

改进的验证正确性 ACTL 性质的限界模型检测方法

徐 亮 余建平

(湖南师范大学数学与计算机科学学院 长沙 410081)

(高性能计算与随机信息处理省部共建教育部重点实验室 长沙 410081)

摘 要 近些年来,基于 SAT 的限界模型检测方法作为基于 BDD 的限界模型检测方法的一种有效补充,已经得到了一定的发展。其中,大部分的研究成果都集中在了使用该方法来进行系统查错方面,而在正确性性质的验证上一直未有突破,原因在于正确性性质的验证依赖于一个完备上界,而这个完备上界在限界模型检测方法中很难实现。对传统限界模型检测中的编码方式进行相应改变,就能够在一定程度上解决这一问题,进行正确性性质的验证。在此基础上对该编码方法进行改进,从而提高它的求解效率,扩大其应用领域。

关键词 限界模型检测,可满足性求解,全局计算树逻辑,验证

中图法分类号 TP301.2 **文献标识码** A

Improved Bounded Model Checking on Verification of Valid ACTL Properties

XU Liang YU Jian-ping

(College of Mathematics and Computer Science, Hunan Normal University, Changsha 410081, China)

(Key Laboratory of High Performance Computing and Stochastic Information Processing, Ministry of Education of China, Changsha 410081, China)

Abstract SAT-based bounded model checking has been introduced as a complementary technique to BDD-based symbolic model checking in recent years, and a lot of successful work has been done with this approach. The success is mostly due to the efficiency of error-detection. Verification of valid properties depends on a completeness threshold that could be too large to be practical. In order to check the valid ACTL properties, the encodings in traditional bounded model checking should be changed. Also, some improvements have been done for this method to improve its efficiency and enlarge its application fields.

Keywords Bounded model checking, SAT, ACTL, Verification

1 引言

模型检测这一概念是由 Clarke 和 Emerson 在 20 世纪 80 年代早期提出的^[1]。在模型检测方法当中,用来验证的系统被设计成一个有限状态的模型,性质则用时序逻辑(Temporal Logic)公式来描述。模型检测通常被认为是一种程序验证方法,但存在一定的局限性。Biere 等人在 1999 年提出了基于可满足性求解(SAT)的 LTL^[2]限界模型检测方法^[3,4]。该方法作为基于 BDD^[5]的符号模型检测方法^[6,7]是一个有效而必要的补充,而它的思想后来也被应用到了对 ACTL(全局计算树逻辑)^[8]性质的验证上^[9]。限界模型检测方法之所以有效,是因为如果一个系统有错,那么我们只需要通过状态空间的一部分就能够找到这个错误。而对于正确性的验证,为了确定系统是没有错误的,我们必须对一条很长的路径进行检测,这条路径的长度称之为完备上界。这个完备上界往往非常大,以至于用上述限界模型检测方法没有办法求解。为了解决这一问题^[10],通过对描述性质的公式进行特定的编码来有

效地避免验证依赖于完备上界的这一问题,从而有效地验证正确性的性质。本文的主要工作就是对这一方法进行进一步的改进,以提高该方法的求解效率。

2 计算树逻辑

计算树逻辑(CTL)是一种分支时序逻辑,被作为描述有限状态系统的一种声明语言^[8]。假设 AP 是表示性质的符号集合,则 CTL 公式的定义如下:

- 原子命题——AP 中的任何一个元素是一个 CTL 公式;

- 逻辑连接符——如果 φ 和 ψ 是 CTL 公式,那么 $\neg\varphi$ 、 $\varphi \wedge \psi$ 、 $\varphi \vee \psi$ 、 $\varphi \rightarrow \psi$ 都是 CTL 公式;

- 时序操作符——如果 φ 和 ψ 是 CTL 公式,那么 EX(φ)、EF(φ)、EG(φ)、E($\varphi U \psi$)、E($\varphi R \psi$)、AX(φ)、AF(φ)、AG(φ)、A($\varphi U \psi$)、A($\varphi R \psi$)都是 CTL 公式。

CTL 公式的一个模型是一个 Kripke 结构 $M = \langle S, T, I, L \rangle$, 其中 S 是有限状态的集合; $T \subseteq S \times S$ 表示迁移关系,每一

本文受国家自然科学基金项目(60903168),湖南省自然科学基金项目(12JJ6063),湖南省科技计划项目(2012FJ6012),长沙市科技计划项目(K1109020-11),湖南省重点学科建设项目(湘教发[2011]76号)资助。

徐 亮(1981—),男,博士,讲师,CCF 会员,主要研究方向为形式化方法、模型检测、安全策略模型研究,E-mail: xul1981@126.com;余建平(1979—),男,博士,副教授,CCF 会员,主要研究方向为网络优化、群体智能、网络安全。

个状态都至少有一个后继; $I \subseteq S$ 是初始状态的集合; $L: S \rightarrow 2^{AP}$ 是一个标记函数, 将 S 中的每一个状态映射到一个在该状态上成立的所有原子命题组成的集合上。对 $i \geq 0$ 有 $w_i \subseteq S$ 并且 $T(w_i, w_{i+1})$ 成立, 则称 $\pi = w_0 w_1 \dots$ 为 M 的一条路径。

定义 1 (CTL 公式的语义) 令 M 表示模型, $s \in S$, $p \in AP$, φ 和 ψ 表示 CTL 公式。 $M, s | = \varphi$ 表示模型 M 中的状态 s 满足 φ 。令 π 为 M 的一条路径, 则关系 $| =$ 的定义如下:

$M, s = p$	$\Leftrightarrow p \in L(s)$
$M, s = \neg \varphi$	$\Leftrightarrow M, s \not\models \varphi$
$M, s = \varphi \wedge \psi$	$\Leftrightarrow (M, s = \varphi) \wedge (M, s = \psi)$
$M, s = \varphi \vee \psi$	$\Leftrightarrow (M, s = \varphi) \vee (M, s = \psi)$
$M, s = \varphi \rightarrow \psi$	$\Leftrightarrow (M, s = \varphi) \rightarrow (M, s = \psi)$
$M, s = EX \varphi$	$\Leftrightarrow \exists \pi. ((w_0 = s) \wedge (M, w_1 = \varphi))$
$M, s = EF \varphi$	$\Leftrightarrow \exists \pi. ((w_0 = s) \wedge \exists k \geq 0. (M, w_k = \varphi))$
$M, s = EG \varphi$	$\Leftrightarrow \exists \pi. ((w_0 = s) \wedge \forall k \geq 0. (M, w_k = \varphi))$
$M, s = E(\varphi U \psi)$	$\Leftrightarrow \exists \pi. ((w_0 = s) \wedge \exists k \geq 0. ((M, w_k = \psi) \wedge \forall j < k. (M, w_j = \varphi)))$
$M, s = E(\varphi R \psi)$	$\Leftrightarrow \exists \pi. ((w_0 = s) \wedge (\forall j \geq 0. (M, w_j = \psi) \vee \exists k \geq 0. ((M, w_k = \psi) \wedge \forall j \leq k. (M, w_j = \varphi))))$
$M, s = AX \varphi$	$\Leftrightarrow \forall \pi. ((w_0 = s) \rightarrow (M, w_1 = \varphi))$
$M, s = AF \varphi$	$\Leftrightarrow \forall \pi. ((w_0 = s) \rightarrow \exists k \geq 0. (M, w_k = \varphi))$
$M, s = AG \varphi$	$\Leftrightarrow \forall \pi. ((w_0 = s) \rightarrow \forall k \geq 0. (M, w_k = \varphi))$
$M, s = A(\varphi U \psi)$	$\Leftrightarrow \forall \pi. ((w_0 = s) \rightarrow \exists k \geq 0. ((M, w_k = \psi) \wedge \forall j < k. (M, w_j = \varphi)))$
$M, s = A(\varphi R \psi)$	$\Leftrightarrow \forall \pi. ((w_0 = s) \rightarrow (\forall j \geq 0. (M, w_j = \psi) \vee \exists k \geq 0. ((M, w_k = \psi) \wedge \forall j \leq k. (M, w_j = \varphi))))$

我们称不含逻辑操作符 $\rightarrow$ 以及逻辑操作符 $\neg$ 只作用在原子命题前的 CTL 公式为 CTL 公式的正则形式。任何 CTL 公式都能转化成为与其等价的正则形式。只包含路径量词 E 的 CTL 公式称为 ECTL 公式, 用 F 表示。同样, 只包含路径量词 A 的 CTL 公式称为 ACTL 公式。这里我们只考虑公式的正则形式。

3 限界模型检测方法

在这一节中, 我们参照以前的一些研究工作^[9-11, 13], 来简要地给出关于 ECTL 的限界模型检测方法。

3.1 ECTL 公式的限界语义

为了定义 ECTL 公式的限界语义以及将限界模型检测问题编码成可满足性求解问题, 我们先给出一些必要的概念。令 $M = \langle S, T, I, L \rangle$ 是一个 Kripke 结构表示的模型, $k \in \mathbb{N}^+$ 是限界模型检测问题的界值, 若对 $0 \leq i \leq k-1$ 有 $(w_i, w_{i+1}) \in T$, 则称 $\pi_k = w_0 \dots w_k$ 为模型 M 的一条 k -路径, 该路径有 $k+1$ 个状态。四元组 $M_k = \langle S, Path_k, I, L \rangle$ 称为模型 M 的 k -模型, 其中 $Path_k$ 表示 M 中所有不同的 k -路径的集合。 $loop(\pi)$ 表示集合 $\{l | (l \leq k) \wedge (w_k, w_l) \in T\}$ 。考虑到时序运算符 EF 和 ER 可以用 EU 和 EG 来定义^[2], 接下来的讨论都省去了这两者。

定义 2 (ECTL 公式的限界语义) 令 M_k 表示模型 M 的 k -模型, $s \in S$, $p \in AP$, $\varphi, \psi \in F$, $\mathbb{N}_i$ 表示集合 $\{0, \dots, i\}$, 则 $M_k, s | =_k \varphi$ 表示 M_k 中的状态 s 满足 φ 。则关系 $| =_k$ 的定义如下:

$M_k, s =_k p$	$\Leftrightarrow p \in L(s)$
$M_k, s =_k \neg p$	$\Leftrightarrow p \notin L(s)$
$M_k, s =_k \varphi \wedge \psi$	$\Leftrightarrow (M_k, s =_k \varphi) \wedge (M_k, s =_k \psi)$
$M_k, s =_k \varphi \vee \psi$	$\Leftrightarrow (M_k, s =_k \varphi) \vee (M_k, s =_k \psi)$
$M_k, s =_k EX \varphi$	$\Leftrightarrow \exists \pi_k. ((w_0 = s) \wedge (M_k, w_1 =_k \varphi))$
$M_k, s =_k EG \varphi$	$\Leftrightarrow \exists \pi_k. ((w_0 = s) \wedge loop(\pi) \wedge \forall i \in \mathbb{N}_k. (M_k, w_i =_k \varphi))$
$M_k, s =_k E(\varphi U \psi)$	$\Leftrightarrow \exists \pi_k. ((w_0 = s) \wedge \exists i \in \mathbb{N}_k. ((M_k, w_i =_k \psi) \wedge \forall j \in \mathbb{N}_{i-1}. (M_k, w_j =_k \varphi)))$

一个 ECTL 公式 φ 在 k -模型 M_k 中成立, 表示成 $M_k | =_k \varphi$, 当且仅当 k -模型的初始状态满足公式 φ , 即 $M_k, s_0 | =_k \varphi$ 。

定理 1 令 $\varphi \in F$, 则有: i) $M | = \varphi \Leftrightarrow \exists k \geq 0. (M_k | =_k \varphi)$; ii) $M_k | =_k \varphi \Rightarrow M_{k+1} | =_{k+1} \varphi$ 。

3.2 ECTL 公式的编码

令 w 是表示状态变量的一个向量, $w = (w[1], \dots, w[n])$, 其中 $w[i], 1 \leq i \leq n$, 表示的是原子命题, $n = \lceil \log_2(|M|) \rceil$, 与模型的状态个数 $|M|$ 有关。对状态的编码可以通过对 $(w[1], \dots, w[n])$ 的真假赋值来表示。令 $k \geq 0$, $N_k(\varphi)$ 表示模型 M 中验证公式 φ 所需的不同 k -路径的条数, 详见文献^[9, 10, 12], 唯一不同的是 $N_k(E(\varphi_0 U \varphi_1)) = k \cdot N_k(\varphi_0) + \max(N_k(\varphi_0), N_k(\varphi_1)) + 1$; 等式 $w_i = w_j$ 定义为 $\bigwedge_{m=1}^n w_i[m] = w_j[m]$; $I(w)$ 表示的是状态 w 是模型的初始状态; $w_{i,j}$ 表示第 i 条路径上的第 j 个状态; $p(w_{i,j})$ 表示状态 $w_{i,j}$ 满足原子命题 p 。

定义 3 (k -模型的编码) 令 M_k 表示模型 M 的 k -模型, $\varphi \in F$, 则原子命题公式 $[[M^{\varphi}]]_k = \bigwedge_{i=1}^{N_k(\varphi)} \bigwedge_{j=0}^{k-1} T(w_{i,j}, w_{i,j+1})$ 。

定义 4 (ECTL 公式的编码) 令 $k \geq 0$, 给定一个状态 w 和一个公式 $\varphi \in F$, 编码 $[[\varphi, w]]_k$ 定义如下:

$[[p, w]]_k$	$= p(w)$
$[[\neg p, w]]_k$	$= \neg p(w)$
$[[\varphi \wedge \psi, w]]_k$	$= [[\varphi, w]]_k \wedge [[\psi, w]]_k$
$[[\varphi \vee \psi, w]]_k$	$= [[\varphi, w]]_k \vee [[\psi, w]]_k$
$[[EX \varphi, w]]_k$	$= \bigvee_{i=1}^{N_k(EX(\varphi))} ((w = w_{i,0}) \wedge [[\varphi, w_{i,1}]]_k)$
$[[EG \varphi, w]]_k$	$= \bigvee_{i=1}^{N_k(EG(\varphi))} ((w = w_{i,0}) \wedge \bigwedge_{j=0}^k [[\varphi, w_{i,j}]]_k)$
$[[E(\varphi U \psi), w]]_k$	$= \bigvee_{i=1}^{N_k(E(\varphi U \psi))} ((w = w_{i,0}) \wedge (\bigvee_{j=0}^k [[\psi, w_{i,j}]]_k \wedge \bigwedge_{t=0}^{j-1} [[\varphi, w_{i,t}]]_k) \vee \bigwedge_{t=0}^{j-1} [[\varphi, w_{i,t}]]_k \vee \bigwedge_{t=0}^k [[\varphi, w_{i,t}]]_k)$

其中当 $j > k$ 时, $[[\varphi, w_{i,j}]]_k$ 赋值为真。

文献^[9]中编码的基本思想是, 如果编码是可满足的, 则表示被编码的问题实例有证明; 反之, 则表示该问题实例没有证明。由上述编码, 我们得到了最终进行可满足性证明的编码 $[[M, \varphi]]_k$ 。

定义 5 令 w 是一个状态变量, $\varphi \in F$, 则 $[[M, \varphi]]_k = I(w) \wedge [[M^{\varphi}]]_k \wedge [[\varphi, w]]_k$ 。

引理 1 令 $\varphi \in F, k \geq 0$, 则:

如果 $M_k | =_k \varphi$, 则 $[[M, \varphi]]_k$ 是可满足的; 如果 $[[M,$

$\varphi]]_{k+1}$ 是可满足的,则 $[[M, \varphi]]_k$ 也是可满足的。

相关证明参见文献[12,13]。

综合引理1和定理1,可以得到如下定理。

定理2 令 $\varphi \in F$, 如果 $M \models \varphi$, 则对所有的 $k \geq 0$, $[[M, \varphi]]_k$ 都是可满足的。

因此,如果对某一个 $k \geq 0$ 有 $[[M, \neg\psi]]_k$ 是不可满足的,其中 ψ 是一个 ACTL 公式,那么就有 $M \models \psi$ 。从而,我们得到了验证正确的 ACTL 性质的限界模型检测方法,即对 $k = 0, 1, 2, \dots$ 看编码 $[[M, \neg\psi]]_k$ 是否是不可满足的。

3.3 验证方法

定理2给出了验证正确性 ACTL 公式的基本思路。为了简化编码后的 SAT 公式,我们基于以下推论来进行验证。我们用 $w_{1,0}$ 来代替定理2中的 w 。令 $[[M, \varphi]]_k^* = I(w_{1,0}) \wedge [[M^p]]_k \wedge [[\varphi, w]]_k$, 则 $[[M, \varphi]]_k^*$ 和 $[[M, \varphi]]_k$ 具有相同的可满足性。由定理2得如下推论:

推论1 令 $\varphi \in F$, 如果 $M \models \varphi$, 则对所有的 $k \geq 0$, $[[M, \varphi]]_k^*$ 都是可满足的。

由此,我们就可以通过检查编码 $[[M, \neg\psi]]_k^*$ 的可满足性来验证一个 ACTL 性质 ψ 。在 $k = 0, 1, 2, \dots$ 时检查 $[[M, \neg\psi]]_k^*$ 的可满足性,只要 $[[M, \neg\psi]]_k^*$ 在某一个 k 值不满足,就可以说性质 ψ 在模型 M 中是有效的;若到达资源的上界还没有找到关于 $[[M, \neg\psi]]_k^*$ 不可满足性的赋值,则不能给出任何结论。

4 改进的限界模型检测方法

由文献[12-14]可知, $[[M, \varphi]]_k^*$ 中的公式具有旋转对称性,当对其每一个成员的标号进行置换 $\sigma = \{k_0 \rightarrow 0, \dots, k_n \rightarrow n\}$, 并对原公式的可满足性赋值进行相应的置换时,不会影响原公式的可满足性。公式的这一特性使得我们只需要考虑一条固定顺序的路径,也即每一层只需要考虑一条相应的路径,而不是所有的 $N_k(\varphi)$ 条路径,且不会改变整个公式的可满足性。

定义6(改进的 k -模型的编码) 令 M_k 表示模型 M 的 k -模型, $\varphi \in F$, 则原子命题公式 $[[M^p]]_k^{a,b} = \bigwedge_{i=a}^{a+b-1} \bigwedge_{j=0}^{b-1} T(w_{i,j}, w_{i,j+1})$ 。

该公式表示在该模型中验证公式 φ 需要有 b 条 k -路径,它们的标号从 a 开始到 $a+b-1$ 。

定义7(改进的 ECTL 公式的编码) 令 $k \geq 0$, 给定一个状态 w 和一个公式 $\varphi \in F$, 编码 $[[\varphi, w_{i,j}]]_k^{i, N_k(\varphi)}$ 定义如下:

$$\begin{aligned} & _u[[p, w]]_k^{i, N_k(\varphi)} = p(w) \\ & _u[[\neg p, w]]_k^{i, N_k(\neg\varphi)} = \neg p(w) \\ & _u[[\varphi \wedge \psi, w]]_k^{i, N_k(\varphi \wedge \psi)} = _u[[\varphi, w]]_k^{i, N_k(\varphi)} \wedge _u[[\psi, w]]_k^{i+N_k(\varphi), N_k(\psi)} \\ & _u[[\varphi \vee \psi, w]]_k^{i, N_k(\varphi \vee \psi)} = _u[[\varphi, w]]_k^{i, N_k(\varphi)} \vee _u[[\psi, w]]_k^{i+N_k(\varphi), N_k(\psi)} \\ & _u[[EX\varphi, w]]_k^{i, N_k(EX\varphi)} = (w = w_{i,0}) \wedge _u[[\varphi, w_{i,1}]]_k^{i+1, N_k(\varphi)} \\ & _u[[EG\varphi, w]]_k^{i, N_k(EG\varphi)} = (w = w_{i,0}) \wedge \bigwedge_{j=0}^k _u[[\varphi, w_{i,j}]]_k^{i+1+j \cdot N_k(\varphi), N_k(\varphi)} \\ & _u[[E(\varphi U \psi), w]]_k^{i, N_k(E(\varphi U \psi))} = (w = w_{i,0}) \wedge ((\bigvee_{j=0}^k _u[[\psi, w_{i,j}]]_k^{i+1, N_k(\psi)} \wedge \bigwedge_{t=0}^{j-1} _u[[\varphi, w_{i,t}]]_k^{i+1+t \cdot N_k(\varphi), N_k(\varphi)}) \vee \bigwedge_{t=0}^k _u[[\varphi, w_{i,t}]]_k^{i+1+t \cdot N_k(\varphi), N_k(\varphi)}) \end{aligned}$$

其中 i 为路径的开始编号,当 $j > k$ 时, $[[\varphi, w_{i,j}]]_k^{i, N_k(\varphi)}$ 赋值为真。

定义8 令 $w_{1,0}$ 表示系统的第一个状态变量, $\varphi \in F$, 则 $[[M, \varphi]]_k^{i, N_k(\varphi)} = I(w_{1,0}) \wedge _u[[M^p]]_k^{i, N_k(\varphi)} \wedge _u[[\varphi, w]]_k^{i, N_k(\varphi)}$ 。

定理3 令 $\varphi \in F$, 如果 $M \models \varphi$, 则对所有的 $k \geq 0$, $[[M, \varphi]]_k^{i, N_k(\varphi)}$ 都是可满足的。

由定理2和推论1可知定理3成立。从而,原问题就由查找 $[[M, \neg\psi]]_k^*$ 的不可满足性转换成了查找 $[[M, \neg\psi]]_k^{i, N_k(\neg\psi)}$ 的不可满足性。而后的编码要比前者简单很多,便于 SAT 的求解。

5 实验比较

下面利用文献[10]中的实验,通过与文献[10]中的结果进行比较,来说明一下改进后方法的优势。在该实验中,令 $p_0, \dots, p_{n-2}, q, r$ 为变量,值域为 $\{0, 1\}$, $\oplus$ 为模二的加法运算。假设系统由 n 个进程组成, $A, B, C_i (i = 0, \dots, n-3)$ 的声明如下:

$$A: r = r \oplus 1; p_0 = p_0 \oplus 1$$

$$B: p_{n-2} = p_{n-2} \oplus 1; q = q \oplus 1$$

$$C_i: p_i = p_i \oplus 1; p_{i+1} = p_{i+1} \oplus 1; q = q \oplus 1$$

考虑如下两个性质:

$$\varphi(n) = A((p_0 \vee \dots \vee p_{n-2})Rq)$$

$$\zeta(n) = AXA((p_0 \vee \dots \vee p_{n-2}U(\neg q \vee \neg r))$$

检查系统是否满足以上两个性质,实验的结果如表1、表2所列。

表1 检测性质 $\varphi(n)$ 的实验结果

性质	k	改进前的方法			改进后的方法			SAT
		变量	子句	时间	变量	子句	时间	
$\varphi(3)$	2	50	360	0.0s	42	227	0.0s	No
$\varphi(4)$	2	65	625	0.0s	54	419	0.0s	No
$\varphi(5)$	2	80	962	0.0s	66	674	0.0s	No
$\varphi(6)$	2	95	1371	0.0s	78	987	0.0s	No
$\varphi(7)$	2	110	1852	0.0s	90	1352	0.0s	No

表2 检测性质 $\zeta(n)$ 的实验结果

性质	k	改进前的方法			改进后的方法			SAT
		变量	子句	时间	变量	子句	时间	
$\zeta(3)$	3	143	1079	0.0s	115	867	0.0s	No
$\zeta(4)$	5	286	3075	0.1s	215	1935	0.1s	No
$\zeta(5)$	7	477	6611	0.6s	343	3773	0.4s	No
$\zeta(6)$	9	716	12119	8.5s	499	6986	6.2s	No
$\zeta(7)$	11	1003	20031	191.8s	683	10928	85.3s	No

实验结果表明改进后的编码方法无论是在变量数、子句数,还是在求解问题的时间上,都比原方法有了很大的提高,便于求解更大规模的问题。

结束语 基于 SAT 的限界模型检测方法作为对基于 BDD 的模型检测方法的补充,在查错方面有着明显的优势,而在验证正确性性质方面由于受到完备上界的限制,在一般情况下问题都很难求解。本文介绍了一种验证正确的 ACTL 性质的基于 SAT 的限界模型检测方法,亦即将模型和性质编码成相应公式,再由 SAT 得知公式的可满足性,从而验证性质的正确性。另外,我们在以前工作的基础上,提出了对该方法的改进,主要是对其路径和性质编码上的改进。通过改进,得到一个更加便于 SAT 求解的公式,从而提高求解问题的规

模和效率。关于这方面,我们也通过小的实验给予了验证。

虽然本文给出的验证正确的 ACTL 性质的限界模型检测方法完全基于完备上界的方法相比要好很多,同时也更利于实现,但是,该方法还是不完备的,这也与限界模型检测本身的特点有一定的关系。在今后的工作当中,我们可以结合多种不同的方法,如规约法^[15]、内插法^[16]等来有效地解决这一问题。同样,还可以将该方法应用在文献[17,18]的工作基础上,以利于基于 SAT 的限界模型检测方法在更广泛的领域得到应用,如安全策略模型的验证、无线传感网中路由协议的安全性验证等。

参考文献

- [1] Clarke E M, Emerson A. Design and synthesis of synchronization skeletons using branching-time temporal logic [C]// Lecture Notes in Computer Science, 1981, 131: 52-71
- [2] Clarke E M, Grunberg O, Peled D A. Model Checking [M]. MIT Press, 1999
- [3] Biere A, Cimatti A, Clarke E, et al. Symbolic model checking without BDDs [C]// Proc. of TACAS 99, 1999: 193-207
- [4] Biere A, Cimatti A, Clarke E, et al. Symbolic Model Checking using SAT procedures instead of BDDs [C]// Proc. of DAC 99, 1999: 317-320
- [5] Bryant R E. Graph-based algorithms for Boolean function manipulation [J]. IEEE Transactions on Computers, 1986, 35(12): 1035-1044
- [6] McMillan K L. Symbolic Model Checking [M]. Kluwer Academic Publishers, Boston, 1993
- [7] Coudert O, Madre J C. A unified framework for the formal verification of sequential circuits [C]// Proc. ICCAD 90, 1990: 126-129
- [8] Emerson E A, Clarke E M. Using branching-time temporal logics to synthesize synchronization skeletons [J]. Science of Computer Programming, 1982, 2(3): 241-266
- [9] Penczek W, Wozna B, Zbrzezny A. Bounded Model Checking for the Universal Fragment of CTL [J]. Fundamenta Informaticae, 2002, 51(1/2): 135-156
- [10] Zhang Wen-hui. Verification of ACTL Properties by Bounded Model Checking [C]// EUROCAST 2007. Lecture Notes in Computer Science, 2007, 4739: 556-563
- [11] Xu Yan-yan, Chen Wei, Xu Liang, et al. Evaluation of SAT-based Bounded Model Checking of ACTL Properties [C]// Proceedings of the 1st Joint IEEE/IFIP Symposium on Theoretical Aspects of Software Engineering, 2007: 339-348
- [12] Xu Liang, Chen Wei, Xu Yan-yan, et al. Improved Bounded Model Checking for Universal Fragment of CTL [J]. Journal of Computer Science and Technology, 2009, 24(1): 96-109
- [13] 徐亮. 限界模型检测方法及其应用[D]. 北京: 中国科学院研究生院, 2009
- [14] Pieprzyk J, Qu Cheng-xin. Rotation-Symmetric Functions and Fast Hashing [C]// Proc. of ACISP 98, 1998: 169-180
- [15] de Moura L, Ruess H, Sorea M. Bounded Model Checking and Induction: From Refutation to Verification [C]// Proc. of CAV 2003, 2003: 14-26
- [16] Jhala R, McMillan K L. Interpolation and SAT-Based Model Checking [C]// Proc. of CAV 2003, 2003: 1-13
- [17] Xu Liang. SMT-based Bounded Model Checking for Real-time Systems [C]// Proceedings of the Eighth International Conference on Quality Software, 2008: 120-125
- [18] Xu Liang. Improved SMT-based Bounded Model Checking for Real-time Systems [J]. Journal of Software, 2010, 21(7): 1491-1502
- [19] Emerson E A, Clarke E M. Using branching-time temporal logics to synthesize synchronization skeletons [J]. Science of Computer Programming, 2000, 2: 1354-1360
- [20] Kim B-I, Wy J. Last two fit augmentation to the well-known construction heuristics for one-dimensional bin-packing problem: an empirical study [J]. The International Journal of Advanced Manufacturing Technology, 2010, 50(9-12): 1145-1152
- [21] Stawowy A. Evolutionary based heuristic for bin packing problem [J]. Computer and Industrial Engineering, 2008, 55(2): 465-474
- [22] Bhatia A K, Basu S K. Packing Bins Using Multi-chromosomal Genetic Representation and Better-Fit Heuristic [J]. Computer Science, 2004, 3316: 181-186
- [23] Chan F T S, Au K C, Chan L Y, et al. Using genetic algorithms to solve quality-related bin packing problem [J]. Robotics and Computer-Integrated Manufacturing, 2007, 23(1): 71-72
- [24] 陈国良, 等. 遗传算法及其应用 [M]. 北京: 人民邮电出版社, 2001: 101-161
- [25] Gupta J N D, Ho J C. A new heuristic algorithm for the one-dimensional bin-packing problem [J]. Production planning & control, 1999, 10(6): 598-603

(上接第 69 页)

- [2] Singh A, Gupta A K. Two heuristics for the one-dimensional bin-packing problem [J]. OR Spectrum, 2007, 29(4): 765-781
- [3] Kuk-Hyun, Kim J-H. Quantum-Inspired Evolutionary Algorithm for a Class of Combinatorial Optimization [J]. Evolutionary Computation, 2002, 6(6): 580-593
- [4] Gu Jin-wei, Gu Man-zhan. Anovel competitive co-evolutionary quantum genetic algorithm for stochastic job shop scheduling problem [J]. Computers and Operations Research, 2010, 37(5): 927-937
- [5] Conffman E G, Galambos J G. Bin packing approximation algorithms; combinatorial analysis [M]. the Netherlands: Kluwer Academic Publishers, 1998: 151-207
- [6] 杨俊安, 庄镇全. 量子遗传算法研究现状 [J]. 计算机科学, 2003, 30(11): 13-15
- [7] Galinier P, Hao J K. Hybrid evolutionary algorithms for graph coloring [J]. Journal of Combinatorial Optimization, 1999, 3(4): 381-383
- [8] Han K-H, Kim J-H. Genetic Quantum Algorithm and its Application to Combinatorial Optimization Problem [J]. Evolutionary