A\varphi$ iff $M_k, \pi(s) \models_{a,k,p}^{i,j} \varphi$ for every k-path $\pi(s)$, according to the compositionality of the relation.

Each $\models_{a,k,p}^{i,j}$ may also be defined by a disjunction of conjunctions of simple relations for path formulas. Let such a definition be as follows.

$$\bigvee_{x=1}^{a_{i,j}} (\bigwedge_{y=1}^{b_{i,j}} \models_{a,k,p}^{i,j,x,y})$$

Suppose that $M, s \models_{a,k} \varphi$ holds. Then there is some i such that for all j and every k-path $\pi(s)$, $M_k, \pi(s) \models_{a,k,p}^{i,j} \varphi$ holds.

Let $R(i)$ be $\bigwedge_{j=1}^{n} \bigvee_{x=1}^{a_{i,j}} (\bigwedge_{y=1}^{b_{i,j}} \models_{a,k,p}^{i,j,x,y})$.

Expanding $R(i)$ to a disjunction of conjunctions of simple relations, we may write $R(i)$ as $\bigvee_{x=1}^{r} (\models_{a,k,p,i,x})$ where $\models_{a,k,p,i,x}$ is a conjunction of simple relations consistent with the definition of $R(i)$.

Suppose that $M_k, s \models_{a,k} \varphi$ holds. Then there is some i and x such that for every k-path $\pi(s)$, $M_k, \pi(s) \models_{a,k,p,i,x} \varphi$ holds.

Lemma 4.1 *Suppose a sound and complete bounded semantics with respect to $\models$ is defined by the family of bounded relations $\models_{a,k}$. Then the following hold:*

1. *If $M_k, \pi \models_{a,k,p,i,x} Gp$, then $p \in L(\pi_i)$ for all $i \in [k]$.*
2. *If $M_k, \pi \models_{a,k,p,i,x} Fp$, then $p \in L(\pi_i)$ for some $i \in [k]$.*

Proof: Suppose that $\models_{a,k}$ is such a family of relations defining the bounded semantics and $M_k, \pi \models_{a,k,p,i,x} Gp$ without requiring every π_i satisfy p. Then we can construct a model M' such that $\pi \in M'_k$ and a formula φ (a disjunction of conjunctions of formulas of the form $X^n q$ where $0 \leq n \leq k$ and q is a propositional formula characterizing the n-th state of a path) characterizing the k-paths starting at π_0 that are not identical to π. Then we have $M'_k, \pi' \models_{a,k,p,i,x} \varphi \vee Gp$ for every k-path π' starting at π_0. Then according to the completeness of $\models_{a,k}$, we obtain that $M', \pi_0 \models A(\varphi \vee Gp)$ which is obviously not true, since not every state along the path starting with π_0, not characterized by φ, satisfies p. This is a contradiction. Therefore the first property must hold. Similarly, the second property must hold.

Theorem 4.1 *There is no sound and complete bounded semantics with respect to the semantics of CTL*.*

Proof: Suppose that $\models_{a,k}$ is such a family of relations defining the bounded semantics. Let M be the model shown in Fig. 1. Let φ be $A(Gp \vee Fr)$.

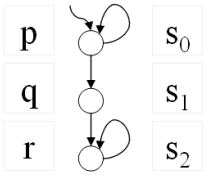

Figure 1. Model with two loops

Since $M, s_0 \models \varphi$, there is a $k \geq 0$ such that $M_k, s_0 \models_{a,k} \varphi$ according to the completeness of $\models_{a,k}$. There are following three types of k-paths in M_k that starts with s_0.

- $(s_0)^{k+1}$
- $(s_0)^k s_1$ for $k \geq 1$.
- $(s_0)^i s_1 (s_2)^j$ for $k \geq 1$ and $i + j = k$.

By Lemma 4.1 and the compositionality of the relation, the only possibility for $M_k, s_0 \models_{a,k} \varphi$ to hold is the case when $k = 0$, since $(s_0)^k s_1$ does not satisfy $Gp \vee Fr$ for any relation corresponding to $\models_{a,k,p,i,x}$. When $k = 0$, there is only one path in M_0, namely s_0. Then $M_0, s_0 \models_{a,k} \varphi$.

Let M' be the modification of M such that a self-loop from s_1 to s_1 is added, as shown in Fig. 2.

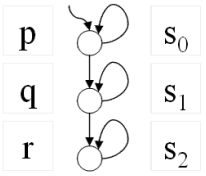

Figure 2. Model with three loops

Let M'_0 be the 0-model of M'. Since M'_0=M_0, we have $M'_0, s_0 \models_{a,k} \varphi$ for $k = 0$ as well. Then we obtain $M', s_0 \models \varphi$ according to the soundness of $\models_{a,k}$. This is not a correct conclusion. Therefore $\models_{a,k}$ does not have the properties as claimed. We conclude that the theorem holds.

On ACTL* The proof above also shows that there are no sound and complete bounded semantics for ACTL*. On the other hand, for ECTL*, a sound and complete bounded semantics is available [25].

5 Applications

The bounded semantics of CTL may serve as a basis for developing a bounded model checking algorithm for checking CTL formulas based on QBF (quantified boolean formulas)-solvers [19]. However, in this section, we will rather concentrate on checking ACTL formulas based on SAT (boolean satisfiability)-solvers, because the universal properties are considered typical in system specifications [11], and of the efficiency of SAT-solvers.

5.1 Further Development for ACTL Properties

For the practical use of the verification principle, the main problem is how to verify $M_k, s \models \varphi$ and $M_k, s \models \neg\varphi$. Since there are many bounded paths in M_k (an over estimation of the number of bounded paths is $|M|^{k+1}$), a brute-force checking of the validity of the two problems is not practical. The development in this section for the verification of ACTL properties is similar to that presented in [22, 30], only that this is now developed under the bounded semantics of CTL that admits bounded model checking and verification principle.

Definition 5.1 (Submodels) *Let $M_k = \langle S, Ph_k, I, L\rangle$ be the k-model of M. $M_k^n = \langle S, Ph_k^n, I, L\rangle$ is a submodel of M_k, if $Ph_k^n \subseteq Ph_k$ where n denotes the size of Ph_k^n. We*

write $M_k^n \leq M_k$ for this relation and call M_k^n a (k,n)-submodel of M_k.

Let M_k^b be a (k,b)-submodel of M_k. Let the relation $M_k^b, s \models \varphi$ be defined similar to the relation $M_k, s \models \varphi$, only with paths restricted to that of M_k^b. For a sufficiently large n, an ACTL formula is satisfied in a k-model iff it is satisfied in all submodels of size n, and an ECTL formula is satisfied in a k-model iff it is satisfied in some submodel of size n. Note that since we do not have $M_k, s \models \varphi$ iff $M_k, s \not\models \neg\varphi$, the above two statements are different, and need to be considered separately. Obviously, if we put $n = |M|^{k+1}$, the statements hold. However, we are interested in smaller n.

Let φ be an ACTL formula and ψ be an ECTL formula. Let $n_k^a(\varphi)$ be the least number such that for all s, $M_k, s \models \varphi$ iff $M_k', s \models \varphi$ for all $(k, n_k^a(\varphi))$-submodels M_k'. Let $n_k^e(\psi)$ be the least number such that for all s, $M_k, s \models \psi$ iff $M_k', s \models \psi$ for some $(k, n_k^e(\psi))$-submodel M_k'. We consider over-approximations of $n_k^a(\varphi)$ and $n_k^e(\psi)$.

Definition 5.2 *Let φ be an ACTL formula. $f_k(\varphi)$ is defined as follows.*

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

Lemma 5.1 *Let φ be an ACTL formula. $n_k^a(\varphi) \leq f_k(\varphi)$.*

Let φ be an ACTL formula. Then $M_k, s \models \varphi$ iff $M_k', s \models \varphi$ for all $(k, f_k(\varphi))$-submodels M_k'. The reasoning is similar to that presented in [22, 30] and is omitted, although the definition of the semantics and the definition of the over-approximation of the necessary number of paths are different[4]. Similarly, we have the following lemma.

Lemma 5.2 *Let ψ be an ECTL formula. $n_k^e(\psi) \leq f_k(\neg\psi)$.*

By Theorem 3.2 and the above two lemmas, we have the following theorem.

Theorem 5.1 *Let φ be an ACTL formula. $M, s \models \varphi$ iff there is a k such that $M_k', s \models \varphi$ for all $(k, f_k(\varphi))$-submodels M_k'' and there is no k such that $M_k'', s \models \neg\varphi$ for some $(k, f_k(\varphi))$-submodel M_k''.*

[4]For simplicity, we do not present functions for calculating over-approximations for $n_k^a(\varphi)$ and $n_k^e(\varphi)$ separately, such that the definition of $f_k()$ may in some cases seem to be unnecessarily large.

Definition 5.3 *Let φ be an ACTL formula. $M_k^b \models \varphi$ iff $M_k^b, s \models \varphi$ for all $s \in I$.*

Definition 5.4 *Let ψ be an ECTL formula. $M_k^b \models \psi$ iff $M_k^b, s \models \psi$ for some $s \in I$.*

The following statement follows from Theorem 5.1.

Corollary 5.1 *Let φ be an ACTL formula. $M \models \varphi$ iff there is a k such that $M_k' \models \varphi$ for all $(k, f_k(\varphi))$-submodels M_k'' and there is no k such that $M_k'' \models \neg\varphi$ for some $(k, f_k(\varphi))$-submodel M_k''.*

Bounded Model Checking and Verification for ACTL Let M be a model and φ an ACTL formula. The corresponding bounded model checking and verification approach is as follows.

Let $k = 0$; .
If $M_k' \models \varphi$ for all $(f_k(\varphi), k)$-models M_k', report that the property holds;
If $M_k' \models \neg\varphi$ for some $(f_k(\varphi), k)$-model M_k', report that the property does not hold;
Increase k, go to the first "if"-test;

5.2 SAT-Based Implementation

A SAT-based characterization of the above approach for ACTL can then be developed[5]. The development follows from the idea of [22, 30] and is therefore omitted. It has then been implemented (the tools is called VERBS[6] hereafter) and an experimental study has been carried out with a comparison to SMV (release 2.5.4.3), an implementation of the BDD-based symbolic model checking technique [20]. The experiments are carried out on a Sun Blade 1000 with 750 MHz and 512 MB. In the experiments, VERBS internally calls MiniSat-1.14 [12].

Model The model consists of global boolean variables $p[0]$,..., $p[n-1]$, $q[0]$, ..., $q[n-1]$, $r[0]$, ..., $r[n-1]$ and three processes p,q,r, each of which has in addition one local variable and has n transitions. The transitions of p written in the first order transition system are as follows:

$$
\begin{array}{lcl}
ss = a_0 & \longrightarrow & (p[0], ss) := (\neg p[0], a_1); \\
ss = a_1 & \longrightarrow & (p[1], ss) := (\neg p[1], a_2); \\
\vdots & & \\
ss = a_{n-2} & \longrightarrow & (p[n-2], ss) := (\neg p[n-2], a_{n-1}); \\
ss = a_{n-1} & \longrightarrow & (p[n-1], ss) := (\neg p[n-1], a_0);
\end{array}
$$

[5]For CTL, as mentioned earlier, a QBF-based characterization maybe developed, however, it is unclear whether it is possible to develop a SAT-based characterization.

[6]This is available from the webpage "http://lcs.ios.ac.cn/∼zwh/verbs/".

Within the process, the variables $p[i]$ are initially set to 0 for all $i \in \{0, ..., n-1\}$, and the variable ss (acting as the program counter, which takes one of the values of $\{a_0, ..., a_{n-1}\}$) is initially a_0 (in practice, a_i is interpreted as number i). The other two processes are similar.

Properties Let $\varphi(i)$ be $\neg p[i] \wedge \neg q[i] \wedge \neg r[i]$. The following types of properties are considered.

$$PT1 : A(\neg\varphi(i)\, R\, A(\neg\varphi(j)\, R\, \varphi(k)))$$
$$PT2 : A(\neg\varphi(i)\, R\, A(\varphi(j)\, U\, \neg\varphi(k)))$$
$$PT3 : A(\varphi(i)\, U\, A(\neg\varphi(j)\, R\, \varphi(k)))$$
$$PT4 : A(\varphi(i)\, U\, A(\varphi(j)\, U\, \neg\varphi(k)))$$

Experimental Results and Discussion There are n^3 properties of each type (i, j, k range from 0 to $n-1$). The experimental data for $n = 9$ (with 729 properties of each type) is summarized in Table 1. The explanation of the symbols in the table is as follows.

A	number of true properties of each of the types in the model
B	number of false properties of each of the types in the model
C	range of time (in seconds) for the true properties by SMV
D	range of time (in seconds) for the false properties by SMV
E	range of time (in seconds) for the true properties by VERBS
F	range of time (in seconds) for the false properties by VERBS
G	percentage of true properties in which VERBS has advantage
H	percentage of false properties in which VERBS has advantage
600+	the time is greater than the given time limit, 600 seconds

The data show that SMV, within each type of properties, is not very sensitive to the concrete properties being verified, with respect to the usage of time, on the other hand, VERBS is sensitive to the concrete properties. As the types of properties are considered, VERBS has an advantage between 18.2 and 67.5 percent (on the other hand, SMV has an advantage between 32.5 and 81.8 percent) for the properties true in the model. In average, for these properties, VERBS has advantage in 41.7 percent of cases, while SMV has advantage in 58.3 percent of the cases. For the properties false in the model, VERBS performs a lot better[7].

[7]Note that VERBS does not have counterexample generation functionality yet, while SMV uses some time on the counterexample generation.

Table 1. Summary of the Experimental Data for $n = 9$

	PT1	PT2	PT3	PT4
A	204	405	324	525
B	525	324	405	204
C	8 - 8	13-16	11-13	13 - 26
D	9 - 12	21-29	13-18	37 - 48
E	0 - 600+	0 - 600+	0- 600+	0-600+
F	0 - 31	0-8	0-41	0 - 28
G	22.5%	18.2%	67.5%	58.8%
H	94.8%	100.0%	66.1%	100.0%

Table 2. Summary of the Experimental Data for $n = 13$

	PT1	PT2	PT3	PT4
A	650	1183	1014	1547
B	1547	1014	1183	650
C	53 - 54	131 - 156	76 - 89	131 - 266
D	59 - 81	232 - 286	89 - 127	419 - 507
E	0 - 600+	0 - 600+	0 - 600+	0 - 600+
F	0 - 340	0 - 600+	0 - 600+	0 - 600+
G	24.3%	18.1%	72.8%	59.5%
H	96.8%	99.6%	72.7%	96.3%

In order to have some idea on the asymptotic behavior of the performance, we have also carried out experiments with $n = 13$ with 2197 properties of each type. The experimental data is summarized in Table 2. As the types of properties are considered, the relative advantage and disadvantage are similar (or slightly better in average) when the size of the problem increases.

For the given time limit and the experimental environment, it is expected that, for instance, when n increases to a relatively big number, the verification of the properties using SMV will be ineffective for all of the problem instances, on the other hand, a significant percentage of the problem instances can still be verified or falsified by VERBS within the time limit.

Mutual Exclusion Experiments have also been carried out with a mutual exclusion algorithm [18], with two processes. Three problem instances are considered, one for verification of mutual exclusion property, one for liveness and one for non-starvation. Let the two processes be identified by $p1$ and $p2$ and let req, cri represent the process states for having just made request for entering the critical region and having just entered the critical region, respectively. The

three properties are as follows:

$AG(\neg(p1.cri \wedge p2.cri))$
$AG((p1.req \vee p2.req) \rightarrow AF(p1.cri \vee p2.cri))$
$AG((p1.req \rightarrow AF(p1.cri)) \wedge (p2.req \rightarrow AF(p2.cri)))$

The verification process correctly verified the first two properties and falsified the last one.

Coherence Experiments have also been carried out with an asynchronous communication mechanism (ACM) with rereading and overwriting [15]. The model is specified as a set of conditional rewriting rules. The coherence property specified in [15] is that if some process starts to read and some starts to write, then the read-process will operates on the first element of the ACM and the write-process will operates on the last element of the ACM. The memory of the ACM of the instance of our model has length 6. Let the memory be denoted by $x[0]x[1]...x[5]$. Let $s \in \{1, .., 6\}$ denote the length of the part of the memory that is in use. Let read-operation on a memory cell containing a be denoted by ra, let the write-operation on a memory cell containing a be denoted by wa. Let $y \in x$ denote $x[0] = y \vee \cdots \vee x[5] = y$. Then one instance of the property is specified as follows.

$AG(s = 6 \rightarrow$
$(ra \in x \rightarrow ra = a[0] \wedge (wa \in x \rightarrow wa = a[5])))$

The verification process correctly verified this property and falsified incorrect ones, for instance, when we change $wa = a[5]$ to $wa = a[4]$ in the property.

6 Concluding Remarks

We have provided a framework for discussion of bounded semantics. This framework has formalized what is the usual understanding of bounded semantics, such that we have a framework to discuss this particular kind of semantics. The traditional bounded semantics presented in [3, 22, 25] fall into this framework and is sound and complete for their target languages, while the bounded semantics of CTL* presented in [24] are considered unsound. In this framework, we have provided a sound and complete bounded semantics for CTL formulas and identified the limitation of such semantics, namely, there are no such sound and complete bounded semantics for CTL*.

The bounded semantics of CTL differs from the previously developed bounded semantics [3, 22, 25, 30] in that the target language is closed under negation such that it can be used to check both a formula and its negation[8], and used as the basis for bounded model checking and verification in the sense discussed in Section 3. The bounded semantics of CTL is then refined in order to develop a SAT-based algorithm for checking ACTL properties. This algorithm is implemented, and experimental comparison with a BDD-based model checking tool SMV is carried out. The experimental results show that this bounded semantics based approach has advantage when a small k is sufficient for verification or error detection of given ACTL properties, while BDD-based approaches has advantage in the rest of cases. One of the important features of this approach based on CTL bounded semantics is that we do not have to be worried about over-approximations of the completeness threshold and the termination criteria which are one of the difficulties of bounded model checking and have been devoted a lot of research effort [17, 9, 16, 10, 1].

Experiments have also been carried out on models for instances of a mutual exclusion algorithm [18] and an asynchronous communication mechanism with rereading and overwriting [15]. For future research, one the theoretical side, we may further investigate bounded semantics of temporal logics, and at the practical side, we may improve the efficiency of the current bounded semantics based approaches[9], including the improvement of SAT-solving techniques and the use of other techniques such as SMT-solvers [21], in order to extend potential advantages of such an approach.

References

[1] Mohammad Awedh, Fabio Somenzi: Termination Criteria for Bounded Model Checking: Extensions and Comparison. Electr. Notes Theor. Comput. Sci. 144(1): 51-66 (2006)

[2] A. Biere, A. Cimmatti, E. Clarke, O. Strichman, and Y. Zhu. Bounded Model Checking. Advances in Computers 58, Academic Press, 2003.

[3] A. Biere, A. Cimmatti, E. Clarke, and Y. Zhu. Symbolic Model Checking without BDDs. LNCS 1579:193-207. TACAS 99.

[4] J. R. Burch, E. M. Clarke, K. L. McMillan, D. L. Dill, and J. Hwang. Symbolic model checking: 10^{20} states and beyond. LICS 1990: 428-439.

[5] R. Bryant. Graph based algorithms for boolean function manipulation. IEEE Transaction on Computers 35(8):677-691. 1986.

[8] The semantics of CTL* presented in [24] can also be used to check a formula and its negation, but it is not a sound semantics in our framework as pointed out in Section 3.2.

[9] One improvement for handling formulas involving EX has been considered in [26]. It can be further developed and used in the approached described above.

[6] Randal E. Bryant. On the Complexity of VLSI Implementations and Graph Representations of Boolean Functions with Application to Integer Multiplication. IEEE Trans. Computers 40(2): 205-213 (1991).

[7] R. Bryant. Binary decision diagrams and beyond: enabling technologies for formal verification. CAD'95:236-243. 1995.

[8] W. Chen, W. Zhang. Bounded Model Checking of ACTL formulae. TASE 2009, to appear.

[9] Edmund M. Clarke, Daniel Kroening, Joel Ouaknine, Ofer Strichman. Completeness and Complexity of Bounded Model Checking. VMCAI 2004: 85-96.

[10] E. M. Clarke, D. Kroening, J. Ouaknine, and O. Strichman. Computational challenges in bounded model checking. STTT 7(2): 174-183. 2005.

[11] E. M. Clarke, O. Grumberg and D. Peled. Model Checking. The MIT Press. 1999.

[12] Niklas Een, Niklas Sorensson: An Extensible SAT-solver. SAT 2003: 502-518.

[13] E. Allen Emerson and E. M. Clarke. Using Branching-time Temporal Logics to Synthesize Synchronization Skeletons. Science of Computer Programming 2(3):241-266. 1982.

[14] E. Allen Emerson, Joseph Y. Halpern: "Sometimes" and "Not Never" revisited: on branching versus linear time temporal logic. J. ACM 33(1): 151-178. 1986.

[15] Kyller Gorgonio, Fei Xia. Modeling and verifying asynchronous communication mechanisms using coloured Petri nets. ACSD 2008: 138-147.

[16] Keijo Heljanko, Tommi A. Junttila, Timo Latvala: Incremental and Complete Bounded Model Checking for Full PLTL. CAV 2005: 98-111.

[17] D. Kroening, O. Strichman. Efficient Computation of Recurrence Diameters. VMCAI 2003: 298-309.

[18] Leslie Lamport. A fast mutual exclusion algorithm. ACM Transactions on Computer Systems 5(1):1-11. 1987.

[19] Daniel Le Berre, Laurent Simon, Armando Tacchella: Challenges in the QBF Arena: the SAT'03 Evaluation of QBF Solvers. SAT 2003: 468-485

[20] K. L. McMillan. Symbolic Model Checking. Kluwer Academic Publisher,1993.

[21] Robert Nieuwenhuis, Albert Oliveras, Cesare Tinelli: Solving SAT and SAT Modulo Theories: From an abstract Davis–Putnam–Logemann–Loveland procedure to DPLL(T). J. ACM 53(6): 937-977 (2006).

[22] W. Penczek, B. Wozna, and A. Zbrzezny. Bounded Model Checking for the Universal Fragment of CTL. Fundamenta Informaticae 51:135-156. 2002.

[23] Mukul R. Prasad, Armin Biere, Aarti Gupta. A survey of recent advances in SAT-based formal verification. STTT 7(2): 156-173 (2005).

[24] Zhi-Hong Tao, Cong-Hua Zhou, Zhong Chen, Li-Fu Wang: Bounded Model Checking of CTL. J. Comput. Sci. Technol. 22(1): 39-43 (2007).

[25] Bozena Wozna. ATCL* properties and Bounded Model Checking. Fundam. Inform. 63(1): 65-87 (2004).

[26] L. Xu, W. Chen, Y. Xu, W. Zhang. Improved Bounded Model Checking for Universal Fragment of CTL. Journal of Computer Science and Technology 24(1):96-109. 2009.

[27] Y. Xu, W. Chen, L. Xu, W. Zhang. Evaluation of SAT-based Bounded Model Checking of ACTL Properties. Proceedings of the 1st Joint IEEE/IFIP Symposium on Theoretical Aspects of Software Engineering (TASE'07):339-348. IEEE Computer Society Press, 2007. Shanghai, China. June 5-8, 2007.

[28] W. Zhang. SAT-based verification of LTL formulas. Lecture Notes in Computer Science 4346 (FMICS 2006):277-292.

[29] W. Zhang. Verification of ACTL Properties by Bounded Model Checking. Lecture Notes in Computer Science 4739 (EUROCAST 2007):556-563.

[30] W. Zhang. Model Checking with SAT-Based Characterization of ACTL Formulas. Lecture Notes in Computer Science 4789 (ICFEM 2007):191-211.