

28th February 2003, NIA, Virginia

Counterexample-Driven Model Checking

N. Shankar and M. Sorea
email: {shankar,sorea}@cs1.sri.com

Computer Science Laboratory
SRI International
Menlo Park, CA

Model Checking

Kripke structure: $\mathbf{M} = \langle \mathbf{AP}, \mathbf{S}, \mathbf{N} \rangle$.

Initial states: I .

Formula: φ .

Model checking problem: $\mathbf{M}, I \models \varphi$.

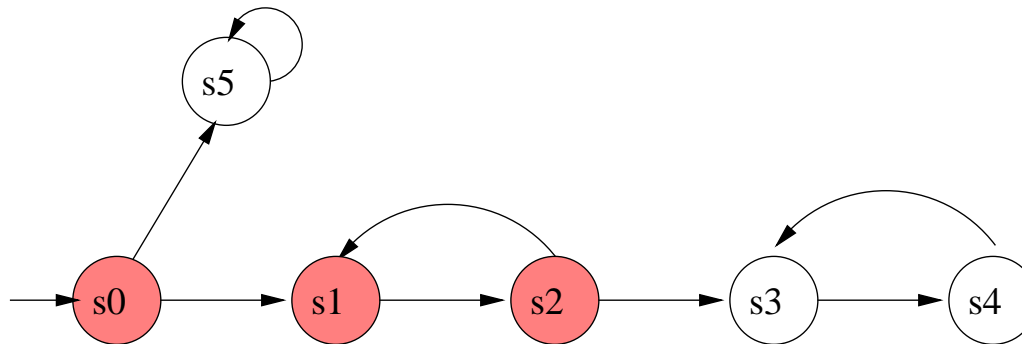
NO \longrightarrow Counterexample

YES \longrightarrow Witness

What is a witness?

What is a counterexample?

Linear Witnesses / Counterexamples

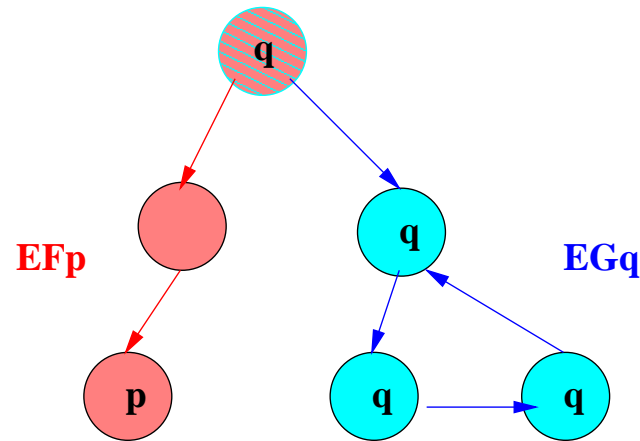


Assume: p holds in the 'pink' states

Witness for $\mathbf{M}, s_0 \models \mathbf{EG}p$: $[s_0, s_1, s_2, s_1]$

Counterexample for $\mathbf{M}, s_0 \models \mathbf{AF}\neg p$

Tree-Like Witnesses / Counterexamples

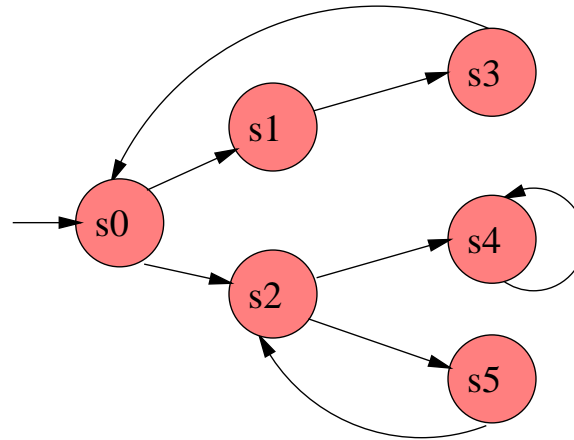


Witness for $\mathbf{M}, s_0 \models \mathbf{EF}p \wedge \mathbf{EG}q$

Counterexample for $\mathbf{M}, s_0 \models \mathbf{AG}\neg p \vee \mathbf{AF}\neg q$

- there is a finite path to a p state
- there is an infinite path along which q is always true

Set-Like Witnesses / Counterexamples

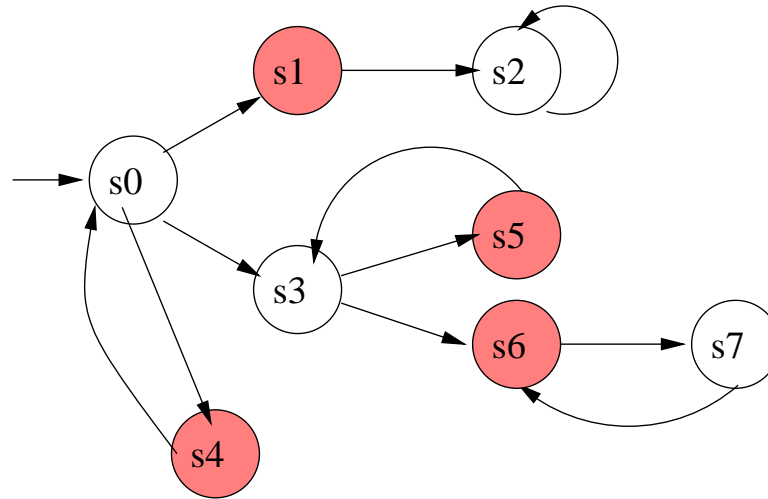


Witness for $\mathbf{M}, s_0 \models \mathbf{AG}p$

Counterexample for $\mathbf{M}, s_0 \models \mathbf{EF}\neg p$

- entire state space
- $\{s_0, s_1, s_2, s_3, s_4, s_5\}$

Set-Like Witnesses / Counterexamples – Cont.



Witness for $\mathbf{M}, s_0 \models \mathbf{AF}p$

Counterexample for $\mathbf{M}, s_0 \models \mathbf{EG}\neg p$

- trace over sets of states
- $[\{s_0\}, \{s_1, s_3, s_4\}, \{s_5, s_6\}]$

Let's Summarize!

Counterexample for universal formulas are traces (linear) or trees.

Only formulas in $ACTL \cap LTL$ have trace (linear) counterexamples.

When model checking succeeds there is no witness for justifying the relation between the model and the property.

Our Approach

Symbolic model checking.

Generation of witnesses if model checking succeeds, and of counterexamples, otherwise.

Witnesses and counterexamples are traces and trees over sets of states, as well as single states.

The model-checking algorithm is local.

Forward + Backward:

Every temporal formula is evaluated by means of a forward unfolding of the state space, starting in I , followed by a local fixpoint computation.

Partition of I in I^+ and I^- , such that $\mathbf{M}, I^+ \models \varphi$ and $\mathbf{M}, I^- \not\models \varphi$.

Symbolic witnesses and counterexamples are constructed from the results produced by the model checker.

Contents

- Preliminaries: CTL, 'naive' model-checking algorithm
- WMC algorithm
- CTL fixpoint characterization vs. WMC
- Characterization and construction of witnesses and counterexamples
- Theoretical and experimental results
- Benefits
- Related work
- Conclusion and work in progress

CTL

AP set of atomic propositions, and $p \in \mathbf{AP}$

CTL in nnf:

$$\begin{aligned} \varphi \quad := \quad & p \mid \neg p \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \\ & \mathbf{EX}\varphi \mid \mathbf{EF}\varphi \mid \mathbf{EG}\varphi \mid \mathbf{E}[\varphi_1 \mathbf{U} \varphi_2] \mid \mathbf{E}[\varphi_1 \mathbf{R} \varphi_2] \mid \\ & \mathbf{AX}\varphi \mid \mathbf{AF}\varphi \mid \mathbf{AG}\varphi \mid \mathbf{A}[\varphi_1 \mathbf{U} \varphi_2] \mid \mathbf{A}[\varphi_1 \mathbf{R} \varphi_2] \end{aligned}$$

Semantics: $\mathbf{M} = \langle \mathbf{AP}, \mathbf{S}, \mathbf{N} \rangle$

Predicate transformers:

$$\begin{aligned} \mathit{post}(\mathbf{N})(S) &:= \mathbf{N}(S) = \{s' \in \mathbf{S} \mid \exists s \in S. s' \in \mathbf{N}(s)\} \\ \mathit{pre}(\mathbf{N})(S) &:= \tilde{\mathbf{N}}(S) = \{s \in \mathbf{S} \mid \exists s' \in S. s' \in \mathbf{N}(s)\} \\ \widehat{\mathit{pre}}(\mathbf{N})(S) &:= \{s \in \mathbf{S} \mid \mathbf{N}(s) \subseteq S\} \end{aligned}$$

CTL – Fixpoint Characterization

[Emerson, Clarke '81] $\mathbf{EG}\varphi = \nu Z. \varphi \wedge \mathbf{EX}Z$

$\llbracket \mathbf{EG}\varphi \rrbracket$ can be computed as a BDD given by $\llbracket \mathbf{EG}\varphi \rrbracket_k$ for the least k such that

$$\llbracket \mathbf{EG}\varphi \rrbracket_k = \llbracket \mathbf{EG}\varphi \rrbracket_{k+1}$$

where

$$\begin{aligned}\llbracket \mathbf{EG}\varphi \rrbracket_0 &= \llbracket \varphi \rrbracket \\ \llbracket \mathbf{EG}\varphi \rrbracket_{i+1} &= \llbracket \mathbf{EG}\varphi \rrbracket_i \wedge \mathit{pre}(\mathbf{N})(\llbracket \mathbf{EG}\varphi \rrbracket_i)\end{aligned}$$

WMC – EG φ (I)

Input: φ, I, N, V, V^+

Output:

- pair of lists $\langle [U_0, \dots, U_k], [W_0, \dots, W_m] \rangle$
- $U_i = \langle S_i, B_i, O'_i, O''_i \rangle$
- $S_0 = I, B_i \subseteq S_i$
- O'_i corresponds to the subformula φ of **EG** φ
- O''_i is empty (relevant only for binary operators)
- W_i set of states representing the stages in the fixpoint computation

WMC – EG φ (II)

$$\mathbf{EGMC}(\varphi, I, \mathbf{N}, V, V^+) = \tag{1}$$

$$\text{let } O' = \mathbf{WMC}(\varphi, I, \mathbf{N}); \tag{2}$$

$$\langle \vec{U}', \vec{W}' \rangle = O'; \tag{3}$$

$$I' = I - (V \cup U'_0.B) \tag{4}$$

$$\text{in (if } I' = \emptyset \tag{5}$$

$$\text{then} \tag{6}$$

$$\text{let } \vec{W}^m = \mathit{pre}(\mathbf{N})^\wedge(V^+) \quad \% \nu Z.V^+ \wedge \mathit{pre}(\mathbf{N})(Z) \tag{7}$$

$$\text{in } \langle [I, I - W_m, O', -], \vec{W} \rangle \tag{8}$$

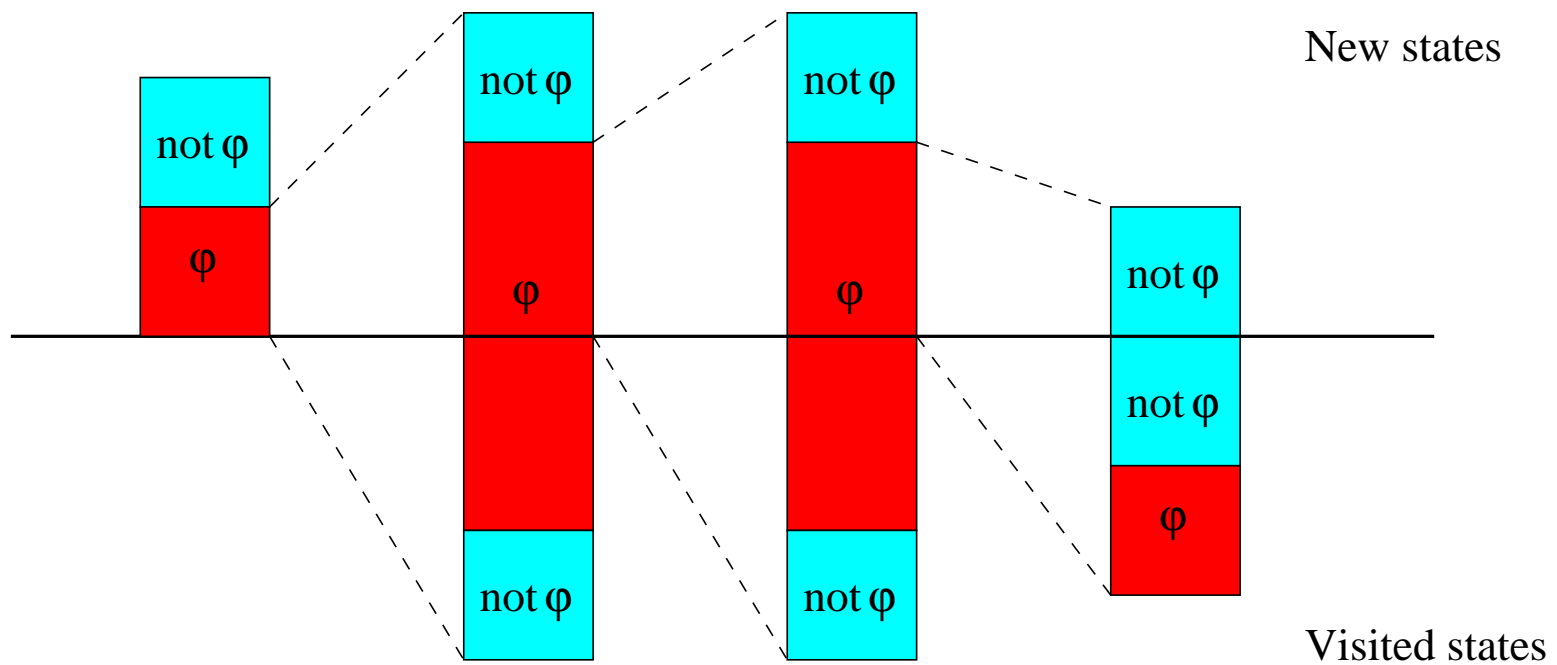
$$\text{else} \tag{9}$$

$$\text{let } \langle \vec{U}, \vec{W}^m \rangle = \mathbf{EGMC}(\varphi, \mathit{post}(\mathbf{N})(I'), \mathbf{N}, V \cup I, V^+ \cup I') \tag{10}$$

$$\text{in } \langle [I, I - W_m, O', -]; \vec{U}, \vec{W} \rangle \tag{11}$$

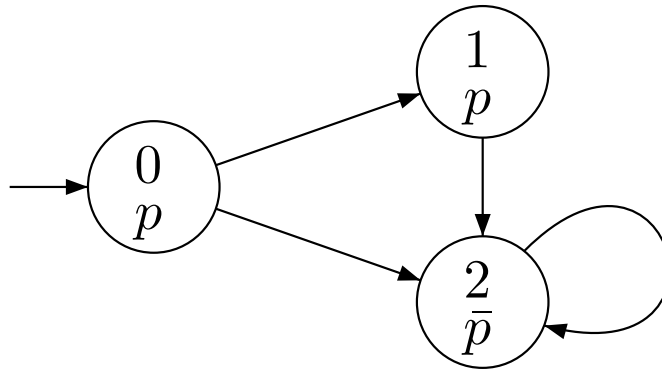
$$\text{endif) } \tag{12}$$

WMC EG_{φ} – graphical illustration forward exploration



Example – EG_p

Model:



Forward exploration:

Step	I	V	V ⁺	I'
0	{0}	\emptyset	\emptyset	{0}
1	{1, 2}	{0}	{0}	{1}
2	{2}	{0, 1, 2}	{0, 1}	\emptyset

Informations about p :

$$O'_0 = \langle [\langle \{0\}, \emptyset, -, - \rangle], [\emptyset] \rangle$$

$$O'_1 = \langle [\langle \{1, 2\}, \{2\}, -, - \rangle], [\{2\}] \rangle$$

$$O'_2 = \langle [\langle \{2\}, \{2\}, -, - \rangle], [\{2\}] \rangle$$

Final value of V^+ is $\{0, 1\}$

Backward computation (of the good states):

$$\vec{W} = pre(\mathbf{N})^\wedge(V^+) = \nu Z.V^+ \wedge pre(\mathbf{N})(Z) = [\{0, 1\}, \{0\}, \emptyset]$$

Output: $\langle \vec{U}, \vec{W} \rangle$ with

$$\vec{U} = [\langle \{0\}, \boxed{\{0\}}, O'_0, - \rangle, \langle \{1, 2\}, \{1, 2\}, O'_1, - \rangle, \langle \{2\}, \{2\}, O'_2, - \rangle]$$

Counterexample: sequence $[C_0, C_1, C_2]$, where $C_i = I_i \cap B_i$

$$[\{0\}, \{1, 2\}, \{2\}]$$

WMC – AG φ

$$\mathbf{AGMC}(\varphi, I, \mathbf{N}, V, V^+) = \tag{1}$$

$$\text{let } O' = \mathbf{WMC}(\varphi, I, \mathbf{N}); \tag{2}$$

$$\langle \vec{U}', \vec{W}' \rangle = O'; \tag{3}$$

$$I' = I - (V \cup U'_0.B) \tag{4}$$

$$\text{in (if } I' = \emptyset \tag{5}$$

$$\text{then} \tag{6}$$

$$\text{let } \vec{W}^m = \widetilde{pre}^\wedge(\mathbf{N})(V^+) \quad \% \nu Z.V^+ \wedge \widetilde{pre}(\mathbf{N})(Z) \tag{7}$$

$$\text{in } \langle [\langle I, I - W_m, O', - \rangle], \vec{W} \rangle \tag{8}$$

$$\text{else} \tag{9}$$

$$\text{let } \langle \vec{U}, \vec{W}^m \rangle = \mathbf{AGMC}(\varphi, post(\mathbf{N})(I'), \mathbf{N}, V \cup I, V^+ \cup I') \tag{10}$$

$$\text{in } \langle [\langle I, I - W_m, O', - \rangle; \vec{U}], \vec{W} \rangle \tag{11}$$

$$\text{endif) } \tag{12}$$

WMC – AF φ

$$\mathbf{AFMC}(\varphi, I, \mathbf{N}, V, V^-) = \tag{13}$$

$$\text{let } O' = \mathbf{WMC}(\varphi, I, \mathbf{N}); \tag{14}$$

$$\langle \vec{U}', \vec{W}' \rangle = O'; \tag{15}$$

$$I' = U'_0.B - V \tag{16}$$

$$\text{in (if } I' = \emptyset \tag{17}$$

$$\text{then} \tag{18}$$

$$\text{let } \vec{W}^m = \text{pre}(\mathbf{N})^\wedge(V^-) \tag{19}$$

$$\text{in } \langle \langle [I, U'_0.B \cap W_m, O', -], \vec{W} \rangle \tag{20}$$

$$\text{else} \tag{21}$$

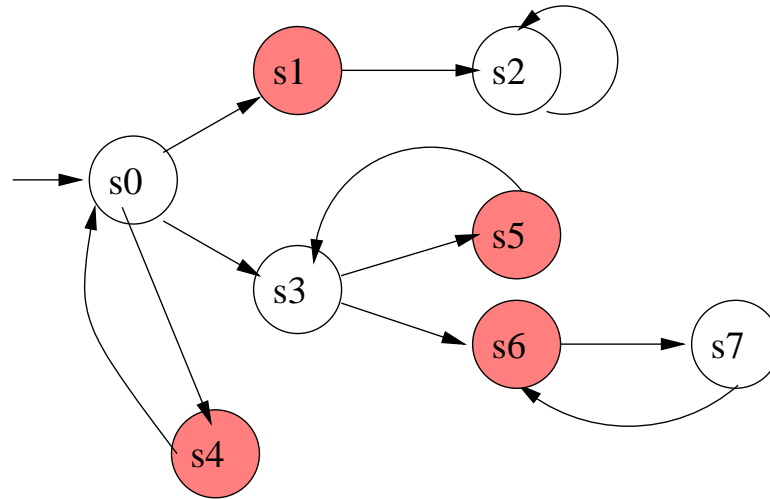
$$\text{let } \langle \vec{U}, \vec{W}^m \rangle = \mathbf{AFMC}(\varphi, \text{post}(\mathbf{N})(I'), \mathbf{N}, V \cup I, V^- \cup I') \tag{22}$$

$$\text{in } \langle \langle [I, (I \cap W_m), O', -]; \vec{U} \rangle, \vec{W} \rangle \tag{23}$$

$$\text{endif) } \tag{24}$$

Example – AFp

Model:



Forward exploration:

Step	I	V	V^-	I'
0	$\{0\}$	\emptyset	\emptyset	$\{0\}$
1	$\{1, 3, 4\}$	$\{0\}$	$\{0\}$	$\{3\}$
2	$\{5, 6\}$	$\{0, 1, 3, 4\}$	$\{0, 3\}$	\emptyset

Informations about p :

$$O'_0 = \langle \langle \{0\}, \{0\}, -, - \rangle, [\{0\}] \rangle$$

$$O'_1 = \langle \langle \{1, 3, 4\}, \{3\}, -, - \rangle, [\{3\}] \rangle$$

$$O'_2 = \langle \langle \{5, 6\}, \emptyset, -, - \rangle, [\emptyset] \rangle$$

Final value of V^- is $\{0, 3\}$

Backward computation (of the bad states):

$$\vec{W} = pre(\mathbf{N})^\wedge(V^-) = \nu Z.V^- \wedge pre(\mathbf{N})(Z) = [\{0, 3\}, \emptyset]$$

Output: $\langle \vec{U}, \vec{W} \rangle$ with

$$\vec{U} = [\langle \{0\}, \boxed{\emptyset}, O'_0, - \rangle, \langle \{1, 3, 4\}, \emptyset, O'_1, - \rangle, \langle \{5, 6\}, \emptyset, O'_2, - \rangle]$$

Witness: sequence $[G_0, G_1, G_2]$, where $G_i = I_i - B_i$

$$[\{0\}, \{1, 3, 4\}, \{5, 6\}]$$

WMC vs. Fixpoint Characterization

Local fixpoint computation in WMC for **EG** φ :

$$\vec{W}^m = pre(\mathbf{N})^\wedge(V^+) = \nu Z. V^+ \wedge pre(\mathbf{N})(Z)$$

Fixpoint definition of **EG** φ :

$$\mathbf{EG}\varphi = \nu Z. \varphi \wedge \mathbf{EX}Z$$

Local fixpoint computation in WMC for **AF** φ :

$$\vec{W}^m = pre(\mathbf{N})^\wedge(V^-) = \nu Z. V^- \wedge pre(\mathbf{N})(Z)$$

Fixpoint definition of **AF** φ :

$$\mathbf{AF}\varphi = \mu Z. \varphi \vee \mathbf{AX}Z$$

Negation: $\nu Z. \neg\varphi \wedge \mathbf{EX}Z$

Theoretical Results

A witness is a justification that $\mathbf{M}, G \models \varphi$.

$$w \vdash \mathbf{M}, G \models \mathbf{EG}\varphi$$

A counterexample is a justification that $\mathbf{M}, C \not\models \varphi$.

$$c \vdash \mathbf{M}, C \not\models \mathbf{EG}\varphi$$

Each execution of the model checker partitions I into I^+ and I^- , such that $\mathbf{M}, I^+ \models \varphi$ and $\mathbf{M}, I^- \not\models \varphi$.

Theoretical Results – Cont.

Theorem 1 [Witness Validity]

If $w \vdash \mathbf{M}, G \models \varphi$ then $\mathbf{M}, G \models \varphi$, and if $c \vdash \mathbf{M}, C \not\models \varphi$, then $\mathbf{M}, C \not\models \varphi$.

Theorem 2 [Correctness]

Let O be $\mathbf{WMC}(\varphi, I, \mathbf{N})$, then there exist disjoint sets G and C such that $I \subseteq G \cup C$, a witness $w = \mathit{witness}(\varphi, \mathbf{N}, O)$, and a counterexample $c = \mathit{counter}(\varphi, \mathbf{N}, O)$, such that $w \vdash \mathbf{M}, G \models \varphi$ and $c \vdash \mathbf{M}, C \not\models \varphi$.

Experimental Results – Synchronous Arbiter

Mutual exclusion property of a buggy version.

No. cells	WMC			MC		
	BDD size	msecs	No. iter	BDD size	msecs	No. iter
5	40	0	3	381	10	3
10	92	130	3	1041	260	4
15	142	80	3	1317	440	3
20	199	229	3	2297	1210	5
25	251	290	3	4838	3510	4
30	301	470	3	5533	2500	3
35	347	530	3	5657	1710	3
40	407	660	3	9976	11360	4
45	455	880	3	6801	4804	3
47	101	150	2	3338	1330	3

Experimental Results – Synchronous Arbiter

Mutual exclusion property of the correct version.

No. cells	WMC			MC		
	BDD size	msecs	No. iter	BDD size	msecs	No. iter
5	25	0	1	186	20	3
10	38	10	1	673	140	3
15	62	20	1	892	320	3
20	86	210	1	1265	250	4
25	102	40	1	2934	890	4
30	121	90	1	3086	1170	4
35	137	140	1	3406	1260	4
40	162	120	1	8164	4040	4
45	181	380	1	3737	2890	4
47	96	80	1	1647	540	3

Benefits

Witnesses and counterexamples generation for the entire CTL.

Fixpoint definitions of the CTL operators are not evaluated on the entire state space.

In general not necessary to compute the reachable states set.

The unfolding of the state space and the fixpoint computation are done according to the given temporal property.

Related Work

[Clarke, Grumberg, McMillan, and Zhao '95] present techniques for the efficient generation of counterexamples for fragments of ACTL and Fair ACTL (CTL with fairness constraints).

Counterexample construction for the three CTL basic operators: **AX**, **AG**, and **AU**, are given, and the problem of finding fair counterexamples is classified as NP-complete. The algorithms for generating counterexamples have been implemented in the SMV model checker [McMillan].

[Kick '96] shows that it is possible to construct tree-like counterexamples for the entire μ -calculus, but the resulting trees are large and quite complicated.

[Clarke, Jha, Lu, and Veith '02] investigate tree-like counterexamples for ACTL based on a backward and then forward exploration of the state space.

[Namjoshi '01] introduced the notion of a *certifying model checker* that can generate independently checkable witnesses for properties verified by a model checker. He defined witnesses for properties of labelled transition systems expressed in the modal μ -calculus based on parity games over alternating tree automata.

[Peled, Pnueli, and Zuck '01] produce deductive proofs for successfully model checked LTL formulas based on identifying the strongly connected components in the model checking tableau and generating a proof for the absence of feasible paths.

[Gurfinkel and Chechik '03] present an approach for annotating model checker witnesses with proof steps and generating proof obligations that can be independently verified with a theorem prover.

All of the above methods generate explanations only when the model checker achieves a verification or a refutation, whereas our approach produces simple and direct witnesses and counterexamples that partition the initial states into good and bad states, respectively. Such a partition is needed for the recursive invocation of the model checker on subformulas of the given formula which do not involve either verification or refutation.

[Iwashita, Nakata, and Hirose '96] present a CTL model-checking algorithm based on forward state traversal for a fragment of CTL. They show that in many situations backward state traversal is more expensive than forward traversal. When combined with BDD-based state traversal techniques using partitioned transition relations, their method could be successfully applied for verifying large finite-state systems.

[Henzinger, Kupferman, and Qadeer '98] investigate the class of specifications that can be checked by symbolic forward state traversal and show that all ω -regular (linear-time) specifications can be verified using forward traversal.

[Biere, Clarke, and Zhu '99] give a local tableau construction for LTL model checking based on sets of states and forward image computation with explicit detection of strongly connected components in the tableau.

Our two-phase symbolic model checking algorithm for CTL consists of a forward traversal that identifies relevant reachable states followed by a backward traversal that partitions the initial set of states into good and bad states.

Conclusions

Simple local model-checking algorithm for CTL that constructs witnesses and counterexamples.

Constrained, forward unfolding of the state space starting from the initial states, followed by a fixpoint computation.

Symbolic witnesses and counterexamples are constructed from the results produced by the model checker.

The model-checking algorithm has been proved to produce valid witnesses and counterexamples.

Preliminary experiments indicate that our method is significantly more efficient than the standard method used for CTL model checking despite the overhead of collecting evidence.

Work in Progress / Future Work

Construct an independent evidence checker for the witnesses and counterexamples produced by our model checker.

Early termination criteria.

Restrict our search to bounded length evidence by bounding the forward unfolding.

Combine counterexample-driven model checking with abstraction to compute over and under-approximations of fixpoint properties.

Application of the algorithm to controller synthesis, where the backward fixpoint computation employs a *controlled* precondition operation.

Adapting the approach to Fair CTL, LTL, CTL*.