Minimizing Counterexample of ACTL Property

ShengYu Shen[1], Ying Qin[1], and SiKun Li[1]

School of Computer Science, National University of Defense Technology of China
{syshen,qy123,skli}@nudt.edu.cn

Abstract. Counterexample minimization tries to remove irrelevant variables from counterexamples, such that they are easier to be understood. For the first time, we proposes a novel approach to minimize loop-like and path-like counterexamples of ACTL properties. For a counterexample $s_0 \dots s_k$, our algorithm tries to extract a succinct cube sequence $c_0 \dots c_k$, such that paths run through $c_0 \dots c_k$ are all valid counterexamples. Experimental result shows that our algorithm can significantly minimize ACTL counterexamples. [1]

1 Preliminaries

BDD contain two terminal nodes and a set of variable nodes. Attribute $\mathbf{value}(\mathbf{u})$ is associated with terminal nodes u. Every variable node has two outgoing edges: $\mathbf{low}(\mathbf{u})$ and $\mathbf{high}(\mathbf{u})$. A variable $\mathbf{var}(\mathbf{u})$ is associated with every node u.

Symbolic model checking with BDD is first proposed by K.McMillan [1], which is implemented by procedure **Check** that takes a CTL formula and returns BDD of those states that satisfy the formula.

Assume the state variable set of Kripke structure $M = \langle S, I, T, L \rangle$ is $V = \{v_0, \dots, v_n\}$. A state $s \in S$ can be seen as assignments to V, which is denoted by $s = \{v_0 \leftarrow b_0, \dots, v_n \leftarrow b_n\}$, with $b_i \in \{0, 1\}$ are boolean constant. Assume $V' = \{v_{i_0}, \dots, v_{i_m}\}$ is a subset of V, then projection of s to V' is defined as

$$s|_{V'} = \{v_{i_0} \leftarrow b_{i_0}, \dots, v_{i_m} \leftarrow b_{i_m}\} \tag{1}$$

A state set $S' \subseteq S$ is a **cube** iff there exists $V' = \{v_{i_0}, \dots, v_{i_m}\} \subseteq V$ and $\{b_{i_0}, \dots, b_{i_m}\}$, such that $S' == \{s | \ s|_{V'} == \{v_{i_0} \leftarrow b_{i_0}, \dots, v_{i_m} \leftarrow b_{i_m}\}\}$

Assume state s is in state set S', then c is a cube guided by s in S' iff $s \in c \subseteq S'$. We denote c by $\mathbf{GuidedCube}(\mathbf{S}', \mathbf{s})$, it can be computed as below.

Algorithm 1: Computing $GuidedCube(S', s)$

1. Assume $s = \{v_0 \leftarrow b_0, \dots, v_n \leftarrow b_n\}$.
2. $c = \phi \quad V' = \phi$ are all empty set
3. cn=root node of BDD of S'
4. while(cn isn't a terminal node)
   (a) assume $var(cn)$ is v_i
   (b) if($b_i == 0$) then $cn = low(cn)$ else $cn = high(cn)$
   (c) $c = c \cup \{v_i \leftarrow b_i\} \qquad V' = V' \cup \{v_i\}$
5. $GuidedCube(S', s) = \{s' | \ s'|_{V'} == c\}$

[1] Supported by Chinese NSF under Grant No.90207019 and No.60403048; the Chinese 863 Program under Grant No. 2002AA1Z1480

2 Minimizing Counterexample of ACTL Property

Existing approaches[2] can only deal with path-like counterexamples of invariant $AG\ f$. For the first time, this paper proposes a novel approach to minimize loop-like and path-like counterexamples of ACTL properties. Due to duality of ACTL and ECTL, we will focus on minimizing witness of ECTL formula.

To make a witness $s_0 \dots s_k$ more easy to be understood, some state variables must be removed. So a minimized witness must be a cube sequence $c_0 \dots c_k$. We define the criteria that it must satisfied.

Definition 1 (Criteria of Minimized Witness of ECTL Property). *Assume $s_0 \dots s_k$ is a witness of an ECTL property f. Cube sequence $c_0 \dots c_k$ is the minimized witness of $s_0 \dots s_k$ iff*

1. $s_i \in c_i (0 \leq i \leq k)$
2. *Every path $s'_0 \dots s'_k$ that satisfy $\bigwedge_{0 \leq i \leq k} s'_i \in c_i$ must be witness of f*

We will discuss minimizing witness of EX, EU and EG below.

2.1 Minimizing Witness of EX and EU

Assume $PreImage(S')$ is a procedure that computes pre-image of S'. We can minimize EXf witness $s_0 s_1$ and $E[fUg]$ witness $s_0 \dots s_{k-1}$ in the following way:

Algorithm 2: Minimizing Witness of $EX\ f$

1. $c_1 = GuidedCube(Check(f), s_1)$
2. $c_0 = GuidedCube(PreImage(c_1), s_0)$

Algorithm 3: Minimizing Witness of $E[f\ U\ g]$

1. $c_{k-1} = GuidedCube(Check(g), s_{k-1})$
2. for $i = k-2$ to 0
3. $\quad c_i = GuidedCube(PreImage(c_{i+1}) \cap Check(f), s_i)$

Correctness proof is omited due to space limitation.

2.2 Minimizing Witness of EG

A loop-like witness of EGf contains two segments: a stem $s_0 \dots s_m$ and a loop $s_m \dots s_n$. We will first prove the following theorem below.

Theorem 1. *Assume a loop-like witness of EGf contains two segments: a stem $s_0 \dots s_m$ and a loop $s_m \dots s_n$. Then a cube sequence $c_0 \dots c_n$ is its minimized witness if the following 4 equations hold true*

$$\bigwedge_{0 \leq i \leq n} s_i \in c_i \tag{2}$$

$$c_n \subseteq PreImage(c_m) \wedge \bigwedge_{m \le i \le n-1} c_i \subseteq PreImage(c_{i+1}) \tag{3}$$

$$\bigwedge_{0 \le i \le m-1} c_i \subseteq PreImage(c_{i+1}) \tag{4}$$

$$\bigwedge_{0 \le i \le n} c_i \subseteq Check(f) \tag{5}$$

Proof. By equation (2), the 1st criteria of Definition 1 is satisfied.

Assume a path $s'_0 \dots s'_n$ satisfy $T(s'_n, s'_m) \wedge \bigwedge_{0 \le i \le n} s'_i \in c_i$. By equation (5), $\bigwedge_{0 \le i \le n} M, s'_i \models f$.

Thus this theorem is proven.

We compute an approximation of $c_m \dots c_n$ with following algorithm.

Algorithm 4 $Min(x)$

1. $c_m = x$
2. $c_n = GuidedCube(PreImage(c_m) \cap Check(f), s_n)$
3. For $i = n - 1$ to m
4. $\quad c_i = GuidedCube(PreImage(c_{i+1}) \cap Check(f), s_i)$
5. return c_m

To compute $c_m \dots c_n$ that satisfies equation (3), we first let

$$C = Check(EGf) \tag{6}$$

And then run $Min(C)$. Cube sequence $c_m \dots c_n$ obtained in this way satisfies almost all $\subseteq$ relation in equation (3), except $c_n \subseteq PreImage(c_m)$.

So we need to run Algorithm 4 iteratively, and obtain the following sequence:

$$Min(C), Min^2(C), \dots Min^t(C), \dots \tag{7}$$

We terminate above iteration only when $Min^{t-1}(C) \subseteq Min^t(C)$, at which $c_n \subseteq PreImage(c_m)$ and equation (3) can be satisfied. So we must prove that iteration in equation (7) is terminable with following theorems.

Theorem 2. *$Min(x)$ is monotonic. (Proof is omited due to space limitation)*

Theorem 3. $C \supseteq Min(C)$

Proof. By Algorithm 4, for every state $s'_m \in Min(C)$, there is a path $s'_m s'_{m+1} \dots s'_n s_m$", such that s_m" $\in C$. That is to say, there is an infinite path p starting from s_m", and f holds true at all states along p.

By Algorithm 4, f holds true on all states of $s'_m s'_{m+1} \dots s'_n s_m$".

Thus, we can concatenate $s'_m s'_{m+1} \dots s'_n s_m$" and p, to form a new path p'. f hold true at all states along p'. Thus, p' is witness of M, $s'_m \models EGf$.

By equation (6), we can conclude that $s'_m \in C$.

Thus, $C \supseteq Min(C)$ is proven.

Table 1. Experimental Result

Cex name	Cex length	Original cex		Minimized cex	
		Number of. Variables.	Run time	Number of Variables	Run time
P1	13	1027	0.12	244	0.12
P2	7	308	0.01	172	0.02
L1	64	975	0.991	791	1.45
L2	76	1140	1.26	942	1.96
L3	75	1125	2.83	929	4.09
L4	22	858	0.19	510	0.24
L5	22	858	0.28	467	0.33
L6	22	858	0.16	455	0.17
L7	22	858	0.12	408	0.17

Theorem 4. *The iteration in equation (7) is terminable.*

Proof. By Theorem 2 and 3, it is obvious that : $C \supseteq Min(C) \ldots \supseteq Min^t(C) \ldots$. So $\exists t.Min^{t-1}(C) == Min^t(C)$ hold true. Thus, this theorem is proven.

Thus, we can construct minimized witness $c_m \ldots c_n$ in the following way:

Algorithm 5: Minimizing Witness of $EG\ f$

1. $c_m = Min^t(C)$
2. $c_n = GuidedCube(PreImage(c_m) \cap Check(f), s_n)$
3. for $i = n - 1$ to 0
4. $\quad c_i = GuidedCube(PreImage(c_{i+1}) \cap Check(f), s_i)$

3 Experimental Result

We implement our algorithm in NuSMV, and perform experiments on NuSMV's benchmarks. All experiments run on a PC with 1GHz Pentium 3.

Table 1 presents experimental result. The 1st column lists the name of counterexamples. P1 and P2 are path-like counterexamples. All others are loop-like counterexamples. The 2nd column lists their length. The 3rd column lists the number of variables in original counterexamples. The 4th column lists the time taken by NuSMV to generate these counterexamples. The 5th column lists the number of variables in minimized counterexamples. The last column lists the run time of our approach.

From the experimental result, it is obvious that our algorithm can significantly minimize counterexamples.

References

1. K.L.McMillan. Symbolic model checking. Kluwer Academic Publishers, 1993.
2. K. Ravi and F. Somenzi. Minimal assignments for bounded model checking. In TACAS'04,LNCS 2988, pages 31-45, 2004.