ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.20, No.8, August 2009, pp.2005−2014
doi: 10.3724/SP.J.1001.2009.03387

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

有界模型检测的优化*

杨晋吉[1,2+], 苏开乐[3], 骆翔宇[4], 林 瀚[2], 肖茵茵[2]

[1](华南师范大学 计算机学院,广东 广州 510631)

[2](中山大学 信息科学与技术学院,广东 广州 510275)

[3](北京大学 信息科学技术学院,北京 100871)

[4](桂林电子科技大学 计算机与控制学院,广西 桂林 541004)

Optimization of Bounded Model Checking

YANG Jin-Ji[1,2+], SU Kai-Le[3], LUO Xiang-Yu[4], LIN Han[2], XIAO Yin-Yin[2]

[1](School of Computer, South China Normal University, Guangzhou 510631, China)

[2](School of Information Science and Technology, Sun Yat-Sen University, Guangzhou 510275, China)

[3](School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China)

[4](College of Computer and Control, Guilin University of Electronic Technology, Guilin 541004, China)

+ Corresponding author: E-mail: yangjj@scnu.edu.cn

Yang JJ, Su KL, Luo XY, Lin H, Xiao YY. Optimization of bounded model checking. *Journal of Software*, 2009,20(8):2005−2014. http://www.jos.org.cn/1000-9825/3387.htm

Abstract: This paper optimizes the encoding of verifying G(p) and G($p\rightarrow$F(q)) which are two important and frequently used modal operators in optimization of encoding for bounded model checking (BMC). Through analysis of the properties of finite state machine (FSM) and LTL (linear-time temporal logic) when verifying these modal operators, it presents a concise recursive formula, which can efficiently translate BMC instances into SAT (satisfiability) instances. The logical properties of these recursion formulas are verified. The experimental comparison between the optimization of BMC and the other two important methods AA_BMC and Timo_BMC for solving these modal operators in BMC shows that the former is superior to the latter in both the scale of instances and the difficulty to solve the problem. Research of this paper is also beneficial to encoding optimization of verifying other modal operators in BMC.

Key words: model checking; bounded model checking; SAT (satisfiability); modal operator; recursion formula

摘 要: G(p)和G($p\rightarrow$F(q))是有界模型检测(bounded model checking,简称BMC)中的两个重要的常用模态算子.对验证G(p)和G($p\rightarrow$F(q))编码转换公式进行优化.通过分析当验证这些模态算子时FSM(finite state machine)的状态转移和线性时序逻辑(linear-time temporal logic,简称LTL)的语义特征.在现有的编码公式的基础上,给出了简

* Supported by the Outstanding Young Research Fund of China under Grant No.60725207 (国家杰出青年基金); the National Natural Science Foundation of China under Grant Nos.60473004, 60763004 (国家自然科学基金); the Guangdong Provincial Natural Science Foundation of China under Grant No.06023195 (广东省自然科学基金), the Guangdong Provincial Research Foundation of Science and Technology of China under Grant No.2007B010400068 (广东省科技攻关项目)

Received 2007-09-29; Revised 2008-02-01; Accepted 2008-04-15

洁、高效的递推公式,该公式有利于高效编码成 SAT(satisfiability)实例;证明了递推公式和原转换公式的逻辑关系.通过实验比较分析,在生成 SAT 实例规模和易求解方面都优于 BMC 中求解这些模态算子的现有的两种重要方法 AA_BMC 和 Timo_BMC.所给出的方法和思想对于 BMC 中验证其他模态算子时的编码优化也有参考价值.

关键词: 模型检测;有界模型检测;可满足性问题;模态算子;递推公式

中图法分类号: TP301 **文献标识码**: A

模型检测(model checking)[1]是一种以形式化方式构造系统的模拟运行过程,自动检测是否满足某些期望规范的一种新兴技术.随着计算机系统越来越复杂,网络应用越来越普及和重要,通过模型检测技术来验证系统和网络的安全性和正确性的应用也逐渐增多.有界模型检测(bounded model checking,简称 BMC)[2,3]是针对前期的 OBDD(ordered binary decision diagram)技术的模型检测的不足,如状态空间爆炸、需要人工设定逻辑公式中的变量顺序才可有效减少逻辑公式相对应的 BDD(binary decision diagram)结构的存储空间、检测变量少等问题而产生的一种新的模型检测技术.其优势之一是把 BMC 问题编码成 SAT(satisfiability)实例,充分利用 SAT 工具进行求解,较好地弥补了上述的不足,使可以验证的变量数提升一个数量级以上;另一优势是由于采用宽度优先搜索,所获得的反例是长度最短、最简明的反例,有利于设计者理解问题,找出原因.当实验验证边界上界 k 小于 60 时,BMC 要优于传统的模型检测[4].近年来,已出现一些有界模型检测的实际应用[5,6].

有界模型检测的主要过程是:先把要验证的系统或模型构造为有限状态自动机(finite state machine,简称 FSM),通过 FSM 状态间的转移来模拟系统或模型运行;要验证的规范说明用时序逻辑 LTL(linear-time temporal logic)进行说明(如 G(p),F(p));设定边界上界 K;FSM 状态间的转移关系和 LTL 逻辑规范否定的 NNF(negation normal form)公式通过逻辑与构成 BMC 转换公式;把 BMC 转换公式编码成 SAT 实例,通过 SAT 工具求解.若有解,则找到反例;反之,若不可满足,则表明要验证的系统或模型运行到 K 阶段时,是安全的、没有错误的.

近年来,有界模型检测成为一个研究热点,围绕 BMC 的许多问题,每年有专门的 BMC 国际会议.目前,主要在 3 个方面提高 BMC 性能.其一在于对 BMC 的转换公式优化[7−9];其二是编码成 SAT 实例时,对其变量和子句的优化[10,11];其三是针对 BMC 问题映射成 SAT 实例后,针对 SAT 子句的特点,优化 SAT 工具,提高 SAT 求解效率[12].前两种方法都是生成尽可能少的变量和子句,句子结构尽可能地简单,方便求解;第 2 种方法在优化时破坏了 BMC 问题的特征,若再用第 3 种方法中针对 BMC 优化的 SAT 工具,就无法取得较好的效果,并且有界模型检测中仅有 20%~30%的时间用于产生 SAT 实例和求解.第 1 种方法是采用把 BMC 转换公式转换为逻辑等价但结构简单、易于实现的 BMC 公式.这样的公式在以后进行编码时,可直接产生变量和子句都较少且易解的 SAT 实例,从而提高 BMC 的效率.

本文的工作属于第 1 种方法,在 BMC 的 LTL 规范说明中,G(p),G($p\rightarrow$F(q))是协议验证中非常重要的 LTL 的模态描述词,如 G(p)验证协议的保密性,验证智能体的认知算子[5];G($p\rightarrow$F(q))验证协议的认证性等.本文根据 FSM 状态间的转移关系和 G(p),G($p\rightarrow$F(q))公式的语义结构,得到与其 BMC 转换公式逻辑等价和蕴含关系的简洁、高效的递推公式,证明了递推公式和原 BMC 转换公式的逻辑关系,分析了其相应的复杂性变化.最后通过实验验证了本文工作的有效性.

本文第 1 节介绍 BMC 原理及相关公式和定理.第 2 节介绍验证 G(p),G($p\rightarrow$Fq)等 LTL 模态词的转换公式及其优化方法.第 3 节介绍与本文相关的其他工作.第 4 节通过实验与两种已有的优化方法进行比较分析.

1 BMC原理及相关公式和定理

BMC 要验证的规范是用线性时序逻辑 LTL 进行描述的.本节首先介绍 LTL 逻辑的语法和语义,其次介绍 FSM 和 LTL 规范描述构成的 BMC 转换公式及原理,最后介绍已有的一些优化方法.

1.1 LTL的语法及语义

定义 1(LTL 语法). 设原子命题的集合为 A,LTL 公式的语法如下定义:

1. 若$\varphi\in A$,则φ是 LTL 公式.

2. 若φ和ϕ是 LTL 公式,则$\neg\varphi,\varphi\wedge\phi,\varphi\vee\phi,\varphi\rightarrow\phi,\mathrm{G}\varphi,\mathrm{F}\varphi,X\varphi,\varphi\mathrm{U}\phi,\varphi R\phi$等是 LTL 公式.

X,G,F,U,R 分别表示下一个(next),全部(global),最终(eventually),直到…全(until),直到…有(release)等模态词;$\neg,\wedge,\vee,\rightarrow$等符号称为连接词.

因为主要是通过 BMC 找反例,因此 BMC 实际处理的是 LTL 规范的否定形式.LTL 规范的否定形式要表示为 NNF(negative normal form)形式的 LTL 公式(简称 NNF 公式).NNF 公式是不包含$\rightarrow$连接词,且否定词只能出现在原子命题前的 LTL 公式.

若 f 为 LTL 公式,则 $depth(f)$为 f 的深度,即 f 中模态词的嵌套层数.

定义 2. BMC 的克里普克结构(Kripke structure)为一四元组 $M=(S,I,T,\ell)$,其中,S 为 BMC 中 FSM 产生的所有状态集合;$I\subseteq S$,I 为初时状态的集合;$T\subseteq S\times S$,T 为状态间转移关系的集合;$\ell:S\rightarrow P(A)$,为标注状态的函数.

定义 3. BMC 的路径$\pi=(s_1,s_2,s_3,\ldots)$,其中,$s_i\in S,i\in N;\pi(i)=s_i;\pi^i=(s_i,s_{i+1},s_{i+2},\ldots)$.

定义 4 (LTL 语义). 设 M 为一克里普克结构,π为 M 的一路径,f 为 LTL 公式,则$\pi\models f$ 如下定义:

$\pi\models p$ iff $p\in\ell(\pi(0))$

$\pi\models\neg p$ iff $p\notin\ell(\pi(0))$

$\pi\models f\wedge g$ iff $\pi\models f$ 且$\pi\models g$

$\pi\models f\vee g$ iff $\pi\models f$ 或$\pi\models g$

$\pi\models\mathrm{G}f$ iff $\forall i,\pi^i\models f$

$\pi\models\mathrm{F}f$ iff $\exists i,\pi^i\models f$

$\pi\models\mathrm{X}f$ iff $\pi^1\models f$

$\pi\models f\mathrm{U}g$ iff $\exists i,\pi^i\models g$ 且$\forall j,j<i,\pi^j\models f$

$\pi\models f\mathrm{R}g$ iff $\forall i,\pi^i\models g$ 或$\exists j,j<i,\pi^j\models f$

定义 5. 布尔公式是各原子命题只通过$\neg,\wedge,\vee,\rightarrow$等连接词连接,而没有使用模态词的公式.

1.2 BMC转换公式及原理

根据 LTL 语义,若 M 为克里普克结构,f 为 BMC 要验证的 LTL 规范说明否定形式的 NNF 公式,k 为整数边界,则可以创建命题公式 $[[M,f]]_k$ (下文称为 BMC 转换公式),路径$\pi=(s_0,s_1,s_2,\ldots,s_k)$为 $[[M,f]]_k$ 的有限状态序列.BMC 的转换公式 $[[M,f]]_k$ 就是对有限状态序列 $s_0,s_1,s_2,\ldots,s_k$ 的约束,使得 $[[M,f]]_k$ 是可满足的当且仅当 f 在路径π中是有效的(valid).

定义 6(BMC 转换公式). 设 M 为克里普克结构,f 为 BMC 要验证的 LTL 规范说明否定形式的 NNF 公式,k 为整数,则 BMC 转换公式为

$$[[M,f]]_k=[[M]]_k\wedge[[f]]_k \tag{1}$$

其中:$[[M]]_k=I(s_0)\wedge\bigwedge_{i=0}^{k-1}T(s_i,s_{i+1}),[[f]]_k=(\neg L_k\wedge[[f]]_k^0)\vee\bigvee_{l=0}^{k}({}_lL_k\wedge{}_l[[f]]_k^0),{}_lL_k=T(s_k,s_l),L_k=\bigvee_{l=0}^{k}{}_lL_k$,$[[f]]_k^0$和${}_l[[f]]_k^0$的递归定义见表 1 的描述;若 f 为布尔公式,则 $[[f]]_k^0$ 和 ${}_l[[f]]_k^0$ 简写为 f_k^0 和 ${}_lf_k^0$.

$[[M,f]]_k$ 是初时状态为 s_0,路径长度为 k,且满足 f 的所有路径.在实际中,因为边界 k 被提前准确计算出来难度很大[13],所以一般 k 依次取 $k_0,k_0+1,k_0+2,k_0+3,\ldots,k_0+\mathrm{MAX}$ 进行试探;k_0 称为边界下界,大多数情况下取 $k_0=0$;$(k_0+\mathrm{MAX})$为边界上界,一般由用户设定.对 $[[M,f]]_k$ 依次求解,若找到反例或到了设定时间$(k_0+\mathrm{MAX})$,则 k 停止增加,算法结束;否则 k 一直加 1 取下去.当 k 每取一个整数 i 时,把 $[[M,f]]_i$ 转换为 SAT 实例,通过 SAT 工具对此 SAT 实例求解,若有解,则算法结束;否则,k 取 $i+1$ 继续前述算法.

因为 LTL 规范 f 否定形式的 NNF 公式被编码成布尔公式 $[[M,f]]_k$ 的一部分,所以与 BDD 方法[1]不同,公式 f 的规模决定 $[[M,f]]_k$ 编码成 SAT 实例的规模.同时,该 SAT 实例的规模与边界 k 有关,边界 k 越大,其 SAT 实例规模越大.前述的第 1 种方法就是对 $[[M,f]]_k$ 的优化,尽可能找到等价或者是可满足性等价的简化的 $[[M,f]]_k$

公式,从而生成较小的 SAT 实例,提高求解的效率.如果能够确定一个理想的边界 k 的下界值 k_0,就可以减少调用 SAT 工具来求解的次数.

Table 1 Recursive definition of $[[f]]_k^i$ and ${}_l[[f]]_k^i$

表 1 $[[f]]_k^i$ 和 ${}_l[[f]]_k^i$ 的递归定义

f	$[[f]]_k^i$	${}_l[[f]]_k^i$
p	p_i	p_i
$\neg p$	$\neg p_i$	$\neg p_i$
$h\wedge g$	$[[h]]_k^i \wedge [[g]]_k^i$	${}_l[[h]]_k^i \wedge {}_l[[g]]_k^i$
$h\vee g$	$[[h]]_k^i \vee [[g]]_k^i$	${}_l[[h]]_k^i \vee {}_l[[g]]_k^i$
$\mathrm{X}g$	$\begin{cases}[[g]]_k^{i+1}, & \text{if } i<k\\ \bot, & \text{otherwise}\end{cases}$	$\begin{cases}{}_l[[g]]_k^{i+1}, & \text{if } i<k\\ {}_l[[g]]_k^{l}, & \text{otherwise}\end{cases}$
$\mathrm{G}g$	$\bot$	$\bigvee_{j=\min(i,l)}^{k} {}_l[[g]]_k^j$
$\mathrm{F}g$	$\bigvee_{j=i}^{k} [[g]]_k^j$	$\bigvee_{j=\min(i,l)}^{k} {}_l[[g]]_k^j$
$h\mathrm{U}g$	$\bigvee_{j=i}^{k}([[g]]_k^j \wedge \bigwedge_{n=i}^{j-1}[[h]]_k^n)$	$\bigvee_{j=i}^{k}({}_l[[g]]_k^j \wedge \bigwedge_{n=i}^{j-1}{}_l[[h]]_k^n) \vee {}_l[[h]]_k^n \bigvee_{j=l}^{i-1}({}_l[[g]]_k^j \wedge \bigwedge_{n=i}^{k}{}_l[[h]]_k^n \wedge \bigwedge_{n=l}^{j-1}{}_l[[h]]_k^n)$
$h\mathrm{R}g$	$\bigvee_{j=i}^{k}([[h]]_k^j \wedge \bigwedge_{n=i}^{j}[[g]]_k^n)$	$\bigwedge_{j=\min(i,l)}^{k}\vee {}_l[[g]]_k^j \vee_{j=i}^{k}({}_l[[h]]_k^j \wedge \bigwedge_{n=i}^{j}{}_l[[g]]_k^n) \vee \bigvee_{j=l}^{i-1}({}_l[[h]]_k^j \wedge \bigwedge_{n=i}^{k}{}_l[[g]]_k^n \wedge \bigwedge_{n=l}^{j}{}_l[[g]]_k^n)$

1.3 BMC转换公式已有的一些优化

这里主要介绍与本文相关的优化方法,其他一些方法将在后文加以介绍.文献[7]给出主要针对 $[[M,f]]_k$ 公式中 $[[f]]_k$ 的一些特性,经过逻辑推导,得出简化的 $[[f]]_k$ 公式的表示.与本文相关的主要有以下几个定理:

定理 1. 对于任意的 LTL 公式 f,任意的 i,l,k,且 $0\le i\le k$ 和 $0\le l\le k$,$[[f]]_k^i \models {}_l[[f]]_k^i$.

定理 2. 公式 $[[f]]_k$ 逻辑等价于 $[[f]]_k^0 \vee \bigvee_{l=0}^{k}({}_lL_k \wedge {}_l[[f]]_k^0)$.

定理 3. 如果 $depth(f)\le 1$,则 ${}_l[[f]]_k^0$ 与 l 无关,$[[f]]_k = [[f]]_k^0 \vee (L_k \wedge {}_l[[f]]_k^0)$. 特殊情况下,当 $[[f]]_k^0 = {}_l[[f]]_k^0$ 时,$[[f]]_k = [[f]]_k^0$.

上述 3 个定理的证明详见文献[7].

2 验证G(*p*),G(*p*→F*q*)的编码优化

2.1 已有的相关的转换公式

G(p)和 G($p\to$Fq)是 BMC 的 LTL 规范公式中,在协议验证中非常重要、用得最多的模态词.G(p)验证协议的机密性,G($p\to$Fq)验证协议的认证性等;它们用来验证智能体系统的多种属性[5].一般 p,q 为布尔公式,G(p)和 G($p\to$Fq)的否定形式的 NNF 公式为 F($\neg p$)和 F($p\wedge$G($\neg q$)).

$depth$(F($\neg p$))=1,由表 1 可知,$[[\mathrm{F}(\neg p)]]_k^0 = {}_l[[\mathrm{F}(\neg p)]]_k^0$,由公式(1)和定理 3 可推出 G($p$)的 BMC 转换公式为

$$[[M,\mathrm{F}(\neg p)]]_k = [[M]]_k \wedge [[\mathrm{F}(\neg p)]]_k^0 \tag{2}$$

$depth$(F($p\wedge$G($\neg q$)))>1,只能由公式(1)和定理 2 得到 G($p\to$Fq)的 BMC 转换公式:

$$[[M,\mathrm{F}(p\wedge \mathrm{G}(\neg p))]]_k = [[M]]_k \wedge \left([[\mathrm{F}(p\wedge \mathrm{G}(\neg p))]]_k^0 \vee \bigvee_{i=0}^{k}({}_lL_k \wedge {}_l[[\mathrm{F}(p\wedge \mathrm{G}(\neg p))]]_k^0)\right) \tag{3}$$

尽管公式(2)、公式(3)这些经过文献[7]优化过的公式,与原公式相比,求解效率有一定的提高,但这些优化方法并未考虑 $[[M]]_k$. 如果同时考虑 $[[M]]_k$,那么根据这些公式明显的递增特性,可推出逻辑等价或至少是有蕴含关系的简洁递推公式.在目前已知的 BMC 工具(NuSMV,VIS 等)的各较新版本中也均未见讨论此类问题的递推公式的实现.本文定理 4 给出了公式 $[[M,\mathrm{F}(\neg p)]]_k$ 简洁的逻辑等价的递推公式的证明,定理 6 给出了公式 $[[M,\mathrm{F}(p\wedge \mathrm{G}(\neg q))]]_k$ 的蕴含公式的递推公式的证明.通过定理 4 和定理 6 的递推公式编码出的 SAT 实例远远小于公式(2)和公式(3)编码成的 SAT 实例,通过用 SAT 公式求解验证,求解效率明显提高,从而提高 BMC 的效率.

2.2 转换公式优化方法和效率分析

定理 4. 如果 $[[M,\mathrm{F}(p)]]_k$ 是不可满足的,则 $[[M,\mathrm{F}(p)]]_{k+1}$ 逻辑等价于 $[[M]]_{k+1}\wedge(p_{k+1}^{k+1})$.

证明:

$$\begin{aligned}[[M,\mathrm{F}(p)]]_{k+1}&=[[M]]_{k+1}\wedge[[\mathrm{F}(p)]]_{k+1}^{0}\\&=I(s_0)\wedge\bigwedge_{i=0}^{k}T(s_i,s_{i+1})\wedge\bigvee_{i=0}^{k+1}p_{k+1}^{i}\\&=I(s_0)\wedge\bigwedge_{i=0}^{k}T(s_i,s_{i+1})\wedge\left(\bigvee_{i=0}^{k}p_k^i\vee p_{k+1}^{k+1}\right)(\text{由表1可知},p_k^i=p_{k+1}^i)\\&=\left(I(s_0)\wedge\bigwedge_{i=0}^{k-1}T(s_i,s_{i+1})\wedge\bigvee_{i=0}^{k}p_k^i\wedge T(s_k,s_{k+1})\right)\vee\left(I(s_0)\wedge\bigwedge_{i=0}^{k}T(s_i,s_{i+1})\wedge p_{k+1}^{k+1}\right)\\&=([[M,\mathrm{F}(p)]]_k\wedge T(s_k,s_{k+1}))\vee([[M]]_{k+1}\wedge p_{k+1}^{k+1})\end{aligned}$$

由前提可知,$[[M,\mathrm{F}(p)]]_k$ 是不可满足的,则 $[[M,\mathrm{F}(p)]]_k\wedge T(s_k,s_{k+1})$ 是不可满足的.因为 $[[M,\mathrm{F}(p)]]_k\wedge T(s_k,s_{k+1})$ 是不可满足的,若 $[[M,\mathrm{F}(p)]]_{k+1}$ 是可满足的,则可由逻辑或关系的性质得出 $[[M]]_{k+1}\wedge(p_{k+1}^{k+1})$ 也是可满足的;反过来,若 $[[M]]_{k+1}\wedge(p_{k+1}^{k+1})$ 是可满足的,则 $[[M,\mathrm{F}(p)]]_{k+1}$ 是可满足的,这也证明了 $[[M]]_{k+1}\wedge p_{k+1}^{k+1}$ 逻辑等价于 $[[M,\mathrm{F}(p)]]_{k+1}$. □

由定理 4 可知,在 $[[M,\mathrm{F}(p)]]_k$ 编码的 SAT 实例无解的前提下,$[[M]]_{k+1}\wedge p_{k+1}^{k+1}$ 逻辑等价于 $[[M,\mathrm{F}(p)]]_{k+1}$.由此得出,$[[M,\mathrm{F}(p)]]_{k+1}$ 编码的 SAT 实例的解就是 $[[M]]_{k+1}\wedge p_{k+1}^{k+1}$ 编码的 SAT 实例的解;反之也成立.若 $[[M]]_{k+1}\wedge p_{k+1}^{k+1}$ 编码的 SAT 实例无解,则 $[[M,\mathrm{F}(p)]]_{k+1}$ 编码的 SAT 实例也无解.而 $[[M]]_{k+1}\wedge p_{k+1}^{k+1}$ 编码的 SAT 实例的规模小于 $[[M,\mathrm{F}(p)]]_{k+1}$ 编码的 SAT 实例规模,因此,本文用求解 $[[M]]_{k+1}\wedge p_{k+1}^{k+1}$ 的 SAT 实例来代替求解 $[[M,\mathrm{F}(p)]]_{k+1}$ 的 SAT 实例,从而提高了求解效率.

下面比较分析编码成的 SAT 子句的效率提高问题.SAT 实例的子句内部是各原子变量逻辑或的关系,子句间是逻辑与的关系.在 LTL 公式中,G,F,X,U 等模态词和蕴含词→可转换为只包含¬,∧,∨等连接词的公式.BMC 转化公式 ϕ 编码成的 SAT 实例的子句数量为 $n(\phi)$,其递归求解过程见表 2.

Table 2 Recursive definition of clause number of ϕ

表 2 ϕ子句数量的递归定义

ϕ	$n(\phi)$	$\neg n(\phi)$
$\neg\phi_1$	$\neg n(\phi_1)$	$n(\phi_1)$
$\phi_1\wedge\phi_2$	$n(\phi_1)+n(\phi_2)$	$n(\phi_1)n(\phi_2)$
$\phi_1\vee\phi_2$	$n(\phi_1)n(\phi_2)$	$n(\phi_1)+n(\phi_2)$

由表 2 得到公式(2)的子句数量公式为

$$n([[M]]_k)+n([[\mathrm{F}(\neg p)]]_k^0)=n([[M]]_k)+n\left(\bigvee_{i=0}^{k}\neg p_k^i\right)=n([[M]]_k)+\prod_{i=0}^{k}n(\neg p_k^i)\tag{4}$$

由表 2 和定理 4 得到 $[[M,\mathrm{F}(p)]]_k$ 的子句数量公式为

$$n([[M]]_k)+n(\neg p_k^i)\tag{5}$$

公式(4)和公式(5)的 $n([[M]]_k)$部分相同,只比较两公式的第 2 部分.由表 1 可知,$n(\neg p_k^i)=n(\neg p_k^j)$.设 $n(\neg p_k^i)=m$,m 为一个整数常数,公式(4)的第 2 部分的复杂性为 k 指数次,$O(m^k)$.文献[10]通过增加原变量近 4 倍的新变量和重命名(renaming)的方式把原公式转换为可满足性等价的公式,使子句数降为线性 tm 级,其中 $t>k$,其复杂性降为线性复杂性 $O(t)$.公式(5)的第 2 部分为常数阶复杂性 $O(m)$,通过比较可看出,公式(4)和公式(5)的第 2 部分的规模由线性阶复杂性降为常数阶复杂性,且没有增加新变量.

本文对 G(p→Fq)进行优化时,充分考虑在协议验证时 G(p→Fq)的语义.若验证规范是 G(p→Fq),其示意图为图 1.若 p 在状态 s_i 为真,则至少存在 1 个 $j\geq i$,使 q 在状态 s_j 为真.

G(p→Fq)否定形式的 NNF 公式是 F($p\wedge$G($\neg q$)),其示意图为图 2.若 p 在状态 s_i 为真,则从状态 s_i 开始的后续

所有状态中,q 都为假.

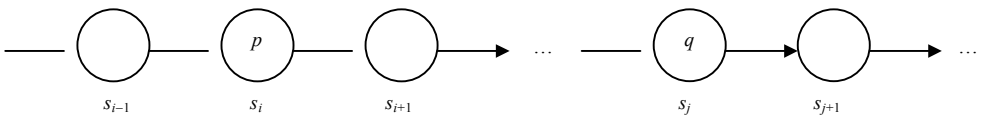

Fig.1 A representation satisfies G(p→Fq)

图 1 G(p→Fq)为真

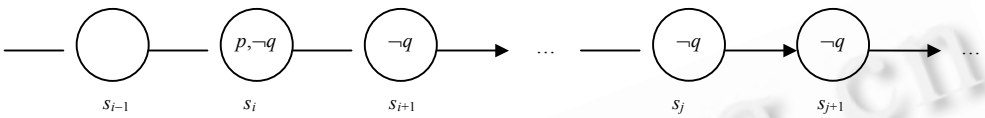

Fig.2 A representation satisfies F(p∧G(¬q))

图 2 F(p∧G(¬q))为真

$[[M,\mathrm{F}(p\wedge\mathrm{G}(\neg q))]]_k=[[M]]_k\wedge\left([[\mathrm{F}(p\wedge\mathrm{G}(\neg q))]]_k^0\vee\bigvee_{l=0}^{k}({}_lL_k\wedge{}_l[[F(p\wedge G(\neg q))]]_k^0)\right)$. 由表 1 可知, $[[\mathrm{G}(\neg q)]]_k^0=\bot$, 所以 $[[\mathrm{F}(p\wedge\mathrm{G}(\neg q))]]_k^0=\bigvee_{i=0}^{k}(p_i\wedge[[\mathrm{G}(\neg q)]]_i^0)=\bot$, 由此可以把公式(3)简化为 $[[M]]_k\wedge\bigvee_{l=0}^{k}({}_lL_k\wedge{}_l[[\mathrm{F}(p\wedge\mathrm{G}(\neg q))]]_k^0)$, 这其实还是一个非常复杂的公式,因为 ${}_lL_k$ 可能产生环,很难直接得到递推公式.根据图 1、图 2 的思想,可以得到公式(3)的一个蕴含公式:

$$[[M]]_k\wedge{}_l[[\mathrm{F}(p\wedge\mathrm{G}'(\neg q))]]_k^0 \tag{6}$$

其中, ${}_l[[\mathrm{F}(p\wedge\mathrm{G}'(\neg q))]]_k^i={}_l[[p]]_k^i\wedge\bigwedge_{j=i}^{k}{}_l[[\neg q]]_k^j$. 本蕴含式与图 1、图 2 语义示意图相一致,且较易得到递推公式.

定理 5. 如果 $[[M]]_k\wedge{}_l[[\mathrm{F}(p\wedge\mathrm{G}'(\neg q))]]_k^0$ 不可满足,则 $[[M]]_k\wedge\left([[\mathrm{F}(p\wedge\mathrm{G}(\neg q))]]_k^0\vee\bigvee_{l=0}^{k}({}_lL_k\wedge{}_l[[\mathrm{F}(p\wedge\mathrm{G}(\neg q))]]_k^0)\right)$不可满足.

证明:因为 $[[M]]_k\wedge\left([[\mathrm{F}(p\wedge\mathrm{G}(\neg q))]]_k^0\vee\bigvee_{l=0}^{k}({}_lL_k\wedge{}_l[[\mathrm{F}(p\wedge\mathrm{G}(\neg q))]]_k^0)\right)$ 蕴含公式 $[[M]]_k\wedge{}_l[[\mathrm{F}(p\wedge\mathrm{G}'(\neg q))]]_k^0$, 根据蕴含性质,当 $[[M]]_k\wedge{}_l[[\mathrm{F}(p\wedge\mathrm{G}'(\neg q))]]_k^0$ 不可满足时, $[[M]]_k\wedge\left([[\mathrm{F}(p\wedge\mathrm{G}(\neg q))]]_k^0\vee\bigvee_{l=0}^{k}({}_lL_k\wedge{}_l[[\mathrm{F}(p\wedge\mathrm{G}(\neg q))]]_k^0)\right)$不可满足,证毕. □

根据定理 5 可知,当边界值为 k 时,若公式(6)不可满足时,有公式(3)也不可满足,则可验证 BMC 问题在边界 k 内是正确的;若公式(6)可满足,则尽管公式(3)不一定满足,但可以得到一个较为理想的 BMC 下界 k_0,从下界 k_0 开始再用公式(3)求解,也一定比当下界 $k_0=0$ 时就一直用公式(3)求解的效率要高.

定理 6. 如果 $[[M]]_k\wedge{}_l[[\mathrm{F}(p\wedge\mathrm{G}'(\neg q))]]_k^0$ 是不可满足的,则 $[[M]]_{k+1}\wedge{}_l[[\mathrm{F}(p\wedge\mathrm{G}'(\neg q))]]_{k+1}^0$ 逻辑等价于 $[[M]]_{k+1}\wedge(p_{k+1}^{k+1}\wedge\neg q_{k+1}^{k+1})$.

证明:

$$[[M]]_{k+1}\wedge{}_l[[\mathrm{F}(p\wedge\mathrm{G}'(\neg q))]]_{k+1}^0=[[M]]_{k+1}\wedge\bigvee_{i=0}^{k+1}\left(p_{k+1}^i\wedge\bigwedge_{j=i}^{k+1}\neg q_{k+1}^j\right)$$

$$=[[M]]_{k+1}\wedge\left(\bigvee_{i=0}^{k}\left(p_{k+1}^i\wedge\bigwedge_{j=i}^{k+1}\neg q_{k+1}^j\right)\vee(p_{k+1}^{k+1}\wedge\neg q_{k+1}^{k+1})\right)$$

$$=[[M]]_{k+1}\wedge\left(\bigvee_{i=0}^{k}\left(p_k^i\wedge\left(\bigwedge_{j=i}^{k}\neg q_{k+1}^j\wedge\neg q_{k+1}^{k+1}\right)\right)\vee(p_{k+1}^{k+1}\wedge\neg q_{k+1}^{k+1})\right)$$

$$=[[M]]_{k+1}\wedge\left(\left(\bigvee_{i=0}^{k}\left(p_k^i\wedge\bigwedge_{j=i}^{k}\neg q_k^j\right)\wedge\neg q_{k+1}^{k+1}\right)\vee(p_{k+1}^{k+1}\wedge\neg q_{k+1}^{k+1})\right)$$

$$=\left([[M]]_k\wedge\left(\bigvee_{i=0}^{k}\left(p_k^i\wedge\bigwedge_{j=i}^{k}\neg q_k^j\right)\wedge T(s_k,s_{k+1})\wedge\neg q_{k+1}^{k+1}\right)\right)\vee([[M]]_{k+1}\wedge(p_{k+1}^{k+1}\wedge\neg q_{k+1}^{k+1}))$$

$$=([[M]]_k\wedge_l[[\mathrm{F}(p\wedge\mathrm{G}'(\neg q))]]_k^0\wedge T(s_k,s_{k+1})\wedge\neg q_{k+1}^{k+1})\vee([[M]]_{k+1}\wedge(q_{k+1}^{k+1}\wedge\neg q_{k+1}^{k+1}))$$

由前提可知,$[[M]]_k\wedge_l[[\mathrm{F}(p\wedge\mathrm{G}'(\neg q))]]_k^0$ 不可满足,则 $[[M]]_k\wedge_l[[\mathrm{F}(p\wedge\mathrm{G}'(\neg q))]]_k^0\wedge T(s_k,s_{k+1})\wedge\neg q_{k+1}^{k+1}$ 也不可满足;若 $[[M]]_{k+1}\wedge_l[[\mathrm{F}(p\wedge\mathrm{G}'(\neg q))]]_{k+1}^0$ 满足,则 $[[M]]_{k+1}\wedge(p_{k+1}^{k+1}\wedge\neg q_{k+1}^{k+1})$ 必满足,反之,若 $[[M]]_{k+1}\wedge(p_{k+1}^{k+1}\wedge\neg q_{k+1}^{k+1})$ 满足,则 $[[M]]_{k+1}\wedge_l[[\mathrm{F}(p\wedge\mathrm{G}'(\neg q))]]_{k+1}^0$ 必满足.由此得证. □

定理 7. 如果当 k 依次取 0,1,2,…,m 时,$[[M]]_k\wedge(p_k^k\wedge\neg q_k^k)$不可满足,当 k 取 m+1 时,$[[M]]_k\wedge(p_k^k\wedge\neg q_k^k)$ 可满足,则当 k 依次取 m+1,m+2,…,$k_{\max}$ 时,$[[M]]_k\wedge\bigvee_{l=0}^{k}({}_lL_k\wedge{}_l[[\mathrm{F}(p\wedge\mathrm{G}(\neg q))]]_k^0)$ 逻辑等价于 $[[M]]_k\wedge\bigvee_{l=0}^{k}\left({}_lL_k\wedge\bigvee_{j=m+1}^{k}(p_k^j\wedge{}_l[[\mathrm{G}(\neg q)]]_k^j)\right)$.

证明:

$$[[M]]_k\wedge\bigvee_{l=0}^{k}({}_lL_k\wedge{}_l[[\mathrm{F}(p\wedge\mathrm{G}(\neg q))]]_k^0)=[[M]]_k\wedge\bigvee_{l=0}^{k}\left({}_lL_k\wedge\bigvee_{j=0}^{k}{}_l[[p\wedge\mathrm{G}(\neg q))]]_k^j\right)$$

$$=[[M]]_k\wedge\bigvee_{l=0}^{k}\left({}_lL_k\wedge\bigvee_{j=0}^{k}({}_lp_k^j\wedge{}_l[[G(\neg q)]]_k^j)\right)(\text{由表1可知},p_k^i={}_lp_k^i)$$

$$=[[M]]_k\wedge\bigvee_{l=0}^{k}\left({}_lL_k\wedge\left(\bigvee_{j=0}^{m}(p_k^j\wedge{}_l[[G(\neg q)]]_k^j)\vee\bigvee_{j=m+1}^{k}(p_k^j\wedge{}_l[[G(\neg q)]]_k^j)\right)\right)$$

(由表1中$p_k^i=p_{k+1}^i$和Gp的性质以及本定理的前提可知,$[[M]]_k\wedge\bigvee_{j=0}^{m}(p_k^j\wedge{}_l[[G(\neg q)]]_k^j)=\bot$)

$$=[[M]]_k\wedge\bigvee_{l=0}^{k}\left({}_lL_k\wedge\bigvee_{j=m+1}^{k}(p_k^j\wedge{}_l[[\mathrm{G}(\neg q)]]_k^j)\right)$$ □

对 $\mathrm{G}(p\rightarrow\mathrm{F}q)$否定形式的 NNF 公式 $\mathrm{F}(p\wedge\mathrm{G}(\neg q))$求解时,用定理 6 和定理 7 的组合来代替转换公式 $[[M]]_k\wedge\bigvee_{l=0}^{k}({}_lL_k\wedge{}_l[[\mathrm{F}(p\wedge\mathrm{G}(\neg q))]]_k^0)$(文献[7]对公式(3)优化过的逻辑等价式).由于公式(6)符合 $\mathrm{F}(p\wedge\mathrm{G}(\neg q))$的语义且是公式(3)的蕴含式;一般情况下,当 k 取 0,1,2,…,n($n\le$MAX)时,公式(3)是无解的,这时一定有当 k 取 0,1,2,…,m($m\le n$)时,公式(6)是无解的,根据定理 5,我们先用公式(6)的逻辑等价式 $[[M]]_k\wedge(p_k^k\wedge\neg q_k^k)$ 来确定公式(6)无解时的 m 值.基于前述相似的理由,当 $[[M]]_k\wedge_l[[\mathrm{F}(p\wedge\mathrm{G}'(\neg q))]]_k^0$ 不可满足时,用求解公式 $[[M]]_{k+1}\wedge(p_{k+1}^{k+1}\wedge\neg q_{k+1}^{k+1})$ 编码的 SAT 实例来代替求解公式 $[[M]]_{k+1}\wedge_l[[\mathrm{F}(p\wedge\mathrm{G}'(\neg q))]]_{k+1}^0$ 编码的 SAT 实例,k 由 0 开始,依次加 1 进行循环迭代;当 k 取 m+1 时,$[[M]]_k\wedge(p_k^k\wedge\neg q_k^k)$ 编码的 SAT 实例有解,则再根据定理 7,使 k 从 m+1 开始,用公式 $[[M]]_k\wedge\bigvee_{l=0}^{k}({}_lL_k\wedge{}_l[[\mathrm{F}(p\wedge\mathrm{G}(\neg q))]]_k^0)$ 的逻辑等价式 $[[M]]_k\wedge\bigvee_{l=0}^{k}\left({}_lL_k\wedge\bigvee_{j=m+1}^{k}(p_k^j\wedge{}_l[[\mathrm{G}(\neg q)]]_k^j)\right)$ 来进行编码求解,若公式 $[[M]]_k\wedge\bigvee_{l=0}^{k}\left({}_lL_k\wedge\bigvee_{j=m+1}^{k}(p_k^j\wedge{}_l[[\mathrm{G}(\neg q)]]_k^j)\right)$ 编码的 SAT 实例有解,此解就是公式 $[[M,\mathrm{F}(p\wedge\mathrm{G}(\neg q))]]_k$ 的 SAT 实例解.

下面我们再来分析编码成 SAT 实例子句的效率.参照表 2,公式(3)产生的子句数量公式为

$$n([[M]]_k)+\left(\prod_{l=0}^{k}(n({}_lL_k)+n({}_l[[\mathrm{F}(p\wedge\mathrm{G}(\neg q))]]_k^0))\right)\tag{7}$$

定理 6 的公式 $[[M]]_k \wedge_l [[\mathrm{F}(p \wedge \mathrm{G}'(\neg q))]]_k^0$ 产生的子句数量公式为

$$n([[M]]_k)+n(p_k^k)+n(\neg q_k^k) \tag{8}$$

公式(7)和公式(8)的第 1 部分都为 $n([[M]]_k)$,同理,只比较第 2 部分.由表 2,设 $n(p_k^i)=m$,m 为一个整数常数,$0\le i\le k$;$n(\neg q_k^j)=r$,r 为一个整数常数,$0\le j\le k$.公式(7)的第 2 部分非常复杂,其复杂性不低于指数级 $O((m+r)^k)$,同样,通过文献[10]增加变量和重命名的方式把原公式转换为可满足性等价的公式,使子句数降为线性 $t(am+br)$级,其中 a,b 各为一个整数常量,$t>k$,使其复杂性降为线性复杂性 $O(t)$;而公式(8)的第 2 部分的复杂性为常数阶 $O(m+r)$,且未增加新变量.

3 相关的工作

本文的工作是文献[7]的工作的继续.文献[7]分析了原始的 BMC 的 SAT 编码问题,主要针对 BMC 转换公式的 $[[f]]_k$ 公式进行了多种优化工作,也提出一些高效的存储结构等.这些工作在 NuSMV2.1.2 中实现,且在后续版本中不断完善.Timo 等人[9]利用 lasso-shaped Kripke Structure 的特点,把求解 CTL(computation tree logic)公式用到的固定点(fixpoint)方法运用到 BMC 的转换公式中.文献[9]中的很多实验结果表明,此方法比文献[7,8]中的方法更为有效.Frisch 等人[8]采用固定点的方法,分别把 BMC 转化公式用 SNF 和固定点的标准形式表示.固定点的标准形式充分利用了标准形式的特点,使用类似于 Tableau_style 的方法.实验结果表明,SAT 编码规模小于文献[7],总体上也优于 SNF 的表示.

Sheridan 等人[10,11]对 BMC 编码成的 SAT 实例进行优化,为了降低编码成 SAT 实例的复杂性,编码时通过增加新变量和重命名,这样编码成的 SAT 实例有很大的冗余,因此,基于 Boy de la Rour 思想,提出了一种紧缩的优化编码的方法,使得产生的 SAT 实例的规模大幅度降低,但 BMC 的一些特征在所产生的 SAT 实例中消失了.

Strichman 等人[12]利用 BMC 产生的 SAT 实例的一些特性,如变量的次序、冲突子句的处理等,对 SAT 工具进行优化,Gupta 等人[14]利用 BDD 模型检测运行使得 SAT 工具自动获取 SAT 实例的一些特征,也取得了较好的效率.

4 实验结果比较

本文是在 NuSMV2.3.1 模型检测工具中完成相应的优化工作.本文的工作主要与 NuSMV2.3.1 原方法和另一种较好的 BMC 方法[9]进行比较.其中,NuSMV2.3.1 原方法是在文献[2]的基础上,经过文献[7]优化的方法,本文简称为 AA_BMC 方法;另一种较好的方法是由 Timo 等人提出,且在 NuSMV2.4.0 中实现的,本文简称为 Timo_BMC 方法.因为 Timo_BMC 方法总体上优于另外几种优化的方法[7,8],因此本文没有与其他方法进行比较.本文给出的对 G(f)优化方法称为 G_BMC,对 G($p\rightarrow$F(q))优化方法称为 G_F_BMC.

实验的环境如下:硬件:Intel Core 4300 处理器,2G 内存;软件:Windows XP 专业版,WinGW32 平台编译 NuSMV.

实验用的模型主要取自文献[15],和自编的 NSPK 协议模型.BMC 具有这样的特点:边界 k 越小,验证时间越短;k 越大,验证时间越长.若模型的边界 k 较小,则算法的验证时间比其他算法要短;当 k 变大时,算法的验证时间仍然比其他算法要短.若模型较小,则在同样的边界 k 下,算法的验证时间比其他算法要短;若模型较大,则在同样的边界 k 下,算法的验证时间仍然比其他算法要短.基于这些特点,为了缩短实验的时间,本文采用的模型较小;若比较费时,就缩短边界上界 K.模型中都包含有 G(p)或 G($p\rightarrow$F(q))验证规范的例子.本实验所用的模型大部分是用 CTL 规范说明,要改为等价的 LTL 规范说明,若在 K 边界内有解,则要对 LTL 规范作适当的修改,使其不可满足,这样更便于验证算法的有效性.NSPK 协议模型当边界 k=14 时可满足属于例外.

每个模型测试的项目包括运行完边界 k 后,其编码生成的 SAT 实例的变量数、子句数、生成 SAT 实例和求解 SAT 实例的总的时间(单位:s).求解用的 SAT 工具为 NuSMV 内置的 SAT 求解工具.

由表 3、表 4 的实验数据可知,在验证模态词 G(p)和 G($p\rightarrow$F(q))时,Timo_BMC 算法与 AA_BMC 算法相比,

并没有明显的优势.本文提出的G_BMC和G_F_BMC在验证这类模态词时比前两类算法优势明显,通过观察所产生的 SAT 实例的变量数和子句数,其效率提高的幅度因不同的模型而有差异,如 BRP 模型提高幅度稍小,SEMAPHORE 模型则提高幅度很大.如前所述,提高的幅度与验证规范公式复杂性密切相关,模态词 G(p)和G($p\rightarrow$F(q))的表达式越复杂,效率提高得越明显.

Table 3 Comparison of encoding G(p)

表 3 各 G(p)编码比较

Model	k	AA_BMC			Timo_BMC			G_BMC		
		Variables	Clauses	Time	Variables	Clauses	Time	Variables	Clauses	Time
DME2	10	6 384	14 727	5	7 053	16 347	7	6 325	14 547	4
	20	12 519	29 277	22	13 727	32 517	29	12 305	28 917	18
BRP	10	4 597	13 294	<1	5 279	15 677	<1	4 543	13 264	<1
	35	16 072	46 169	16	18 479	54 577	19	15 443	46 064	15
	50	23 257	65 894	39	26 699	77 917	51	21 983	65 744	37
Semaphore	10	1 248	1 959	2	1 349	2 329	2	534	1 404	<1
	14	2 188	2 759	315	2 329	3 277	215	742	1 956	7

Table 4 Comparison of encoding G($p\rightarrow$F(q))

表 4 各 G($p\rightarrow$F(q))编码比较

Model	k	AA_BMC			Timo_BMC			G_F_BMC		
		Variables	Clauses	Time	Variables	Clauses	Time	Variables	Clauses	Time
Mutex	30	23 033	9 817	3	23 181	9 972	3	1 457	4 161	<1
	50	87 387	20 867	19	87 631	21 122	17	2 417	6 921	3
DME4	10	25 099	37 263	3	54 964	58 493	5	12 805	29 217	3
	16	51 505	59 862	10	123 652	93 830	25	19 945	46 551	7
NSPK	10	41 443	96 840	12	42 064	98 700	12	25 423	86 231	9
	14 satisfied	66 715	135 714	33	67 584	138 318	32	35 255	120 531	24

G_BMC 算法是基于定理 4 的,其递推公式与公式(2)是逻辑等价的,因此,当 G_BMC 算法不可满足时,模态词 G(p)在 k 阶段成立;当 G_BMC 算法有可满足解时,此解就是 G(p)不成立的反例.G_F_BMC 算法是基于定理 6 和定理 7 的,其中定理 6 的前提条件公式(6)与公式 $[[M,\mathrm{F}(p\wedge\mathrm{G}(\neg q))]]_k$ 不等价,是蕴含关系(公式 $[[M,\mathrm{F}(p\wedge\mathrm{G}(\neg q))]]_k$ 蕴含公式(6)),这在验证模型的正确性上是一致的;而当定理 7 使公式(6)可满足时,则可获得正确的反例.当 G_F_BMC 算法不可满足时,公式 $[[M,\mathrm{F}(p\wedge\mathrm{G}(\neg q))]]_k$ 一定不可满足,模态词 G($p\rightarrow$F(q))在 k 阶段一定成立;而当 G_F_BMC 算法中的定理 7 使得公式(6)可满足时,则可正确求出公式 $[[M,\mathrm{F}(p\wedge\mathrm{G}(\neg q))]]_k$ 的 SAT 解,即 G($p\rightarrow$F(q))性质不成立时的反例解.由表 4 可以看出,G_F_BMC 算法比另两种方法的效率提高得较明显.

5 结束语

本文对有界模型检测中比较重要且很常用的 LTL 的模态词 G(p)和 G($p\rightarrow$F(q))编码进行优化,结合 FSM 的状态转移关系和这些模态词的语义,推出简洁且高效的递推式,证明了递推式和原公式的等价或蕴含关系,分析了优化后的方法产生的 SAT 实例的复杂性,最后通过实验,用具体的模型验证了本方法的有效性.根据本文优化所采用的递推思想,应该可以对部分其他 LTL 模态词编码进行优化,这是下一步的工作.

致谢 衷心感谢论文评审专家中肯而有益的评审意见和相关编辑严谨、热情的工作;感谢 Biere,Cimatti 等人提供的 NuSMV 系统;感谢田艳玲副教授和陈清亮同学的帮助和支持.

References:

[1] Clarke EM, Grumberg O, Peled DA. Model Checking. Cambridge: The MIT Press, 1999. 61−87.

[2] Biere A, Cimatti A, Clarke EM, Zhu Y. Symbolic model checking without BDDs. In: Cleaveland R, ed. Proc. of the 5th Int'l Conf. on Tools and Algorithms for the Constructions and Analysis of Systems (TACAS'99). LNCS 1579, Berlin: Springer-Verlag, 1999. 193−207.

[3] Cimatti A, Clarke EM, Giunchiglia E, Giunchiglia F, Pistore M, Roveri M, Sebastiani R, Tacchella A. NuSMV 2: An opensource tool for symbolic model checking. In: Brinksma E, Larsen KG, eds. Proc. of the 14th Int'l Conf. on Computer Aided Verification

(CAV 2002). LNCS 2404, Berlin: Springer-Verlag, 2002. 359−364.

[4] Amla N, Kurshan R, McMillan K, Medel R. Experimental analysis of different techniques for bounded model checking. In: Hubert G, Hatcliff J, eds. Proc. of the 9th Int'l Conf. on Tools and Algorithms for the Constructions and Analysis of Systems (TACAS 2003). LNCS 2619, Berlin: Springer-Verlag, 2003. 34−48.

[5] Luo XY, Su KL, Yang JJ. Bounded model checking for temporal epistemic logic in synchronous multi-agent systems. Journal of Software, 2006,17(12):2485−2498 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/17/2485.htm

[6] Daniel G, Ulrich K, Rolf D. HW/SW co-verification of embedded systems using bounded model checking. In: Qu G, Ismail YI, Vijaykrishnan N, Zhou H, eds. Proc. of the 16th ACM Great Lakes Symp. on VLSI 2006. Cairns: ACM Press, 2006. 43−48.

[7] Cimatti A, Pistore M, Roveri M, Sebastiani R. Improving the encoding of LTL model checking into SAT. In: Agostino C, ed. Proc. of the 3rd Int'l Workshop on Verification, Model Checking, and Abstract Interpretation (VMCAI 2002). LNCS 2294, Berlin: Springer-Verlag, 2002. 196−207.

[8] Frisch A, Sheridan D, Walsh T. A fixpoint encoding for bounded model checking. In: Aagaard MD, OLeary JW, eds. Proc. of the 4th Int'l Conf. on Formal Methods in Computer-Aided Design (FMCAD 2002). LNCS 2517, Berlin: Springer-Verlag, 2002. 238−255.

[9] Latvala T, Biere A, Heljanko K, Junttila T. Simple bounded LTL model checking. In: Hu AJ, Martin AK, eds. Proc. of the 5th Int'l Conf. on Formal Methods in Computer-Aided Design (FMCAD 2004). LNCS 3312, Berlin: Springer-Verlag, 2004. 186−200.

[10] Jackson P, Sheridan D. Clause form conversions for boolean circuits. In: Hoos HH, Mitchell DG, eds. Proc. of the 7th Int'l Conf. on Theory and Applications of Satisfiability Testing (SAT 2004). LNCS 3542, Berlin: Springer-Verlag, 2004. 183−198.

[11] Jackson P, Sheridan D. The optimality of a fast CNF conversion and its use with SAT. In: Hoos HH, Mitchell DG, eds. Proc. of the 7th Int'l Conf. on Theory and Applications of Satisfiability Testing (SAT 2004). LNCS 3709, Berlin: Springer-Verlag, 2005. 827−831.

[12] Strichman O. Accelerating bounded model checking of safety properties. Formal Methods in System Design, 2004,24(1):5−24.

[13] Clarke EM, Kroenig D, Oukanine J, Strichman O. Completeness and complexity of bounded model checking. In: Steffen B, Levi G, eds. Proc. of the 5th Int'l Conf. on Verification, Model Checking, and Abstract Interpretation (VMCAI 2004). LNCS 2937, Berlin: Springer-Verlag, 2004. 85−96.

[14] Gupta A, Ganai M, Wang C, Yang Z, Ashar P. Learning from BDDs in SAT-based bounded model checking. In: Proc. of the 40th Conf. on Design Automation (DAC 2003). Montreal: IEEE Computer Society Press, 2003. 824−829.

[15] http://nusmv.irst.itc.it/examples/examples.html

附中文参考文献:

[5] 骆翔宇,苏开乐.杨晋吉.有界模型检测同步多智体系统的时态认知逻辑.软件学报,2006,17(12):2485−2498. http://www.jos.org.cn/1000-9825/17/2485.htm

杨晋吉(1968－),男,山西太原人,博士,副教授,主要研究领域为模型检测,协议验证.

苏开乐(1964－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为模型检测,智能体系统.

骆翔宇(1974－),男,博士,副教授,主要研究领域为模型检测,多智体系统.

林瀚(1979－),男,博士,CCF 会员,主要研究领域为人工智能逻辑及其算法.

肖茵茵(1983－),女,博士生,CCF 学生会员,主要研究领域为模型检测,安全协议验证.