基于 Craig 插值的线性混成系统符号化模型检测

陈祖希[1],徐中伟[1],霍伟伟[2],喻 钢[2]
(1.同济大学电子与信息工程学院,上海 201804;2.上海大学悉尼工商学院,上海 201800)

摘 要: 最强后件的计算是模型检测算法的核心.本文使用一阶逻辑可满足性模线性算术理论给出线性混成自动机的有界模型检测表示公式,利用一阶逻辑公式不可满足情况下的插值存在性定理,对线性混成自动机的有界模型检测公式进行指定的划分,使用支持线性算术插值计算的可满足性模理论后端证明引擎的线性时间复杂度的消解反证技术获得这两部分公式间的插值公式,按一阶逻辑 Craig 插值的性质,所得到的插值公式就是模型检测过程中最强后件公式的上近似表示.有效地避免了使用逻辑编码方案实现线性混成自动机模型检测过程中需要双指数时间复杂度的量词消去操作求取最强后件公式,也不需像有界模型检测按步长展开变迁公式进行可满足性判定.最后本文在此最强后件计算的基础上,以有界模型检测技术作为反例确认方法,实现了一种无假反例的混成系统近似可达集计算算法.实验证明该算法与目前已经得到广泛工业应用的有界模型检测算法相比具有更优的性能.

关键词: Craig 插值; 可满足性模理论; 线性混成自动机; 符号模型检验; 混成系统
中图分类号: TP311.1 **文献标识码:** A **文章编号:** 0372-2112 (2014)07-1338-09
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2014.07.014

Symbolic Model Checking for Linear Hybrid Systems Base on Craig Interpolation

CHEN Zu-xi[1], XU Zhong-wei[1], HUO Wei-wei[2], YU Gang[2]
(1. *School of Electronics and Information, Tongji University, Shanghai* 201804, *China*; 2. *SHU-UTS Business School, Shanghai University, Shanghai* 201800, *China*)

Abstract: The key to model checking algorithm is the computation of strongest post condition. This article encodes the bounded model checking problem for linear hybrid automata as formula of SAT Modulo theories for linear arithmetic. We divided the formula into two specific parts to obtain the interpolation with a linear time complexity. According to the properties of Craig interpolation theorem for first order logic, the interpolation as an over-approximation strongest post condition and can replace the original strongest post condition used in symbolic model checking for hybrid automata with exponential time complexity. This method does not require to the transition relation is fully expanded the same as bounded model checking to check satisfiability, also. We implement the hybrid systems model checking algorithm without false counter-example using the over-approximation strongest post condition operator together with bounded model checking algorithm. Experiments show that our approach can be more efficient than bounded model checking for hybrid systems.

Key words: Craig interpolation; satisfiability modulo theories; linear hybrid automata; model checking; hybrid systems

1 引言

混成系统(Hybrid Systems)广泛存在于各种安全性要求极高的领域,如航空航天、轨道交通、核电站等.在这样的系统中即使是一个极小的系统错误都可能引起灾难性的后果.因此,混成系统设计的正确性显得极为重要.然而,混成系统是由包含软、硬件和通信网络的计算实体与现实的物理环境之间构成的离散与连续行为交互的系统.这种连续和离散行为,以及行为之间的交互使得混成系统表现出了比一般系统更高的复杂性,混成系统的设计和验证面临巨大挑战[1~3].

目前,为保证混成系统的正确性广泛采用的是仿真和测试技术,不幸的是要进行相对充分的仿真和测试会带来设计成本过高和设计周期过长等诸多问题,最为关键的是表示混成系统连续空间的实数系是稠密的,测试和仿真对这样的混成系统而言只能是一种查找系统错

收稿日期:2013-10-18;修回日期:2014-04-27;责任编辑:孙瑶
基金项目:国家自然科学基金(No.60674004,No.61075002,No.71302048);十二五国家科技支撑计划(No.2011BAG01B03);国家 863 高技术研究发展计划(No.2012AA112801);铁道部重点研究发展计划(No.2012AA112801)

误的技术,并不能满足混成系统高安全性的要求.与仿真和测试技术不同,基于形式化技术的混成系统模型检测方法提供了一种覆盖混成系统全部可能输入和所有执行路径的方法,能够有效地证明混成系统是否设计正确,是更适合于类似安全苛求系统这样对安全性要求极高的复杂系统设计正确的保障技术.

混成自动机是一种能够准确刻画混成系统行为特征的形式化模型,混成系统的模型检测算法主要建立在混成自动机模型之上.因为混成自动机模型检测需要验证的性质都通过可达集计算实现,所以混成自动机的可达集计算是混成自动机模型检测方法研究的焦点.目前,混成自动机的可达集计算都是采用前向计算的方法,即以初始状态集作为当前可达状态集出发,迭代计算从当前状态集经过一步变迁关系可以到达的状态集(即当前状态集的最强后件)实现.所以,怎样表示无穷状态空间上的状态集和实现这种表示上高效的最强后件计算就成了混成系统模型检测算法的核心.

因为非线性微分方程在数学上不一定可解,并且非线性系统都可以近似为线性系统.所以,目前的混成系统模型检测方法的研究都主要集中在线性混成系统.对于非线性混成系统模型检测,通常都是采用线性化方法将非线性行为近似为相对简单的线性行为,进而在线性混成系统上进行模型检测.然而,即使线性混成自动机比非线性混成自动机行为简单,线性混成自动机也依然不满足有限互模拟属性的,其可达集计算是不可判定的.现有的针对一般线性混成自动机的模型检测工具,如 HyTech, CHAEON 等[1,4,5],大部分都是使用凸多面体来近似可达状态集.但是,这种方法除了在可达集迭代过程中为保证可达状态的凸性需要使用多个凸多面体凸包操作外,为了保证算法的可终止性还需使用多面体抽象解释的外推操作,这些操作都可能导致计算出的可达状态中包含假的反例.UPPAAL 等工具针对线性混成系统中的特殊子类的实时系统采用了基于 DBM(Difference Bound Matrices)的符号化可达性计算方法,在 DBM 上构造的可达集计算相对于多面体有更高的效率.但是,为了保证算法能够在有限步内终止,基于 DBM 的算法要求对超过指定界值的时钟值变量使用外推操作,这就导致当时钟变量之间存在对角约束时可能会产生可达集计算错误的情况,而要消除时钟变量的对角约束又会导致模型的规模呈指数级增长,所以在很大程度限制了该方法的应用[6~8].

近年来,随着可满足性求解技术取得的巨大进步,基于 SAT(Boolean Satisfiability)的有界模型检测算法作为极少数成功应用在工业应用中的形式化方法,越来越多地在实际的工业领域得以推广.作为命题 SAT 求解技术的扩展,SMT(Satisfiability Modulo Theories)技术也得到了极大的发展,并被用到了一些比命题约束具有更丰富约束关系的系统的有界模型检测中,线性混成系统就是其中之一.由于线性混成自动机不具有有限互模拟属性,所以线性混成系统的有界模型检测不具有完备性,只能是一种混成系统查错技术,不能证明混成系统设计的正确性[9,10].

为了解决上述问题,本文基于目前工业界比较成熟应用的混成系统有界模型检测方法[4,11,12]和文献[13,14]提出的有限状变迁态系统的近似最强后件计算的思想,在线性混成自动机的可满足性模线性算术理论编码的基础上,结合一阶逻辑的插值理论,给出了混成系统近似最强后件计算的实现过程,并建立近似最强后件操作所对应的线性混成系统可达性算法.具体来说,本文的主要工作如下:

(1)在使用一阶逻辑中可判定的可满足性模线性算术理论编码线性混成自动机的 k 的变迁公式 BMC(R,T,F,k)基础上,将 k 步变迁公式划分从状态 R 过一步变迁 T 可以到达状态的表示公式 $A = R \wedge T_0$ 和在 k-1 步内可以到达失效状态的表示公式 $B = T_1 \wedge T_2 \wedge \cdots \wedge T_k \wedge F_1 \wedge F_2 \wedge \cdots \wedge F_k$ 两部分,通过求取这两个公式得插值公式获得 R 的最强后件的上近似 R',从而实现了一种较为高效的近似最强后件操作 R'计算方法.

(2)在近似最强后件迭代计算线性混成系统的可达状态的过程中,初始从 $R = I$ 开始,在公式 $A \wedge B$ 不满足的情况使用近似最强后件操作计算 R 的最强后件 R',并将其状态 R 中.重复上述迭代直至可达状态达到不动点或者在迭代到第 i 次公式 $A \wedge B$ 可满足(存在长度为 $i + k$ 的反例).因为公式 $A \wedge B$ 中的公式 R 是可达状态的上近似,所以当迭代达到不动点停止,就说明系统是不能到达公式 F 表示的失效状态,系统模型检测算法结束.而当迭代是因为公式 $A \wedge B$ 可满足而停止,就需要使用公式 BMC$(I,T,F,i+k)$判定反例的真实性,如果 BMC$(I,T,F,i+k)$可满足则返回满足性赋值(反例),否则将 k 设置为 $k+1$,重新从 $R = I$ 开始新的一轮迭代.从而实现了一个无假反例的近似模型检测算法.

上述工作的实现,避免了使用凸多面体等作为状态集表示,实现可达性计算过程中引入假的反例的问题.同时,该算法的近似最强前件的计算也不像传统符号模型检测算法的精确最强前件计算那样,需要用到具有指数时间复杂度的量词消去操作,相比采用逻辑编码的混成系统模型检测算法具有更高的算法执行效率.

相对于有界模型检测算法,该算法除了解决混成系统有界模型检测的不完备性问题外,该算法在第 i 次迭代时是使用从初始状态 I 出发的 i 步可达状态集 R

作为出发状态集,使用公式 BMC(R,T,F,k)的可满足性判断步长为 $i+k$ 的反例的存在性,而只有在 BMC(R,T,F,k)可满足的情况下才完全按照有界模型检测的方法从初始状态集 I 出发完全展开变迁公式 $i+k$ 次判断反例的真假.而有界模型检测算法对任意长度 k 的反例都需要从初始状态 I 出发完全展开 k 次关系判断公式的可满足性进行确认,所以该算法在通常情况下都具有比有界模型检测算法更高的性能.

2 背景知识

2.1 线性混成自动机

为了阅读的方便,本文将向量缩写为粗体,如 $\boldsymbol{x}$ 是向量$(x_1,\cdots,x_n)$的缩写;引入状态变量 $\boldsymbol{s}=(v,\boldsymbol{x})$,其中 v 的取值为库所集 $V=\{v_1,\cdots,v_m\}$,$\boldsymbol{x}=(x_1,\cdots,x_n)$.本文中凸线性约束均是指形如$\{\boldsymbol{x}\in\mathbb{R}^n\mid P\boldsymbol{x}\leqslant q\}$的有限多个整系数的线性约束公式的合取,其中 $P\in\mathbb{Z}^{m\times m}$,$q\in\mathbb{Z}^m$;线性转换是形如$\{(\boldsymbol{x},\boldsymbol{x}')\in\mathbb{R}^n\times\mathbb{R}^n\mid \boldsymbol{x}'=A\boldsymbol{x}+\boldsymbol{b}\}$的空间变换,其中 $A\in\mathbb{Z}^{n\times n}$,$b\in\mathbb{Z}^n$.

定义 1 线性混成自动机 H 是一个六元组$(\boldsymbol{X},V,E,l_0,\boldsymbol{x}_0,G,A,I,R)$[12,15],其中:

①$\boldsymbol{X}=\{x_1,\cdots,x_n\}$是由 n 个实值变量构成的集合,其中 $n>0$;

②(V,E)是有限个顶点构成的有向图,其中顶点集 $V=\{v_1,\cdots,v_n\}$表示线性自动机的 n 个库所构成的集合,边集 E 表示自动机的离散变迁集合;

③l_0 代表初始库所;

④$\boldsymbol{x}_0$ 代表初始域,由 $\boldsymbol{X}=\{x_1,\cdots,x_n\}$中 n 个实值变量的凸线性约束定义;

⑤G 与 A 分别为变迁集 E 中的每一变迁指定一个由凸线性约束定义的触发条件和线性转换指定的连续向量 $\boldsymbol{x}$ 的连续演变规则;

⑥I 和 R 分别为库所集 V 中的每一库所指定一个由凸线性约束表示的不变量和形如 $l\leqslant \mathrm{d}\boldsymbol{x}\leqslant u$ 表示的连续向量 $\boldsymbol{x}$ 在该库所内的连续演化规则,其中$(l,u)\in\mathbb{Z}^n\times\mathbb{Z}^n$ 是一个超多面体,$\mathrm{d}\boldsymbol{x}$ 表示 $\boldsymbol{x}$ 的一阶导.

定义 1 给出了线性混成自动机的语法的形式定义,有了线性混成自动机的语法定义后,我们就可以使用线性混成自动机的语法建立混成系统的线性混成自动机模型.下面给出线性混成自动机的语义模型,线性混成自动机 $H:=(\boldsymbol{X},L,E,l_0,\boldsymbol{x}_0,G,A,I,R)$的语义模型被定义为一个变迁系统$(Q,Q_0,(\rightarrow_\tau\cup\rightarrow_\xi))$,其中:

①$Q=L\times\mathbb{R}^n$ 是线性混成自动机状态的集合;

②$Q_0=\{(l_0,\boldsymbol{x})\in Q\mid \boldsymbol{x}\in\boldsymbol{X}\cap I(l_0)\}$是初始状态集合;

③连续变迁关系$\rightarrow_\tau Q\times Q$ 使得$(l,\boldsymbol{x})\rightarrow_\tau(l',\boldsymbol{x}')$,当且仅当 $l=l'$,存在 $t>\mathbb{R}^+$使得 $\boldsymbol{x}+tl\leqslant\boldsymbol{x}'\leqslant\boldsymbol{x}+tu$ 并且$(l,u)\in R(l)$,$\boldsymbol{x}'\in I(l)$,使用 $\mathrm{flow}_l(\boldsymbol{x},t,\boldsymbol{x}')$表示连续变迁$\rightarrow_\tau$ 在库所 l 经过时间 t 后,连续向量的值从 $\boldsymbol{x}$ 连续变化为$\boldsymbol{x}'$;

④离散变迁关系$\rightarrow_\xi Q\times Q$ 使得$(l,\boldsymbol{x})\rightarrow_\xi(l',\boldsymbol{x}')$,当且仅当存在 $e\in E$,使得 $e=(l,l')$,$\boldsymbol{x}\in G(e)$,$(\boldsymbol{x},\boldsymbol{x}')\in A(e)$,并且 $\boldsymbol{x}'\in I(l')$,使用 $\mathrm{Jump}_{l,l'}(\boldsymbol{x},\boldsymbol{x}')$表示在离散变迁$\rightarrow_\xi$ 的作用下,线性混成自动机从库所 l 发生离散跳变到库所l',跳变前连续向量值 $\boldsymbol{x}$ 和跳变后的值 $\boldsymbol{x}'$.

在上述线性混成自动机的语法和语义的定义中,线性混成自动机 H 的状态包含一个控制库所 $l\in L$ 和实值连续向量 $\boldsymbol{x}\in\mathbb{R}^n$($\boldsymbol{x}$ 中变量的一次赋值),线性混成自动机的对应库所 l 的数据域 S_l 是$\mathbb{R}^n$ 中凸多面体集的有限并.一个域 $S=\bigcup_{l\in L}(l,S_l)$是由数据域 $S_l\in\mathbb{R}^n$ 构成的集族,其中每一个库所 l 对应一个数据域 S_l.

2.2 可满足性模理论[9,10,16]

本节所介绍的可满足性模理论是线性混成自动机符号化模型检测的基础,本文将依据节 2.1 给出的线性混成自动机的语义模型在节 2.3 使用可满足性模理论建立线性混成自动机的符号化表示,最后通过对可满足性模理论的公式可满足性判定和插值等操作的基础上实现线性混成自动机的符号化模型检测.

可满足性模理论在线性算术、未定义函数、位向量及其他一系列可判定的算术理论的支持下,通过研究具有可判定性的算术理论的可判定性保持扩展和组合方法,从而实现对命题逻辑的一般化推广.其中,类别基调(sorted signature)Σ 是一个三元组 $<\boldsymbol{S},\Sigma^F,\Sigma^{P>}$,$\boldsymbol{S}$ 是类别符号的集合,Σ^F与 Σ^P分别是互不相交的函数符号和谓词符号的集合.每个函数符号和谓词符号都带有一个被称为元的非负整数 n,n 元函数符号$f\in\Sigma^F$的元是一个有序序列$(S_1,S_2,\cdots,S_n,S)$,通常记作 $f:S_1,S_2,\cdots,S_n\rightarrow S$,0 元函数表示类别为 $\boldsymbol{S}$ 的个体常量;n 元谓词符号$p\in\Sigma^P$的元是一个有序序列$(S_1,S_2,\cdots,S_n)$,通常记作 $p:S_1,S_2,\cdots,S_n$,0 元谓词表示命题常元.假设 V 是由 $\boldsymbol{S}$ 诱导的变量集合族,并且每个类别符号 $S(S\in\boldsymbol{S})$诱导的 $\boldsymbol{V}$ 中的变量集合为 V_S.那么项的集合可以递归定义如下:如果变量 $v\in V_S$,那么 v 是类别为 S 的项;如果个体常量 c 是元为(S)的函数符号,那么 c 是类别为 S 的项;如果 $f:S_1,S_2,\cdots,S_n\rightarrow S$ 和 t_i是类别为 S_i的项,那么 $f(t_1,\cdots,t_n)$是类别为 S 的项.原子公式的集合定义如下:true 与 flase 是原子公式;如果 $P:S_1,S_2,\cdots,S_n$ 和 t_i 是类别为 S_i 的项,那么 $p(t_1,\cdots,t_n)$是原子公式;如果$\approx$代表逻辑相等符号,并且 t 与 s 都是类别为 S 的项,那么 $t\approx s$ 是一个原子公式.文字是原子公式或原

子公式的否定.公式集合的递归定义如下:原子公式是一个公式;如果 φ 是公式,那么$\neg\varphi$是公式;如果 φ_1 与 φ_2 是公式,那么 $\varphi_1\wedge\varphi_2$ 是公式;如果 φ_1 与 φ_2 是公式,那么 $\varphi_1\vee\varphi_2$ 是公式;如果 φ_1 与 φ_2 是公式,那么 $\varphi_1\rightarrow\varphi_2$ 是公式;如果 φ_1 与 φ_2 是公式,那么 $\varphi_1\leftrightarrow\varphi_2$是公式;如果 $\boldsymbol{x}$ 是变量φ是公式;那么$\forall\boldsymbol{x}.\varphi$是一个公式;如果 $\boldsymbol{x}$ 是变量φ是公式;那么 $\boldsymbol{x}.\varphi$ 是一个公式.如果公式 ϕ 是形如$\forall\boldsymbol{x}.\varphi$或$\boldsymbol{x}.\varphi$的公式,那么变量 $\boldsymbol{x}$ 在公式φ中的出现称为约束出现.一个公式被称为封闭的公式,如果它的所以变量都是约束出现的.不包含变量的公式称为句子.

基调 $\Sigma=<\boldsymbol{S}^1,\Sigma^F,\Sigma^P>$ 的结构(解释)$\tilde{A}$ 是一个序对 $<D,\rho>$,其中 D 称作论域,是由类别集合 $\boldsymbol{S}^1$ 诱导的非空集合族,ρ 是定义在 $\Sigma^F\cup\Sigma^P$ 上的完全函数,使得如果 $f:S_1,S_2,\cdots,S_n\rightarrow S$ 那么 $\rho(f):D_{S_1},D_{S_2}\times\cdots\times D_{S_n}\rightarrow D_S$ 和如果 $p:S_1,S_2,\cdots,S_n$ 那么 $\rho(p):D_{S_1}\times D_{S_2}\times\cdots\times D_{S_n}$.一个变量赋值 μ 是一个定义在变量集合 V 上的完全函数,它使得对所有 $v\in V_S$有 $\mu(v)\in D_S$.一个项 t 对于结构 $\tilde{A}$ 和变量赋值μ 的指称用符号$[\![t]\!]_\sigma^u$ 表示,它被递归定义如下:如果 t 是一个变量,那么$[\![t]\!]_\sigma^u=\mu(t)$;如果 t 是一个个体常量,那么$[\![t]\!]_\sigma^u=\rho(t)$;如果 $t=f(t_1,\cdots,t_n)$,那么$[\![t]\!]_\sigma^u=\rho(f)([\![t]\!]_\sigma^u,\cdots,[\![t]\!]_\sigma^u)$;我们将$\varphi$能够被$\sigma$和$\mu$所满足记作$(\tilde{A},\mu)\models\varphi$,关系$(\tilde{A},\mu)\models\varphi$能够被递归定义如下:当 $\varphi=\text{true}$ 时,$(\tilde{A},\mu)\models\varphi$ 平凡成立;当 $\varphi=p(t_1,\cdots,t_n)$时$(\tilde{A},\mu)\models\varphi$ 成立,当且仅当$([\![t]\!]_\sigma^u,\cdots,[\![t]\!]_\sigma^u)\in\rho(p)$;如果 $\varphi=t\approx s$ 时$(\tilde{A},\mu)\models\varphi$ 成立,当且仅当$[\![t]\!]_\sigma^u=[\![s]\!]_\sigma^u$;当公式 $\varphi=\neg\phi$ 时$(\tilde{A},\mu)\models\varphi$ 成立,当且仅当$(\tilde{A},\mu)\not\models\varphi$;当公式 $\varphi=\phi_1\vee\phi_1$ 时$(\tilde{A},\mu)\models\varphi$ 成立,当且仅当$(\tilde{A},\mu)\models\varphi_1$ 并且$(\tilde{A},\mu)\models\varphi_2$;当公式 $\varphi=\phi_1\vee\phi_1$ 时$(\tilde{A},\mu)\models\varphi$ 成立,当且仅当$(\tilde{A},\mu)\models\varphi_1$或$(\tilde{A},\mu)\models\varphi_2$;当公式 $\varphi=\phi_1\rightarrow\phi_1$ 时$(\tilde{A},\mu)\models\varphi$ 成立,当且仅当$(\tilde{A},\mu)\models\varphi_1$或$(\tilde{A},\mu)\models\varphi_2$;当公式 $\varphi=\forall\boldsymbol{x}.\phi$时$(\tilde{A},\mu)\models\forall\boldsymbol{x}.\phi$ 成立,当且仅当 $\boldsymbol{x}\in X_s$ 时每一个$e\in D_S$有$(\tilde{A},\mu[\boldsymbol{x}\rightarrow e])\models\phi$;当公式 $\varphi=\boldsymbol{x}.\phi$ 时$\models_\sigma^u\boldsymbol{x}.\phi$ 成立,当且仅当 $\boldsymbol{x}\in X_S$ 时存在$e\in D_S$ 使得$(\tilde{A},\mu[\boldsymbol{x}\rightarrow e])\models\phi$.当对所有的变量赋值 μ 有$(\tilde{A},\mu)\models\varphi$ 成立,那么称解释 $\tilde{A}$ 是公式φ的模型,记作 $\tilde{A}\models\varphi$:.假设 Γ 是一个公式的集合和φ是一个公式,对所有使得$(\tilde{A},\mu)\models\varphi$ 成立的 $\tilde{A}$ 和μ 也使得对所有$\phi\in\Gamma$ 有$(\tilde{A},\mu)\models\phi$,则称公式 φ 是Γ 的语义后承,记作 $\Gamma\models\varphi$.

一个 Σ-理论 J 是一个在语义后承下封闭的Σ-句子的集合,也就是说对每个公式 φ 有$J\models\varphi$,当且仅当 $\varphi\in J$.假设 J 是一个理论,如果存在一个 Σ-理论 J 的模型 $\tilde{A}$ 和一个变量赋值μ 使得$(\tilde{A},\mu)\models\varphi$,那么称 φ 是 J-可满足的.对于 Σ-理论 J 的可满足性模理论问题 SMT(J)就是判定给定无量词 Σ-公式 φ 是否是 J-可满足的,有了一阶理论相关的定义,接下来我们给出本文将要用到的实数域上的线性算术理论的定义.

定义 2　实数域的线性算术理论 $J_{LA(\mathbb{R})}$ 的基调 $\Sigma_{LA(\mathbb{R})}$对每个有理数 $r\in\mathbb{Q}$有一个对应的常量符号 c_r,$\Sigma_{LA(\mathbb{R})}$-句子的集合是全部在以实数集$\mathbb{R}$作为论域 D 的结构 $\tilde{A}$ 下按照标准的解释为真的句子的集合.

实数域上无量词的线性算术理论的可满足性问题是多项式时间复杂度的,本文将主要使用实数域上的线性算术公式给出线性混成自动机的有界模型检测表示公式,并在后续章节利用其线性时间复杂插值算法构建符号化模型检测算法.

3　线性混成自动机限界模型检测

3.1　线性混成自动机的符号表示[11,12]

表示线性混成自动机语义的变迁系统 $M\overset{\text{def}}{=}(Q,Q_0,(\rightarrow_\tau U\rightarrow_\zeta))$可以使用带有论域 A 的一阶结构 $\tilde{A}$ 和变量集合 $\boldsymbol{X}$($\boldsymbol{X}$ 包含库所变量和连续变量)符号化表示(对本文所论述的线性混成自动机,结构 A 的是指线性算术结构),并通过求取结构 $\tilde{A}$ 上的公式的可满足性来实现变迁结构 M 的符号化模型检测过程,具体过程如下:

①每一个状态 $s\in Q$ 是$[\boldsymbol{X}\rightarrow A]$中的映射;

②初始状态的集合 I 用无量词的公式 $I[\boldsymbol{x}]$编码表示,使得 $s\in Q_0$,当且仅当 $\tilde{A}\models I([S]$;

③变迁关系$(\rightarrow_\tau U\rightarrow_\zeta)$用无量词的公式 $T[\boldsymbol{x},\boldsymbol{x}']$编码表示,使得$(s,s')\in(\rightarrow_\tau U\rightarrow_\zeta)$,当且仅当 $\tilde{A}\models T[s,s']$.

通过上面定义的编码方法,就可以将混成自动机的状态可达性问题转换为判定一组由布尔变量和在实值变量上线性表达式的布尔组合公式的可满足性问题,并利用可满足性模线性算术理论的插值获得新的可达状态.

在具体的符号化表示的过程中,为了能够区别库所变量和连续实值变量在线性混成自动机 $H:=(\boldsymbol{X},L,E,l_0,\boldsymbol{x}_0,G,A,I,R)$上,本文使用有序对$(at,\boldsymbol{x})$表示状态向量,其中 at 是解释在库所集 L 上的指定当前状态所在库所的库所变量,实值向量 $\boldsymbol{x}$ 中的变量 $x_i\in\boldsymbol{X}$ 是解释在实数集$\mathbb{R}$上的连续实值变量.

定义 3　对于库所集 $L=\{l_1,\cdots,l_m\}$与实值变量集 $\boldsymbol{X}=\{x_1,\cdots,x_n\}$($m,n\in\mathbb{N}$)的线性混成自动机 $H:=(\boldsymbol{X},L,E,l_0,\boldsymbol{x}_0,G,A,I,R)$,其状态和变迁公式使用可满足性模线性算术理论公式定义如下:

①初始状态集:

$$\varphi_I\overset{\text{def}}{=}(at=l_0\wedge\boldsymbol{x}=x_0)$$

②离散变迁：

$$J(s,s) \overset{\text{def}}{=} \vee_{l,l' \in L}(at = l \wedge at' = l' \wedge \text{Jump}_{l,l'}(\boldsymbol{x},x') \wedge \text{Inv}_{l'}(\boldsymbol{x}))$$

③连续变迁：

$$F(s,s') \overset{\text{def}}{=} \vee_{l \in L}(at = l \wedge at' = l \wedge t > 0 \wedge \text{Flow}_l(\boldsymbol{x},t,\boldsymbol{x}') \wedge \text{Inv}_l(\boldsymbol{x}'))$$

④变迁公式：

$$\varphi_R(s,s) \overset{\text{def}}{=} J(s,s) \vee F(s,s)$$

其中，$S=(at,S)$和 $S'=(at,\boldsymbol{x})$是状态变量，t 是大于 0 的实值变量表示连续变迁连续变化的时间. 需要注意的是对于离散变迁通过离散跳变 $\text{Jump}_{l,l'}(\boldsymbol{x},\boldsymbol{x})$进入状态 s'和连续变迁在连续变化 $\text{Flow}_l(\boldsymbol{x},t,\boldsymbol{x})$的作用下经过 t 个时间单位后的状态 s'必须检查满足所在库所的不变量约束.

3.2 符号化模型检测

3.1 节在线性混成自动机的语义变迁模型基础上，使用可满足性可判定的可满足性模理论的线性算术理论公式给出了线性混成自动机一阶符合化表示，本节将讨论基于该可满足性模理论线性算术理论公式表示的线性混成自动机状态可达性的符号化模型检测方法[11,12].

由 3.1 节的线性混成自动机的符号化表示可知，一个状态 s 是线性混成自动机可达的，当且仅当在线性混成自动机的符号化编码表示中存在一个状态序列 $s_0,\cdots,s_{k-1},s_k=s$ 使得

$$\check{A} \models I[s_0] \wedge T[s_0,s_1] \wedge \cdots \wedge T[s_{k-1},s_k] \quad (1)$$

$F[\boldsymbol{x}]$是一个表示状态集的公式，满足公式 $F[\boldsymbol{x}]$的状态 s 是可达的，当且仅当存在一个状态序列 $s_0,\cdots,s_{k-1},s_k=s$ 使得

$$\check{A} \models I[s_0] \wedge T[s_0,s_1] \wedge \cdots \wedge T[s_{k-1},s_k] \wedge F[s_k] \quad (2)$$

因此，当公式 $F[\boldsymbol{x}]$表示的是不安全状态的集合时，就可以使用式(2)来判断被验证系统是否存在可执行变迁到达不安全的状态. 一个变迁系统 M 对于一个公式 F 所表示的状态是安全的，当且仅当公式对任意的 $k \geqslant 0$ 是不可满足的.

$$I[x_0] \wedge T[x_0,x_1] \wedge \cdots \wedge T[x_{k-1},x_k] \wedge F[x_k] \quad (3)$$

由于线性混成自动机状态空间是无穷的，在式(3)中一般不存在一个具体的 k 值使得线性混成自动机的所有可达状态都能够被式(3)所表示. 所以，在线性混成自动机的模型检测过程中就不能使用式(3)来为线性混成自动机提供一个完备的模型检测算法. 对于线性混成自动机这类系统，只能通过公式 l 和公式 T 求线性混成自动机语义变迁系统 M 的最强递归不变量(strongest inductive invariant) φ_{IT}.

定义 4 一个变迁结构 $M \overset{\text{def}}{=} (S,I,R,L:S \to 2^{AT})$的可达状态的集合 $\boldsymbol{R}$ 是满足下面两个条件的最小集合：

① $I \subseteq \boldsymbol{R}$(初始状态是可达状态)

② $\boldsymbol{R} \to^R \subseteq \boldsymbol{R}$(可达状态集 $\boldsymbol{R}$ 经过 $\boldsymbol{R}$ 变迁后的状态任是 $\boldsymbol{R}$ 中的状态)

定义 5 集合 $\boldsymbol{R}$ 是变迁结构 $M \overset{\text{def}}{=} (S,I,R,L:S \to 2^{AT})$的可达状态集，$M$ 的最强递归不变量是公式 φ_{IT}，使得 $\check{A} \models \varphi_{IT}[S]$，当且仅当 $s \in \boldsymbol{R}$.

从定义 2 和定义 3 可知对线性混成自动机的任意的状态 $s \in S$，如果状态 s 是线性混成自动机可达的，那么就有线性混成自动机对应的一阶结构 A，使得式(4)成立.

$$\check{A} \models \varphi_{IT}[s] \quad (4)$$

因此，要证明线性混成自动机所对应的变迁系统 M 对于表示不安全属性的公式 ψ 是安全的，只需要证明

$$\check{A} \models \varphi_{IT}[x] \wedge \psi[x] \quad (5)$$

是不可满足的.

定义 6 公式 ψ 对于变迁关系 T 的最强后件 $sp_T(\psi)$是公式 φ，使得 $\psi \wedge T \models \varphi$，并且对一切满足 $\psi \wedge T \models \varphi'$的 φ'有 $\varphi \models \varphi'$.

由定义 6 可知，当公式 ψ 约束表示状态的集合时公式 φ 所表示的状态可以看作是公式 I 表示状态对于公式 T 表示的变迁转换后的最强后件. 由公式 φ_{IT}的定义知道，公式 φ_{IT}是从公式 T 表示的状态出发进行最强后件操作的最小不动点，因此，对于线性混成自动机符号表示的变迁系统 M 的最强递归不变量 φ_{IT}就能通过计算最强后件 $sq_T(I)$的不动点得到：

$$\varphi_{IT} = \mu \boldsymbol{x}.\, I \vee sp_T(\boldsymbol{x}) \quad (6)$$

从式(6)可知，公式 φ_{IT}的计算要点就是计算公式 sp_T，公式 sp_T的计算如下：

$$sp_T(\varphi) = \exists \boldsymbol{x}(\varphi \wedge T) \quad (7)$$

也就是必须采用量词消去这类方法来计算可达集合 $\boldsymbol{R}$，然而，线性算术公式量词消去运算的计算复杂度是双指数级，如果直接采用上述方法来计算线性混成自动机的可达状态必然导致效率太低. 为了解决这个问题，本文主要是采用具有线性时间复杂度 Craig 插值技术实现公式 sp_T的上近似计算.

4 基于 Craig 插值的模型检测算法

4.1 Craig 插值

3.2 节给出的符号化模型检测方法中最强后件计算过程的量词消除算法的复杂性是整个符号化模型检测算法的效率的主要障碍，针对这个问题，本节主要采用基于 Craig 插值[6]的方法用过近似的 sp'_T 代替精确的

sp_T计算.

定义 7　假设(ψ,φ)是一个满足条件 $\psi\wedge\varphi$ 不可满足的公式对,(ψ,φ)的插值公式 J 是满足如下条件的公式:

①ψ 蕴含 J;

②$J\wedge\varphi$ 不可满足;

③$OCC(J)OCC(J)\cap OCC(J)$(其中 $OCC(F)$表示公式 F 的非逻辑符号).

Craig 插值定理指出对任意的不一致的一介公式对(ψ,φ)插值总是存在,在实际应用中更让研究人员感兴趣的是 Craig 插值总是可以通过特定证明系统中 $\psi\wedge\varphi$ 的不可满足性在线性时间内消解反证得到.

4.2　基于 Craig 插值的模型检测算法

基于命题逻辑公式 Craig 插值的模型检测算法已广泛的用在有限状态系统模型检测中[13,14].本文进一步将扩展命题逻辑的可满足性模线性算术理论应用到线性混成自动机进行符号化编码,将线性混成自动机的语义变迁模型表示成可满足性模线性算术理论公式,从而利用可满足性模线性算术理论的可判定性和一阶逻辑不可满足性公式插值的存在性实现线性混成自动机的基于插值的符号化模型检测算法.

由于在线性混成自动机的符号化模型检测算法的核心操作是最强后件公式 sp_T的计算,所以提高线性混成系统的模型检测算法的效率直接取决于最强后件公式 sp_T的计算时间.本节的主要目的是利用 Craig 插值给出精确地最强后件公式 sp_T的近似公式 sp_T'的计算方法,然后根据该方法给出基于插值的线性混成自动机的符号化模型检测算法.

在基于插值的线性混成自动机的符号化模型检测中,首先给定一个界值 k,然后类似线性混成自动机的限界模型检测过程,按照界值 k 将线性混成自动机的变迁关系展开成公式 BMC(I,T,F,k):

$$I(s_0)\wedge T(s_0,s_1)\wedge\cdots\wedge T(s_{k-2},s_{k-1})\wedge T(s_{k-1},s_k)\wedge(F(s_1)\vee\cdots\vee F(s_{k-1})\vee F(s_k)) \tag{8}$$

由于线性混成自动机的符号化编码所得的公式都是一阶可满足性模线性算术理论和命题逻辑中的公式,所以式(8)的可满足性是可判定的,可以使用 SMT 求解器进行可满足性求解.如果式(8)可满足,那么混成自动机中至少存在一条长度为 k 的反例.如果式(8)不可满足,那么就表明不存在一个从初始状态出发的状态序列可以在 k 步之内到达用公式 F 表示的具有不安全性质的状态集.因此,可以将式(8)划分为由 A 和 B 表示的如下两个部分:

$$\begin{aligned}A&\equiv I(s_0)\wedge T(s_0,s_1)\\B&\equiv T(s_1,s_2)\wedge\cdots\wedge T(s_{k-2},s_{k-1})\wedge T(s_{k-1},s_k)\\&\quad\wedge(F(s_1)\vee\cdots\vee F(s_{k-1})\vee F(s_k))\end{aligned} \tag{9}$$

因为公式 A 与公式 B 是不一致的,所以根据插值定理可知公式 A 与公式 B 之间必定存在一个插值公式ψ,并且有 $A[s_1]\models\psi[s_1]$,$\psi[s_1]\wedge B[s_1]$不可满足,所以,所有从状态 s_0 经过一步可达的状态集都包含在公式 ψ 所表示的状态集合内,因而公式 ψ 所代表的状态集合包含公式 I 的最强后件 $sp_T(I)$所表示的状态集合,也就是公式 ψ 是公式 I 所表示的向前一步可达状态集合的上近似,所以公式 ψ 也就是我们所求得最强后件近似公式 sp_T'.此外,因为公式 ψ 与公式 B 是不一致的,所以公式 F 所表示的状态集合相对于公式 ψ 所表示的状态集合在 k-1 步内是不可达的.

如果将上述 BMC(I,T,F,k)公式中的表示初值状态的公式 I 替换成表示当前状态的公式 R,并将初始的可达状态 R 赋值为 I,那么就可以利用不动点迭代获得系统的可达状态的不动点.接下来本文将根据 sp_T' 给出基于插值的符号化模型检测算法.基于插值的符号化模型检测的主要思想是:在给定的一个整数值 k 的前提下,将初始可达状态 R 设置为系统初始状态 I,使用 SMT 求解器求解表示有界模型检测的公式 BMC(R,T,F,k)的可满足性,如果公式 BMC(R,T,F,k)可满足算法返回可满足赋值(反例),程序终止.当公式 BMC(R,T,F,k)不可满足时,将公式 BMC(R,T,F,k)划分为公式 A 与公式 B 两个部分,计算公式 A 与公式 B 的插值得到 sp_T'.因为有 $A\models sp_T'$,所以所有从 $R(s_0)$一步变迁可达的状态在公式 sp_T' 下都为真,也就是说公式 sp_T' 表示的状态集是从 $R(s_0)$经过一步变迁可达状态集合的上近似.同时,因为 $sp_T'\wedge B\models\perp$,所以所有满足公式 sp_T' 的状态不包含在 k-1 步内能够到达公式 F 的状态.如果公式 $sp_T'\models R$ 成立,表示当前可达状态在经过一步变迁后没有新的状态可达,那么得到线性混成系统可达状态的不动点.如果公式 $sp_T'\models R$ 不成立,就将新的状态加入当前可达状态构成新的可达状态集的上近似 R'.算法用公式 R'替换 BMC(R,T,F,k)公式中的原有状态集 R,继续迭代求解路径深度为 k 的有界模型检测公式的可满足性和在不可满足情况下继续迭代可达状态.在可满足的情况那么就说明反例出现,因为计算的是可达状态的上近似,而当前得到的反例深度实际是变迁展开深度 k 加上迭代次数 i,所以就需要用迭代的次数 i 加上当前变迁展开长度 k 来获得新的 k 值,并使用原始的限界模型检测判定反例真假.这个时候算法就又回到算法开始的开始部分,因为这个时刻的 k 值是上一轮迭代过程中的 k 值加上一轮迭代次数 i 的值,所以在不可满足的情况下,继续的新一轮迭代实际上就通过将 k 值增加到 $k+i$ 实现了反例的精细化,从而将所有长度小于新的 k 值的反例在新的不动点迭代过

程中移除.算法的具体过程如算法 1 所示.

算法 1 基于插值近似的可达性算法

输入:初始状态公式 I;变迁关系公式 T;不安全状态公式 F.

输出:当不安全状态是可达状态,返回"Yes",否则返回"No"

(1)判断谓词公式 I 与 F 的合取是否可满足,可满足返回"Yes",否则将公式 I 赋值给表示可达状态的公式 R,设置迭代计数器 $i=0$,设置变迁展开深度 $k=1$

(2)将公式 $\mathrm{BMC}(R,T,F,k)=R(s_0)\wedge T(s_0,s_1)\wedge\cdots\wedge T(s_{k-2},s_{k-1})\wedge T(s_{k-1},s_k)\wedge(F(s_1)\vee\cdots\vee F(s_{k-1})\vee F(s_k))$划分为公式 $A=R(s_0)\wedge T(s_0,s_1)$和公式 $B=T(s_1,s_2)\wedge\cdots\wedge T(s_{k-2},s_{k-1})\wedge T(s_{k-1},s_k)\wedge(F(s_1)\vee\cdots\vee F(s_{k-1})$两个部分,判断公式 A 与公式 B 是否可满足

(3)如果公式 $A\wedge B$ 可满足,判断公式 $R(s_0)$与公式 $I(s_0)$是否等价,如果等价返回"Yes",否则设置 $k=k+i$ 和 $R(s_0)=I(s_0)$,继续执行步 2

(4)如果公式 $A\wedge B$ 不可满足,计算公式 A 与公式 B 的插值公式 Φ 和设置 $i=i+1$,并对公式 Φ 使用变量换名操作使得公式 $R'(s_0)=R(s_0)\vee\Phi(s_{0/s_1})$

(5)判断公式 $R'(s_0)\Rightarrow R(s_0)$是否为真,如果为真返回"No",如果不为真用公式 $R'(s_0)$替换公式 $R(s_0)$,继续执行步 2

上述算法最强后件公式的计算复杂是与公式规模成线性关系的,而公式规模的复杂度又是随公式展开深度成线性关系增长.所以本文的算法的最强后件计算复杂度极大的优于采用量词消去操作实现最强后件计算的混成系统有界模型检测算法.

4.3 算法的加宽操作

因为线性混成自动机的符号化模型检测是通过计算当前可达状态的最强后件公式来计算线性混成自动机的可达状态集,而公式的最强后件公式所表示的只是公式表示的状态经过一步变迁可达的状态.所以,当线性混成自动机具有无限变迁结构,且每一次从当前可达状态经过一步变迁后都有新的状态可达时,使用不动点迭代可达状态的算法就可能不终止.

为了保证多面体抽象域上的不动点计算收敛,抽象解释理论框架给出了基于多面体抽象域的外推操作过近似不动点的方案.然而,本文的基于插值的线性混成系统模型检测算法在不动点跌代过程中实际获得的是有限多个多面体的并集(析取),因此本文采用文献[17]提供的一种更加一般化的针对多个多面体的外推操作以保证基于插值的可达状态迭代过程的收敛,但是这种外推操作也可能引入假的反例,鉴于版面有限本文不以详述,有兴趣的读者可以参考相应文献.

5 实验分析

为了评估本文提出的基于插值的线性混成系统模型检测算法,在 DELL A840 PC(Intel Core2 T CPU T5457 1.60GHz/ 3GB RAM)Ubuntu10.04 平台上使用 C 语言调用支持实数域上的线性算术插值的可满足性模理论求解引擎 Mathsat5[18]的 API 函数分别实现了基于插值的线性混成系统模型检测算法 imc 和文献[12]给出的有界模型检测算法 bmc 进行对比分析两个算法的优、缺点.

考虑到 imc 和 bmc 算法中都需要在一定程度上根据反例深度展开被验证系统的变迁关系,这种变迁关系的展开也会使得变量和子句数目跟着相应的增加,而影响可满足性模理论求解引擎求解效率的主要因素就是变量和子句数目.所以,为了保证实验对象覆盖范围的全面性,本文分别选择了变迁关系展开比较有限可终止线性混成自动机基准模型和变迁关系可以无限展开的不可终止线性混成自动机基准模型,分别模拟简单和复杂两类线性混成系统,然后按照不同的反例的深度进行对比分析 imc 和 bmc 算法执行时间和内存占用情况.两种类型线性混成自动机在不同反例深度的对比分析分别如表 1 和表 2 所示.

表 1 bmc 和 imc 算法在可终止线性混成自动机模型资源占用对比表

No.	location	x	y	时间		内存		反例深度
				bmc(Sec)	imc(Sec)	bmc(M)	imc(M)	
1	—	2	12	0.02	0.05	2.83	3.24	3
2	3	0	5	0.04	0.08	3.18	3.45	4
3	—	2	10	0.07	0.09	3.42M	3.72	5
4	—	1	3	0.08	0.12	3.76	4.56	6
5	—	12	11	—	0.13	—	6.54	无反例

表 2 bmc 和 imc 算法在不可终止线性混成自动机模型资源占用对比表

No.	location	x	y	时间		内存		反例深度
				bmc(Sec)	imc(Sec)	bmc(M)	imc(M)	
1	—	1	2	0.01	0.02	2.62	2.81	4
2	—	1	3	0.02	0.04	2.69	2.89	6
3	—	1	4	0.04	0.05	2.71	3.02	8
4	—	1	5	0.10	0.06	2.81	3.24	10
5	—	1	10	1.21	0.13	3.81	3.48	20
6	—	1	15	265.89	0.22	88.45	3.74	30
7	—	1	20	—	0.32	—	4.18	40
8	—	1	25	—	0.41	—	4.53	50
9	—	1	50	—	1.09	—	6.34	100
10	—	20	140	—	0.03	—	2.78	无反例

基于以上表 1 和表 2 给出的数据,可以看出 bmc 算法和 imc 算法的优缺点如下:

(1)在可终止模型上,因为实验中给出的不终止模型是一个比较简单的模型,模型中反例深度都较小.所以 imc 和 bmc 算法在这种情况下进行模型检测展开变

迁关系的深度不能形成太大差距,而 bmc 算法在展开变迁关系后只需进行可满足性判定操作,而 imc 算法除了要做同样的可满足性判定之外,还需要划分公式进行消解反证获取插值公式.所以在变迁关系展开比较有限的可终止模型上,bmc 算法在执行时间和占用的内存空间上都表现出比 imc 算法略优.

(2)在不可终止模型上,因为反例深度可以是任意整数,而查找相应深度的反例,bmc 算法需要完全的展开变迁关系相应的次数,然后判断公式的可满足性,故变迁公式的完全展开必然导致计算效率降低.而 imc 算法更多的则是用经 i 次迭代计算得到可达状态的上近似可达状态公式与 $k-1$ 次展开的变迁公式之间的可满足性证明是否存在一个长度为 $k+i$ 深度的反例,不需要完全展开 $k+i$ 变迁关系.因而在反例深度较大时,imc 算法能够较明显的显示出比 bmc 算法具有更优的执行时间和内存占用.

(3)无论是在可终止和不可终止模型上,imc 算法都能给出系统满足安全属性的证明,而 bmc 算法只能是一种类似仿真的查找错误的技术.

因为实际验证中的难点是复杂系统,而一般简单的系统略微的多消耗资源也不会对验证过程产生实质性的影响.所以,基于插值的线性混成系统符号化模型检测算法是一种比线性混成系统有界模型检测算法更接近实际使用的算法.

6　结语

本文使用可满足性模线性算术理论公式给出了线性混成自动机的步长为 k 的有界模型检测表示公式,然后利用一阶逻辑的 Craig 插值定理,获得线性时间复杂度的一步可达状态计算方法,避免了使用双指数时间复杂度的量词消除方法的过程.此外,该方法在迭代不动点的过程中该算法均是按照一个同样的 k 值展开变迁公式,只有在出现了反例时候才将变迁公式按照反例的实际长度进行展开,从而比有界模型检测使用更小的公式进行可满足性判定.因此,该算法的执行时间和占用内存也比有界模型检测算法具有更大的优势.

在本文给出的基于插值的线性混成自动机模型检测算法中还有一些问题需要进一步研究.首先,尽管采用可满足性模理论中的线性算术理论公式作为线性混成自动机的状态集合的表示手段,能够有效的表示一些无穷状态的集合,但是这种表示还不够简约,而且在不动点迭代的过程中可能会出现大量冗余公式,使得可满足性模理论引擎判定公式的可满足性的效率极大的降低,还需要在今后的工作中展开进一步研究.

参考文献

[1] T A Henzinger, V Rusu. Reach ability verification for hybrid automata[A]. Proceedings of Hybrid Systems: Computation and Control[C]. Berlin: Springer-Verlag, 1998. 190 – 204.

[2] A B TEWODROS, CORNELIU P, ANDREY R. Solving existentially quantified horn clauses[A]. Proceedings of the 25th International Conference on Computer Aided Verification[C]. Saint Petersburg, Russia: Springer-Verlag, 2013. 869 – 882.

[3] 林惠民,张文辉.模型检测:理论、方法与应用[J].电子学报,2002,30(3):1907 – 1912.
Lin Huimin, Zhang Wenhui. Model checking: Theories, techniques and applications[J]. Acta Electronica Sinica, 2002, 30(3):1907 – 1912. (in Chinese)

[4] T A Henzinger, P H Ho. A note on abstract interpretation strategies for hybrid automata[A]. Proceedings of Hybrid Systems: Computation and Control[C]. Berlin: Springer-Verlag, 1995. 252 – 264.

[5] Eggers A, et al. Challenges in constraint-based analysis of hybrid systems[A]. Proceedings of Recent Advances in Constraints[C]. Berlin: Springer-Verlag, 2009. 51 – 65.

[6] M Sorea. Bounded model checking for timed automata[J]. Proceedings of 3rd Workshop on Models for Time-Critical Systems, 2003, 68(5): 116 – 134.

[7] Bouyer P. From qualitative to quantitative analysis of timed systems[D]. ENS Cachan: University Paris, 2009. 7.

[8] 尹传龙,庄雷,王从银.实时模型检测中基于驻留环的精确加速[J].电子学报,2011,39(3):489 – 493.
YIN Chuan-long, ZHUANG Lei, WANG Cong-yin. Exact acceleration of real-time model checking based on parking cycle[J]. Acta Electronica Sinica, 2011, 39(3): 489 – 493. (in Chinese)

[9] L De Moura, N Bjorner. Satisfiability modulo theories: An appetizer[A]. Proceedings of Formal Methods: Foundations and Applications[C]. Berlin: Springer-Verlag, 2009. 23 – 36.

[10] L De Moura, B Dutertre, N Shankar. A tutorial on satisfiability modulo theories[A]. Proceedings of Computer Aided Verification[C]. Berlin: Springer-Verlag, 2007. 20 – 36.

[11] Audemard G, et al. Verifying industrial hybrid systems with MathSAT[J]. Electronic Notes in Theoretical Computer Science, 2005, 119(2): 17 – 32.

[12] Abraham E, et al. Optimizing bounded model checking for linear hybrid systems[J]. Verification, Model Checking, and Abstract Interpretation Lecture Notes in Computer Science, 2005, 33(5): 396 – 412.

[13] McMillan K L. Applications of Craig interpolants in model checking[A]. Proceedings of Tools and Algorithms for the Construction and Analysis of Systems[C]. Heidelberg: Springer-Verlag, 2005. 1 – 12.

[14] McMillan K L. Interpolation and SAT-based model checking [A]. Proceedings of Computer Aided Verification[C]. Berlin: Springer-Verlag, 2003. 1 – 13.

[15] B Boigelot, L Bronne, S Rassart. An improved reachability analysis method for strongly linear hybrid systems[A]. Proceedings of Computer Aided Verification [C]. Berlin: Springer-Verlag, 1997. 167 – 178.

[16] A Cimatti, A Griggio, R Sebastiani. Efficient generation of craig interpolants in satisfiability modulo theories[J]. ACM Transactions on Computational Logic, 2010, 12(1): 1 – 54.

[17] T Bultan, R Gerber, W Pugh. Model-checking concurrent systems with unbounded integer variables: Symbolic representations, approximations, and experimental results [J]. ACM Transactions on Programming Languages and Systems, 1999, 21(4): 747 – 789.

[18] Alessandro Cimatti, Alberto Griggio, Bastiaan Joost Schaafsma, Roberto Sebastiani. The MathSAT5 SMTsolver[A]. Proceedings of the 19th International Conference on Tools and Algorithms for the Construction and Analysis of Systems[C]. Rome, Italy: Springer-Verlag, 2013. 124 – 138.

作者简介

陈祖希 男. 1981 年 3 月出生, 贵州赫章人. 2009 年进入同济大学电子与信息工程学院就读博士, 2012 年于美国科罗拉多大学博尔德分校 PL group 作为访问学者, 从事安全评估与形式化验证技术方面的研究.

E-mail: chenzuxi814@hotmail.com

徐中伟 男. 1964 年 6 月出生, 江苏无锡人. 同济大学电子与信息工程学院教授, 博士生导师, 铁道部计算机联锁检验站站长, 《铁道学报》编委. 研究方向包括: 基于通信的列车控制系统, 软件、通信协议的安全性形式验证和测试评估.