Bounded Semantics

Wenhui Zhang
State Key Laboratory of Computer Science
Institute of Software, Chinese Academy of Sciences
Beijing, China

Abstract. The concept of bounded semantics of temporal logics has not been sufficiently specified, and this has led to definitions of bounded semantics of temporal logics that may not be appropriate with respect to the potential usefulness as a basis for developing bounded model checking approaches. On the other side, the research effort on bounded semantics has mainly focused on existentially interpreted fragments of temporal logics, due to the intricacy of defining appropriate bounded semantics for universally interpreted fragments, or for temporal logics that are closed under negation. This work[1] addresses these two problems, by defining the characteristics of bounded semantics for clarifying the concept of bounded semantics, and presenting a bounded semantics for the full set of CTL, a logic closed under negation, including possibility for specifying both existential and universal properties.
Keywords. temporal logics; formal semantics; model checking; formal methods;

1 Introduction

Bounded semantics of LTL with existential interpretation has been studied and used as the theoretical basis for SAT-based bounded model checking [6, 13]. The successfulness of this kind of model checking has led to extensive research on bounded semantics for various (fragments of) temporal logics [33, 3, 40, 28, 38, 24, 34]. This kind of approaches is considered complementary to BDD-based model checking [10, 9, 26, 12] for combating the state explosion problem [11, 7], esp. for efficient error detection [37]. However, there are two problems with this kind of research, one is that the concept of bounded semantics of temporal logics has not been properly defined; the other is that the research effort on bounded semantics has mainly focused on existentially interpreted fragments of temporal logics. The first problem may lead to definitions of bounded semantics of temporal logics (or fragments of such logics) that are not appropriate with respect to the potential usefulness as a basis for developing bounded model checking approaches, for instance, the one defined in [38]. The second problem makes it difficult to use bounded semantics as a basis for verification of universally

[1] This work merges and extends parts of the preliminary works presented at ICFEM 2007 and ICFEM 2009 [43, 45], and it was supported by the National Natural Science Foundation of China under Grant Nos. 60833001 and 61272135, and the CAS Innovation Program.

specified properties. For verification purposes, one needs to reach a completeness threshold or some termination criteria [23, 14, 20, 15, 1, 30] in order to show the non-existence of a counter-example. Ideally, 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 to make a conclusion, we increase the bound, until we have a conclusion or we run out of resources. This work addresses these two problems, by defining the characteristics of bounded semantics for clarifying the concept of bounded semantics, and presenting a bounded semantics for the full set of CTL, a logic closed under negation, including possibility for specifying both existential and universal properties. A QBF-encoding of CTL formulas based on the bounded semantics and an algorithm for QBF-based bounded correctness checking of CTL properties are also provided.

2 Bounded Semantics

Bounded semantics of temporal logics are considered, and we provide characteristics of such semantics, and an analysis of the potential applications of such semantics.

Given a set of models $\mathcal{M}$ and a set $\mathcal{L}$ of formulas interpreted on $\mathcal{M}$. A semantic relation R of $\mathcal{L}$ over $\mathcal{M}$ is a subset of $\mathcal{M} \times \mathcal{L}$. Let $\mathbf{N}$ denote the set of natural numbers $\{i \mid i \geq 0\}$. A bounded semantics is represented by a family of semantic relations $(R_i)_{i \in \mathbf{N}}$ each of which is a subset of $\mathcal{M} \times \mathcal{L}$.

The desirable properties of such a bounded semantics include: $\bigcup_{i \geq 0} R_i = R$ and $\bigcup_{i=0}^{n} R_i \neq \bigcup_{i \geq 0} R_i$ for all $n \in \mathbf{N}$. The former provides a good relation between the bounded semantics and the given semantics. The latter means that $\bigcup_{i \geq 0} R_i$ is not closed within $\bigcup_{i=0}^{n} R_i$ by any n, and is essential for the concept of bounded semantics, such that each R_i may be thought of as an approximation of $\bigcup_{i \geq 0} R_i$ and the amount of information available up to R_n is not complete for all $n \in \mathbf{N}$.

The first property is divided into soundness and completeness, and the second (referred to as the *uncloseness* property in the sequel) is refined to the well-definedness condition of bounded semantics. For formal definitions of these concepts, we narrow down our scope and assume that the temporal logics under consideration are interpreted over Kripke structures [16], which are also called transition systems in [21].

Definition 1 (Models). *Let AP be a set of propositions. A Kripke structure over AP 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 non-empty 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.*

A computation of M is an infinite sequence $s_0 s_1 \cdots$ such that $s_0 \in I$ and $(s_i, s_{i+1}) \in T$ for $i \geq 0$. A state s is reachable if it appears in some computation of M.

Definition 2 (Semantic Relations). *Let $\mathcal{M}$ be a set of models, and $\mathcal{L}$ be the set of formulas of some temporal logic interpreted on $\mathcal{M}$. A semantic relation $\models$ of $\mathcal{L}$ over $\mathcal{M}$ is a subset of $\mathcal{M} \times \mathcal{L}$. $M \models \varphi$ denotes that M satisfies φ.*

2.1 Characteristics of Bounded Semantics

For the purpose of defining the well-definedness condition, we first define the concepts of k-expansion, distinguishability and boundedness.

Definition 3. *The k-expansion of the Kripke structure $M = \langle S, T, I, L\rangle$ is a Kripke structure $M' = \langle S', T', I', L'\rangle$ defined as follows.*

S'	$= \bigcup_{i=0}^{k-1}(S_i) \cup S'_k$
T'	$= \bigcup_{i=0}^{k-1}(T_i) \cup T'_k$
I'	$= \{(s,0) \mid s \in I\}$
$L'((s,i))$	$= L(s)\ for\ all\ (s,i) \in S'$
S_0	$= I \times \{0\}$
S_i	$= \{s' \mid (s,s') \in T, s \in S_i\} \times \{i\}\ for\ all\ i \geq 1$
T_i	$= \{((s,i),(s',i+1)) \mid (s,s') \in T\}\ for\ all\ i \geq 0$
S'_k	$= S \times \{k\}$
T'_k	$= \{((s,k),(s',k)) \mid (s,s') \in T\}$

The 0-expansion of M is a copy of M with $S' = S \times \{0\}$. The $(k+1)$-expansion of M is a model such that S_0 is the set of the initial states, and the states reached in the first k-steps of the computation of M are collected respectively in $S_1, ..., S_k$, while S'_{k+1} together with T'_{k+1} is a copy of M, which constrains the rest of the computation.

Definition 4. *A Kripke structure $M'' = \langle S'', T'', I'', L''\rangle$ is a variant of the k-expansion of $M = \langle S, T, I, L\rangle$, if the following hold (where S_i and T_i are as that previously defined).*

$\bigcup_{i=0}^{k}(S_i)$	$\subseteq S''$
$\bigcup_{i=0}^{k}(T_i)$	$\subseteq T''$
I''	$= \{(s,0) \mid s \in I\}$
$L''((s,i))$	$= L(s)\ for\ all\ (s,i) \in \bigcup_{i=0}^{k}(S_i)$
$(T'' \setminus \bigcup_{i=0}^{k}(T_i)) \cap \{(s,s') \mid s' \in \bigcup_{i=0}^{k-1}(S_i)\}$	$= \emptyset$
$(T'' \setminus \bigcup_{i=0}^{k}(T_i)) \cap \{(s,s') \mid s \in \bigcup_{i=0}^{k-1}(S_i)\}$	$= \emptyset$

Informally, a variant of the k-expansion of M has the same structure as the k-expansion within the first $k+1$ levels (with initial states at level 1) of the expansion, or in other words, within the first k steps of the computations.

Definition 5. *Let $\models$ be a semantic relation of $\mathcal{L}$ over $\mathcal{M}$. Let $\varphi \in \mathcal{L}$. Two models M and M' are not distinguishable by $(\models, \varphi)$, if $M \models \varphi$ iff $M' \models \varphi$. Two models M and M' are not distinguishable by $(\models, \mathcal{L})$, if for all $\varphi \in \mathcal{L}$, they are are not distinguishable by $(\models, \varphi)$.*

Definition 6. *A semantic relation $\models_k$ of $\mathcal{L}$ over $\mathcal{M}$ is bounded, if for all $\varphi \in \mathcal{L}$, for all model $M \in \mathcal{M}$, there is an m, such that the variants of the m-expansion of M are not distinguishable by $(\models_k, \varphi)$.*

If $\models_k$ is bounded, then $\models_k$ has limited capability for utilizing the whole state space of a large model in order to evaluate the truth of a formula in the model, and therefore cannot distinguish models of which the initial parts (up to some size) are identical, for instance, only the first $m+1$ levels of an expanded model are relevant in such an evaluation.

Definition 7. *A family of semantic relations $(\models_i)_{i\in\mathbf{N}}$ is a well-defined bounded semantics, if for all $i \geq 0$, $\models_i$ is bounded.*

The well-definedness condition ensures that there is a hierarchy of the amount of information in the different semantic relations of a bounded semantics that may be used to assert the correctness of a property. However this does not directly imply the uncloseness property. For the discussion of uncloseness, one needs some additional assumptions on the semantics, that are to be defined in the next subsection.

Definition 8. *A bounded semantics $(\models_i)_{i\in\mathbf{N}}$ is sound with respect to $\models$, if the following condition holds: for every $i \geq 0$, if $M \models_i \varphi$, then $M \models \varphi$.*

The condition is equivalent to the following: if there exists $i \geq 0$ such that $M \models_i \varphi$ holds, then $M \models \varphi$ also holds.

Definition 9. *A bounded semantics $(\models_i)_{i\in\mathbf{N}}$ is complete with respect to $\models$, if the following holds: if $M \models \varphi$ then there is a $i \geq 0$ such that $M \models_i \varphi$.*

2.2 Uncloseness Property

Definition 10. *A family of semantic relations $(\models_i)_{i\in\mathbf{N}}$ satisfies the uncloseness property, if $\bigcup_{i=0}^{n}(\models_i) \neq \bigcup_{i\geq 0}(\models_i)$ for all $n \in \mathbf{N}$.*

In order to be able to state additional assumptions for reasoning about uncloseness, we define the concepts of structural equivalence, k-equivalence, expansion-equivalence, and bounded formulas.

Definition 11. *Let $M_1 = \langle S_1, T_1, I_1, L_1\rangle$ and $M_2 = \langle S_2, T_2, I_2, L_2\rangle$ be two models. M_1 and M_2 are structurally equivalent, if there is a bijective map $f : S_1 \to S_2$ such that*

$$\boxed{\begin{array}{ll} (x,y) \in T_1 & \leftrightarrow (f(x), f(y)) \in T_2 \\ x \in I_1 & \leftrightarrow f(x) \in I_2 \\ L_1(x) & = L_2(f(x)) \end{array}}$$

Let a k-model of M be a simple variant of the corresponding k-expansion of M, such that S_k only contains those states that are reachable from S_{k-1} by one step, and the transitions within S_k are all self-loops.

Definition 12. *M and M' are k-equivalent, if their k-models are structurally equivalent. M and M' are expansion equivalent, if their k-models are structurally equivalent, for all $k \geq 0$.*

Expansion-equivalence is entailed by structural equivalence as well as k-expansion. On the other hand, it is a strong form of equivalence that entails trace-equivalence [2] (a model is viewed as a set of infinite traces) and bisimulation equivalence [4] (the reader is referred to Appendix A.2 for a discussion of the relations), and therefore formulas of many commonly used temporal logics, such as LTL [35, 36], CTL [17], CTL* [18] and μ-calculus [22], are not able to distinguish expansion equivalent models.

Definition 13. *Let $\models$ be a semantic relation of $\mathcal{L}$ over $\mathcal{M}$. A formula $\varphi \in \mathcal{L}$ is bounded, if there is a k such that k-equivalent models are not distinguishable by $(\models, \varphi)$.*

A bounded formula φ is a formula such that, for every model M, when M is expanded as a set of infinite trees (with roots on the top), the nodes below level $k+1$ for some k (roots are nodes of level 1) do not affect the truth of φ.

Lemma 1. *Let $\models$ be a semantic relation of $\mathcal{L}$ over $\mathcal{M}$. Assume that expansion equivalent models are not distinguishable by $(\models, \mathcal{L})$. Let $(\models_i)_{i \in \mathbf{N}}$ be a well-defined, sound and complete bounded semantics with respect to $\models$. Then for every unbounded formula φ, for all $i \geq 0$, there exists M such that for all $j \leq i$, $M \not\models_j \varphi$, and there is a $k > i$ such that $M \models_k \varphi$.*

Proof. Let φ be an unbounded formula. Then for any m, there exist two models M^* and M^{**} such that $M^* \models \varphi$ and $M^{**} \not\models \varphi$.

- Let $M^{*(m-)}$ and $M^{**(m-)}$ denote respectively the m-model of M^* and that of M^{**}, then $M^{*(m-)}$ and $M^{**(m-)}$ are structurally equivalent. Therefore we have

$$M^{*(m-)} \models_j \varphi \text{ iff } M^{**(m-)} \models_j \varphi, \text{ for all } j \leq i.$$

- Let $M^{*(m)}$ and $M^{**(m)}$ be respectively the m-expansions of M^* and M^{**}. Then we have

$$M^{*(m)} \models \varphi \text{ iff } M^* \models \varphi, \text{ and } M^{**(m)} \models \varphi \text{ iff } M^{**} \models \varphi.$$

- Since $M^{*(m-)}$ is a variant of $M^{*(m)}$ and $M^{**(m-)}$ is a variant of $M^{**(m)}$, and $\models_0, ..., \models_i$ are bounded, let m be large enough, then for all $j \leq i$, we have

$$M^{*(m-)} \models_j \varphi \text{ iff } M^{*(m)} \models_j \varphi, \text{ and } M^{**(m-)} \models_j \varphi \text{ iff } M^{**(m)} \models_j \varphi.$$

- Combining the above with the conclusion obtained at the first item that guarantees the equivalence of $M^{*(m-)} \models_j \varphi$ and $M^{**(m-)} \models_j \varphi$, we have

$$M^{*(m)} \models_j \varphi \text{ iff } M^{**(m)} \models_j \varphi, \text{ for all } j \leq i.$$

– Since we have $M^{**} \not\models \varphi$, we also have $M^{**(m)} \not\models \varphi$. Therefore $M^{**(m)} \not\models_j \varphi$ for all $j \leq i$ (by soundness), and then $M^{*(m)} \not\models_j \varphi$ for all $j \leq i$. On the other hand, we have $M^* \models \varphi$. Therefore $M^{*(m)} \models \varphi$, and then there is a k such that $M^{*(m)} \models_k \varphi$ (by completeness). Let $M = M^{*(m)}$, then we have that needed to be proved. Therefore the lemma holds.

Theorem 1. *Let $\models$ be a semantic relation of $\mathcal{L}$ over $\mathcal{M}$. Assume that expansion equivalent models are not distinguishable by $(\models, \mathcal{L})$, and that $\mathcal{L}$ has a nonempty subset of unbounded formulas. Then a well-defined, sound and complete semantics with respect to $\models$ satisfies the uncloseness property.*

This theorem follows from Lemma 1.

Example Let $\models$ be a semantic relation of $\mathcal{L}$ over $\mathcal{M}$. Assume that expansion equivalent models are not distinguishable by $(\models, \mathcal{L})$, and that $\mathcal{L}$ has a nonempty subset of unbounded formulas. Let $(\models_i)_{i \in \mathbf{N}}$ be a family of semantic relations, where $M \models_i \varphi$ is defined as $M \models \varphi$, for all $i \in \mathbf{N}$. Obviously, $(\models_i)_{i \in \mathbf{N}}$ is sound and complete with respect to $\models$, however, it is not a valid bounded semantics for $\models$, because it does not satisfy the uncloseness property, and therefore is not a well-defined bounded semantics. This uncloseness property is intended for excluding non-interesting definitions of bounded semantics that violate the incrementality of the ability to use the information provided by the models. Examples of the definitions that satisfied this property is to be found in Section 3.2 and Section 4.1.

Discussion The above example of bounded semantics is a trivially non-valid one. Looking at the existing bounded semantics, the bounded semantics of various fragments of CTL* [18], including that of LTL with existential interpretation [6], that of the existential fragment of CTL [33], and that of the existential fragment of CTL* [40] are all well-defined, sound and complete with respect to their respective target languages. However, the bounded semantics for CTL* defined in [38] is a family of semantic relations that does not satisfy the proposed soundness condition. Properties of the bounded semantics of LTL and that of CTL* are to be further discussed in Section 4.1 and Section 4.2, respectively.

2.3 Potential Applications of Bounded Semantics

Let $\mathcal{L}$ be a language and $\mathcal{M}$ be a set of models. Let $\models$ be a semantic relation of $\mathcal{L}$ over $\mathcal{M}$. In general, a corresponding bounded semantics may be defined on a subset of $\mathcal{L}$ over a subset of $\mathcal{M}$. The potential applications of the bounded semantics may be different depending on whether it is only well-defined, sound and complete with respect to a given part of the standard semantics $\models$. We explain model checking approaches denoted *bounded verification*, *bounded model checking* and a combined approach denoted *bounded correctness checking*.

Bounded Verification Let $(\models_k)_{k\in\mathbf{N}}$ be a well-defined, sound and complete bounded semantics with respect to the semantic relation $\models$. Such a bounded semantics may be applied to bounded verification with falsification based on the used of completeness thresholds, a concept similar to the one defined in [5].

Definition 14. *The completeness threshold of the problem $M \models \varphi$ for $(\models_k)_{k\in\mathbf{N}}$ is defined as the least k such that if $M \models_k \varphi$ does not hold then $M \models_{k'} \varphi$ does not hold for all $k' > k$.*

The completeness threshold of the problem $M \models \varphi$ for $(\models_k)_{k\in\mathbf{N}}$ exists. Let ct denote this completeness threshold. The argument for the existence is divided into two cases as follows.

- Case 1: $M \models_i \varphi$ does not hold for all i.
  In this case, we have $ct = 0$, according to the definition.
- Case 2: $M \models_i \varphi$ holds for some i, and i_0 is the least one of such i's.
  In this case, we have $ct = i_0$. Therefore the completeness threshold exists.

If the completeness threshold ct of the problem $M \models \varphi$ is known, then the problem is almost solved. We have two cases.

- If $ct = 0$, then we only need to check whether $M \models_0 \varphi$ holds.
- If $ct = i_0 > 0$, then we know that $M \models_{i_0} \varphi$ holds, and therefore $M \models \varphi$ holds according to the soundness of the bounded semantics.

Since computing the completeness threshold is difficult (at least as difficult as the model checking problem), we may consider over-approximations of such a threshold. Let $ct(M, \varphi)$ denote the completeness threshold of the problem $M \models \varphi$.

Definition 15. *m is an over-approximations of $ct(M, \varphi)$, if $m \geq ct(M, \varphi)$.*

The following proposition follows from the definition of the completeness threshold and that of over-approximation.

Proposition 1. *Let ct_0 be an over-approximation of $ct(M, \varphi)$. $M \models \varphi$ holds iff $M \models_k \varphi$ for some $k \leq ct_0$.*

Let $\mathbf{N}$ be the set of natural numbers with 0 as the least element.

Let $\mu\mathcal{K}.h$ denote the least fixpoint of $h : \mathbf{N} \to \mathbf{N}$.

Let $x/2$ denote the integer division by 2.

Let $odd(x)$ denote 1 if x is odd, 0 otherwise.

Let $inc(x, y)$ denote $x + y - odd(x)$. Then $inc(x, 1)$ is the function that moves x to the closest odd number (including itself), and $inc(x, 2)$ is the function that moves x to the closest even number larger than x.

Let $ite(c, x, y)$ denote the function that returns x if c is true, and y otherwise.

The following are a fixpoint formulation and an algorithmic formulation for the bounded verification approach. Let M be a model, and $\varphi \in \mathcal{L}$ be a formula. The fixpoint formulation with the use of completeness threshold is as follows.

Let ct_0 be an over-approximation of $ct(M, \varphi)$.
Let $h^{bv}_{M,\varphi} : \mathbf{N} \to \mathbf{N}$ be a monotonic function defined by

$$h^{bv}_{M,\varphi}(\mathcal{K}) = ite((\mathcal{K}/2 \leq ct_0), ite((M \models_{\mathcal{K}/2} \varphi), inc(\mathcal{K}, 1), inc(\mathcal{K}, 2)), \mathcal{K}).$$

Corollary 1 (Bounded Verification). *Let M be a model, and $\varphi \in \mathcal{L}$ be a formula. $M \models \varphi$ iff $odd(\mu\mathcal{K}.h^{bv}_{M,\varphi}) = 1$.*

This corollary characterizes the bounded verification approach. The existence of such a least fixpoint and the correctness are guaranteed by Proposition 1. A corresponding algorithmic formulation (with some simplification with respect to the fixpoint formulation) is as follows.

Let ct_0 be an over-approximation of $ct(M, \varphi)$;
for $(k = 0; k \leq ct_0; k{+}{+})$ { if $(M \models_k \varphi$ holds) break; }
report that $M \models \varphi$ holds iff $k \leq ct_0$;

This bounded verification approach may be applied to quickly report that φ holds when a small k is sufficient for proving $M \models_k \varphi$. Note that for the algorithmic efficiency, different optimizations may be considered, for instance, by using other termination criteria, or increasing the value of k differently, however, discussions around such optimizations are not essential for the purpose of this paper.

Bounded Model Checking For bounded model checking, we may distinguish between a specification langauge $\mathcal{L}_0 \subseteq \mathcal{L}$, a language $\mathcal{L}_1 \subseteq \mathcal{L}$ for which bounded semantics is defined. In this setting, $\mathcal{L}_1$ is thought of as the negated set of formulas of $\mathcal{L}_0$, and the bounded semantics of $\mathcal{L}_1$ is defined over a subset $\mathcal{M}_1 \subseteq \mathcal{M}$. Such a bounded semantics may be applied to bounded falsification of formulas of $\mathcal{L}_0$, while verification of such formulas may be done based on the used of completeness thresholds. Before presenting this bounded model checking approach, we define a concept called *ng*-consistency for formally dealing with the mapping between $\mathcal{L}_0$ and $\mathcal{L}_1$.

Definition 16 (*ng*-Consistency). *Let $ng : \mathcal{L}_0 \to \mathcal{L}_1$ be a partial function on $\mathcal{L}$. A semantic relation $\models$ of $\mathcal{L}$ over $\mathcal{M}$ is ng-consistent, if the following holds: there is a function $wf : \mathcal{M} \to 2^{\mathcal{M}}$ such that for all $\varphi \in \mathcal{L}_0$, $M \not\models \varphi$ iff there is some $M' \in wf(M)$ such that $M' \models ng(\varphi)$.*

The function wf is called a witness function. The function ng is meant to be the negation of a formula, however, it is not necessarily a syntactically negated formula. Let $(\models_k)_{k \in \mathbf{N}}$ be a well-defined, sound and complete bounded semantics with respect to the part of semantic relation $\models$ restricted to formulas of $\mathcal{L}_1$ and models of $\mathcal{M}_1 \subseteq \mathcal{M}$.

Proposition 2. *Let $\models$ be ng-consistent with f as a witness function satisfying $wf(M) \subseteq \mathcal{M}_1$ for every $M \in \mathcal{M}$. Let $\varphi \in \mathcal{L}_0$. Let $(\models_k)_{k \in \mathbf{N}}$ be as stated above. Then $M \not\models \varphi$ iff there is a k such that $\exists M' \in wf(M).(M' \models_k ng(\varphi))$.*

Let M be a model, and φ be a formula of $\mathcal{L}_0 \subseteq \mathcal{L}$. Suppose that $\models$ is ng-consistent with wf as a witness function such that $wf(M) \subseteq \mathcal{M}_1$ for every $M \in \mathcal{M}$. The fixpoint formulation of the bounded model checking approach is as follows.

Let ct_0 be an over-approximation of $ct(M', ng(\varphi))$ for all $M' \in wf(M)$.

Let $h^{bmc}_{M,\varphi} : \mathbf{N} \to \mathbf{N}$ be a monotonic function defined by

$$h^{bmc}_{M,\varphi}(\mathcal{K}) = \\ ite((\mathcal{K}/2 \leq ct_0), ite((\exists M' \in wf(M).(M' \models_{\mathcal{K}/2} ng(\varphi))), \mathcal{K}, inc(\mathcal{K}, 2)), inc(\mathcal{K}, 1)).$$

Corollary 2 (Bounded Model Checking). *Let M be a model, and φ be a formula of $\mathcal{L}_0 \subseteq \mathcal{L}$. $M \models \varphi$ iff $odd(\mu\mathcal{K}.h^{bmc}_{M,\varphi}) = 1$.*

This corollary characterizes the bounded model checking approach. The existence of such a least fixpoint and the correctness are guaranteed by Proposition 2. A corresponding algorithmic formulation is as follows.

Let ct_0 be an over-approximation of $ct(M', ng(\varphi))$ for all $M' \in wf(M)$;
for $(k = 0; k \leq ct_0; k{+}{+})$ { if $(\exists M' \in wf(M).(M' \models_k ng(\varphi)))$ break; }
report that $M \models \varphi$ does not hold iff $k \leq ct_0$;

This bounded model checking approach may be applied to quickly report that φ does not hold when a small k is sufficient for proving $M' \models_k ng(\varphi)$ for some $M' \in wf(M)$. This approach corresponds to the usual bounded model checking approach developed for, for instance, checking of LTL, ACTL, and ACTL* properties [6, 33, 40].

Example Considering the bounded semantics in [6] as an example. Let φ be an LTL formula. Let $\models_k$ denote the bounded semantics of LTL (for the details, the reader is referred to Section 4.1). Let $\langle M, s\rangle$ denote M with I replaced by $\{s\}$. By interpreting an LTL formula φ in the context $M \models \varphi$ as a universally quantified state formula $A\varphi$ in CTL*, essentially we have the following:

$$\begin{aligned} & M \not\models A\varphi \\ \Leftrightarrow\ & \exists s \in I.(\langle M, s\rangle \models E\neg\varphi) \\ \Leftrightarrow\ & \exists k \geq 0.\exists s \in I.(\langle M, s\rangle \models_k E\neg\varphi) \\ \Leftrightarrow\ & \exists k \geq 0.\exists \langle M, s\rangle \in wf(M).(\langle M, s\rangle \models_k E\neg\varphi) \end{aligned}$$

This complies with the above bounded model checking approach with

$$\begin{aligned} \mathcal{L}_0 &= \{A\varphi \mid \varphi \in LTL\} \\ \mathcal{L}_1 &= \{E\varphi \mid \varphi \in LTL\} \\ ng(A\varphi) &= E\neg\varphi \\ \mathcal{M}_1 &= \{\langle S, T, \{s\}, L\rangle \in \mathcal{M} \mid s \in S\} \\ \models_k &\subseteq \mathcal{M}_1 \times \mathcal{L}_1 \\ wf(M) &= \{\langle M, s\rangle \in \mathcal{M}_1 \mid s \in I\} \end{aligned}$$

Let φ be an LTL formula. Then according to Proposition 2, we have a bounded model checking approach, which can be refined to be as follows.

Let ct_0 be an over-approximation of $ct(\langle M, s\rangle, E\neg\varphi)$ for all $s \in I$;
for $(k = 0; k \leq ct_0; k{+}{+})$ { if $(\exists s \in I.(\langle M, s\rangle \models_k E\neg\varphi))$ break; }
report that $M \models A\varphi$ does not hold iff $k \leq ct_0$;

Bounded Correctness Checking For a bounded semantics defined for a language closed under negation, we may apply the bounded semantics to both bounded verification and bounded model checking. Let $(\models_k)_{k\in\mathbf{N}}$ be a well-defined, sound and complete bounded semantics with respect to the semantic relation $\models$.

Proposition 3. *Let $ng : \mathcal{L} \to \mathcal{L}$ be a total function on $\mathcal{L}$. Let $\models$ be ng-consistent with wf as a witness function. Let $(\models_k)_{k\in\mathbf{N}}$ be as stated above. Then $M \models \varphi$ iff there is a k such that $M \models_k \varphi$, and $M \not\models \varphi$ iff there a k such that $\exists M' \in wf(M).(M' \models_k ng(\varphi))$.*

Let M be a model, and φ be a formula of $\mathcal{L}$. Suppose that $\models$ is ng-consistent with wf as a witness function. The fixpoint formulation of the bounded correctness checking approach, that applies the bounded semantics to checking the satisfiability of both of φ and $ng(\varphi)$ in the model, is as follows.

Let $h^{bcc}_{M,\varphi} : \mathbf{N} \to \mathbf{N}$ be a monotonic function defined by

$$h^{bcc}_{M,\varphi}(\mathcal{K}) =$$
$$ite((M \models_{\mathcal{K}/2} \varphi), inc(\mathcal{K}, 1), ite((\exists M' \in wf(M)).(M' \models_{\mathcal{K}/2} ng(\varphi))), \mathcal{K}, inc(\mathcal{K}, 2)).$$

Corollary 3 (Bounded Correctness Checking). *Let M be a model, and $\varphi \in \mathcal{L}$ be a formula. $M \models \varphi$ iff $odd(\mu\mathcal{K}.h^{bcc}_{M,\varphi}) = 1$.*

This corollary characterizes the bounded correctness checking approach. The existence of such a least fixpoint and the correctness are guaranteed by Proposition 3. One of the features of this formulation is that there is no need of completeness thresholds. This is explained as follows.

- If $M \models \varphi$, then according to completeness of the bounded semantics, we have $M \models_n \varphi$ for a sufficiently large n, and then the least fixpoint calculation terminates with $\mathcal{K} = 2k + 1$ for some $k \leq n$.
- If $M \not\models \varphi$, then $(\exists M' \in wf(M)).(M' \models ng(\varphi)$ according the ng-consistency of $\models$. Then according to completeness of the bounded semantics, we have $(\exists M' \in wf(M)).(M' \models_n ng(\varphi))$ for a sufficiently large n, and the least fixpoint calculation terminates with $\mathcal{K} = 2k$ for some $k \leq n$.

A corresponding algorithmic formulation is as follows.

for $(k = 0; 1; k{+}{+})$ { if $(M \models_k \varphi$ or $\exists M' \in wf(M)).(M' \models_k ng(\varphi)))$ break; }
report that $M \models \varphi$ holds iff $M \models_k \varphi$ holds;

Discussion Bounded verification is a direct approach for verification of given formulas. Bounded model checking is an approach aiming at quickly showing unsatisfiability of given formulas, by defining bounded semantics for the negation of such formulas. Bounded correctness checking is a combined approach relying on bounded semantics defined for both the given formulas and the negation of such formulas. Bounded model checking is more complicated than bounded verification, because the properties to be checked are not directly represented by the formulas for which the bounded semantics is defined.

3 Bounded Correctness Checking of CTL Formulas

The temporal logic CTL is considered, and we present a well-defined, sound and complete bounded semantics for CTL. Such a bounded semantics naturally leads to a bounded correctness checking approach for the verification of CTL formulas.

3.1 Computation Tree Logic

CTL is a propositional branching-time temporal logic [17] 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 represented by a Kripke structure.

Paths Let $M = \langle S, T, I, L\rangle$ be a Kripke structure. An infinite path of M is an infinite sequence of states $\pi = \pi_0\pi_1\cdots$ such that $(\pi_i, \pi_{i+1}) \in T$ for all $i \geq 0$. A computation of M is then an infinite path π of M such that $\pi_0 \in I$. 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.

Definition 17 (Semantics of CTL). *Let p be a propositional symbol, φ and ψ CTL formulas. Let $\pi = \pi_0\pi_1\cdots$ denote 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* $\exists\pi(s).(\forall k \geq 0.(M, \pi_k \models \psi \vee\ \exists j < k.(M, \pi_j \models \varphi)))$

Definition 18. *$M \models \varphi$ iff $M, s \models \varphi$ for all $s \in I$.*

Negation Normal Form 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, by applying the following equivalences.

$\neg(\neg\varphi_0)$	$\equiv (\varphi_0)$
$\neg(\varphi_0 \wedge \varphi_1)$	$\equiv (\neg\varphi_0 \vee \neg\varphi_1)$
$\neg AX\varphi_0$	$\equiv EX\neg\varphi_0$
$\neg AF\varphi_0$	$\equiv EG\neg\varphi_0$
$\neg AG\varphi_0$	$\equiv EF\neg\varphi_0$
$\neg A(\varphi_0 U\varphi_1)$	$\equiv E(\neg\varphi_0 R\neg\varphi_1)$
$\neg A(\varphi_0 R\varphi_1)$	$\equiv E(\neg\varphi_0 U\neg\varphi_1)$

Without loss of generality, in the following, we only consider formulas in NNF. Formulas not in NNF are considered as an abbreviation of the equivalent one in NNF.

Example of Unbounded Formulas CTL contains unbounded formulas. AFp is an unbounded formula, since for any k, there exists k-equivalent models that are distinguishable by AFp, for instance, we may construct $\langle S, T, I, L\rangle$ and $\langle S, T, I, L'\rangle$ such that each has $k+2$ states with a single computation $(s_0...s_k)(s_{k+1})^\omega$, and in the first model, $L(s_i) = \{p\}$ for all i, while in the second model, L' is defined by $L'(s_i) = \{p\}$ for $i \leq k$ and $L'(s_{k+1}) = \{\}$, then the two models are k-equivalent and distinguishable by AFp.

3.2 Bounded Semantics

The bounded semantics of CTL in this subsection has been presented in [45]. This bounded semantics extends that of ACTL presented in [43]. The latter

was inspired by the previous works on bounded model checking and bounded verification [6, 33, 41, 42]. In addition to the soundness and completeness discussed in [45], this subsection also discusses the well-definedness and uncloseness properties of the bounded semantics. In the following, we fix the model under consideration to be $M = \langle S, T, I, L\rangle$.

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

rs-Paths Let ζ be a finite path. ζ is a path with repeating states is denoted by $rs(\zeta)$. Then $rs(\zeta)$ implies that the number of different states appearing in ζ is less that the length of ζ. An important property is that if ζ is a prefix of ζ', then $rs(\zeta) \rightarrow rs(\zeta')$. For the idea of the use of rs-path, the reader is referred to [43] (in which it is denoted eqs).

k-Paths Let $k \geq 0$. A k-path of M is a finite path of M with length $k+1$. ζ is a k-path, if $\zeta = \zeta_0 \cdots \zeta_k$ such that $\zeta_i \in S$ for $i = 0, ..., k$ and $(\zeta_i, \zeta_{i+1}) \in T$ for $i = 0, ..., k-1$. For the idea of k-path, the reader is referred to [6]. The set of all k-paths of M is denoted M_k.

Definition 19 (Bounded Semantics of CTL). *Let s be a state, p a propositional symbol, φ and ψ CTL formulas. Let $k \geq 0$. Let $\zeta = \zeta_0 \cdots \zeta_k \in M_k$ denote a k-path. The semantics relation $M, s \models_k \varphi$ is defined as follows.*

$M, s \models_k p$	*iff* $p \in L(s)$.
$M, s \models_k \neg p$	*iff* $p \notin L(s)$
$M, s \models_k \varphi \wedge \psi$	*iff* $(M, s \models_k \varphi)$ *and* $(M, s \models_k \psi)$
$M, s \models_k \varphi \vee \psi$	*iff* $(M, s \models_k \varphi)$ *or* $(M, s \models_k \psi)$
$M, s \models_k AX\varphi$	*iff* $k \geq 1 \wedge \forall\zeta(s).(M, \zeta_1 \models_k \varphi)$
$M, s \models_k AF\psi$	*iff* $\forall\zeta(s).(\exists i \leq k.(M, \zeta_i \models_k \psi))$
$M, s \models_k AG\psi$	*iff* $\forall\zeta(s).(rs(\zeta) \wedge (\forall i \leq k.(M, \zeta_i \models_k \psi)))$
$M, s \models_k A(\varphi U \psi)$	*iff* $\forall\zeta(s).(\exists i \leq k.(M, \zeta_i \models_k \psi \wedge \forall j < i.(M, \zeta_j \models_k \varphi)))$
$M, s \models_k A(\varphi R \psi)$	*iff* $\forall\zeta(s).((\forall i \leq k.(M, \zeta_i \models_k \psi \vee \exists j < i.(M, \zeta_j \models_k \varphi))) \wedge (\exists j \leq k.(M, \zeta_j \models_k \varphi) \vee rs(\zeta)))$
$M, s \models_k EX\varphi$	*iff* $k \geq 1 \wedge \exists\zeta(s).(M, \zeta_1 \models_k \varphi)$
$M, s \models_k EF\psi$	*iff* $\exists\zeta(s).(\exists i \leq k.(M, \zeta_i \models_k \psi))$
$M, s \models_k EG\psi$	*iff* $\exists\zeta(s).(rs(\zeta) \wedge (\forall i \leq k.(M, \zeta_i \models_k \psi)))$
$M, s \models_k E(\varphi U \psi)$	*iff* $\exists\zeta(s).(\exists i \leq k.(M, \zeta_i \models_k \psi \wedge \forall j < i.(M, \zeta_j \models_k \varphi)))$
$M, s \models_k E(\varphi R \psi)$	*iff* $\exists\zeta(s).((\forall i \leq k.(M, \zeta_i \models_k \psi \vee \exists j < i.(M, \zeta_j \models_k \varphi))) \wedge (\exists j \leq k.(M, \zeta_j \models_k \varphi) \vee rs(\zeta)))$

Definition 20. *$M \models_k \varphi$ iff $M, s \models_k \varphi$ for all $s \in I$.*

Example Consider the Kripke structure in Fig. 1 and the two properties $AFEFq$ and $AGEFq$. The first one holds and the second one does not hold.

– For checking that $AFEFq$ holds, we need to check for a k, all k-paths starting from the initial states on whether there is a state on each such path satisfying EFq. Let $k = 1$. The following are all such 1-paths that need to be checked.

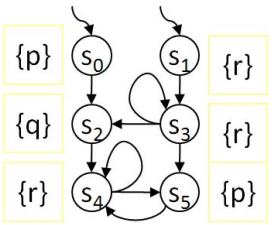

Fig. 1. Example Kripke Structure 1

$$s_0 s_2$$
$$s_1 s_3$$

Then it is sufficient to check that the state s_2 and the state s_3 satisfy EFq. It is sufficient to check that there is a state satisfying q on each of the following 1-paths, which is true according to the bounded semantics.

$$s_2 s_4$$
$$s_3 s_2$$

This proves that the model satisfies $AFEFq$ with $k = 1$.

– For checking that $AGEFq$ does not hold, we have to check there is an initial state that does not satisfy $AGEFq$, i.e., there is an initial state that satisfies $EFAG\neg q$. Then according to the bounded semantics, it is sufficient to check that there is a state on the following 2-path satisfying $AG\neg q$.

$$s_0 s_2 s_4$$

Then for the state s_4, the 2-paths starting from s_4 are as follows.

$$s_4 s_4 s_4$$
$$s_4 s_5 s_4$$

It is easily checked that all states on these two paths satisfy $\neg q$, and $rs(s_4 s_4 s_4)$ and $rs(s_4 s_5 s_4)$ hold. This proves that the model does not satisfy $AGEFq$ with $k = 2$.

In the following, we establish that the bounded semantics of CTL given in Definition 20 is well-defined, sound and complete.

Lemma 2. *Let φ be a formula. For all $M \in \mathcal{M}$, there is an m such that the variants of the m-expansion of M are not distinguishable by $(\models_k, \varphi)$.*

Let m be larger than k times the number of the occurrences of temporal operators in φ. Let M' and M'' be two variants of the m-expansion of M. Then the evaluation of $M' \models_k \varphi$ is not able to use the information beyond the first m levels of M' (according to the definition of $\models_k$). Since the first m levels of M' and that of M'' are structurally equivalent, we have that $M' \models_k \varphi$ iff $M'' \models_k \varphi$.

Theorem 2 (Well-definedness). *The family of semantic relations* $(\models_k)_{k\in\mathbf{N}}$ *is a well-defined bounded semantics.*

The well-definedness condition follows from Lemma 2.

Lemma 3. *If* $M, s \models_k \varphi$*, then* $M, s \models_{k+1} \varphi$*.*

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 $M, s \models_k A(\varphi R\psi)$ holds and $M, s \models_{k+1} A(\varphi R\psi)$ does not hold. Then there is a $\zeta \in M_{k+1}$ with $\zeta_0 = s$ such that

$$(\forall i \leq k+1.(M, \zeta_i \models_{k+1} \psi \vee \exists j < i.(M, \zeta_j \models_{k+1} \varphi)))\wedge$$
$$(\exists j \leq k+1.(M, \zeta_j \models_{k+1} \varphi) \vee rs(\zeta))$$

(denote hereafter by (*)) does not hold. Let ζ' be the k-path that is at the same time a prefix of ζ.

- Suppose that $rs(\zeta')$ does not hold.
  Then by assumption, we have
  $(\forall i \leq k.(M, \zeta_i \models_k \psi \vee \exists j < i.(M, \zeta_j \models_k \varphi))) \wedge (\exists j \leq k.(M, \zeta_j \models_k \varphi))$.
  Then by the induction hypothesis, we have
  $\forall i \leq k.(M, \zeta_i \models_{k+1} \psi \vee \exists j < i.(M, \zeta_j \models_{k+1} \varphi)) \wedge (\exists j \leq k.(M, \zeta_j \models_{k+1} \varphi))$.
  This contradicts to that (*) does not hold.
- Suppose that $rs(\zeta')$ holds.
  Then by assumption, we have
  $(\forall i \leq k.(M, \zeta_i \models_k \psi \vee \exists j < i.(M, \zeta_j \models_k \varphi))) \wedge rs(\zeta')$.
  Similarly, by the induction hypothesis, we have
  $(\forall i \leq k.(M, \zeta_i \models_{k+1} \psi \vee \exists j < i.(M, \zeta_j \models_{k+1} \varphi)))$.
  Since $rs(\zeta)$ is implied by $rs(\zeta')$, the only possible case that may fail (*) is that $(M, \zeta_{k+1} \models_{k+1} \psi \vee \exists j < k+1.(M, \zeta_j \models_{k+1} \varphi))$ does not hold.
  Let $\zeta = \zeta_0 \cdots \zeta_k \zeta_{k+1}$.
  Since $rs(\zeta')$ holds, we have $\zeta_i = \zeta_j$ for some $0 \leq i < j \leq k$.
  Let $\zeta'' = \zeta_0 \cdots \zeta_i \zeta_{j+1} \cdots \zeta_k \zeta_{k+1}$.
  Then ζ'' is a prefix (not necessarily a proper one) of some k-path starting with s.
  Since $M, s \models_k A(\varphi R\psi)$, we have
  $\forall i \leq k.(M, \zeta''_i \models_k \psi \vee \exists j < i.(M, \zeta''_j \models_k \varphi)) \wedge (\exists j \leq k.(M, \zeta''_j \models_k \varphi) \vee rs(\zeta''))$.
  Let the position of ζ_{k+1} in ζ'' be $l+1$ (i.e. $\zeta''_l = \zeta_{k+1}$).
  Then we have $(M, \zeta''_l \models_k \psi \vee \exists j < l.(M, \zeta''_j \models_k \varphi))$.
  Again, by the induction hypothesis, we have
  $(M, \zeta''_l \models_{k+1} \psi \vee \exists j < l.(M, \zeta''_j \models_{k+1} \varphi))$.
  Since $\zeta''_l = \zeta_{k+1}$ and every elements of ζ'' is an element of ζ, we have
  $(M, \zeta_{k+1} \models_{k+1} \psi \vee \exists j < k+1.(M, \zeta_j \models_{k+1} \varphi))$.
  This contradicts to that (*) does not hold.

For the case of ER, the reasoning is similar. A complete proof involving all of the different cases can be formalized based on structural induction, and is to be found in Appendix A.1.

Corollary 4. *If $M \models_k \varphi$, then $M \models_{k+1} \varphi$.*

Lemma 4. *If $M, s \models_i \varphi$ for some $i \geq 0$, then $M, s \models \varphi$.*

According to Lemma 3, if $M, s \models_i \varphi$ for some i, then $M, s \models_k \varphi$ 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 largest number among the lengths of such paths, the number of reachable states of M and the number i. We have $M, s \models_k \varphi$. Let π be an infinite path. 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 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.

Corollary 5. *If $M \models_i \varphi$ for some $i \geq 0$, then $M \models \varphi$.*

Lemma 5. *If $M, s \models \varphi$, then $M, s \models_k \varphi$ 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 $rs(\pi)$ 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 $M, s \models_k \varphi$ holds for a sufficiently large k (large enough to make $rs(\pi)$ true for all k-paths). In particular, the number of reachable states of M will be such a k.

Corollary 6. *If $M \models \varphi$, then $M \models_k \varphi$ for some $k \geq 0$.*

Theorem 3 (Soundness and Completeness). *$M \models \varphi$ iff $M \models_k \varphi$ for some $k \geq 0$.*

This theorem is a combination of Corollary 5 and Corollary 6. It was established in [45], and that restricted to ACTL formulas was established in [43].

Lemma 6. *Expansion equivalent models are not distinguishable by* $(\models, CTL)$.

This follows from that expansion equivalent models are bisimulation equivalent, and CTL does not distinguish such models. The proof is to be found in Appendix A.2.

Corollary 7 (Uncloseness). $\bigcup_{k=0}^{n}(\models_k) \neq \bigcup_{k\geq 0}(\models_k)$ *for all* $n \geq 0$.

This corollary follows from Theorem 1, since $(\models_k)_{k\in\mathbf{N}}$ is well-defined, sound and complete with respect to $\models$, and $(\models, CTL)$ does not distinguish models that are expansion equivalent , according to Lemma 6, and there are unbounded formulas in CTL (an example of unbounded formulas was given at the end of Section 3.1).

3.3 Bounded Correctness Checking

The bounded semantics of CTL may be applied to the verification of finite state systems.

Bounded Verification According to Proposition 1, we have a bounded verification algorithm as follows.

Let ct_0 be an over-approximation of $ct(M, \varphi)$;
for $(k = 0; k \leq ct_0; k{+}{+})$ { if $(M \models_k \varphi$ holds) break; }
report that $M \models \varphi$ holds iff $k \leq ct_0$;

Bounded Correctness Checking Since CTL is a language closed under negation, we may also develop a bounded correctness checking algorithm.

Proposition 4. *Let ng be a function that maps an NNF formula to an NNF formula equivalent to the negation of the formula. The standard semantics of CTL is ng-consistent with a witness function wf defined by* $wf(\langle S, T, I, L\rangle) = \{\langle S, T, \{s\}, L\rangle \mid s \in I\}$.

Then according to Proposition 3, we have a bounded correctness checking approach, which can be refined to be as follows.

for $(k = 0; 1; k{+}{+})$ { if $(M \models_k \varphi$ or $\exists s \in I.(M, s \models_k \neg\varphi))$ break; }
report that $M \models \varphi$ holds iff $M \models_k \varphi$ holds;

Note that $\neg\varphi$ represents the NNF formula equivalent to $\neg\varphi$. The remaining problem is then to have an appropriate algorithm for checking $M \models_k \varphi$. This may be done by transforming such a problem into a QBF validity checking problem [8], and using QBF-reasoning [25, 19] to check the problem.

3.4 QBF-based Bounded Correctness Checking

The components of the model $M = \langle S, T, I, L\rangle$ can be represented by Boolean formulas as follows. Let $|X|$ denote the size of the set X. Let $m \geq 0$ such that $|S| \leq 2^m$. Let $x = \{x_1, ..., x_n\}$ be a set of propositional variables. Let $\Phi(x)$ denote the set of propositional formulas over x. Let $x' = \{z' \mid z \in x\}$. Let $\Phi(x, x')$ denote the set of propositional formulas over the set of variables $x \cup x' = \{x_1, ..., x_m, x'_1, ..., x'_m\}$. Let $\Sigma = \{0, 1\}^m$ and $\Sigma^2 = \Sigma \times \Sigma$.

- Let $\sigma = (a_1, ..., a_m) \in \Sigma$.
  Let $\sigma \models \varphi$ denote $\varphi^{a_1,....,a_m}_{x_1,...,x_m} = 1$ for $\varphi \in \Phi(x)$.
  For brevity, we also use $(\varphi)^\sigma_x$ to denote $\varphi^{a_1,....,a_m}_{x_1,...,x_m}$ when $\sigma = (a_1, ..., a_m)$.
- Let $\sigma = (a_1, ..., a_m, a'_1, ..., a'_m,) \in \Sigma^2$.
  Let $\sigma \models \varphi$ denote $\varphi^{a_1,....,a_m,a'_1,....,a'_m}_{x_1,...,x_m,x'_1,...,x'_m} = 1$ for $\varphi \in \Phi(x, x')$.
- Let $[[\varphi]]$ denote $\{\sigma \mid \sigma \models \varphi\}$.
  Let $f : S \to \Phi(x)$ satisfy $|[[f(s)]]| = 1$, and $s_1 \neq s_2 \to [[f(s_1)]] \neq [[f(s_2)]]$.
  Let $f(s, v)$ denote $(f(s))^v_x$, i.e., $(f(s))^{v_1,....,v_m}_{x_1,...,x_m}$ with $v = \{v_1, ..., v_m\}$.
  Let $f_2 : S^2 \to \Phi(x, x')$ be defined by $f_2(s_1, s_2) = (f(s_1) \wedge f(s_2, x'))$.

The mapping f may be given explicitly by making an enumeration of S and then assigning a formula to each $s \in S$, however for our purpose, it is sufficient to know that f establishes an injective mapping from S to Σ via $\Phi(x)$ such that every state of S is represented by a unique element of Σ. The mapping f_2 depends on f and establishes an injective mapping from $T \subseteq S^2$ to Σ^2. Let ρ_T, ρ_I and ρ_p for each $p \in AP$ be defined as follows.

$$\boxed{\begin{array}{l}\rho_I(x) = \bigvee_{s\in I}(f(s)).\\ \rho_T(x, x') = \bigvee_{(s_1,s_2)\in T}(f_2(s_1, s2)).\\ \rho_p(x) = \bigvee_{\{s \mid p\in L(s)\}}(f(s)).\end{array}}$$

Then ρ_I represents the set of the initial states I. ρ_T represents the transition relation T. ρ_p represents the set of states that satisfy p. Let $g : S \cup T \to \Sigma \cup \Sigma^2$ be defined by

$g(w) = \sigma$ such that
if $w \in S$, then $[[f(w)]] = \{\sigma\}$, and if $w \in T$, then $[[f_2(w)]] = \{\sigma\}$.

Then g is an injective mapping from S and T to Σ and Σ^2. Following from the definition, we have the following.

$$\boxed{\begin{array}{ll}s \in I & \text{iff } g(s) \models \rho_I\\ (s_1, s_2) \in T & \text{iff } g(s_1, s_2) \models \rho_T\\ p \in L(s) & \text{iff } g(s) \models \rho_p\end{array}}$$

Symbolic Representation of k-Paths Let $k \geq 0$. Let $u_0, ..., u_k$ be a finite sequence of state variables (each of the state variables is represented by a set of m propositional variables, i.e., a copy of x). The sequence $u_0, ..., u_k$ (denoted by $\vec{u}$) is intended to be used as a representation of a k-path of M. This is captured by the following definition of $P_k(\vec{u})$.

Definition 21.

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

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 $rs_k(\vec{u})$ denote that the k-path represented by $\vec{u}$ is a repeating state path. Formally, we have the following definition of $rs_k(\vec{u})$.

Definition 22. *Let u_i be represented by $\{u_i^1, ..., u_i^m\}$ for $i = 0, ..., k$. Let $u_i = u_j$ denote $u_i^1 \leftrightarrow u_j^1 \wedge \cdots \wedge u_i^m \leftrightarrow u_j^m$.*

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

Definition 23 (Transformation of CTL Formulas). *Let $k \geq 0$. Let v be a state variable and φ be an CTL formula. The encoding $[[\varphi, v]]_k$ is defined as follows.*

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

In the above encoding, each time $\forall\vec{u}$ or $\exists\vec{u}$ is encountered, a fresh set of variables is used, such that it does not quantify over variables that already have been given a meaning in the formula.

The following theorem follows from the transformation scheme.

Theorem 4. *Let φ be a CTL formula. $M, s \models_k \varphi$ iff $f(s, v) \rightarrow [[\varphi, v]]_k$ is valid.*

The formula $f(s, v)$ restricts the satisfying assignment of v to $g(s)$, the representation of s in Σ. $f(s, v) \rightarrow [[\varphi, v]]_k$ is valid iff $([[\varphi, v]]_k)_v^{g(s)}$ is valid.

Corollary 8. *Let φ be a CTL formula. $M \models_k \varphi$ iff $\forall v.(\rho_I(v) \rightarrow [[\varphi, v]]_k)$, and $\exists s \in I.(M, s \models \neg\varphi)$ iff $\exists v.(\rho_I(v) \wedge [[\neg\varphi, v]]_k)$.*

The formula $\rho_I(v)$ restricts the satisfying assignment of v to the representations of $s \in I$ in Σ. $\rho_I(v) \rightarrow [[\varphi, v]]_k$ is valid iff $([[\varphi, v]]_k)^\sigma_v$ is valid for all $\sigma \models \rho_I(v)$, and $\rho_I(v) \wedge [[\neg\varphi, v]]_k$ is satisfiable iff $([[\neg\varphi, v]]_k)^\sigma_v$ is satisfiable for some $\sigma \models \rho_I(v)$. Combining Corollary 8 and the bounded correctness checking approach given at the beginning of this section, we have the following algorithm.

QBF-based Bounded Correctness Checking Algorithm Let φ be a CTL formula. The corresponding QBF-based bounded correctness checking algorithm for $M \models \varphi$ is then as follows.

for $(k = 0;1;k{+}{+})$ if $(\forall v.(\rho_I(v) \rightarrow [[\varphi, v]]_k)$ or $\exists v.(\rho_I(v) \wedge [[\neg\varphi, v]]_k))$ break;
report that $M \models \varphi$ holds iff $\forall v.(\rho_I(v) \rightarrow [[\varphi, v]]_k)$ holds;

3.5 An Illustrative Example of Bounded Correctness Checking

The example is a concurrent program representing a formulation of Peterson's mutual exclusion algorithm [27] as a first order transition system [32]. Let a, b be variables of enumeration type which have respectively the domain $\{s_0, ..., s_3\}$ and $\{t_0, ..., t_3\}$. Let x, y, t be variables of Boolean type. The program consists of two processes: A and B with the following specification:

Process A:	
$a = s_0$	$\longrightarrow (y, t, a) := (1, 1, s_1)$
$a = s_1 \wedge (x = 0 \vee t = 0)$	$\longrightarrow (a) := (s_2)$
$a = s_2$	$\longrightarrow (y, a) := (0, s_3)$
$a = s_2$	$\longrightarrow (a) := (s_2)$
$a = s_3$	$\longrightarrow (y, t, a) := (1, 1, s_1)$
Process B:	
$b = t_0$	$\longrightarrow (x, t, b) := (1, 0, t_1)$
$b = t_1 \wedge (y = 0 \vee t = 1)$	$\longrightarrow (b) := (t_2)$
$b = t_2$	$\longrightarrow (x, b) := (0, t_3)$
$b = t_2$	$\longrightarrow (b) := (t_2)$
$b = t_3$	$\longrightarrow (x, t, b) := (1, 0, t_1)$

Let the formula specifying the set of the initial states be $a = s_0 \wedge b = t_0 \wedge x = y = 0$. The value of t is arbitrary at the initial state. The following explains the meaning of some of the constants.

$a = s_1$: process A waits for the critical region
$a = s_2$: process A is in the critical region
$a = s_3$: process A has left the critical region
$b = t_1$: process B waits for the critical region
$b = t_2$: process B is in the critical region
$b = t_3$: process B has left the critical region

Let α_i denote $a = s_i$ and β_i denote $b = t_i$. We formulate a set of properties for the program as follows.

	T/F	k
$AF(\alpha_2 \vee \beta_2)$	T	3
$AG(\neg(\alpha_2 \wedge \beta_2))$	T	10
$AG((\alpha_1) \rightarrow AF(\alpha_2 \vee \beta_2))$	T	10
$AG((\alpha_1) \rightarrow AF(\alpha_2))$	F	2
$AG((\alpha_1) \rightarrow EF(\alpha_2))$	T	10
$A((\neg\alpha_2)UA((\alpha 2)U(\neg\alpha_2)))$	T	0
$A((\neg\alpha_2)U((\alpha_2) \wedge A((\alpha_2)U(\neg\alpha_2))))$	F	2
$A((\neg\alpha_2)U((\alpha_2) \wedge E((\alpha_2)U(\neg\alpha_2))))$	F	3
$E((\neg\alpha_2)U((\alpha_2) \wedge E((\alpha_2)U(\neg\alpha_2))))$	T	2

Explanation The first 5 properties are the usual ones for mutual exclusion algorithms, including safety property, liveness property, and non-starvation (which does not necessarily hold for various formulations of mutual exclusion algorithms). The rest are properties trying to establish whether the computation tree has certain patterns involving $(\neg\alpha_2)$-states and (α_2)-states. The column indicated by T/F shows whether the property holds in the program model. The column indicated by k shows the value of the least k in the semantic relation $\models_k$ for proving or falsifying each of the properties. The table shows that 6 of the 9 properties hold in the program model, and the other 3 properties do not hold. The values of k for proving or falsifying the 9 properties range from 0 to 10.

Discussion This kind of approaches has advantages, comparing with the traditional symbolic model checking [26, 16], when it is possible to determine the truth of a property (either verified or falsified) with a relatively small k (in the extreme cases, $k = 0$ is sufficient for verifying or falsifying a property).

Complexity Issues The worst-case complexity of the QBF-based bounded correctness checking is the complexity of solving a PSPACE-complete problem with the input size exponential in the number of nested temporal operators. This complexity is much higher than that of symbolic model checking. The potential practical value of this approach is that it may achieve advantages when a relatively small k is sufficient for determining the truth of a property, and therefore may be used as a complementary approach to symbolic model checking. The existence of such advantageous situations has been shown by the above example.

4 Discussions and Related Works

This section provides a discussion on two of the existing bounded semantics that are mentioned in Section 2, the bounded semantics of LTL [6] and that of CTL* [38]. A discussion on the difficulty of defining a sound and complete bounded semantics for CTL* , and a general description of related works are provided.

4.1 On a Bounded Semantics of LTL

The bounded semantics of LTL to be discussed is the one defined in [6]. LTL is a logic introduced by Pnueli as a specification language for concurrent programs [35, 36].

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

$$\Phi ::= p \mid \neg\Phi \mid \Phi \wedge \Phi \mid \Phi \vee \Phi \mid X\,\Phi \mid F\,\Phi \mid G\,\Phi \mid \Phi\,U\,\Phi \mid \Phi\,R\,\Phi$$

Semantics The semantics of LTL is defined with respect to paths of Kripke structured. Let $M = \langle S, T, I, L\rangle$ be a Kripke structure.

Definition 24. *Let p be a propositional symbol, φ and ψ LTL formulas. Let $\pi = \pi_0\pi_1\cdots$ be an infinite path of M. The relation $\pi \models \varphi$ is defined as follows.*

$\pi \models p$	*iff* $p \in L(\pi_0)$
$\pi \models \neg\varphi$	*iff* $\pi \not\models \varphi$
$\pi \models \varphi \wedge \psi$	*iff* $\pi \models \varphi$ *and* $\pi \models \psi$
$\pi \models \varphi \vee \psi$	*iff* $\pi \models \varphi$ *or* $\pi \models \psi$
$\pi \models X\varphi$	*iff* $\pi^1 \models \varphi$
$\pi \models F\varphi$	*iff* $\exists k \geq 0.\pi^k \models \varphi$
$\pi \models G\varphi$	*iff* $\forall k \geq 0.\pi^k \models \varphi$
$\pi \models \varphi U\psi$	*iff* $\exists k \geq 0.\forall j < k.(\pi^k \models \psi \wedge \pi^j \models \varphi)$
$\pi \models \varphi R\psi$	*iff* $\forall j \geq 0.(\pi^j \models \psi) \vee\ \exists k \geq 0.((\pi^k \models \varphi) \wedge (\forall j \leq k.(\pi^j \models \psi))$

Definition 25. *$M, s \models^E \varphi$ iff there is an infinite path π with $\pi_0 = s$ such that $\pi \models \varphi$.*

This semantic definition uses an existential interpretation for the satisfiability, in order to make a correspondence with the bounded semantics defined later.

Definition 26. *$M \models^E \varphi$ iff $M, s \models^E \varphi$ for some $s \in I$.*

Negation Normal Form An LTL formula is in negation normal form (NNF), if $\neg$ is applied only to proposition symbols. Let *true* denote $p \vee \neg p$ for a given proposition symbol p. Every LTL formula can be transformed into an equivalent LTL formula in NNF without the use of the temporal operators F, R by applying the following equivalences.

$\neg\neg\varphi$	$= \varphi$
$\neg(\varphi \wedge \psi)$	$= (\neg\varphi \vee \neg\psi)$
$\neg X\varphi$	$= X\neg\varphi$
$\neg F\varphi$	$= G\neg\varphi$
$\neg(\varphi U\psi)$	$= \neg\varphi R\neg\psi$
$F\varphi$	$= \mathit{true}\, U\varphi$
$\varphi R\psi$	$= (\psi U(\varphi \wedge \psi)) \vee G\psi$

Without loss of generality, in the bounded semantics, we only consider NNF formulas constructed from propositions and negation of propositions with $\vee$, $\wedge$, X, G, and U. A formula not constructed this way is considered as an abbreviation of the equivalent one constructed this way.

Example of Unbounded Formulas LTL contains unbounded formulas. Ap is an unbounded formula, since for any k, there exists k-equivalent models that are distinguishable by Fp, for instance, we may construct $\langle S, T, I, L\rangle$ and $\langle S, T, I, L'\rangle$ such that each has $k+2$ states with a single computation $(s_0...s_k)(s_{k+1})^\omega$, and in the first model, $L(s_i) = \{p\}$ for all i, while in the second model, L' is defined by $L'(s_i) = \{p\}$ for $i \leq k$ and $L'(s_{k+1}) = \{\}$, then the two models are k-equivalent and distinguishable by Fp.

(k,l)-Loops A (k,l)-loop is a k-path $\pi = \pi_0 \cdots \pi_k$ such that $\pi' = (\pi_0 \cdots \pi_k)(\pi_l \cdots \pi_k)^\omega$ is an infinite path of M.

k-Loops A k-loop is a k-path such that it is a (k,l)-loop for some $0 \leq l \leq k$. For the idea of (k,l)-loops and k-loops, the reader is referred to [6].

Bounded Semantics The following definition of bounded semantics and the proof of its soundness and completeness are according to [6].

Definition 27 (Bounded Semantics for a Loop). *Let $k \geq 0$ and π be a k-loop. Then an LTL formula φ is true on π, written $\pi \models_k \varphi$, iff $\pi' \models \varphi$ with $\pi' = (\pi_0 \cdots \pi_k)(\pi_l \cdots \pi_k)^\omega$ for some $0 \leq l \leq k$.*

Definition 28 (Bounded Semantics without a Loop). *Let $k \geq 0$ and π be a k-path which is not a k-loop. Then an LTL formula φ is true on π, written $\pi \models_k \varphi$, iff $\pi \models_k^0 \varphi$ where:*

$\pi \models_k^i p$	iff $p \in L(\pi_i)$
$\pi \models_k^i \neg p$	iff $\pi \not\models_k^i p$
$\pi \models_k^i \varphi \wedge \psi$	iff $\pi \models_k^i \varphi$ *and* $\pi \models_k^i \psi$
$\pi \models_k^i \varphi \vee \psi$	iff $\pi \models_k^i \varphi$ *or* $\pi \models_k^i \psi$
$\pi \models_k^i X\varphi$	iff $i < k$ *and* $\pi \models_k^{i+1} \varphi$
$\pi \models_k^i G\varphi$	iff *false*.
$\pi \models_k^i \varphi U \psi$	iff $\exists j \in \{i, ..., k\}.\ \forall n \in \{i, ..., j-1\}.(\pi \models_k^j \psi \wedge \pi \models_k^n \varphi)$

Note that $\pi \models_k^i G\varphi$ is *false* by definition if the k-path is not a k-loop. This is explained by that a global property can only be witnessed by an infinite path (or a path with a loop).

Definition 29. *$M, s \models_k \varphi$ iff there is a k-path π with $\pi_0 = s$ such that $\pi \models \varphi$.*

Definition 30. *$M \models_k \varphi$ iff $M, s \models_k \varphi$ for some $s \in I$.*

Proposition 5. *The bounded semantics $(\models_k)_{k \in \mathbf{N}}$ is well-defined, sound and complete with respect to the existential semantics $\models^E$.*

This is explained as follows.

Well-definedness The family of semantic relations $(\models_k)_{k\in\mathbf{N}}$ is a well-defined bounded semantics. This follows from that for each k, the relation $\models_k$ is only able of utilizing partial information of a given model, when the model is large enough. This is further explained as follows. Let φ be a formula. For all $M \in \mathcal{M}$, then the variants of the $(k+1)$-expansion of M are not distinguishable by $(\models_k, \varphi)$, since the first $k+1$ levels of such variants are identical and loop-free, and $\models_k$ cannot utilize the information beyond the first $k+1$ levels.

Soundness $M \models^E \varphi$ if $M \models_k \varphi$ for some $k \geq 0$. This follows from that $\pi \models_k \varphi$ implies that π can be extended to an infinite path π' such that $\pi' \models \varphi$ [6].

Completeness If $M \models^E \varphi$, then $M \models_k \varphi$ for some $k \geq 0$. This follows from that if there is a computation $\pi = \pi_0\pi_1 \cdots$ of M such that $\pi \models \varphi$ holds, then there is a k-loop π' starting π_0 with k bounded by $|S| \cdot 2^{|\varphi|}$ such that $\pi' \models_k \varphi$ holds [6].

Uncloseness We have the following lemma.

Lemma 7. *Expansion equivalent models are not distinguishable by* $(\models^E, LTL)$.

This follows from that expansion equivalent models are bisimulation equivalent (according to Lemma 8), which implies that they are trace equivalent [2], and LTL does not distinguish trace equivalent models.

Corollary 9 (Uncloseness). $\bigcup_{k=0}^{n}(\models_k) \neq \bigcup_{k\geq 0}(\models_k)$ *for all* $n \geq 0$.

This corollary follows from Theorem 1, since $(\models_k)_{k\in\mathbf{N}}$ is well-defined, sound and complete with respect to $\models$, and $(\models^E, LTL)$ does not distinguish models that are expansion equivalent , according to Lemma 7, and there are unbounded formulas in LTL.

Bounded Model Checking That remains is to relate the bounded semantics to the bounded model checking approach. Let $\mathcal{L} = \{A\varphi, E\varphi \mid \varphi \in LTL\}$.

Definition 31. $M \models E\varphi$ *iff* $M \models^E \varphi$, *and* $M \models A\varphi$ *iff* $\pi \models \varphi$ *for all every computation* π *of* M.

Then we have a well-defined, sound and complete bounded semantics for the language $\mathcal{L}_1 = \{E\varphi \mid \varphi \in LTL\}$ with respect to $\models$, and therefore a bounded model checking approach for the target langauge $\mathcal{L}_0 = \{A\varphi \mid \varphi \in LTL\}$ as explained in Section 2.3.

4.2 On a Bounded Semantics of CTL*

The bounded semantics of CTL* to be discussed is the one defined in [38]. The temporal logic CTL* was proposed in [18] as a unifying framework subsuming both CTL and LTL. There are two types of formulas in CTL*. One is state formulas and the other is path formulas.

Syntax 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.

Semantics Let $M = \langle S, T, I, L\rangle$ be a Kripke structure. Let s be a state of M, and π an infinite path of M. The relation that ψ holds on π of M for a path formula ψ is denoted by $M, \pi \models \psi$, and the relation that φ holds on s of M for a state formula φ is denoted by $M, s \models \varphi$.

Definition 32. *Let p denote a proposition symbol, φ_0, φ_1 denote state formulas, and ψ_0, ψ_1 denote path formulas. 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*
$\forall j \geq 0. (M, \pi^j \models \psi_1) \vee \exists k \geq 0. ((M, \pi^k \models \psi_0) \wedge (\forall j \leq k. (M, \pi^j \models \psi_1))$

A CTL* formula is in negation normal form (NNF), if $\neg$ is applied only to proposition symbols. Every CTL* formula can be transformed into an equivalent CTL* formula in NNF by applying the following equivalences.

$\neg\neg\varphi$	$= \varphi$
$\neg(\varphi \wedge \psi)$	$= (\neg\varphi \vee \neg\psi)$
$\neg X\varphi$	$= X\neg\varphi$
$\neg F\varphi$	$= G\neg\varphi$
$\neg(\varphi U \psi)$	$= \neg\varphi R \neg\psi$
$\neg E\varphi$	$= A\neg\varphi$

Without loss of generality, in the bounded semantics, we only consider NNF formulas. A formula not in NNF is considered as an abbreviation of the equivalent one in NNF.

Refined k-Paths A refined k-path is a pair (ζ, l) where ζ is a k-path and $l \leq k$, which denotes the (k,l)-loop of ζ if $(\zeta_k, \zeta_l) \in T$, otherwise the k-path ζ. For the idea of this notation, the reader is referred to [40].

Bounded Semantics Let $M, [(\zeta, l), n)] \models \psi$ denote the relation that the path formula ψ holds along the the suffix of (ζ, l) starting at position $n+1$ (the state at the first position is ζ_0). Let $M, s \models_k \varphi$ denote that φ holds on s of M by the semantic relation $\models_k$. The following definition of these relations is according to [38].

Definition 33. *Let $k > 0$. Let p denote a proposition symbol, φ_0, φ_1 denote state formulas, and ψ_0, ψ_1 denote path formulas. Let $[i, j]$ denote the set $\{i, i+1, ..., j\}$ of numbers. Let $loop(\zeta) = \{l \mid (\zeta_k, \zeta_l) \in T\}$. The relation $M, s \models_k \varphi$ and the auxiliary relation $M, [(\zeta, l), n)] \models_k \psi$ (for brevity, M is omitted in the following) are defined as follows.*

$s \models_k p$ *iff* $p \in L(s)$
$s \models_k \neg p$ *iff* $p \notin L(s)$
$s \models_k \varphi_0 \wedge \varphi_1$ *iff* $s \models_k \varphi_0$ *and* $s \models_k \varphi_1$
$s \models_k \varphi_0 \vee \varphi_1$ *iff* $s \models_k \varphi_0$ *or* $s \models_k \varphi_1$
$s \models_k E\psi_0$ *iff* $\exists \zeta(s). \exists l \leq k. ([(\zeta, l), 0] \models \psi_0)$
$s \models_k A\psi_0$ *iff* $\forall \zeta(s). \forall l \leq k. ([(\zeta, l), 0] \models \psi_0)$
$[(\zeta, l), n] \models \psi$ *iff* $\zeta_n \models_k \psi$ *if* ψ *is a state formula*
$[(\zeta, l), n] \models \psi_0 \wedge \psi_1$ *iff* $[(\zeta, l), n] \models \psi_0$ *and* $[(\zeta, l), n] \models \psi_1$
$[(\zeta, l), n] \models \psi_0 \vee \psi_1$ *iff* $[(\zeta, l), n] \models \psi_0$ *or* $[(\zeta, l), n] \models \psi_1$
$[(\zeta, l), n] \models X\psi_0$ *iff*
$((k < n) \wedge ([(\zeta, l), n+1] \models \psi_0)) \vee ((k = n) \wedge (l \in loop(\zeta)) \wedge ([(\zeta, l), l] \models \psi_0))$
$[(\zeta, l), n] \models F\psi_0$ *iff*
$(\exists j \in [n, k]. ([(\zeta, l), j] \models \psi_0)) \vee ((l \in loop(\zeta)) \wedge (\exists j \in [l, n-1]. ([(\zeta, l), j] \models \psi_0)))$
$[(\zeta, l), n] \models G\psi_0$ *iff*
$l \in loop(\zeta) \wedge \bigwedge_{j=min(n,l)}^{k} ([(\zeta, l), j] \models \psi_0)$
$[(\zeta, l), n] \models \psi_0 U \psi_1$ *iff*
$\exists j \in [n, k]. (([(\zeta, l), j] \models \psi_1) \wedge \forall i \in [n, j-1]. ([(\zeta, l), i] \models \psi_0)) \vee$
$((l \in loop(\zeta)) \wedge \exists j \in [l, n-1]. (([(\zeta, l), j] \models \psi_1) \wedge$
$\forall i \in [n, k]. ([(\zeta, l), i] \models \psi_0) \wedge \forall i \in [l, j-1]. ([(\zeta, l), i] \models \psi_0)))$
$[(\zeta, l), n] \models \psi_0 R \psi_1$ *iff*
$\exists j \in [n, k]. (([(\zeta, l), j] \models \psi_0) \wedge \forall i \in [n, j]. ([(\zeta, l), j] \models \psi_1)) \vee$
$((l \in loop(\zeta)) \wedge (\bigwedge_{j=min(n,l)}^{k} ([(\zeta, l), j] \models \psi_1) \vee$
$\exists j \in [l, n-1]. (([(\zeta, l), j] \models \psi_0) \wedge$
$\forall i \in [n, k]. ([(\zeta, l), i] \models \psi_1) \wedge \forall i \in [l, j]. ([(\zeta, l), i] \models \psi_1))))$

Definition 34. *Let $k > 0$. $M \models_k \varphi$ iff $M, s \models_k \varphi$ for all $s \in I$.*

On the Problem of Soundness

Proposition 6. *The defined bounded semantics does not have the following property: if* $M \models_k \varphi$ *for some* $k \geq 1$*, then* $M \models \varphi$*.*

Proof. Suppose that M is as shown in Fig. 2. Then we have $M \not\models AGp$, since $p \notin L(s_2)$. On the other hand, we have $M \models_1 AGp$. The latter is argued as follows.

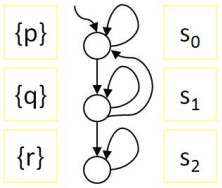

Fig. 2. Example Kripke Structure 2

- The only 1-paths starting from s_0 are $\zeta = s_0 s_0$ and $\zeta' = s_0 s_1$.
- According to the bounded semantics, $M, s_0 \models AGp$ iff the following 4 conditions hold

$$[(\zeta, 0), 0] \models Gp$$
$$[(\zeta, 1), 0] \models Gp$$
$$[(\zeta', 0), 0] \models Gp$$
$$[(\zeta', 1), 0] \models Gp$$

- We have $loop(\zeta) = loop(\zeta') = \{0, 1\}$. Then $[(\zeta, 0), 0] \models Gp$ holds, since all states on ζ satisfy p and $0 \in loop(\zeta)$. Similarly, the other three conditions can be proved, according to the bounded semantics.

In the definition of the bounded semantics, $k > 0$ is assumed, and therefore the relation $\models_0$ is not defined, however, no matter how this relation is defined and added to the family of the defined relations $(\models_k)_{k>0}$, as a consequence of the above proposition, this bounded semantics does not have the property defined in Definition 8, and it support the claim made in the discussion part at the end of Section 2.2.

4.3 On Defining a Bounded Semantics for CTL*

In this subsection, we discuss the possibility of defining a sound and complete bounded semantics for CTL*, and prove that there are no such possibility, if we impose additional conditions on such a bounded semantics. This presentation is a simplification of that presented in [45].

Let $M = \langle S, T, I, L\rangle$ be a model. Let $M_{[k]} = \langle S, P_k, I, L\rangle$ be the restricted model of M where P_k is the set of all k-paths of M. For the idea of this restricted model, the reader if referred to [33]. In the following, we assume that, if $M_{[k]}$ is used as the context in a definition, then we can use the paths in P_k and cannot use T in the definition.

Definition 35 (Compositionality w.r.t. Prop. Connectives). *Let ζ denote a k-path of $M_{[k]}$. Let $\models^p_k$ be a relation defined for path formulas. The relation $\models^p_k$ is compositional with respect to propositional connectives, if the following hold:*

- $M_{[k]}, \zeta \models^p_k \varphi \vee \psi$ *iff* $M_{[k]}, \zeta \models^p \varphi$ *or* $M_{[k]}, \zeta \models^p_k \psi$.
- $M_{[k]}, \zeta \models^p_k \varphi \wedge \psi$ *iff* $M_{[k]}, \zeta \models^p_k \varphi$ *and* $M_{[k]}, \zeta \models^p_k \psi$.

Definition 36 (Consistency w.r.t. Path Operators). *Let $\zeta = \zeta_0 \cdots \zeta_k$ denote a k-path of $M_{[k]}$. Let $\models^p_k$ be a relation defined for path formulas. The relation $\models^p_k$ is consistent with respect to path operators, if the following hold:*

- *If* $M_{[k]}, \zeta \models^p_k Gp$, *then* $p \in L(\zeta_n)$ *for all* $n \in \{0, ..., k\}$.
- *If* $M_{[k]}, \zeta \models^p_k Fp$, *then* $p \in L(\zeta_n)$ *for some* $n \in \{0, ..., k\}$.

Let $(\models_k)_{k\in\mathbf{N}}$ be a family of semantic relations defined for CTL* state formulas over Kripke structures.

Proposition 7. *Let $\models^p_k$ be a family of relations satisfying the compositionality and consistency conditions. If $\models_k$ satisfies the following state-to-path transition conditions, then $(\models_k)_{k\in\mathbf{N}}$ is not a sound and complete bounded semantics for CTL* .*

- $\langle M, s\rangle \models_k A\varphi$ *iff* $M_{[k]}, \zeta \models^p_k \varphi$ *for every k-path ζ of $M_{[k]}$ starting at s.*
- $\langle M, s\rangle \models_k E\varphi$ *iff* $M_{[k]}, \zeta \models^p_k \varphi$ *for some k-path ζ of $M_{[k]}$ starting at s.*

Proof by contradiction: Suppose that $\models_k$ is such a family of relations defining a sound and complete bounded semantics. Let $M = \langle M, s_0\rangle$ be the model shown in Fig. 3. Let φ be $A(Gp \vee Fr)$.

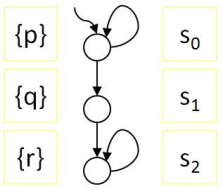

Fig. 3. Model with two loops

- It is easy to check that $M \models A(Gp \vee Fr)$.

– Then there is a $k \geq 0$ such that $M \models_k A(Gp \vee Fr)$, according to the completeness of $\models_k$.
– There are following three types of k-paths in M starting at s_0.

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

According to the state-to-path transition conditions on the relation $\models_k$, we have $M_{[k]}, \zeta \models_k^p (Gp \vee Fr)$ for every such k-path ζ starting at s_0.
– According to the compositionality and consistency conditions of $\models_k^p$, we have that $(s_0)^k s_1$ does not satisfy $Gp \vee Fr$ for $k \geq 1$, i.e.,

$$M_{[k]}, (s_0)^k s_1 \not\models_k^p Gp \vee Fr$$

– Then the only possibility for $M_{[k]}, \zeta \models_k^p Gp \vee Fr$ to hold for all k-path ζ starting at s_0 is the case when $k = 0$. Therefore we have

$$M_{[0]}, s_0 \models_0^p Gp \vee Fr$$

– Let M' be the modification of M such that a self-loop from s_1 to s_1 is added, as shown in Fig. 4.

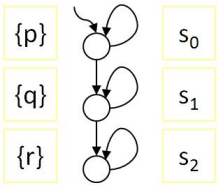

Fig. 4. Model with three loops

– Then the following holds:
$M'_{[0]}, s_0 \models_0^p Gp \vee Fr$, since $M_{[0]}, s_0 \models_0^p Gp \vee Fr$ and $M'_{[0]}$=$M_{[0]}$.
$M' \models_0 A(Gp \vee Fr)$, according to the state-to-path transition conditions on the relation $\models_k$.
$M' \models A(Gp \vee Fr)$, according to the soundness of $\models_k$.
– On the other hand, it is easy to check that $M' \not\models A(Gp \vee Fr)$. Therefore the proposition is proved by contradiction.

Remarks The conditions imposed on the bounded semantics includes compositionality, consistency and the state-to-path transition conditions. For defining a sound and complete bounded semantics for CTL* , it is necessary to look for relations that do not satisfy these conditions, and generally, that not definable by relations that satisfy these conditions.

4.4 Related Works

Bounded model checking of LTL properties was proposed in [6]. It is a technique for overcoming the state explosion problem for quickly identifying unsatisfiability of universally quantified properties. Along this line of research, there have been works on bounded model checking of ACTL properties [33], ACTL* properties [40] and $\forall\mu$-calculus [39]. Among these temporal logics, μ-calculus is most expressive and subsumes all the other ones. For efficient bounded model checking of $\forall\mu$-calculus, a new approach was proposed [30] based on the proof system proposed in [29].

Within this framework, for the verification of universally quantified properties, one looks for termination criteria, which may be considered as criteria for determining whether an over-approximation of the completeness threshold has been reached. Static termination criteria may be used for terminating a bounded model checking process with a conclusion on the correctness of the universally quantified property under bounded model checking [16]. Improvements of such over-approximations are made by using dynamic termination criteria, for instance, in the proposed approach for bounded model checking of $\forall\mu$-calculus properties [30], a dynamic termination criterium is used, such that it is able to prove a $\forall\mu$-calculus property by first transforming it into an $\exists\mu$-calculus formula (or equivalently an existential alternating parity tree automaton) and show that the model does not satisfy this formula for all $k \leq m$ and that the termination criterium is satisfied at the m-th round of the bounded model checking process.

On the other hand, bounded verification of LTL properties has been addressed in [41], and a corresponding bounded semantics was discussed in [44]. Similar idea was used in [42] for bounded verification of ACTL properties. However, these approaches are not complete for their respective target langauge LTL and ACTL. In [43], the approach for bounded verification of ACTL properties was improved and a bounded semantics for ACTL was developed, and it provides a basis for a complete approach for the bounded verification of ACTL properties.

The above approaches are all aimed at either falsifying or verifying universally quantified properties. For verifying and falsifying formulas with mixed path-quantifiers, we have to develop approaches that can deal with both path-quantifiers. This has been achieved by the development of a bounded semantics for CTL [45]. The paper [45] and the one on bounded semantics of ACTL [43] form the basis of the presentation of the bounded semantics of CTL in this current paper, and how to develop a well-defined, sound and complete bounded semantics for a more expressive logics remains as a further research issue.

5 Concluding Remarks

Characteristics of bounded semantics has been presented for clarifying the concept of bounded semantics. Then a bounded semantics for CTL has been presented, with an application of the bounded semantics to QBF-based bounded correctness checking of finite state systems.

Remarks The kind of bounded correctness checking approaches may be considered as complementary to the traditional symbolic model checking. A distinguished feature of the bounded semantics of CTL is that it covers the full set of CTL which is closed under negation, and this semantics may be used to both bounded verification and bounded model checking, while the earlier developed bounded semantics for, for instance, existentially interpreted LTL and the existential fragment of CTL [6, 33], focus mainly on their potentials for bounded model checking (falsification) of universally specified properties.

Experimental Tools An implementation of the QBF-based bounded correctness checking approach is available in an experimental tool named *verbs* [46] for verification of finite state systems.

Open Problems One of the interesting problems on bounded semantics is whether there exists a well-defined, sound and complete bounded semantics for CTL* [18]. This is particularly interesting, because, there has been an attempt to develop a bounded semantics for CTL* [38], however such an attempt has not been successful with respect to the requirements of a well-defined, sound and complete bounded semantics. On the other hand, it can be proved that such a bounded semantics cannot be defined with additional restrictions on the bounded semantic relations. A similar question may be raised for μ-calculus for which SAT-based model checking approaches had been considered for a subset of μ-calculus formulas [39, 30].

Acknowledgement The author would like to thank the anonymous referees for their valuable comments on an earlier version of this paper.

A Proofs

This appendix provides a proof of Lemma 3 and a proof of Lemma 6.

A.1 Proof of Lemma 3

Recall that M_k denote the set of k-paths of M. For simplicity, we use $M_k(s)$ to denote the set of k-paths starting at s.

Lemma 3. If $M, s \models_k \varphi$, *then* $M, s \models_{k+1} \varphi$.

Proof. This is trivial for φ being a proposition or the negation of that. Assume the induction hypothesis, i.e., $M, s' \models_k \varphi'$ implies $M, s' \models_{k+1} \varphi'$ for all state s' and all proper subformulas φ' of φ. We have the following cases.

Case 1. $\varphi = \varphi_0 \vee \varphi_1$.

According to the induction hypothesis, we obtain

$$
\begin{aligned}
& M, s \models_k \varphi \\
\Leftrightarrow\ & M, s \models_k \varphi_0 \text{ or } M, s \models_k \varphi_1 \\
\Rightarrow\ & M, s \models_{k+1} \varphi_0 \text{ or } M, s \models_{k+1} \varphi_1 \\
\Leftrightarrow\ & M, s \models_{k+1} \varphi
\end{aligned}
$$

Case 2. $\varphi = \varphi_0 \wedge \varphi_1$.

The proof is similar to that of the previous case.

Case 3. $\varphi = AX\varphi_0$.

According to the induction hypothesis, for $k \geq 1$, we obtain

$$
\begin{aligned}
& M, s \models_k \varphi \\
\Leftrightarrow\ & \forall \zeta \in M_k(s),\ M, \zeta_1 \models_k \varphi_0 \\
\Rightarrow\ & \forall \zeta \in M_k(s),\ M, \zeta_1 \models_{k+1} \varphi_0 \\
\Leftrightarrow\ & \forall \zeta' \in M_{k+1}(s),\ M, \zeta'_1 \models_{k+1} \varphi_0 \\
\Leftrightarrow\ & M, s \models_{k+1} \varphi
\end{aligned}
$$

For $k = 0$, $M, s \models_k \varphi$ is false by definition, therefore $M, s \models_k \varphi$ implies $M, s \models_{k+1} \varphi$ also holds in this case.

Case 4. $\varphi = EX\varphi_0$.

According to the induction hypothesis, we obtain

$$
\begin{aligned}
& M, s \models_k \varphi \\
\Leftrightarrow\ & k \geq 1 \wedge \exists \zeta \in M_k(s),\ M, \zeta_1 \models_k \varphi_0 \\
\Rightarrow\ & k \geq 1 \wedge \exists \zeta \in M_k(s),\ M, \zeta_1 \models_{k+1} \varphi_0
\end{aligned}
$$

Assume $M, s \models_k \varphi$. We need to prove $M, s \models_{k+1} \varphi$. Since we already have $k \geq 1$, we only need to prove that there is some $\zeta' \in M_{k+1}(s)$ such that

$M, \zeta'_1 \models_{k+1} \varphi_0$.

Let $\zeta \in M_k(s)$ be a path such that $M, \zeta_1 \models_{k+1} \varphi_0$ holds. Then we construct a path $\zeta' \in M_{k+1}(s)$ by appending a state to the path ζ (this is always possible, since the transition relation of M is total). Then $\zeta_1 = \zeta'_1$ (since $k \geq 1$) and we have $\zeta' \in M_{k+1}(s)$ and $M, \zeta'_1 \models_{k+1} \varphi_0$. Therefore $M, s \models_{k+1} \varphi$.

Case 5. $\varphi = AF\varphi_1$.

According to the induction hypothesis, we obtain

$$
\begin{aligned}
& M, s \models_k \varphi \\
\Leftrightarrow\ & \forall \zeta \in M_k(s), \text{ there is some } 0 \leq n \leq k \text{ such that } M, \zeta_n \models_k \varphi_1 \\
\Rightarrow\ & \forall \zeta \in M_k(s), \text{ there is some } 0 \leq n \leq k \text{ such that } M, \zeta_n \models_{k+1} \varphi_1
\end{aligned}
$$

Assume $M, s \models_k \varphi$. We need to prove $M, s \models_{k+1} \varphi$, i.e., for all $\zeta' \in M_{k+1}(s)$,

there is some $0 \leq n \leq k+1$ such that $M, \zeta_n \models_{k+1} \varphi_1$.

Consider a path $\zeta' \in M_{k+1}(s)$. Let $\zeta'' = \zeta'_0 \cdots \zeta'_k$. Then since $\zeta'' \in M_k(s)$, there is some $0 \leq n \leq k$ such that $M, \zeta'_n \models_{k+1} \varphi_1$. Therefore there is also some $0 \leq n \leq k+1$ such that $M, \zeta'_n \models_{k+1} \varphi_1$.

Case 6. $\varphi = EF\varphi_1$.

According to the induction hypothesis, we obtain

$$\begin{array}{l} M, s \models_k \varphi \\ \Leftrightarrow \exists \zeta \in M_k(s), \text{ there is some } 0 \leq n \leq k \text{ such that } M, \zeta_n \models_k \varphi_1 \\ \Rightarrow \exists \zeta \in M_k(s), \text{ there is some } 0 \leq n \leq k \text{ such that } M, \zeta_n \models_{k+1} \varphi_1 \end{array}$$

Assume $M, s \models_k \varphi$. We need to prove $M, s \models_{k+1} \varphi$, i.e., there is some $\zeta' \in M_{k+1}(s)$ such that

there is some $0 \leq n \leq k+1$ such that $M, \zeta_n \models_{k+1} \varphi_1$.

Let $\zeta \in M_k(s)$ be a path such that there is some $0 \leq n \leq k$ such that $M, \zeta_n \models_{k+1} \varphi_1$. Then we construct a path $\zeta' \in M_{k+1}(s)$ by appending a state to the path ζ. Then we have $\zeta' \in M_{k+1}(s)$ and $M, \zeta'_n \models_{k+1} \varphi_1$ for some $0 \leq n \leq k+1$. Therefore $M, s \models_{k+1} \varphi$.

Case 7. $\varphi = AG\varphi_1$.

According to the induction hypothesis, we obtain

$$\begin{array}{l} M, s \models_k \varphi \\ \Leftrightarrow \forall \zeta \in M_k(s), \text{ for all } 0 \leq j \leq k,\ M, \zeta_j \models_k \varphi_1 \text{ and } rs(\zeta) \\ \Rightarrow \forall \zeta \in M_k(s), \text{ for all } 0 \leq j \leq k,\ M, \zeta_j \models_{k+1} \varphi_1 \text{ and } rs(\zeta) \end{array}$$

Assume $M, s \models_k \varphi$. We need to prove $M, s \models_{k+1} \varphi$, i.e., for all $\zeta' \in M_{k+1}(s)$,

for all $0 \leq j \leq k+1$, $M, \zeta'_j \models_{k+1} \varphi_1$ and $rs(\zeta')$.

Consider a path $\zeta' \in M_{k+1}(s)$. Let $\zeta'' = \zeta'_0 \cdots \zeta'_k$. Then since $\zeta'' \in M_k(s)$, for all $0 \leq j \leq k$, $M, \zeta'_j \models_{k+1} \varphi_1$ and $rs(\zeta'')$. Since ζ'' is a prefix of ζ', $rs(\zeta')$ follows from $rs(\zeta'')$ and it is sufficient to prove that $M, \zeta'_{k+1} \models_{k+1} \varphi_1$ holds (the only case not covered by the induction hypothesis).

Since $rs(\zeta'')$ holds in this case, there are $x < y \leq k$ such that $\zeta'_x = \zeta'_y$. Then $\zeta'_0 \cdots \zeta'_x \zeta'_{y+1} \cdots \zeta'_{k+1}$ is a prefix (not necessary a proper one) of some k-path $\zeta''' \in M_k(s)$ with $\zeta'''_0 = \zeta'_0 = s$. Then for all $0 \leq j \leq k$, $M, \zeta'''_j \models_{k+1} \varphi_1$. Since ζ'_{k+1} is ζ'''_j for some $0 \leq j \leq k$, we obtain that $M, \zeta'_{k+1} \models_{k+1} \varphi_1$ holds.

Case 8. $\varphi = EG\varphi_1$.

According to the induction hypothesis, we obtain

$$\begin{array}{l} M, s \models_k \varphi \\ \Leftrightarrow \exists \zeta \in M_k(s), \text{ for all } 0 \leq j \leq k,\ M, \zeta_j \models_k \varphi_1 \text{ and } rs(\zeta) \\ \Rightarrow \exists \zeta \in M_k(s), \text{ for all } 0 \leq j \leq k,\ M, \zeta_j \models_{k+1} \varphi_1 \text{ and } rs(\zeta) \end{array}$$

Assume $M, s \models_k \varphi$. We need to prove $M, s \models_{k+1} \varphi$, i.e., there is some $\zeta' \in M_{k+1}(s)$ such that

for all $0 \leq j \leq k+1$, $M, \zeta'_j \models_{k+1} \varphi_1$ and $rs(\zeta')$.

Let $\zeta \in M_k(s)$ be a path such that for all $0 \leq j \leq k$, $M, \zeta_j \models_{k+1} \varphi_1$ and $rs(\zeta)$. Since $rs(\zeta)$, we have $\zeta_x = \zeta_y$ for some $0 \leq x < y \leq k$. Let $\zeta' \in M_{k+1}(s)$ be a $(k+1)$-path which is at the same time a prefix of the infinite path $\zeta_0 \cdots (\zeta_x \cdots \zeta_{y-1})^\omega$. Then we have that for all $0 \leq j \leq k+1$, $M, \zeta'_j \models_{k+1} \varphi_1$ and $rs(\zeta')$. Therefore $M, s \models_{k+1} \varphi$.

Case 9. $\varphi = A(\varphi_0 U \varphi_1)$.

According to the induction hypothesis, we obtain

$$
\begin{array}{l}
\quad M, s \models_k \varphi \\
\Leftrightarrow \forall \zeta \in M_k(s), \\
\quad \text{there is some } 0 \leq n \leq k \\
\quad \text{such that } M, \zeta_n \models_k \varphi_1 \text{ and for all } l < n,\ M, \zeta_l \models_k \varphi_0 \\
\Rightarrow \forall \zeta \in M_k(s), \\
\quad \text{there is some } 0 \leq n \leq k \\
\quad \text{such that } M, \zeta_n \models_{k+1} \varphi_1 \text{ and for all } l < n,\ M, \zeta_l \models_{k+1} \varphi_0
\end{array}
$$

Assume $M, s \models_k \varphi$. We need to prove $M, s \models_{k+1} \varphi$, i.e., for all $\zeta' \in M_{k+1}(s)$,

there is some $0 \leq n \leq k+1$
such that $M, \zeta'_n \models_{k+1} \varphi_1$ and for all $l < n$, $M, \zeta'_l \models_{k+1} \varphi_0$.

Consider a path $\zeta' \in M_{k+1}(s)$. Let $\zeta'' = \zeta'_0 \cdots \zeta'_k$. Then since $\zeta'' \in M_k(s)$, there is some $0 \leq n \leq k$ such that $M, \zeta'_n \models_{k+1} \varphi_1$ and for all $l < n$, $M, \zeta'_l \models_{k+1} \varphi_0$. Therefore there is also some $0 \leq n \leq k+1$ such that $M, \zeta'_n \models_{k+1} \varphi_1$ and for all $l < n$, $M, \zeta'_l \models_{k+1} \varphi_0$. Therefore $M, s \models_{k+1} \varphi$.

Case 10. $\varphi = E(\varphi_0 U \varphi_1)$.

According to the induction hypothesis, we obtain

$$
\begin{array}{l}
\quad M, s \models_k \varphi \\
\Leftrightarrow \exists \zeta \in M_k(s), \\
\quad \text{there is some } 0 \leq n \leq k \\
\quad \text{such that } M, \zeta_n \models_k \varphi_1 \text{ and for all } l < n,\ M, \zeta_l \models_k \varphi_0 \\
\Rightarrow \exists \zeta \in M_k(s), \\
\quad \text{there is some } 0 \leq n \leq k \\
\quad \text{such that } M, \zeta_n \models_{k+1} \varphi_1 \text{ and for all } l < n,\ M, \zeta_l \models_{k+1} \varphi_0
\end{array}
$$

Assume $M, s \models_k \varphi$. We need to prove $M, s \models_{k+1} \varphi$, i.e., there is some $\zeta' \in M_{k+1}(s)$ such that

there is some $0 \leq n \leq k+1$
such that $M, \zeta'_n \models_{k+1} \varphi_1$ and for all $l < n$, $M, \zeta'_l \models_{k+1} \varphi_0$.

Let $\zeta \in M_k(s)$ be a path such that there is some $0 \leq n \leq k$ such that $M, \zeta_n \models_{k+1} \varphi_1$ and for all $l < n$, $M, \zeta_l \models_{k+1} \varphi_0$. Then we construct a path $\zeta' \in M_{k+1}(s)$ by appending a state to the path ζ. Then we have $\zeta' \in M_{k+1}(s)$ and there is some $0 \leq n \leq k+1$ such that $M, \zeta'_n \models_{k+1} \varphi_1$ and for all $l < n$, $M, \zeta'_l \models_{k+1} \varphi_0$. Therefore $M, s \models_{k+1} \varphi$.

Case 11. $\varphi = A(\varphi_0 R \varphi_1)$.

According to the induction hypothesis, we obtain

$M, s \models_k \varphi$
$\Leftrightarrow \forall \zeta \in M_k(s)$,
for all $0 \leq i \leq k$, $(M, \zeta_i \models_k \varphi_1 \vee \exists j < i.(M, \zeta_j \models_k \varphi_0)))$, and
there is some $0 \leq j < k$ such that $M, \zeta_j \models_k \varphi_0$ or $rs(\zeta)$
$\Rightarrow \forall \zeta \in M_k(s)$,
for all $0 \leq i \leq k$, $(M, \zeta_i \models_{k+1} \varphi_1 \vee \exists j < i.(M, \zeta_j \models_{k+1} \varphi_0))$, and
there is some $0 \leq j < k$ such that $M, \zeta_j \models_{k+1} \varphi_0$ or $rs(\zeta)$

Assume $M, s \models_k \varphi$. We need to prove $M, s \models_{k+1} \varphi$, i.e., for all $\zeta' \in M_{k+1}(s)$,

(a): for all $0 \leq i \leq k+1$, $(M, \zeta'_i \models_{k+1} \varphi_1 \vee \exists j < i.(M, \zeta'_j \models_{k+1} \varphi_0))$, and
(b): there is some $0 \leq j < k+1$ such that $M, \zeta'_j \models_{k+1} \varphi_0$ or $rs(\zeta')$.

Consider a path $\zeta' \in M_{k+1}(s)$. Let $\zeta'' = \zeta'_0 \cdots \zeta'_k$. Then since $\zeta'' \in M_k(s)$, the following hold:

(a'): for all $0 \leq i \leq k$, $(M, \zeta'_i \models_{k+1} \varphi_1 \vee \exists j < i.(M, \zeta'_j \models_{k+1} \varphi_0))$, and
(b'): there is some $0 \leq j < k$ such that $M, \zeta'_j \models_{k+1} \varphi_0$ or $rs(\zeta'')$.

We consider two cases of (b'):

- Suppose that $M, \zeta'_j \models_{k+1} \varphi_0$ holds for some $0 \leq j < k$.
  Then we also have $M, \zeta'_j \models_{k+1} \varphi_0$ for some $j < k+1$. The correctness of (a) and (b) follows from this and (a').
- Suppose that $M, \zeta'_j \models_{k+1} \varphi_0$ does not hold for all $0 \leq j < k$.
  Then following from (a') and (b'),
  we have that for all $0 \leq i \leq k$, $(M, \zeta'_i \models_{k+1} \varphi_1)$ and $rs(\zeta'')$.
  Since ζ'' is a prefix of ζ', $rs(\zeta')$ follows from $rs(\zeta'')$, and then the correctness of (a) and (b) is implied by $M, \zeta'_{k+1} \models_{k+1} \varphi_1$, which is to be proved as follows.
  Since $rs(\zeta'')$ holds in this case, there are $x < y \leq k$ such that $\zeta'_x = \zeta'_y$. Then $\zeta'_0 \cdots \zeta'_x \zeta'_{y+1} \cdots \zeta'_{k+1}$ is a valid path.
  This path is a prefix (not necessary a proper one) of some k-path $\zeta''' \in M_k(s)$ with $\zeta'''_0 = \zeta'_0 = s$, which has the property that $M, \zeta'''_l \models_{k+1} \varphi_0$ does not hold for all states in ζ''' up to (may not include) the state ζ'_{k+1}.
  For this k-path ζ''', we have the following (by the induction hypothesis):
  for all $0 \leq i \leq k$, $(M, \zeta'''_i \models_{k+1} \varphi_1 \vee \exists j < i.(M, \zeta'''_j \models_{k+1} \varphi_0))$.
  Let l be the position index such that $\zeta'''_l = \zeta'_{k+1}$.
  Then we have $M, \zeta'''_l \models_{k+1} \varphi_1 \vee \exists j < l.(M, \zeta'''_j \models_{k+1} \varphi_0)$.
  Since $M, \zeta'''_l \models_{k+1} \varphi_0$ does not hold for all states in ζ''' up to the state $\zeta'_{k+1} = \zeta'''_l$, we have $M, \zeta'''_l \models_{k+1} \varphi_1$, i.e., $M, \zeta'_{k+1} \models_{k+1} \varphi_1$.

Case 12. $\varphi = E(\varphi_0 R \varphi_1)$.

According to the induction hypothesis, we obtain

$$
\begin{array}{l}
\quad M, s \models_k \varphi \\
\Leftrightarrow \exists \zeta \in M_k(s), \\
\quad \text{for all } 0 \leq i \leq k,\ (M, \zeta_i \models_k \varphi_1 \vee \exists j < i.(M, \zeta_j \models_k \varphi_0))), \text{ and} \\
\quad \text{there is some } 0 \leq j < k \text{ such that } M, \zeta_j \models_k \varphi_0 \text{ or } rs(\zeta) \\
\Rightarrow \exists \zeta \in M_k(s), \\
\quad \text{for all } 0 \leq i \leq k,\ (M, \zeta_i \models_{k+1} \varphi_1 \vee \exists j < i.(M, \zeta_j \models_{k+1} \varphi_0)), \text{ and} \\
\quad \text{there is some } 0 \leq j < k \text{ such that } M, \zeta_j \models_{k+1} \varphi_0 \text{ or } rs(\zeta)
\end{array}
$$

Assume $M, s \models_k \varphi$. We need to prove $M, s \models_{k+1} \varphi$, i.e., there is some $\zeta' \in M_{k+1}(s)$ such that

(a): for all $0 \leq i \leq k+1$, $(M, \zeta'_i \models_{k+1} \varphi_1 \vee \exists j < i.(M, \zeta'_j \models_{k+1} \varphi_0))$, and
(b): there is some $0 \leq j < k+1$ such that $M, \zeta'_j \models_{k+1} \varphi_0$ or $rs(\zeta')$.

Let $\zeta \in M_k(s)$ be a path such that the following hold:

(a'): for all $0 \leq i \leq k$, $(M, \zeta_i \models_{k+1} \varphi_1 \vee \exists j < i.(M, \zeta_j \models_{k+1} \varphi_0))$, and
(b'): there is some $0 \leq j < k$ such that $M, \zeta_j \models_{k+1} \varphi_0$ or $rs(\zeta)$.

We consider two cases of (b'):

- Suppose that $M, \zeta_j \models_{k+1} \varphi_0$ holds for some $0 \leq j < k$.
  Then we construct a path $\zeta' \in M_{k+1}(s)$ by appending a state to the path ζ.
  Then we also have $M, \zeta'_j \models_{k+1} \varphi_0$ for some $j < k+1$. The correctness of (a) and (b) follows from this and (a').
- Suppose that $M, \zeta_j \models_{k+1} \varphi_0$ does not hold for all $0 \leq j < k$.
  Then following from (a') and (b'),
  we have that for all $0 \leq i \leq k$, $(M, \zeta_i \models_{k+1} \varphi_1)$ and $rs(\zeta)$.
  Since $rs(\zeta)$ holds in this case, there are $x < y \leq k$ such that $\zeta_x = \zeta_y$.
  Let $\zeta' \in M_{k+1}(s)$ be a $(k+1)$-path which is at the same time a prefix of the infinite path $\zeta_0 \cdots (\zeta_x \cdots \zeta_{y-1})^\omega$.
  Then we have that for all $0 \leq i \leq k+1$, $(M, \zeta'_i \models_{k+1} \varphi_1)$ and $rs(\zeta')$.
  Therefore we have $\zeta' \in M_{k+1}(s)$ that satisfies (a) and (b).
  Therefore $M, s \models_{k+1} \varphi$.
  This concludes the proof of Lemma 3. □

A.2 Proof of Lemma 6

Firstly, we recall the concept of bisimulation equivalence [4, 31, 16].

Definition 37. *Let $M_1 = \langle S_1, T_1, I_1, L_1 \rangle$ and $M_2 = \langle S_2, T_2, I_2, L_2 \rangle$ be two models with the same set of atomic propositions AP. A relation $B \subseteq S_1 \times S_2$ is a bisimulation relation between M_1 and M_2, if the following hold:*

- *for every $(s_1, s_2) \in R$, $L_1(s_1) = L_2(s_2)$.*

– *for every* $(s_1, s_2) \in R$, *if* $(s_1, s_1') \in T_1$, *there is an* s_2' *such that* $(s_2, s_2') \in T_2$ *and* $(s_1', s_2') \in R$, *and if* $(s_2, s_2') \in T_2$, *there is an* s_1' *such that* $(s_1, s_1') \in T_1$ *and* $(s_1', s_2') \in R$.

M_1 and M_2 are bisimulation equivalent, if there is a bisimulation relation B between M_1 and M_2 such that the following two conditions for the correspondence of initial states is satisfied:

– *for every* $s_1 \in I_1$, *there is an* $s_2 \in I_2$ *such that* $(s_1, s_2) \in B$,
– *for every* $s_2 \in I_2$, *there is an* $s_1 \in I_1$ *such that* $(s_1, s_2) \in B$.

Secondly, we prove a relation between expansion equivalent models and bisimulation equivalent models.

Lemma 8. *Expansion equivalent models are bisimulation equivalent.*

Proof. Let $M_1 = \langle S_1, T_1, I_1, L_1 \rangle$ and $M_2 = \langle S_2, T_2, I_2, L_2 \rangle$ be two expansion equivalent models.

– Let $M_1^m = \langle S_1^m, T_1^m, I_1^m, L_1^m \rangle$ be the m-model of M_1.
  Let $M_2^m = \langle S_2^m, T_2^m, I_2^m, L_2^m \rangle$ be the m-model of M_2.
  Let the ω-model of M be the natural extension of an m-model to an infinite loop-free directed diagram, such that the first m-levels of the directed diagram is the same as that of the m-model for all $m \geq 1$.
  Let $M_1^\omega = \langle S_1^\omega, T_1^\omega, I_1^\omega, L_1^\omega \rangle$ be the ω-model of M_1.
  Let $M_2^\omega = \langle S_2^\omega, T_2^\omega, I_2^\omega, L_2^\omega \rangle$ be the ω-model of M_2.
  Since M_1^m and M_2^m are structurally equivalent for all $m \geq 0$, it follows that M_1^ω and M_2^ω are structurally equivalent.
– Then there is a one-to-one map $f_\omega : S_1^\omega \to S_2^\omega$ preserving the properties defined in Definition 11 with $S_1^\omega \subseteq S_1 \times \mathbf{N}$ and $S_2^\omega \subseteq S_2 \times \mathbf{N}$.
  Let $B = \{(s_1, s_2) \mid \exists j.(f_\omega(s_1, j) = (s_2, j))\}$.
– That B is a bisimulation relation between M_1 and M_2 is checked as follows.
  Suppose that $B(s_1, s_2)$ holds.
  Then $\exists j.(f_\omega(s_1, j) = (s_2, j))$.
  We have $L(s_1) = L((s_1, j)) = L((s_2, j)) = L(s_2)$, according to the construction of the ω-model and the construction of f_ω.
  Suppose that $(s_1, s_1') \in T_1$ holds.
  Then $((s_1, j), (s_1', j+1)) \in T_1^\omega$, according to the construction of the ω-model.
  Then $(f(s_1, j), f(s_1', j+1)) \in T_2^\omega$, according to the construction of f_ω.
  Let $f(s_1', j+1) = (s_2', j+1)$.
  According to the construction of B, we have $B(s_1', s_2')$.
  On the other hand, since $(f(s_1, j), f(s_1', j+1)) \in T_2^\omega$, we have $((s_2, j), (s_2', j+1)) \in T_2^\omega$, and according to the construction of the ω-model, we have $(s_2, s_2') \in T_2^\omega$.
  Therefore B is a bisimulation relation.
– In addition, according to the construction of f_ω, for any initial state $s_1 \in I_1$, there is some $s_2 \in I$ such that $f_\omega(s_1, 0) = f_\omega(s_2, 0)$ and therefore $B(s_1, s_2)$.
  Similarly, for any initial state $s_2 \in I_1$, there is some $s_1 \in I$ such that $f_\omega(s_1, 0) = f_\omega(s_2, 0)$ and $B(s_1, s_2)$.
  Therefore M_1 and M_2 are bisimulation equivalent.

Finally, Lemma 6 is proved as follows.

Lemma 6. Expansion equivalent models are not distinguishable by $(\models, CTL)$.

Proof. Since CTL does not distinguish bisimulation equivalent models [16], this lemma follows from that expansion equivalent models are also bisimulation equivalent (Lemma 8). □

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. C. Baier and J.-P. Katoen. Principles of Model Checking. MIT Press. 2008.
3. M. Benedetti, A. Cimatti: Bounded Model Checking for Past LTL. TACAS 2003: 18-33.
4. J. van Benthem. Modal Correspondence Theory. PhD thesis, Mathematisch Instituut & Instituut voor Grondslagenonderzoek, University of Amsterdam, 1976.
5. A. Biere, A. Cimmatti, E. Clarke, O. Strichman, and Y. Zhu. Bounded Model Checking. Advances in Computers 58, Academic Press, 2003.
6. A. Biere, A. Cimmatti, E. Clarke, and Y. Zhu. Symbolic Model Checking without BDDs. LNCS 1579:193-207. TACAS 99.
7. Beate Bollig. A very simple function that requires exponential size nondeterministic graph-driven read-once branching programs. Inf. Process. Lett. 86(3): 143-148 (2003).
8. Hans Kleine Buning and Uwe Bubeck. Theory of Quantified Boolean Formulas. Handbook of Satisfiability (Armin Biere, Marijn Heule, Hans van Maaren and Toby Walsh (Eds.)):735-760, IOS Press, 2009.
9. 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.
10. Randal E. Bryant. Graph based algorithms for boolean function manipulation. IEEE Transaction on Computers 35(8):677-691. 1986.
11. 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).
12. Randal E. Bryant. Binary decision diagrams and beyond: enabling technologies for formal verification. CAD'95:236-243. 1995.
13. Edmund M. Clarke, Armin Biere, Richard Raimi, Yunshan Zhu. Bounded Model Checking Using Satisfiability Solving. Formal Methods in System Design 19(1): 7-34 (2001).
14. Edmund M. Clarke, Daniel Kroening, Joel Ouaknine, Ofer Strichman. Completeness and Complexity of Bounded Model Checking. VMCAI 2004: 85-96.
15. E. M. Clarke, D. Kroening, J. Ouaknine, and O. Strichman. Computational challenges in bounded model checking. STTT 7(2): 174-183. 2005.
16. E. M. Clarke, O. Grumberg and D. Peled. Model Checking. The MIT Press. 1999.
17. 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.
18. 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.

19. Enrico Giunchiglia, Paolo Marin and Massimo Narizzano. Reasoning with Quantified Boolean Formulas. Handbook of Satisfiability (Armin Biere, Marijn Heule, Hans van Maaren and Toby Walsh (Eds.)):761-780, IOS Press, 2009.
20. Keijo Heljanko, Tommi A. Junttila, Timo Latvala: Incremental and Complete Bounded Model Checking for Full PLTL. CAV 2005: 98-111.
21. I. Hodkinson and M. Reynolds. Temporal Logics. In: Handbook of Modal Logic (eds.: P. Blackburn, J. F.A.K. van Benthem, F. Wolter): 655-720. Elsevier, 2007.
22. D. Kozen. Results on the propositional μ-calculus. In: Proc. of the 9th International Colloq. on Automata, Languages and Programming. 1982. 384-359.
23. D. Kroening, O. Strichman. Efficient Computation of Recurrence Diameters. VMCAI 2003: 298-309.
24. Alessio Lomuscio, Wojciech Penczek, Bozena Wozna. Bounded model checking for knowledge and real time. Artif. Intell. 171(16-17):1011-1038 (2007).
25. Daniel Le Berre, Laurent Simon, Armando Tacchella: Challenges in the QBF Arena: the SAT'03 Evaluation of QBF Solvers. SAT 2003: 468-485
26. K. L. McMillan. Symbolic Model Checking. Kluwer Academic Publisher,1993.
27. G. L. Peterson. Myths About the Mutual Exclusion Problem. Information Processing Letters 12(3):115-116. 1981.
28. X. Luo, K. Su, A. Sattar, M. Reynolds. Verification of Multi-agent Systems Via Bounded Model Checking. Australian Conference on Artificial Intelligence 2006: 69-78.
29. K. S. Namjoshi. Certifying Model Checkers. CAV 2001, LNCS 2102, pp. 2-13, 2001.
30. R. Oshman and O. Grumberg. A New Approach to Bounded Model Checking for Branching Time Logics. ATVA 2007, LNCS 4762, pp. 410-424, 2007.
31. D. Park. Concurrency and automata on infinite sequences. 5th GI-Conference on Theoretical Computer Science: 167-183. Springer. 1981.
32. D. A. Peled. Software Reliability Methods. Springer-Verlag. 2001.
33. W. Penczek, B. Wozna, and A. Zbrzezny. Bounded Model Checking for the Universal Fragment of CTL. Fundamenta Informaticae 51:135-156. 2002.
34. W. Penczek, B. Wozna-Szczesniak, A. Zbrzezny. Towards SAT-based BMC for LTLK over Interleaved Interpreted Systems. Proc. of the international workshop on concurrency, specification, and programming (CS&P 2011):565-576.
35. A. Pnueli: The Temporal Logic of Programs. FOCS 1977: 46-57.
36. A. Pnueli. The temporal semantics of concurrent programs. Theoretical Computer Science 13:45-60. 1981.
37. Mukul R. Prasad, Armin Biere, Aarti Gupta. A survey of recent advances in SAT-based formal verification. STTT 7(2): 156-173 (2005).
38. 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).
39. Bow-Yaw Wang. Proving $\forall\mu$-Calculus Properties with SAT-Based Model Checking. FORTE 2005: 113-127.
40. Bozena Wozna. ATCL* properties and Bounded Model Checking. Fundam. Inform. 63(1): 65-87 (2004).
41. W. Zhang. SAT-based verification of LTL formulas. Lecture Notes in Computer Science 4346 (FMICS 2006):277-292.
42. W. Zhang. Verification of ACTL Properties by Bounded Model Checking. Lecture Notes in Computer Science 4739 (EUROCAST 2007):556-563.
43. W. Zhang. Model Checking with SAT-Based Characterization of ACTL Formulas. Lecture Notes in Computer Science 4789 (ICFEM 2007):191-211.

44. W. Zhang. Weak Bounded Semantics and Bounded Verification of LTL Formulas. Proceedings of CHINA 2008 Workshop (Concurrency metHods: Issues aNd Applications), held in Xian, China, 2008. Jetty Kleijn, Maciej Koutny (Eds.).
45. W. Zhang. Bounded Semantics of CTL and SAT-based Verification. Lecture Notes in Computer Science 5885 (ICFEM 2009):286-305. Springer-Verlag. 2009.
46. W. Zhang. *verbs*: Verification of Finite State Systems by Bounded Correctness Checking. Manuscript, availabe at http://lcs.ios.ac.cn/∼zwh/verbs/.