ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.17, No.1, January 2006, pp.1−10
DOI: 10.1360/jos170001

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

有限精度时间自动机的可达性检测*

晏荣杰[1,2+], 李广元[1], 徐雨波[1,2], 刘春明[1,2], 唐稚松[1]

[1](中国科学院 软件研究所 计算机科学重点实验室,北京 100080)

[2](中国科学院 研究生院,北京 100049)

Reachability Checking of Finite Precision Timed Automata

YAN Rong-Jie[1,2+], LI Guang-Yuan[1], XU Yu-Bo[1,2], LIU Chun-Ming[1,2], TANG Zhi-Song[1]

[1](Laboratory of Computer Science, Institute of Software, The Chinese Academy of Sciences, Beijing 100080, China)

[2](Graduate School, The Chinese Academy of Sciences, Beijing 100049, China)

+ Corresponding author: Phn: +86-10-62562796, Fax: +86-10-62563894, E-mail: yrj@ios.ac.cn, http://www.ios.ac.cn

Yan RJ, Li GY, Xu YB, Liu CM, Tang ZS. Reachability checking of finite precision timed automata. *Journal of Software*, 2006,17(1):1−10. http://www.jos.org.cn/1000-9825/17/1.htm

Abstract: To relieve the state space explosion problem, and accelerate the speed of model checking, this paper introduces the concept of finite precision timed automata (FPTAs) and proposes a data structure to represent its symbolic states. FPTAs only record the integer values of clock variables together with the order of their most recent resets to reduce the state space. The constraints under which the reachability checking of a timed automaton can be reduced to that of the corresponding FPTA are provided, and then an algorithm for reachability analysis is presented. Finally, the paper presents some preliminary experimental results, and analyzes the advantages and disadvantages of the new data structure.

Key words: finite precision timed automata; symbolic method; model checking; reachability

摘 要: 为了缓解状态空间爆炸问题,减小模型检测过程中生成的状态空间,加快模型检测速度,引入有限精度时间自动机(finite precision timed automata,简称 FPTA)作为实时系统的形式模型,并提出了一种数据结构 SDS(series of delay sequence)符号化表示状态空间中的状态集.FPTA 只记录时钟变量的整数值及时钟变化的先后次序,从而减小生成的状态空间.在一定的时间约束下,Alur 与 Dill 提出的时间自动机的可达性检测可简化为 FPTA 的可达性检测.举例描述了状态空间的生成过程和表示方法.最后,列出部分初步的实验结果,分析了 SDS 的特点及不足.

关键词: 有限精度时间自动机;符号化方法;模型检测;可达性

中图法分类号: TP301 文献标识码: A

时间自动机[1]为实时系统的自动化分析和验证提供了一种形式化的理论模型.在对实时系统运用模型检

* Supported by the National Natural Science Foundation of China under Grant Nos.60273025, 60223005, 60421001 (国家自然科学基金); the National Grand Fundamental Research 973 Program of China under Grant No.2002cb312200 (国家重点基础研究发展规划(973))

Received 2004-04-28; Accepted 2005-07-11

测方法进行研究的过程中,出现了许多基于时间自动机的模型检测工具,如 Uppaal[2],Kronos[3,4],Red[5−7],Rabbit[8]等.然而,在实时系统的模型检测过程中,状态空间爆炸几乎是所有工具都必须面对的问题.为了减小检测过程中占用的存储空间,加快检测速度,一般的做法包括将状态空间划分成等价类[1,9],从模型本身特点出发,利用对称归约(symmetry reduction)[10,11]、偏序归约(partial order reduction)[12]或活动时钟归约(active clock reduction)[13]等方法减少系统的状态,或将连续时间离散化[8,14]等.

运用离散语义下的时间自动机(discrete timed automaton)[15]进行模型检测虽然效率较高,却不适合对异步系统进行检测.连续语义下的时间自动机[1]既可以描述同步系统,也可以描述异步系统,但一般情况下进行模型检测的复杂度很高[16].针对这些问题,本文提出有限精度时间自动机(finite precision timed automaton,简称 FPTA)模型,用以进行实时系统的模型检测.“有限精度”是指该模型中的时钟仅取整数值.可以证明,在一定的时间约束下(参见定理 2.1),时间自动机的可达性可以归结为 FPTA 的可达性.

在与模型检测相关的研究工作中,还有许多是针对如何符号化表示状态空间,从而加快检测速度方面的.其中,文献[1,9]中提到的划分状态等价类的方法是符号化方法的基础.它根据时钟的取值将状态归为不同的 region,把因无限连续时间导致的无限状态空间分成有限的等价类.基于此类划分方法,出现了 zone,使用不等式组成的约束集表示满足该组约束的状态集.目前,大部分实时系统模型检测工具都采用 zone 来表示状态集,如 Uppaal,Kronos 和 Red 等.对于 zone 的表示存在不同的数据结构.如采用 DBM(difference-bounded matrix)表示 zone 的工具有 Uppaal,Kronos 等.同时,这两种工具也可以用类似 BDD(binary decision diagram)[17]的数据结构来表示 zone[18,19].

根据 FPTA 的相应特点,本文设计了一种与 DBM 及 BDD 不同的数据结构符号化表示状态的集合,显式地记录了模型检测过程中生成的状态空间.基于 FPTA 模型及对应的数据结构,本文还给出其模型检测工具中可达性的分析算法,并列出实验结果.

本文第 1 节简单介绍时间自动机的相关定义.第 2 节首先介绍 FPTA 的基本概念,然后给出 FPTA 的语义解释,并讨论了 FPTA 与时间自动机就可达性的等价关系.第 3 节给出根据 FPTA 特点符号化表示状态空间的数据结构及可达性分析算法,并举例说明可达性分析中状态空间的生成过程.第 4 节与相关工作进行比较,并给出部分实验结果,进而引出下一步工作.

1 基本概念

由 Alur 与 Dill 提出的时间自动机[1]是对一般的有限状态自动机的扩充,其时钟变量的取值为非负实数.时间自动机的语法定义如下:

定义 **1.1**(时间自动机). X 是时钟的有穷集合.X 上的时间约束集 $C(X)$是由语法定义$\phi::=(x\sim c)|\phi_1\wedge\phi_2|$true 生成的所有公式的集合,其中 $x\in X,\sim\in\{<,\le,\ge,>\},c\in N^+,N^+$为非负整数集.$X$ 上的时间自动机是一个六元组 $A=\langle L,l_0,\Sigma,X,I,E\rangle$,其中

- L 是结点的有穷集合,$l_0\in L$ 是初始结点;
- 映射 $I:L\mapsto C(X)$ 为每个结点 l 赋以一个时间约束,这个约束称为结点的不变式;
- Σ是同步标号的有限集合;
- $E\subseteq L\times C(X)\times\Sigma\times 2^X\times L$ 是迁移的集合.每条迁移(l,ϕ,σ,Y,l')表示在满足约束条件 ϕ 时,通过标号为$\sigma\in\Sigma$的迁移,结点 l 可以迁移到结点 l',同时属于 $Y(Y\subseteq X)$ 的时钟被重置为 0.ϕ 称为迁移的约束条件.

时间自动机的时钟赋值为 $\mu:X\mapsto R^+$,其中 R^+ 为非负实数集.时钟赋值 $\mu+t$ 表示对于所有的时钟变量 $x\in X$,$\mu(x+t)=\mu(x)+t,t\in R^+$.时钟赋值 μ 满足时间约束 ϕ (记为 $\mu\models\phi$),当且仅当在 μ 的赋值下 ϕ 为真.对于 $Y\subseteq X,\mu[Y:=0]$ 表示对每个 $x\in Y$ 的时钟赋值为 0,其余的时钟值保持不变.

时间自动机 $A=\langle L,l_0,\Sigma,X,I,E\rangle$ 的语义为迁移系统 $[[A]]_C=\langle S,s_0,\Sigma\cup R^+,\to\rangle$,其中 $S=L\times U_X$,U_X 为 X 上所有时钟赋值的集合,初始状态 $s_0=(l_0,\mu_0)$,对于所有 $x\in X$,$\mu_0(x)=0$,迁移关系 $\to$ 的定义如下:

定义 **1.2**(迁移关系 $\to$).

- 延迟迁移: $(l,\mu)\xrightarrow{\delta}(l,\mu')$,若 $\mu \models I(l)$ 且 $\mu' \models I(l)$,其中$\mu' = \mu + \delta, \delta \in R^+$;
- 离散迁移: $(l,\mu)\xrightarrow{\sigma}(l',\mu')$,若 $(l,\phi,\sigma,Y,l') \in E, \mu \models \phi, \mu' \models I(l')$,其中$\mu' = \mu[Y := 0]$.

直观上解释,延迟迁移表示时间在某个结点中的推移,而离散迁移表示在约束条件满足时结点之间的转换.

定义 **1.3**(时间自动机的可达状态).

对于时间自动机 A 的某个状态 $s=(l,\mu)$,其中 $l \in L, \mu \in U_X$.称 s 是可达的,若 A 的迁移系统中存在一个有穷状态序列 $s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} ... \xrightarrow{\alpha_{k-1}} s_k$ $(\alpha_i \in \Sigma \cup R^+, i \in \{0,\ldots,k-1\})$,使得

- s_0 是初始状态;
- $s = s_k$;
- 对于任意 $i \in \{0,\ldots,k-1\}$, $(l_i,\mu_i)\xrightarrow{\alpha_i}(l_{i+1},\mu_{i+1})$,要么是 A 的一条离散迁移,要么是 A 的一条延迟迁移.

对于结点 l 及时间约束 $\phi \in C(X)$,如果存在可达状态 (l,μ),使得 $\mu \models \phi$,则称 (l,ϕ) 在时间自动机 A 的连续语义下是可达的.

2 FPTA 的相关概念

2.1 FPTA的基本概念

FPTA 的特点在于它能够区分时钟变化的相对次序,而这是通过引入序来实现的.因此,本节介绍 FPTA 的基本概念、语义以及可达性的定义,并举例解释 FPTA 的语义.

FPTA 的语法定义与定义 1.1 相同.为了解释 FPTA 的语义,下面首先给出序的定义.

定义 **2.1**(序). 时钟集合 X 上的一个序 o 是指从 X 到 N^+ 的一个映射, O_X 是 X 上所有序的集合.对于时钟变量 $x_1, x_2 \in X$,若 $o(x_1) < o(x_2)$,则称 x_1 的序小于 x_2 的序,即时钟 x_1 的变化比 x_2 的变化早发生.

定义 **2.2**(序标准化).

1. 设 o 为 X 上的一个序,若存在 $k \in [0,|X|-1]$,使得 $\{o(x) \mid x \in X\} = \{0,1,...,k-1,k\}$,则称 o 为标准序.
2. 设 o_1, o_2 为 X 上的两个序,其中 $o_1 = (o_{11}, o_{12}, ..., o_{1n}), o_2 = (o_{21}, o_{22}, ..., o_{2n}), n = |X|$,称 o_1 与 o_2 等价,若对于任意的 $i, j \le n$,有 $o_{1i} \le o_{1j} \Leftrightarrow o_{2i} \le o_{2j}$ 成立.
3. 若 o_1 等价于 o_2,且 o_2 是标准序,则称 o_2 是 o_1 的标准化,记为 o_2=*norm*(o_1).

例如,对于某个序 o_1=(1,0,3,4,3),则标准化后的序为 *norm*(o_1)=(1,0,2,3,2).

FPTA 中的时钟赋值为 $v: X \mapsto N^+$,定义 V_X 为 X 上所有时钟赋值的集合.给定有限精度时间自动机 $A = \langle L, l_0, \Sigma, X, I, E \rangle$,$A$ 中的一个状态 s 是一个三元组$(l,(v,o))$,其中 l 为状态 s 所在的结点,v 为一个时钟赋值,o 为一个标准序,(v,o)称为 s 的时钟状态.对于一个状态 $s=(l,(v,o))$ 及约束 ϕ,如果 $v \models \phi$,则称状态 s 满足(l,ϕ),记作 $s \models (l,\phi)$.

A 的迁移系统为$[[A]]_{FP}=\langle S, s_0, \Sigma, \rightarrow \rangle$,其中 $S=L \times (V_X \times O_X)$,初始状态为 $s_0=(l_0,(v_0,o_0))$,对于所有的 $x \in X$, $v_0(x)=0$ 且 $o_0(x)=0$.由离散迁移和延迟迁移构成的迁移关系 $\rightarrow$ 的定义如下:

定义 **2.3**(迁移关系 $\rightarrow$).

- 离散迁移: $(l_1,(v_1,o_1))\xrightarrow{\sigma}(l_2,(v_2,o_2))$,若存在 $(l_1,\phi,\sigma,Y,l_2) \in E$,使得 $v_1 \models \phi, v_2 \models I(l_2)$,且 $(v_2,o_2) = reset((v_1,o_1),Y)$,其中 $reset((v,o),Y) = (v[Y:=0], norm((o+1)[Y:=0]))$,

$$(o+1)[Y:=0](x) = \begin{cases} 0 & x \in Y \\ o(x)+1 & x \in X - Y \end{cases}.$$

- 延迟迁移: $(l,(v,o))\xrightarrow{\varepsilon}(l,(v,o)\oplus k)$ (或简记为 $(l,(v,o)) \rightarrow (l,(v,o)\oplus k)$),若 $k \in N^+$ 且 $(v,o)\oplus k \models I(l)$, 其中 $(v,o)\oplus k$ 定义为 $((v,o)\oplus k)(x) = (v(x)+(o(x)+k) \text{ div } m, (o(x)+k) \bmod m)$,其中 $m=1+\max\{o(x) \mid x \in X\}$.

设 $A = \langle L, l_0, \Sigma, X, I, E \rangle$ 为一个 FPTA,则从初始状态开始由离散迁移或延迟迁移构成的无穷序列:

$$(l_0,(v_0,o_0))\xrightarrow{\alpha_0}(l_1,(v_1,o_1))\xrightarrow{\alpha_1}(l_2,(v_2,o_2))\xrightarrow{\alpha_2}\ldots,$$

是 A 的一个状态迁移序列,其中 $\alpha_i\in\Sigma\cup\{\varepsilon\}$.

为了更清楚地描述 FPTA 的迁移语义,下面以图 1 中的模型为例,给出它的一条迁移序列.为叙述方便,我们将图 1 中从 l_0 迁移到 l_1 的离散迁移称为 t_1;从 l_1 迁移到 l_0 的离散迁移称为 t_2.

下面是图 1 的一条状态迁移序列(为了简化时钟状态的表示,这里将时钟变量的值和序按照所有时钟的赋值、所有时钟序的赋值方式排列.那么 0000 中前两位表示时钟 x,y 的值,后两位表示它们的序,后面也使用这种表示方法):

$(l_0,0000)\to(l_0,1100)\to(l_0,2200)\xrightarrow{a}(l_1,2010)\to(l_1,3001)\to(l_1,3110)\to(l_1,4101)\to(l_1,4210)\to(l_1,5201)\to(l_1,5310)\xrightarrow{b}(l_0,0301)\to\ldots$

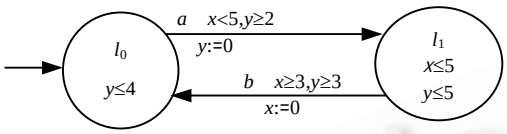

Fig.1 A simple timed automaton

图 1 一个简单的时间自动机

在没有离散迁移发生且序相同时,每个时钟值在经历一个时间单位后加 1,序保持不变.状态$(l_0,2200)$生成后,满足离散迁移 t_1 的约束条件,t_1 发生后生成状态$(l_1,2010)$.因为它不满足 t_2 的约束条件,在满足 l_1 不变式的前提下进行延迟迁移生成新的状态,在状态$(l_1,5310)$生成后满足 t_2 的约束条件,进行跳转生成新的后续状态.

由于如安全性、有界活性等[20]这样的性质可以表示成可达性,许多实时系统的模型检测工具主要对可达性进行分析和验证.

定义 **2.4(FPTA** 的可达状态). 设 A 为一个 FPTA,若对于它的某个状态$(l,(v,o))$,存在一条从初始状态出发的有穷状态序列$(l_0,(v_0,o_0))\xrightarrow{\alpha_0}(l_1,(v_1,o_1))\xrightarrow{\alpha_1}\ldots\xrightarrow{\alpha_{n-1}}(l_n,(v_n,o_n))$,使得 $(l_n,(v_n,o_n))=(l,(v,o))$,则称$(l,(v,o))$是可达的.对于结点 l 和时间约束 $\phi\in C(X)$,若存在可达状态 $(l,(v,o))$ 使得 $v\models\phi$,则称 (l,ϕ) 是可达的.

2.2 FPTA与TA在可达性方面的关系

在一定的时间约束下,TA 的可达性分析可以简化为 FPTA 中的可达性分析.

定义 **2.5**(左闭右开时间自动机). 称时间约束是左闭右开的,若 ϕ 可由语法 $\phi::=(x\ge c)|x<c|\phi_1\wedge\phi_2|\text{true}$ 生成,其中,$x\in X,c\in N^+$.称时间自动机 A 是左闭右开的,若 A 中出现的所有时间约束都是左闭右开的.

定理 **2.1**. 设 A 是左闭右开的时间自动机,ϕ 为左闭右开的时间约束,(l,ϕ) 在 A 的连续语义中是可达的当且仅当 (l,ϕ) 在 A 的有限精度语义中是可达的.

在 FPTA 中的两个时钟状态(v,o)和(v',o')称为等价的(记为$(v,o)\cong(v',o')$),如果它们满足下面的关系:

- 对于所有的 $x\in X$,或者 $v(x)=v'(x)$或者$(v(x)\ge c_x+1)\wedge(v'(x)\ge c_x+1)$($c_x$ 为 A 中与时钟变量 x 有关的约束中出 现的常数的最大值).
- 对于所有的 $x\in X$,$o(x)=o'(x)$.

等价类的划分可以保证状态空间的有限性,即时钟增长到大于或等于 c_x+1 后的赋值与 c_x+1 是相同的,不再无限增长.需要说明的是,FPTA 根据时钟值及序划分状态的等价类与 TA 如文献[1,9]中根据时钟小数部分划分等价类的方法不同.文献[6,7]提出的模型与 FPTA 极为相象,即也采用整数记录时钟值,整数标识时钟变化的相对次序.但它们基于文献[1,9]中的方法来划分等价类,为连续时间模型.

3 基于 FPTA 的可达性分析

一般而言,可达性分析算法基于图的搜索过程,其顶点为符号化的状态.在状态空间的搜索过程中存在前向搜索和后向搜索两种方法.FPTA 工具的可达性检测目前是对状态空间进行前向搜索.从初始状态开始,在展开状态空间的同时,检测新生成的状态是否满足性质.若满足,则搜索过程结束.否则,继续对新生成的状态展开所有的后继,直到不再生成新的状态为止,整个过程结束.

本节根据状态空间生成过程的特点,首先分析一段由连续的延迟迁移构成的状态序列(简称为延迟序列)的特点和表示形式,接着,根据延迟序列经过离散迁移而发生结点的转换,并重置时钟后生成的新状态之间的关

系,提出一种在 FPTA 中符号化表示状态集的数据结构,然后给出可达性分析算法.

3.1 延迟迁移序列中状态的符号化

基于 FPTA 模型,在前向搜索过程中,状态空间的生成首先从初始状态$(l_0,(v_0,o_0))$开始,在满足 $I(l_0)$的前提下,可以由延迟迁移生成以 l_0 为结点的状态序列$(l_0,(v_0,o_0)\oplus 0)\to(l_0,(v_0,o_0)\oplus 1)\to\ldots\to(l_0,(v_0,o_0)\oplus k)$,其中$(v_0,o_0)\oplus i|=I(l_0),i\in\{1,\ldots,k\}$.下面给出符号化表示该序列的定义.

定义 **3.1**(延迟迁移序列中状态的符号化表示).

- $(l,(v,o),k)$表示状态集合$\{(l,(v,o)\oplus 0),(l,(v,o)\oplus 1),\ldots,(l,(v,o)\oplus(k-1))\}$,其中 $k\in N^{>0}$($N^{>0}$ 表示正整数集).
- $(l,(v,o),\infty)$表示状态集合$\{(l,(v,o)\oplus 0),(l,(v,o)\oplus 1),\ldots\}$.根据等价类的划分方法,对于任意 $x\in X$,大于 c_x+1 的时钟值都被认为是相同的,即使允许时间的无限延迟,形成的延迟序列中状态的数目也还是有限的.

这样,$(l,(v,o),k)$组成了由状态$(l,(v,o))$经过延迟迁移生成的一段延迟序列(delay sequence,简称 DS),其中 $k\in N^{>0}\cup\{\infty\}$.

在由时间延迟构成的状态序列中,若存在部分状态能够满足与所在结点相连的某条离散迁移上的约束条件,则可发生离散迁移.因此有必要判断由延迟迁移生成的哪些状态满足离散迁移的约束条件.

由定义可知,约束条件 ϕ 是 $x\sim c(\sim\in\{<,\le,\ge,>\})$这种形式不等式的交集.其中 $x>c,x\ge c$ 形式的不等式限定了可以进行离散迁移的状态中时钟 x 必须满足的最小值,而含 $x<c,x\le c$ 形式的不等式确定了时钟 x 可取的最大值.若能依次求出满足这两种约束的状态集,就可以得出当前结点由延迟迁移生成的状态中哪些能进行离散迁移.

延迟序列中的时钟状态是有序的.只要找到满足$<,\le$ 约束的时钟状态最大值,则该时钟状态以前的序列都满足这种约束.同样,找到满足$>,\ge$ 约束的时钟状态的最小值,则该时钟状态以后的序列都满足这种约束.

为方便计算,这里首先给出如下约定:设 $c\in N^+$,则 $\infty-c=\infty$,且 $\infty>c$.

定义 **3.2**(受约束后的延迟序列). 若延迟序列为$(l,(v,o),k)$,约束条件为 ϕ .令 $m=\max\{o(x)|x\in X\}+1$.

- 若 ϕ 为 true,则$(l,(v,o),k)|_\phi$ 仍为$(l,(v,o),k)$.
- 若 ϕ 为$(x<c)$,令 $l_c=\min\{(c-v(x))\times m-o(x),k\}$,则受 $x<c$ 约束后的延迟序列$(l,(v,o),k)|_{x<c}$ 为$(l,(v,o),l_c)$.
- 若 ϕ 为$(x\ge c)$,如果 $v(x)<c$,令 $d=(c-v(x))\times m-o(x)$.若 $d<k$,定义$(v',o')=(v,o)\oplus d,g_c=k-d$,则约束后的序列$(l,(v,o),k)|_{x\ge c}$ 为$(l,(v',o'),g_c)$;若 $d\ge k$,$(l,(v,o),k)|_{x\ge c}$ 为空.若 $v(x)\ge c$,则原序列不变.
- 若 $\phi=\phi_1\wedge\phi_2$,则$(l,(v,o),k)|_\phi$ 为$((l,(v,o),k)|_{\phi_1})|_{\phi_2}$.

受 $x\le c,x>c$ 形式约束后的状态序列计算方法与上面的过程类似,不再详述.这样,可以将$(l,(v,o),\infty)|_{I(l)}$定义为由$(l,(v,o))$生成的最大延迟序列.

例 3.1:满足约束条件的状态集的计算.

这里,计算能够进行离散迁移 t_1 的状态集,其约束条件为 $x<5\wedge y\ge 2$.由于 $I(l_0)=y\le 4$,从初始状态开始的延迟迁移形成的延迟序列记为$(l_0,0000,5)$.根据上面的计算方法,首先求出延迟序列$(l_0,0000,5)$受 $x<5$ 约束后的状态集为$(l_0,0000,5)$.它受 $y\ge 2$ 约束后的状态集为$(l_0,2200,3)$.因此属于$(l_0,2200,3)$的状态可以进行离散迁移.

在判断时钟状态是否满足$(v_1,o_1)\in((v,o),k)$时,相当于判断是否存在 k',使得$(v,o)\oplus k'=(v_1,o_1)$,且 $k'<k$,其中$k'=\max\{(v_1(x)-v(x))\times m+o_1(x)-o(x)|x\in X$ 且 $v_1(x)\le c_x\}$.两个序列的交集是否为空可以转化为求是否时钟状态$(v,o)\in((v_1,o_1),k_1)$或$((v_1,o_1)\in((v,o),k)$.

3.2 符号化状态的组织

若只使用 DS 这种结构作为状态空间的符号化表示,状态空间完全展开后符号化状态的数目仍较大.若根据延迟序列之间的某种联系,将它们组织成粒度更粗的符号化表示,则可以有效地减少符号化状态的个数.

在图 1 的模型中,从初始状态出发的延迟序列中可以进行离散迁移 t_1 的状态集为$(l_0,2200,3)$.这些状态在经过 t_1 后生成的新状态集合为$\{(l_1,2010),(l_1,3010),(l_1,4010)\}$.从直观上比较,新生成的每个状态中 y 的值都为 0,而 x 的值递增变化.因为这些状态是由一段连续的延迟序列经过时钟重置得到的,可以将其时钟状态记为

$((v,o),k,Y)=\{(v_r,o_r)|(v_r,o_r)=reset((v,o)\oplus i,Y),0\le i<k\}$,其中 Y 为对应离散迁移中重置时钟的集合.但这样要对每个原有状态进行时钟的重置并作序的标准化操作才能生成新状态,比较费时.而符号化方法的目标是使操作简单,且尽量减少表示状态集合的符号化状态的数量.

根据分析,由离散迁移生成的一个状态,可以算出原延迟迁移形成的序列经过离散迁移后生成的其他状态.设某个经过离散迁移的时钟状态为(v_r,o_r),其中重置时钟集合为 Y,则(v_r,o_r)后第 i 个状态的时钟状态为$(v_{ir},o_{ir})=((v_r,o_r)\otimes i)\backslash Y$,其中

$$(((v,o)\otimes i)\backslash Y)(x)=\begin{cases}(v(x)+(o(x)+i-1)\ div\ m',(o(x)+i-1)\bmod m'+1), & 若x\in X-Y\\(0,0), & 若x\in Y\\(v(x),o(x)), & 若Y=\varnothing\end{cases},\ m'=\max\{o(x)|x\in X\}.$$

下面给出 FPTA 的符号化状态定义:

定义 **3.3**(延迟序列的系列化). 在 FPTA 中,给定状态$(l,(v,o)),\theta\in N^{>0},Y\subseteq X$,则把$(l,(v,o),\theta,Y)$作为状态集合$\{(l,(v',o'))|(v',o')=((v,o)\otimes i)\backslash Y$,其中 $0\le i<\theta\}$的符号化表示形式.我们将这种表示称为延迟序列的系列化(series of delay sequences,简称 SDS).

例如,设在某时间自动机中, $X=\{x,y,z\}$, $l\in L$,它的某个状态集合为$\{(l,(230120)),(l,(240210)),(l,(340120)),(l,(350210)),(l,(450120))\}$,其中时钟 z 被重置,则这个状态集合可以表示成$(l,230120,5,\{z\})$.

这里的符号化状态是由延迟序列经过离散迁移生成的,它所表示的集合中的状态是后继延迟序列的起点,称为起始状态.θ 表示集合的大小,即原延迟序列经过离散迁移后生成的非重复起始状态的个数.θ 可以通过原延迟序列及重置时钟集合计算得出,由于篇幅有限,这里略去计算过程.

由离散迁移生成的所有状态不一定都满足新结点的不变式,故需计算出满足新结点不变式的符号化状态.

定义 **3.4**(受不变式约束后的符号化状态$(\boldsymbol{l'},(\boldsymbol{v},\boldsymbol{o}),\boldsymbol{\theta},\boldsymbol{Y})\|_{I(l')}$).

设经过某离散迁移后生成的符号化状态为$(l',(v,o),\theta,Y)$,不妨设结点 l'的不变式 $I(l')=\phi$ 为 $x<c$ 与 $x\ge c$ 这种不等式的交集.

1. 若 $Y=\varnothing$.设$(l',(v,o),\infty)|_\phi=(l',(v',o'),g)$,则$(l',(v,o),\theta,Y))\|_\phi$ 为$(l',(v',o'),1,\varnothing)$.

2. 若 $Y\ne\varnothing$.

- 若 ϕ 为 true,$(l',(v,o),\theta,Y)\|_\phi$ 仍为$(l',(v,o),\theta,Y)$.
- 若 ϕ 为$(x<c)$且 $x\notin Y$,令$\theta'=\min\{\theta,(c-v(x))\times m'-o(x)+1\}$,则$(l',(v,o),\theta,Y)\|_{x<c}$ 为$(l',(v,o),\theta',Y)$.
- 若 ϕ 为$(x\ge c)$且 $x\notin Y$,令 $d=(c-(v(x))\times m'+1-o(x)$,若 $0\le d<\theta'$,定义$(v',o')=((v,o)\otimes d)\backslash Y,\theta'''=\theta'-d$,则$(l',(v,o),\theta',Y)\|_{x\ge c}$ 为$(l',(v',o'),\theta''',Y)$;若 $d\ge\theta'$,则$(l',(v,o),\theta',Y)\|_{x\ge c}$ 为空.
- 若 $\phi=\phi_1\wedge\phi_2$,则$(l',(v,o),\theta,Y)\|_\phi$ 为$((l',(v,o),\theta,Y)\|_{\phi_1})\|_{\phi_2}$.

根据以上定义,由$(l_0,2200,3)$表示的状态集进行离散迁移 t_1 后,生成的符号化状态可以表示为$(l_1,2010,3,\{y\})$.

因此,从直观上解释,在满足离散迁移 $e=(l,\phi,\sigma,Y,l')$的约束条件的情况下,若干延迟迁移形成的状态集经过该离散迁移生成的满足 $I(l')$的起始状态形成了状态空间的一个符号化状态.这个符号化状态所包含的起始状态在满足 $I(l')$的情况下,生成的新的延迟序列又可以在满足下一离散迁移约束的条件下生成新的起始状态,构成新的符号化状态.由于符号化状态的含义是若干起始状态的集合,所以整个状态空间可以看作是所有符号化状态中的所有状态生成的最大延迟序列中所有状态的并集.

下面还是以图 1 的模型为例说明符号化状态的产生过程.

例 3.2:符号化状态的生成.

图 1 模型中的符号化初始状态为 $s_0=(l_0,0000,1,\varnothing)$.那么它只含有一条由初始状态生成的延迟序列$(l_0,0000,5)$,其中$(l_0,2200,3)$可以进行离散迁移.经过离散迁移 t_1 后生成的符号化状态为 $s_1=(l_1,2010,3,\{y\})$.s_1 包含的 3 个延迟序列的起始状态可以在满足 l_1 的不变式时进行延迟迁移,由于这些满足 l_1 约束的延迟序列中只有$(l_1,2010)$通过延迟迁移生成的状态$(l_1,5310)$满足 t_2 的约束条件,可以跳转生成 $s_2=(l_0,0301,1,\{x\})$…

在以这种数据结构表示的状态集合中,由新的起始状态根据延迟迁移生成的多个状态序列之间也具有一

定的规律.例如,在计算出由$(l,(v,o))$按延迟迁移生成的状态序列中可以进行离散迁移的状态集后,因为受同一约束条件限制,可以很快求出以$(l,((v,o)\otimes i)\backslash Y)$生成的延迟序列中能够进行同一离散迁移的状态集,这样加快了后继状态的生成.

此外,用这种数据结构表示的延迟序列集之间的包含关系也能很快判断出来,做法与判断延迟序列的相互关系类似,但要考虑对重置时钟的处理.

3.3 可达性分析算法

这一小节主要介绍基于 FPTA 的符号化状态表示形式,根据可达性分析展开状态空间的主要算法,并给出图 1 的模型展开整个状态空间后的符号化状态.

在状态空间的前向搜索过程中,这里使用两个链表 W 和 P 分别表示待展开的状态集及已经展开的状态集.W 中的每个数据表示未展开的若干延迟序列的起始状态集,而 P 的作用是记录已经生成的符号化状态,并避免状态的重复生成.根据 W中存取方式(先入先出,先入后出)的不同,搜索过程可分为广度优先搜索和深度优先搜索两种方式.根据 SDS 这种数据结构的可达性分析过程为算法 1.

算法 1. Reachability Analysis.

```
SDS: wa;
List of state: Succ;
List of SDS: P,W=∅;
wa:=(l0,(v0,o0),1,∅);
W:={wa};
WHILE{W≠∅}
    get wa from W;
    IF Satisfy(wa,l,φ)
        return(true)
    ELSE
        IF(wa∉P)
            Add(wa,P);
            Succ:=Unfolding(wa);
            CreateSuccessive(Succ)
        ENDIF
    ENDIF
ENDWHILE
```

算法 2. CreateSuccessive(Succ).

```
SDS: wa;
  List of DS: ds;
  SDS: wa;
  FORALL(discrete transitions δ from the current location)
      ds0:=get the δ-enabled state set from Succ0;
      get xl,xu∈X which are used to compute ds0;
      FOR(i:=1 to |Succ|−1)
          dsi:=GetDelaySet(ds0,xl,xu,Succi)
      ENDFOR
      FOR(i:=0 to |Succ|−1)
          wa:=GenerateNext(dsi,δ);
          IF(wa∉W∪P)
              W:=W+wa
          ENDIF
      ENDFOR
  ENDFOR
```

算法 1 中,Satisfy 返回 true,若 wa 中的某个起始状态生成的最大延迟序列中存在状态 s,使得 $s|=(l,\phi)$.$Succ$ 存放从 W中取出的符号化状态展开后所有起始状态的集合.这些状态是通过 Unfolding 这个函数展开的,它根据 $(v,o),k,Y$ 的信息计算前一延迟序列经过离散迁移后生成的满足当前结点不变式的所有状态.CreateSuccessive 根据 $Succ$ 的每个起始状态,首先计算满足迁移约束条件的状态集,从而生成新的符号化状态.若新生成的符号化状态与 P 中的符号化状态没有相等或包含关系,则放到 W 中等待展开.

为了更清楚地说明 FPTA 中状态空间的生成过程,还是以图 1 中给出的模型为例进行说明.表 1 列出了该模型在状态空间生成过程中,P 及 W 的变化情况.

Table 1 Symbolic states in W and P

表 1 W 及 P 中的符号化状态

Step	W	P
1	$\{(l_0,0000,1,\varnothing)\}$	$\{\}$
2	$\{(l_1,2010,3,\{y\})\}$	$\{(l_0,0000,1,\varnothing)\}$
3	$\{(l_0,0301,1,\{x\})\}$	$\{(l_0,0000,1,\varnothing),(l_1,2010,3,\{y\})\}$
4	$\{(l_1,0010,2,\{y\})\}$	$\{(l_0,0000,1,\varnothing),\{(l_1,2010,3,\{y\}),(l_0,0301,1,\{x\})\}$
5	$\{(l_0,0401,1,\{x\})\}$	$\{(l_0,0000,1,\varnothing),\{(l_1,2010,3,\{y\}),(l_0,0301,2,\{x\}),(l_1,0010,2,\{y\})\}$
6	$\{\}$	$\{(l_0,0000,1,\varnothing),\{(l_1,2010,3,\{y\}),(l_0,0301,2,\{x\}),(l_1,0010,2,\{y\})\}$

说明:在第 5 步中,由$(l_1,0010,2,\{y\})$生成的后继实际为$(l_0,0301,2,\{x\})$,但它包含 P 中已经存在的符号化状态

$(l_0,0301,1,\{x\})$.两者通过比较,得出新的未展开序列.所以在对 P 中的数据进行修改后,把未展开过的$(l_0,0401,1,\{x\})$加入 W 中等待展开.最后第 6 步中生成的后继已经存在于 P 中,不再存入 W,所以 W 为空,整个过程终止.

在 Uppaal 中,对应模型状态空间完全展开得出的 zone 为(不含死锁)

$$l_0: x\in[0,4], y\in[0,4], y=x,$$
$$l_1: x\in[2,5], y\in[0,3], x-y\in[2,4],$$
$$l_0: x\in[0,1], y\in[3,4], y-x=3,$$
$$l_1: x\in[0,5], y\in[0,5], y-x\in[-1,0],$$
$$l_0: x\in[0,1], y\in[3,4], y-x\in[3,4],$$
$$l_1: x\in[3,5], y\in[3,5], y-x\in[-1,0].$$

其中第 3 步中的 zone 包含在第 5 步生成的 zone 中,而第 6 步的 zone 被第 4 步的 zone 包含.后面形成的 zone 与前面的重复.也就是说,图 1 的模型在 Uppaal 中可以用 4 个 zone 表示整个状态空间.

将 FPTA 最终形成的符号化状态与 Uppaal 最终生成的 zone 进行比较:两者的符号化状态个数相同,表示每个符号化状态所用的存储空间是不同的.仅对时钟状态比较,设时钟个数为 n,每个时钟状态用一个字节表示:FPTA 用 $2n$ 个字节表示时钟值及序,而 zone 使用$(n+1)^2$ 个字节表示时钟状态.虽然 FPTA 中的符号化状态多了重置时钟的信息,但在最坏情况下共用 $3n$ 个字节表示一个符号化状态,小于 zone 占用的存储空间.

4 相关工作与实验

4.1 相关工作

就模型而言,虽然 FPTA 的时钟变量取整数值,但它与离散时间自动机模型并不完全相同.离散时间自动机不关心时钟在单位时间内的变化次序.文献[21]中的两种离散化方法虽然能够区分发生在不同时间片中的时钟变化,但在同一时间片中的变化是无法区分的.即固定的时间片划分无法区分所有的时钟变化.文献[22]中提到的离散时间自动机,以一个全局时钟为参考,在时钟刚发生变化的一个时间单位内其先后顺序是可以区分的,但只要经过全局时钟的整点时刻,时钟值都增加后,就无法再区分若干时钟变化的先后顺序.在文献[23]采用的离散时间模型中,它的语义迁移与 FPTA 不同点在于:在离散迁移中没有任何时钟被重置时,各个时钟变量都增加一个时钟单位;若在迁移过程中至少一个时钟发生重置,则其余的时钟值不变,仅重置的时钟变为 0.

与目前存在的数据结构比较,DBM 对时钟变量的最大值不敏感,但只能描述非凹的 zone,同时不利于数据的共享.类似 BDD 的数据结构,如 IDD[24],CDD[18],RED[6,7]和 CRD[8]等虽然可以表示非凸的 zone,并且有利于数据共享,但变量之间的顺序对它们占用的存储空间影响较大.将连续时间离散化以后用 BDD 的结构,如 NDD[14],及 Rabbit 中采用的 BDD 方法[8]对时钟变量、共享变量等进行编码表示,整个状态空间虽然可以达到各种整型变量的统一表示,但对时间约束中出现的最大常量比较敏感.SDS 这种数据结构相当于把状态空间中存在的所有延迟序列进行枚举.最后 P 中存放的是状态空间中具有的所有延迟序列的起始状态.这与 zone 从约束角度表示状态空间的做法是不同的.与 DBM 比较,SDS 描述若干条延迟序列占用的空间要小,各种计算的时间复杂度一般都是线性的,比 DBM 中的操作复杂度要低[25].不足之处是,一个 SDS 中表示的状态集可能要少于一个 DBM 中表示的状态集.与类似 BDD 的数据结构相比,它避免了各进程、变量之间顺序排列不当引起的空间复杂度指数级的增长,也无须考虑向标准形式的转化.不足是对时钟变量的最大取值比较敏感(最大值增加时,状态空间可能呈指数级增长),在数据的共享方面仍要进一步加以改进.

4.2 实 验

以 FPTA 为基础的模型检测工具原型(可从 http://lcs.ios.ac.cn/~xyz/FPTA 下载)首先对使用 xml 脚本语言描述的系统模型进行编译,再根据编译后的模型文件进行状态空间的前向搜索.本文使用 Fischer 的互斥协议模型[26](如图 2 所示),在状态空间完全展开的情况下,与 Uppaal 3.4.6 的实验结果进行比较.实验的硬件环境为 Intel

P4 2.60GHz 处理器,512MB 内存.FPTA 的相关数据是在 Windows 环境下得出的,而 Uppaal 在 Windows 中不易得出精确的运行时间,其实验数据是在 Linux 下得出的.表 2 中列出 a,b 取不同值时两者生成整个状态空间耗费的时间(a=4,b=2 时会出现多个进程同时共享临界资源的情况).“−”表示运行时间大于 600s.

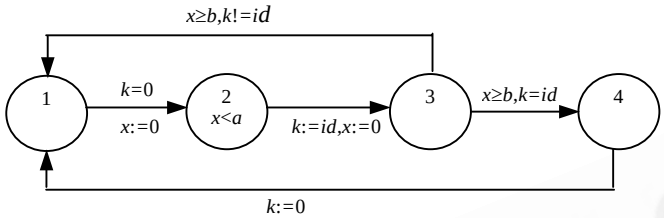

Fig.2 Fischer’s mutual exclusive protocol

图 2 Fischer 互斥协议模型

Table 2 Results with Fischer’s mutual exclusive protocol

表 **2** Fischer 互斥协议实验数据

No.	Uppaal (a=2,b=4)		FPTA (a=2,b=4)		Uppaal (a=4,b=2)		FPTA (a=4,b=2)	
	Depth (s)	Breadth (s)	Depth (s)	Breadth (s)	Depth (s)	Breadth (s)	Depth (s)	Breadth (s)
2	0.11	0.12	0.02	0	0.12	0.13	0.02	0.02
3	0.11	0.12	0.05	0.06	0.12	0.11	0.11	0.13
4	0.18	0.14	0.61	0.61	0.24	0.17	12.81	9.312
5	1.00	1.13	36.97	27.13	8.59	2.32	−	−
6	137.09	41.09		−	−	197.90	−	−

目前,该原型除采用文献[13]中的时钟归约技术减少状态空间以外,还没有采用其他方法对模型进行简化,也未使用任何数据压缩技术,从这个角度来说,在性能上与使用了很多方法提高模型检测效率的 Uppaal[27]相比还有一定差距.通过对实验结果的横向比较可以得出,FPTA 的工具在进程个数较少时,运行速度与 Uppaal 非常接近.但随着进程数量的增加,FPTA 的时间耗费迅速增加.因此,FPTA 在时钟变量比较少时,状态空间的展开速度较快.而随着时钟变量和进程数量的增加,FPTA 的性能下降得较快.因此需要在数据共享、降低对时钟变量的敏感度方面做进一步的工作.

虽然目前已经有很多种方法表示实时系统模型检测过程中产生的状态集合,但在如何将时钟变量与其他变量统一符号化表示方面仍未有很好的方法.SDS 这种数据结构只是众多结构中针对 FPTA 模型特点的一种尝试.因此,我们的下一步工作,首先是完善目前正在开发的 FPTA 模型检测工具,进行大量的实例研究,与现有的其他相关工具进行比较;其次,改善当前的数据结构,还要尝试其他便于数据间共享的数据结构,减小对时间约束最大值的敏感程度.

References:

[1] Alur R, Dill, DL. A theory of timed automata. Theoretical Computer Science, 1994,126(2):183−235.

[2] Larsen KG, Pettersson P, Wang Y. UPPAAL in a nutshell. Int’l Journal on Software Tools for Technology Transfer, 1997,1(1−2): 134−152.

[3] Daws C, Olivero A, Tripakis S, Yovine S. The tool KRONOS. In: Hybrid Systems III. LNCS 1066, New Brunswick: Springer-Verlag, 1996. 208−219.

[4] Bozga M, Daws C, Maler O, Olivero A, Tripakis S, Yovine S. Kronos: A model-checking tool for real-time systems. In: Hu AJ, Vardi MY, eds. CAV. London: Springer-Verlag, 1998. 298−302.

[5] Wang F. Efficient data structure for fully symbolic verification of real-time software systems. In: TACAS. LNCS 1785, London: Springer-Verlag, 2000. 157−171.

[6] Wang F. Region encoding diagram for fully symbolic verification of real-time systems. In: COMPSAC. Taipei: IEEE Computer Society, 2000. 509−515.

[7] Wang F. Efficient verification of timed automata with BDD-like data-structures. In: VMCAI. London: Springer-Verlag, 2003. 189−205.

[8] Beyer D, Lewerentz C, Noack A. Rabbit: A tool for BDD-based verification of real-time systems. In: CAV. LNCS 2725, London: Springer-Verlag, 2003. 122−125.
[9] Katoen JP. Concepts, Algorithms and Tools for Model Checking. Erlangen-Nurnberg: Friedrich-Alexander University, 1999.
[10] Bosnacki D, Dams D, Holenderski L. A heuristic for symmetry reductions with scalarsets. In: FME. LNCS 2021, London: Springer-Verlag, 2001. 518−533.
[11] Hendriks M, Behrmann G, Larsen KG, Vaandrager F. Adding symmetry reduction to uppaal. In: Larsen KG, Niebert P, eds. FORMATS. London: Springer-Verlag, 2003. 46−59.
[12] Bengtsson J, Jonsson B, Lilius J, Wang Y. Partial order reductions for timed system. In: Sangiorgi D, de Simone R, eds. CONCUR. London: Springer-Verlag, 1998. 485−500.
[13] Daws C, Yovine S. Reducing the number of clock variables of timed automata. In: IEEE RTSS. IEEE Computer Society, 1996. 208−219.
[14] Asarin E, Bozga M, Kerbrat A, Maler O, Pnueli A, Rasse A. Data-Structures for the verification of timed automata. In: Proc. of the Int'l Workshop on Hybrid and Real-Time Systems. LNCS 1201, London: Springer-Verlag, 1997. 346−360.
[15] Alur R, Henzinger TA. A really temproal logic. In: IEEE FOCS. IEEE Computer Society, 1989. 164−169.
[16] Wang F. Formal verification of timed systems: A survery and perspective. Proc. of the IEEE, 2004,92(8):1283−1307.
[17] Bryant R. Graph-Based algorithms for boolean function manipulation. IEEE Trans. on Computers, 1986,35(8):677−691.
[18] Behrmann G, Larsen KG, Weise C, Wang Y, Pearson J. Efficient timed reachability analysis using clock difference diagrams. In: CAV. LNCS 1633, London: Springer-Verlag, 1999. 341−353.
[19] Bozga M, Maler O, Pnueli A, Yovine S. Some progress in the symbolic verification of timed automata. In: CAV. LNCS 1254, London: Springer-Verlag, 1997. 179−190.
[20] Berard B, Bidoit M, Finkel A, Laroussinie F, Petit A, Petrucci L, Schnoebelen P. Systems and Software Verification: Model Checking Techniques and Tools. Springer-Verlag, 2001.
[21] Göllü A, Puri A, Varaiya P. Discretization of timed automata. In: Proc. of the 33rd IEEE Conf., 1994,1(14-16):957−958.
[22] Raskin JF, Schoebbens P. Real-Time logics: Fictitious clock as an abstraction of dense time. In: Tools and Algorithms for the Construction and Analysis of Systems. LNCS 1217, London: Springer-Verlag, 1997. 165−182.
[23] Dang Z, Ibarra OH, Bultan T, Kemmerer RA, Su J. Binary reachability analysis of discrete pushdown timed automata. In: CAV. LNCS 1855, London: Springer-Verlag, 2000. 69−84.
[24] Strehl K, Thiele L. Symbolic model checking of process networks using interval diagram techniques. In: ICCAD. New York: ACM Press, 1998. 686−692.
[25] Bengtsson J, Wang Y. Timed automata: Semantics, algorithms and tools. Technical Report, UNU-IIST No. 316, 2004.
[26] Lamport L. A fast mutual exclusion algorithm. ACM Trans. on Computer Systems, 1987,5(1):1−11.
[27] Gerd B, Johan B, Alexandre D, Larsen KG, Paul P, Wang Y. UPPAAL implementation secrets. In: FTRTFT. LNCS 2469, London: Springer-Verlag, 2002. 3−22.

晏荣杰(1977－),女,河北保定人,博士生,主要研究领域为形式化方法,时序逻辑,实时系统的模型检测方法.

刘春明(1979－),男,硕士生,主要研究领域为实时系统的模型检测.

李广元(1962－),男,博士,副研究员,CCF高级会员,主要研究领域为时序逻辑,实时及混成系统的建模,形式化验证及模型检测.

唐稚松(1925－),男,研究员,博士生导师,中国科学院院士,主要研究领域为时序逻辑,程序验证,数理逻辑,自动机理论及软件工程.

徐雨波(1980－)男,硕士生,主要研究领域为实时系统的模型检测.