软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2017,28(5):1080−1090 [doi: 10.13328/j.cnki.jos.005209]

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

异步多进程时间自动机的可覆盖性问题*

刘 立, 李国强

(上海交通大学 软件学院,上海 200240)

通讯作者: 李国强, E-mail: li.g@sjtu.edu.cn

摘 要: 已有的实时系统模型无法动态创建新进程.为此,基于时间自动机模型,提出了异步多进程时间自动机模型,将每个进程抽象为进程时间自动机,其部分状态能够触发新进程.考虑到队列会导致模型图灵完备,进程都被缓存在集合中,但仍可建模许多实时系统.通过将其编码到可读边时间 Petri 网,证明了该模型的可覆盖性问题可判定.

关键词: 实时;异步多进程时间自动机;时间自动机;可读边时间 Petri 网;可覆盖性

中图法分类号: TP301

中文引用格式:刘立,李国强.异步多进程时间自动机的可覆盖性问题.软件学报,2017,28(5):1080−1090. http://www.jos.org.cn/1000-9825/5209.htm

英文引用格式: Liu L, Li GQ. Coverability problem of asynchronous multi-process timed automata. Ruan Jian Xue Bao/ Journal of Software, 2017,28(5):1080−1090 (in Chinese). http://www.jos.org.cn/1000-9825/5209.htm

Coverability Problem of Asynchronous Multi-Process Timed Automata

LIU Li, LI Guo-Qiang

(School of Software, Shanghai Jiaotong University, Shanghai 200240, China)

Abstract: Existing models for real-time system are not able to create new process at runtime dynamically. This paper proposes a new model named asynchronous multi-process timed automata based on timed automata. Each process is abstracted as a process timed automata, and new process is triggered by parts of the states. Consider that queuing can make the model Turing complete, processes are buffered in a set. However the model is still powerful enough to model many real-time systems. The paper proves that model's coverability is decidable by encoding it to read-arc timed petri net.

Key words: real-time; asynchronous multi-process timed automata; timed automata; read-arc timed petri net; coverability

过去几十年中,实时系统在日常生活、工业生产、航天航空等领域得到了广泛的使用,因此,实时系统的验证也变得越来越重要.时间自动机[1,2]是由 Rajeev 等人提出的一种描述实时系统进程的模型,在有限状态自动机基础上扩展了时钟用以精确刻画时间,其格局可达性可判定.基于该理论模型有许多,如 UPPAAL[3,4]的实时系统验证工具被设计出来并应用于各种实时系统.但在这些系统中,进程都是运行前静态创建的,不允许进程运行时动态创建其他进程.

时间正则任务自动机是一种允许动态创建进程的模型,它扩展了时间自动机,每个状态对应一种任务,变迁后触发新状态对应任务的一个实例.但该模型侧重可调度性分析,每个任务实例有相对截止时间,如果都能在相对截止时间前完成,则称可调度.通过编码至嵌套时间自动机[5],将其可调度性问题转化为嵌套时间自动机的可达性问题,从而证明其可调度性问题是可判定的.

另一类对实时系统建模的模型基于传统的 Petri 网进行了扩展,如时间 Petri 网[6,7],用变迁来记录时间;如边

* 基金项目: 国家自然科学基金(61472240, 61672340, 91318301)
Foundation item: National Natural Science Foundation of China (61472240, 61672340, 91318301)
收稿时间: 2016-07-14; 修改时间: 2016-09-25; 采用时间: 2016-12-07; jos 在线出版时间: 2017-01-20
CNKI 网络优先出版: 2017-01-20 16:06:41, http://www.cnki.net/kcms/detail/11.2560.TP.20170120.1606.017.html

时间 Petri 网[8,9],用令牌来记录时间.其可达性都不可判定,仅有边时间 Petri 网的可覆盖性可判定.相对于时间自动机,它们表达能力更强,但很多性质不可判定,而且不能自然地建模实时进程.一般都是将它们的受限版本规约到时间自动机,利用已有的工具进行验证.

受时间正则任务自动机的启发,本文提出了异步多进程时间自动机模型.在异步多进程时间自动机模型里,每一个进程是一个进程时间自动机实例.在时间自动机的基础上,它的部分状态可触发新的进程,由于只关注可覆盖性本身,忽略所有输入字符信息.当进程时间自动机变迁至触发状态后,根据一个函数将该状态映射到进程时间自动机,触发其实例,即可描述运行时动态创建新进程的行为;进程实例缓存在其进程类型的多重集合中,从所有多重集中不确定地挑选进程激活或者执行,新触发的实例插入到对应集合.因为允许动态创建进程,格局中可能有任意多的进程实例,所以本模型无法通过时间自动机网络(等价于时间自动机)编码,即无法利用已有的工具进行验证.而边时间 Petri 网格局可以描述任意多的进程实例,其可覆盖性可判定.通过将本模型编码到边时间 Petri 网的变种模型可读边时间 Petri 网[10],证明了本模型可覆盖性可判定,为以后设计验证工具打下了理论基础.

通过一个实例来说明如何用异步多进程时间自动机来建模问题,并用可覆盖性来验证某些性质.如图 1 所示,系统有两种进程 $P1$ 和 $P2$,其中,$P1$ 是主进程,当其运行至 p_1 或 p_2 状态时,根据 R 函数触发 $P2$ 进程的实例并缓存在其多重集缓冲区中.$P1$ 有两个时钟 x 和 y,$P2$ 有一个时钟 x,初始值都为 0.一个关于系统的性质是系统可能运行至某种格局,其中,$P1$ 的激活实例处于状态 p_2,其 x 和 y 时钟值分别为 6 和 30;$P2$ 的激活实例处于状态 q_1,其 x 时钟值为 6.$P2$ 的未激活实例中,从触发到现在历时 24 的实例至少有 1 个,从触发到现在历时 6 的实例至少有 2 个.这个性质就可以由系统格局的可覆盖性来描述.系统的格局由各类进程的子格局组成,子格局可以理解为三元组(q,ν,M),分别是激活实例的所处状态和时钟估值函数以及非激活实例的多重集缓冲区.那么上述性质可描述为是否存在一个系统运行时可达的格局覆盖格局 $P1:(p_2,[x\to 6,y\to 30],[]),P2:(q_1,[x\to 6],[24,6,6])$.格局覆盖分两部分定义:对于激活实例部分,要求严格相等,类似于可达性;对于非激活实例部分,要求满足多重集上的$\geqslant$序关系.

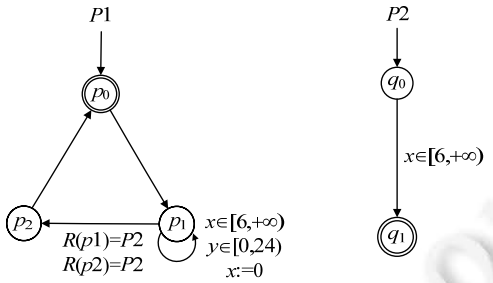

Fig.1 Instance of asynchronous multi-process timed automata

图 1 异步多进程时间自动机实例

本文第 1 节介绍用到的一些基础符号及标签变迁系统、时间自动机、可读边时间 Petri 网的定义.第 2 节介绍本文提出的异步多进程时间自动机的定义和语义.第 3 节介绍异步多进程时间自动机到可读边时间 Petri 网的编码.第 4 节证明异步多进程时间自动机的可覆盖性可判定.第 5 节介绍关于本研究的一些相关工作.第 6 节总结全文,展望未来的工作.

1 预备知识

令$\mathbb{R}^{\geqslant 0}$为非负实数集,$\mathbb{N}$为自然数集.定义$\mathbb{N}^{\omega}=\mathbb{N}\cup\{\omega\}$,$\omega$为正无穷.令$\mathcal{I}$表示区间集合,定义为$\mathcal{I}=\{I|I=(a,b)\vee I=[a,b]\vee I=[a,b)\vee I=(a,b]\wedge a\in\mathbb{N}\wedge b\in\mathbb{N}^{\omega}\}$.给定实数$r\in\mathbb{R}^{\geqslant 0}$及区间$I\in\mathcal{I}$,$r\in I$表示$r$在区间$I$范围内.

令$X=\{x_1,\ldots,x_n\}$为时钟的有限集合,X上的时钟约束定义为ϕ::=True$|x\in I|\phi\wedge\phi$,其中,$r\in X$且$I\in\mathcal{I}$,令ϕ_X为X上

时钟约束的集合.时钟估值函数$v:X\to\mathbb{R}^{\geqslant 0}$给出每个时钟$x\in X$当前记录的时间.$v_0$为初始估值函数,每个时钟记录的时间都为 0.给定时钟估值函数v和时钟约束ϕ,$v\vDash\phi$表示v满足约束ϕ.给定时钟估值函数v和实数 $t\in\mathbb{R}^{\geqslant 0}$,$\forall x\in X,(v+t)(x)=v(x)+t$.给定时钟子集$\lambda\subseteq X$和估值函数$v$,时钟重置函数$v[\lambda]$定义如下:

$$(v[\lambda])(x)=\begin{cases}0, & \text{if } x\in\lambda\\ v(x), & \text{otherwise}\end{cases}.$$

令 $MS[U]$为集合 U 上的多重集合,写作[·].例如,$[u,u,v,v]$表示包含两个 u 和两个 v 的多重集合.给定集合 U 上的多重集合 $Ms_1,Ms_2,Ms_1\oplus Ms_2$ 表示两者的并集,$Ms_1\ominus Ms_2$ 表示两者的差集.

给定 $u\in U$,令 $Ms_1(u)$表示 u 在 Ms_1 中出现的次数.多重集合上的序关系定义为 $Ms_1\geqslant Ms_2$ 当且仅当$\forall u\in U$,$Ms_1(u)\geqslant Ms_2(u)$.给定函数$f:U\to V$,$Ms_1.map(f)=[f(e)|e\in Ms_1]$表示对 Ms_1 中每个元素作用f之后的多重集合.

1.1 标签变迁系统

定义 1. 一个标签变迁系统是一个元组$\langle S,A,\to,r\rangle$,S 是状态的集合,A 是动作的集合,$\to\subseteq S\times A\times S$ 是变迁关系,r 是初始状态.如果$(s,a,s')\in\to$,也写作 $s\xrightarrow{a}s'$,表示系统通过 a 动作从状态 s 变迁到状态 s'.

1.2 时间自动机

时间自动机[1,2]是扩展后的有限状态自动机,配有有限个时钟变量,用以刻画实时系统的行为.

定义 2. 一个时间自动机是一个元组$\langle Q,q_0,F,X,I,\Delta\rangle$,其中,

- Q 是状态的有限集合;
- q_0 是初始状态,$q_0\in Q$;
- F 是终止状态的有限集合,$F\subseteq Q$;
- X 是时钟的有限集合;
- $I:Q\to\phi_X$给每一个状态设置了时钟约束;
- $\Delta\subseteq Q\times\phi_X\times 2^X\times Q$.

$\langle q_1,\phi,\lambda,q_2\rangle\in\Delta$可写为 $q_1\xrightarrow{\phi,\lambda}q_2$.由于只考虑可覆盖性,去除所有输入字符.

定义 3(时间自动机的语义). 一个时间自动机的格局是(q,v)对,其中,q 是当前的状态,v 是所有时钟的估值函数.对应的标签变迁系统(LTS)如下所示.

- 连续变迁: $(q,v)\xrightarrow{t}(q,v+t)$,其中,$t\in\mathbb{R}^{\geqslant 0}$且$(v+t)\vDash I(q)$.
- 离散变迁: $(q_1,v)\longrightarrow(q_2,v[\lambda])$,其中,$q_1\xrightarrow{\phi,\lambda}q_2$,$v\vDash\phi$且$v[\lambda]\vDash I(q_2)$.

初始格局为(q_0,v_0).对于连续变迁,时间流逝了 t,估值函数需要整体加上 t,并且更新后的估值函数要满足当前状态的时钟约束;对于离散变迁,类似于普通有限状态自动机,变迁前的估值函数要满足守卫时钟约束ϕ,更新后的估值函数要满足新状态的时钟约束.

1.3 可读边时间Petri网

定义 4. 一个可读边时间 Petri 网是一个元组 $N=\langle P,T,F,R,times\rangle$,其中,

- P 是库所的有限集合;
- T 是变迁的有限集合,$P\cap T=\varnothing$;
- F 是边的有限集合,$F\subseteq(P\times T)\cup(T\times P)$;
- R 是读边的有限集合,$R\subseteq P\times T\wedge R\subseteq F$;
- $times$ 是边与其时间区间的关联函数,$times:F\to\mathcal{I}$.

定义 5(可读边时间 Petri 网的语义). 一个可读边时间 Petri 网的格局是一个函数 $M:P\to MS(\mathbb{R}^{\geqslant 0})$,$MS(\mathbb{R}^{\geqslant 0})$表示非负实数上的多重集合,对应输入库所中的所有令牌及其年龄.

- 连续变迁: $M\xrightarrow{t}M'$.$\forall p\in P,M'(p)=[n+t|n\in M(p)]$.

- 离散变迁: $M \xrightarrow{a} M'$.定义$\bullet a=\{p|(p,a)\in F\}$,$a\bullet=\{p|(a,p)\in F\}$.变迁 a 可触发当且仅当$\forall p\in\bullet a,\exists x_p\in M(p)\wedge x_p\in times(p,a)$,即对于变迁 a 的每一个前置库所,至少存在 1 个年龄在时间区间约束范围内的令牌.

如果变迁 a 可触发,则触发后$\forall p\in P,M'(p)=M(p)\oplus C^+(a,p)\ominus C^-(p,a)$,其中,

- $C^+(a,p)=\begin{cases}[x_n], p\in a\bullet\wedge x_n\in times(a,p)\\ \varnothing, \text{ otherwise}\end{cases}$;
- $C^-(p,a)=\begin{cases}[x_r], p\in\bullet a\wedge x_r\in M(p)\wedge x_r\in times(p,a)\wedge(p,a)\notin R\\ \varnothing, \text{ otherwise}\end{cases}$.

其中,$C^+(a,p)$表示变迁 a 触发后库所 p 应该增加的令牌,令牌的初始年龄从时间区间 $times(a,p)$中随机选取;$C^-(p,a)$表示变迁 a 触发后库所 p 应该移除的令牌,被移除令牌在前置库所中随机选取并且要求满足时间区间 $times(p,a)$约束.$(p,a)\notin R$ 表示当前边不是读边,读边对应的令牌在变迁后不会被消耗.

定义 6(可读边时间 Petri 网的可覆盖性问题). 给定格局 M,是否存在一条从初始格局 M_0 到 M'的变迁路径且满足 $M'\geqslant M.M'\geqslant M$ 当且仅当$\forall p\in P,M'(p)\geqslant M(p)$.

定理 1. 可读边时间 Petri 网的可覆盖性问题是可判定的.

根据文献[10]可以得到.

2 异步多进程时间自动机

在实时系统中,进程运行时可触发其他进程实例,新生成的进程或打断当前进程,或被暂时挂起.所有的进程实例缓存在一个队列中等待调度.为了描述以上行为,我们提出异步多进程时间自动机模型,每一类进程由一个进程时间自动机描述.在时间自动机的基础上,它去除了所有输入字符,其部分状态可触发新的进程.考虑到队列会导致模型图灵完备,采用集合缓存所有进程实例.当进程时间自动机变迁至触发状态时,对应的新进程时间自动机实例被插入到集合.刚被触发的实例处于释放状态,还需要进一步激活才能运行.每次不确定地从集合中挑选进程实例激活或者运行,直到集合为空.但为了能编码到可读边时间 Petri 网,对语义做了限制,一个进程时间自动机的多个实例中,最多只能有 1 个处于激活状态.

定义 7. 一个进程时间自动机是一个元组$\langle Q_C,Q,q_0,F,X,I,A\rangle$,其中,

- $Q_C\subseteq Q$,表示所有用于触发新进程的有限状态集合;
- 其余部分与时间自动机一致,只是去除了所有输入字符.

定义 8. 一个异步多进程时间自动机是一个元组$\langle P_A,A_M,R\rangle$,其中,

- P_A 是进程时间自动机的有限集合,对应系统中每一种可能触发的任务进程类型;
- A_M 是对应主进程的进程时间自动机,$A_M\in P_A$;
- $R:\bigcup_{A\in P_A}Q_C(A)\to P_A$,将所有触发状态集合映射到进程时间自动机.

定义 9(异步多进程时间自动机的语义). 一个异步多进程时间自动机的格局是一个函数 $M:P_A\to(\{(Q,X\to\mathbb{R}^{\geqslant 0}),empty\},MS(\mathbb{R}^{\geqslant 0}))$,将每一类进程时间自动机映射到一个组对.组对的第 1 部分是(q,v)对,表示给定进程时间自动机存在一个激活的实例,其所处状态与时钟估值函数分别为 q 和 v;或者是 $empty$,表示给定进程时间自动机不存在激活的实例.组对的第 2 部分是非负实数上的多重集合,对应给定进程时间自动机的所有未激活实例,集合中每个非负实数表示其从释放到当前的时间.对应的标签变迁系统(LTS)如下所示.

- $M \xrightarrow{t} M'$.其中,$t\in\mathbb{R}^{\geqslant 0}$;

$$\forall p\in P_A.M'(p)=\begin{cases}(empty,rts.map(x\Rightarrow x+t)), & M(p)=(empty,rts)\\ ((q,v+t),rts.map(x\Rightarrow x+t)), & M(p)=((q,v),rts)\end{cases}.$$

- $M \xrightarrow{active} M'$.要求触发前$\exists p_{active}\in P_A,M(p_{active})=(empty,rts)\wedge\exists rt\in rts$;

$$\forall p\in P_A,M'(p)=\begin{cases}((q_0^{P_{active}},v_0),rts\ominus[rt]), & p=p_{active}\wedge M(p)=(empty,rts)\wedge\exists rt\in rts\\ M(p), & \text{otherwise}\end{cases}.$$

- $M \xrightarrow{a} M'$.要求触发前 $\exists p_a \in P_A.M(p_a)=((q,v),rts) \wedge q \xrightarrow{\phi,\lambda} q' \wedge v \vDash \phi \wedge q' \notin Q_C(p_a)$;
  $\forall p \in P_A, M'(p)=\begin{cases}((q',v'),rts), & p=p_a \wedge M(p)=((q,v),rts)\\ M(p), & \text{otherwise}\end{cases}$,其中,$v'=v[\lambda], v' \vDash I_{p_a}(q')$.
- $M \xrightarrow{r} M'$.要求触发前 $\exists p_r \in P_A.M(p_r)=((q,v),rts) \wedge q \xrightarrow{\phi,\lambda} q' \wedge v \vDash \phi \wedge q' \in Q_C(p_r)$;
  $\forall p \in P_A, M'(p)=\begin{cases}((q',v'),rts), & p=p_r \wedge M(p)=((q,v),rts)\\ (any, rts \oplus [0]), & p=R(q') \wedge M(p)=(any,rts)\\ M(p), & \text{otherwise}\end{cases}$,其中,$v'=v[\lambda], v' \vDash I_{p_a}(q')$.
- $M \xrightarrow{\varepsilon} M'$.要求触发前$\exists p_\varepsilon \in P_A, M(p_\varepsilon)=((q,v),rts) \wedge q \in F(p_\varepsilon)$;
  $\forall p \in P_A, M'(p)=\begin{cases}(empty,rts), & p=p_\varepsilon \wedge M(p)=((q,v),rts) \wedge q \in F(p_\varepsilon)\\ M(p), & \text{otherwise}\end{cases}$.

初始格局为 $M_0(p)=\begin{cases}((q_0^{A_M}, v_0),[]), & p=A_M\\ (empty,[]), & p \in P_A/\{A_M\}\end{cases}$,只有主进程对应的进程时间自动机存在激活的实例,所有进程时间自动机都没有未激活实例.

第 1 条是连续变迁,经过时间 t 后,所有未激活实例的释放时间增加 t,所有激活实例的时钟估值函数增加 t;第 2 条是激活变迁,从 P_A 中随机选出一个进程时间自动机,要求其不存在激活实例且存在未激活实例,变迁后从未激活实例集合 rts 中随机选择一个删除,同时将 $empty$ 变为 $(q_0^{P_{active}}, v_0)$,分别表示所选进程时间自动机的初始状态和初始估值函数;第 3 条是动作变迁,从 P_A 中随机选出一个进程时间自动机,要求其存在激活实例且该实例能做一步变迁动作,变迁后状态不属于触发状态集合,变迁后更新激活实例的状态和估值函数;第 4 条是触发变迁,与第 3 条类似,但要求变迁后状态属于触发状态集合,变迁后被 R 映射到的进程时间自动机的未激活实例集合增加 0;第 5 条是结束变迁,从 P_A 中随机选出一个进程时间自动机,要求其存在激活实例且状态处于终止状态,变迁后将(q,v)重设为 $empty$.

定义 10(异步多进程时间自动机的可覆盖性问题). 给定格局 M,是否存在一条从初始格局 M_0 到 M'的变迁路径且满足 $M' \geqslant M.M' \geqslant M$ 当且仅当$\forall p \in P_A, M'(p) \geqslant M(p)$.定义$((q,v),rts) \geqslant ((q',v'),rts')$,其中,$q=q' \wedge v=v' \wedge rts \geqslant rts'$.定义$(empty,rts) \geqslant (empty,rt')$,其中,$rts \geqslant rts'$.

3 异步多进程时间自动机到可读边时间 Petri 网的编码

本节介绍异步多进程时间自动机到可读边时间 Petri 网的编码方法,从而证明异步多进程时间自动机的可覆盖性问题可判定.

3.1 自动机状态与时钟

本节介绍单个进程时间自动机实例到可读边时间 Petri 网的编码.进程时间自动机分为有限状态自动机和时钟两部分:有限状态自动机部分与去时间的可读边时间 Petri 网直接对应;时钟可编码到令牌,时钟值对应令牌的年龄.

如图 2 所示,有限状态自动机的状态及变迁与可读边时间 Petri 网的库所及变迁一一对应.构造唯一一个令牌,其所处的库所对应有限状态自动机当前所处状态.库所与变迁之间的时间区间都是[0,∞),因此可完全忽略令牌的年龄.

对于每一个时钟,构造一个新的库所,其中存放唯一一个令牌.在此基础之上,可编码时钟守卫和时钟重置.如图 3 所示,对于 $x \geqslant 2$ 时钟守卫,在可读边时间 Petri 网中找到 x 时钟对应的库所,在该库所与变迁之间添加一条读边(没有箭头).读边的时间区间与时间守卫区间一致,当该库所中的令牌年龄大于等于 2 时,变迁才可触发,由于是读边,触发后令牌不会被消耗并保持其年龄;如图 4 所示,对于 x=0 时钟重置,为了更新 x 对应库所中的令牌年龄,需添加一条该库所到变迁的边,时间区间是[0,∞),用于消耗原令牌.另外,还需添加一条变迁到该库所的边,用于生成新的令牌,令牌的初始年龄通过时间区间限制到 0.

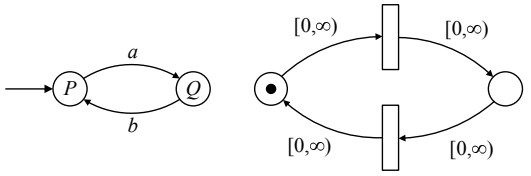

Fig.2 Encoding of finite state automata

图 2 有限状态自动机的编码

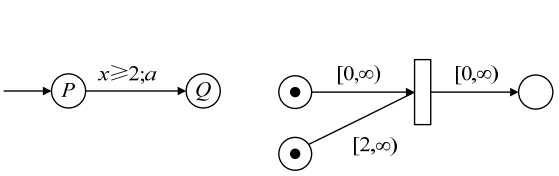

Fig.3 Encoding of clock guard

图 3 时钟守卫的编码

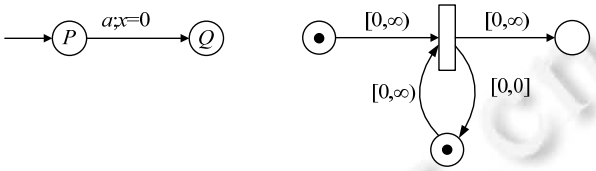

Fig.4 Encoding of clock resetting

图 4 时钟重置的编码

如果一个时钟在变迁中同时参与了时钟守卫和时钟重置,编码方式类似时钟重置,只需修改时钟库所到变迁的边上的时间区间,使其与时钟守卫一致.

3.2 进程实例触发激活结束

上一节中使用一个令牌来编码一个进程时间自动机实例当前所处的状态,如果进程时间自动机有多个实例,直觉的想法是使用多个令牌来编码多个实例所处的状态.在异步多进程时间自动机模型中,每个进程时间自动机实例分为激活与未激活两种状态,进程实例刚被触发时处于未激活状态.因此,对于每个进程时间自动机,额外构造一个库所用于编码未激活实例,新的实例被触发后令牌流入该库所;当令牌从该库所流出至初始状态对应库所时,该实例被激活.对于每个终止状态对应的库所,构造一个变迁用于消耗令牌,表示进程实例结束.

具体的编码方法如图 5 所示.

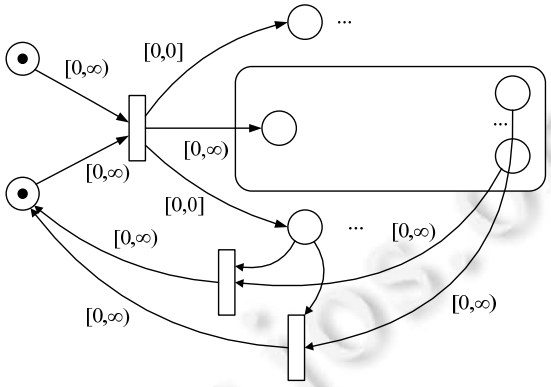

Fig.5 Triggering, activation and finishing of a task

图 5 任务触发激活结束

方框内是自动机状态对应的库所,其中左边是初始状态库所,右边是终止状态库所.方框边缘上下是时钟库所.左上角是未激活实例库所,其他所有触发当前进程时间自动机的变迁都会有一条边指向该库所的边,触发后新的令牌流入库所,令牌初始年龄为 0;方框左边的变迁是激活变迁,暂时忽略左下角库所,变迁触发后,未激活实例库所流出一个令牌,初始状态库所和所有时钟库所流入新的令牌,其中,流入时钟库所的令牌初始年龄通过时钟区间限制到 0;根据异步多进程时间自动机模型的语义,只有进程时间自动机的所有实例都处于未激活状态

时才可以激活其中某一个.因此新增了左下角同步控制库所,如果其存有令牌,表明当前不存在已激活实例,激活变迁才能触发.方框下方的变迁是结束变迁,触发后当前激活实例结束,终止状态库所及所有时钟库所里的令牌都被消耗(图中只画了一部分边),为了让其他未激活实例能够被激活,新的令牌流入同步控制库所.

4 定理及证明

定义 11. 令 M_t 表示异步多进程时间自动机格局的集合,M_p 表示可读边时间 Petri 网格局的集合;给定 $p\in P_A$,记 p 编码后对应的子可读边时间 Petri 网为 $Ratp_p$, l_r^p 为其未激活进程库所, l_y^p 为其同步控制库所, l_c^p 为其 c 时钟库所, l_s^p 为其 s 状态库所,L^p 为其所有库所集合.

定义函数 F_1:$M_t\to M_p$ 及函数 F_2:$M_p\to M_t$.

给定 $M\in M_t,\forall p\in P_A,\forall l\in L^p$,

- 如果 $M(p)=((q,v),rts)$, $F_1(M)(l)=\begin{cases}rts, & l=l_r^p\\ [], & l=l_y^p\\ [v(c)], & l=l_c^p\\ \exists tk\in\mathbb{R}^{\geqslant 0}.[tk], & l=l_q^p\\ [], & \text{otherwise}\end{cases}$;
- 如果 $M(p)=(empty,rts)$, $F_1(M)(l)=\begin{cases}rts, & l=l_r^p\\ \exists tk\in\mathbb{R}^{\geqslant 0}.[tk], & l=l_y^p\\ [], & \text{otherwise}\end{cases}$.

其中,$\exists tk\in\mathbb{R}^{\geqslant 0}.[tk]$表示可以设置任意非负实数值.

给定 $M\in M_p,\forall p\in P_A,\forall l\in L^p$,

- 如果 $M(l)=\begin{cases}rts, & l=l_r^p\\ [], & l=l_y^p\\ [t_c], & l=l_c^p\\ \exists tk\in\mathbb{R}^{\geqslant 0}.[tk], & l=l_q^p\\ [], & \text{otherwise}\end{cases}$, $F_2(M)(p)=((q,v),rts)\wedge v(c)=t_c$;
- 如果 $M(l)=\begin{cases}rts, & l=l_r^p\\ \exists tk\in\mathbb{R}^{\geqslant 0}.[tk], & l=l_y^p\\ [], & \text{otherwise}\end{cases}$, $F_2(M)(p)=(empty,rts)$.

引理 1. $F_1.F_2=id_t,id_t:M_t\to M_t\wedge\forall M\in M_t,id_t(M)=M$.

证明:给定 $M\in M_t,\forall p\in P_A,\forall l\in L^p$,

- 如果 $M(p)=((q,v),rts)$, $F_1(M)(l)=\begin{cases}rts, & l=l_r^p\\ [], & l=l_y^p\\ [v(c)], & l=l_c^p\\ \exists tk\in\mathbb{R}^{\geqslant 0}.[tk], & l=l_q^p\\ [], & \text{otherwise}\end{cases}$, $F_2(F_1(M))(p)=((q,v'),rts)\wedge v'(c)=v(c)$;
- 如果 $M(p)=(empty,rts)$, $F_1(M)(l)=\begin{cases}rts, & l=l_r^p\\ \exists tk\in\mathbb{R}^{\geqslant 0}.[tk], & l=l_y^p\\ [], & \text{otherwise}\end{cases}$, $F_2(F_1(M))(p)=(empty,rts)=M(p)$.

即,$F_2(F_1(M))(p)=((q,v),rts)=M(p)$. □

引理 2. 假定异步多进程时间自动机格局 M_1,M_2 满足 $M_1\xrightarrow{apta}M_2$,则 $F_1(M_1)\xrightarrow{ratp}F_1(M_2)$,其中,$\xrightarrow{apta}$ 是异步多进程时间自动机中的任意变迁,$\xrightarrow{ratp}$ 是可读边时间 Petri 网中与之对应的变迁.假定可读边

时间 Petri 网格局 M_1,M_2 满足 $M_1 \xrightarrow{ratp} M_2$,则 $F_2(M_1) \xrightarrow{apta} F_2(M_2)$,其中,$\xrightarrow{ratp}$ 是可读边时间 Petri 网中的任意变迁,$\xrightarrow{apta}$ 是异步多进程时间自动机中与之对应的变迁.

证明:

- 证明 $M_1 \xrightarrow{apta} M_2$,则 $F_1(M_1) \xrightarrow{ratp} F_1(M_2)$:

连续变迁: $\forall p \in P_A, M_2(p) = \begin{cases} (empty, rts.map(x \Rightarrow x+t)), & M_1(p) = (empty, rts) \\ ((q, v+t), rts.map(x \Rightarrow x+t)), & M_1(p) = ((q, v), rts) \end{cases}$,

如果 $M_1(p)$=$(empty,rts)$∧$M_2(p)$=$(empty,rts.map(x⇒x+t))$,$\forall l \in L^p$:

$$F_1(M_1)(l) = \begin{cases} rts, & l = l_r^p \\ \exists tk \in \mathbb{R}^{\geqslant 0}.[tk], & l = l_y^p \\ [], & \text{otherwise} \end{cases}, F_1(M_2)(l) = \begin{cases} rts.map(x \Rightarrow x+t), & l = l_r^p \\ \exists tk' \in \mathbb{R}^{\geqslant 0}.[tk'], & l = l_y^p \\ [], & \text{otherwise} \end{cases}.$$

令 tk'=tk+t,则有 $F_1(M_1) \xrightarrow{t} F_1(M_2)$.

$M_1(p)$=$((q,v),rts)$∧$M_2(p)$=$((q,v+t),rts.map(x⇒x+t))$的情况类似,此处省略.

激活变迁: $\forall p \in P_A, M_2(p) = \begin{cases} ((q_0^{p_{active}}, v_0), rts \ominus [rt]), & p = p_{active} \wedge M_1(p) = (empty, rts) \wedge \exists rt \in rts \\ M_1(p), & \text{otherwise} \end{cases}$,

$M_1(p_{active}) = (empty, rts) \wedge M_2(p_{active}) = ((q_0^{p_{active}}, v_0), rts \ominus [rt]), \forall l \in L^{p_{active}}$,

$$F_1(M_1)(l) = \begin{cases} rts, & l = l_r^p \\ \exists tk \in \mathbb{R}^{\geqslant 0}.[tk], & l = l_y^p \\ [], & \text{otherwise} \end{cases}, F_1(M_2)(l) = \begin{cases} rts \ominus [rt], & l = l_r^p \\ [], & l = l_y^p \\ [0], & l = l_c^p \\ \exists tk' \in \mathbb{R}^{\geqslant 0}.[tk'], & l = l_{q_0^{p_{active}}}^p \\ [], & \text{otherwise} \end{cases}.$$

满足激活变迁,即 $F_1(M_1) \xrightarrow{active} F_1(M_2)$.

动作变迁: $\forall p \in P_A, M'(p) = \begin{cases} ((q', v[\lambda]), rts), & p = p_a \wedge M(p) = ((q, v), rts) \\ M(p), & \text{otherwise} \end{cases}$,

$M_1(p_a) = ((q,v),rts) \wedge M_2(p_a) = ((q',v[\lambda]),rts), \forall l \in L^{p_a}$,

$$F_1(M_1)(l) = \begin{cases} rts, & l = l_r^p \\ [], & l = l_y^p \\ [v(c)], & l = l_c^p \\ \exists tk \in \mathbb{R}^{\geqslant 0}.[tk], & l = l_q^p \\ [], & \text{otherwise} \end{cases}, F_1(M_2)(l) = \begin{cases} rts, & l = l_r^p \\ [], & l = l_y^p \\ [v[\lambda](c)], & l = l_c^p \\ \exists tk' \in \mathbb{R}^{\geqslant 0}.[tk'], & l = l_{q'}^p \\ [], & \text{otherwise} \end{cases}.$$

根据时钟编码规则及相应变迁语义, $F_1(M_1) \xrightarrow{a} F_1(M_2)$.

触发变迁:与动作变迁类似,只是 $R(q')$上的格局也有变化,$M_1(R(q'))$=(any,rts)∧$M_2(R(q'))$=$(any,rts⊕[0])$,根据变迁语义,$R(q')$对应的子网中 $l_r^{R(q')}$ 增加了一个年龄为 0 的令牌,因此 $F_1(M_1) \xrightarrow{r} F_1(M_2)$.

结束变迁: $\forall p \in P_A, M'(p) = \begin{cases} (empty, rts), & p = p_\varepsilon \wedge M(p) = ((q, v), rts) \wedge q \in F(p_\varepsilon) \\ M(p), & \text{otherwise} \end{cases}$,

$M_1(p_\varepsilon) = ((q,v),rts) \wedge M_2(p_\varepsilon) = (empty, rts), \forall l \in L^{p_\varepsilon}$,

$$F_1(M_1)(l) = \begin{cases} rts, & l = l_r^p \\ [], & l = l_y^p \\ [v(c)], & l = l_c^p \\ \exists tk \in \mathbb{R}^{\geqslant 0}.[tk], & l = l_q^p \\ [], & \text{otherwise} \end{cases}, F_1(M_2)(l) = \begin{cases} rts, & l = l_r^p \\ \exists tk' \in \mathbb{R}^{\geqslant 0}.[tk'], & l = l_y^p \\ [], & \text{otherwise} \end{cases}.$$

根据变迁语义,p_ε对应的子网中所有时钟令牌和状态令牌清空,新增一个同步控制令牌,因此,

$$F_1(M_1)\xrightarrow{\varepsilon}F_1(M_2).$$

- 证明 $M_1\xrightarrow{ratp}M_2$,则 $F_2(M_1)\xrightarrow{apta}F_2(M_2)$:

同样,根据变迁类型归纳,易证.此处略. □

引理 3. 假定异步多进程时间自动机格局 M_1,M_2 满足 $M_1\geqslant M_2$,则 $F_1(M_1)\geqslant F_1(M_2)$;假定可读边时间 Petri 网格局 M_1,M_2 满足 $M_1\geqslant M_2$,则 $F_2(M_1)\geqslant F_2(M_2)$.

证明:

- 证明 $M_1\geqslant M_2$,则 $F_1(M_1)\geqslant F_1(M_2)$.

$\forall p\in P_A, M_1(p)\geqslant M_2(p)$,如果 $M_1(p)=((q,v),rts)\wedge M_2(p)=((q,v),rts')\wedge rts\geqslant rts'$,根据 F_1 的定义,

$$F_1(M_1)(l)=\begin{cases}rts, & l=l_r^p\\ [], & l=l_y^p\\ [v(c)], & l=l_c^p\\ \exists tk\in\mathbb{R}^{\geqslant 0}.[tk], & l=l_q^p\\ [], & \text{otherwise}\end{cases},F_1(M_2)(l)=\begin{cases}rts, & l=l_r^p\\ [], & l=l_y^p\\ [v(c)], & l=l_c^p\\ \exists tk'\in\mathbb{R}^{\geqslant 0}.[tk'], & l=l_{q'}^p\\ [], & \text{otherwise}\end{cases}.$$

$rts\geqslant rts'$已知,取 $tk\geqslant tk'$,则$\forall l\in L^p, F_1(M_1)(l)\geqslant F_1(M_2)(l)$.

如果 $M_1(p)=(empty,rts)\wedge M_2(p)=(empty,rts')\wedge rts\geqslant rts'$,根据 F_1 的定义,

$$F_1(M_1)(l)=\begin{cases}rts, & l=l_r^p\\ \exists tk\in\mathbb{R}^{\geqslant 0}.[tk], & l=l_y^p\\ [], & \text{otherwise}\end{cases},F_1(M_2)(l)=\begin{cases}rts', & l=l_r^p\\ \exists tk'\in\mathbb{R}^{\geqslant 0}.[tk'], & l=l_y^p\\ [], & \text{otherwise}\end{cases}.$$

$rts\geqslant rts'$已知,取 $tk\geqslant tk'$,则$\forall l\in L^p, F_1(M_1)(l)\geqslant F_1(M_2)(l)$.

- 证明 $M_1\geqslant M_2$,则 $F_2(M_1)\geqslant F_2(M_2)$.

$\forall p\in P_A,\forall l\in L^p$,如果

$$M_1(l)=\begin{cases}rts, & l=l_r^p\\ [], & l=l_y^p\\ [t_c], & l=l_c^p\\ \exists tk\in\mathbb{R}^{\geqslant 0}.[tk], & l=l_q^p\\ [], & \text{otherwise}\end{cases},M_2(l)=\begin{cases}rts, & l=l_r^p\\ [], & l=l_y^p\\ [t_c'], & l=l_c^p\\ \exists tk'\in\mathbb{R}^{\geqslant 0}.[tk'], & l=l_{q'}^p\\ [], & \text{otherwise}\end{cases}.$$

$\forall l\in L^p, M_1(l)\geqslant M_2(l)$,则 $rts\geqslant rts', [t_c]\geqslant[t_c']$,即 $t_c=t_c', tk\geqslant tk'$.

$F_2(M_1)(p)=((q,v),rts), F_2(M_2)(p)=((q,v'),rts'), v=v', rts\geqslant rts'$.所以$\forall p\in P_A, F_2(M_1)(p)=F_2(M_2)(p)$.

如果

$$M_1(l)=\begin{cases}rts, & l=l_r^p\\ \exists tk\in\mathbb{R}^{\geqslant 0}.[tk], & l=l_y^p\\ [], & \text{otherwise}\end{cases},M_2(l)=\begin{cases}rts', & l=l_r^p\\ \exists tk'\in\mathbb{R}^{\geqslant 0}.[tk'], & l=l_y^p\\ [], & \text{otherwise}\end{cases}.$$

$F_2(M_1)(p)=(empty,rts), F_2(M_2)(p)=(empty,rts'), rts\geqslant rts'$.所以$\forall p\in P_A, F_2(M_1)(p)=F_2(M_2)(p)$. □

定理 2. 异步多进程时间自动机的可覆盖性问题是可判定的.

证明:令异步多进程时间自动机初始格局为 M_0,读边时间 Petri 网初始格局为 M_0',显然有:

$$F_1(M_0)=M_0', F_2(M_0')=M_0.$$

给定异步多进程时间自动机格局 M_2,根据定义 11 得到可读边时间 Petri 网中对应的格局 $F_1(M_2)$,再根据定理 1,可判定是否存在 M_1' 满足 $M_1'\geqslant F_1(M_2)\wedge M_0'\longrightarrow^* M_1'$:

- 若存在,根据引理 2 和引理 3 可得, $F_2(M_1')\geqslant F_2(F_1(M_2))\wedge F_2(M_0')\longrightarrow^* F_2(M_1')$,根据引理 1,

$$F_2(M_1')\geqslant M_2\wedge M_0\longrightarrow^* F_2(M_1').$$

- 若不存在,则异步多进程时间自动机中也一定不存在 M_1 满足 $M_1 \geqslant M_2 \wedge M_0 \longrightarrow^* M_1$.假设存在,根据引理 2 和引理 3, $F_1(M_1) \geqslant F_1(M_2) \wedge F_1(M_0) \longrightarrow^* F_1(M_1)$,即 $F_1(M_1) \geqslant F_1(M_2) \wedge M_0' \longrightarrow^* F_1(M_1)$,与假设矛盾,即得证. □

5 相关工作

近年来,基于时间自动机理论的扩展模型与方法在实时系统验证领域得到了广泛的应用,目前已有的研究成果如下.

实时系统验证工具 UPPAAL 通过时间自动机网络模型对系统建模,使用 TCTL(时间计算树逻辑)的子集作为查询语言描述需要验证的性质.已成功地应用于通信协议、实时控制器、多媒体应用等系统中.由于使用时间自动机网络建模,UPPAAL 不允许系统运行时动态创建新的进程.为了改进这一点,可调用时间自动机[11]被提出来.在时间自动机的基础上,它做了以下扩展:时间自动机被数据集合参数化;时间自动机能触发其他时间自动机实例;被调用者能向调用者返回结果.可调用时间自动机的部分子集已被翻译到 UPPAAL 的时间自动机.

部分实时系统更适合用 Petri 网风格的模型来建模,但时间 Petri 网或边时间 Petri 网等模型很多性质都不可判定.通过增加限制,将其受限版本编码到时间自动机,就可利用 UPPAAL 进行验证.因此,编码的效率就显得非常重要,许多研究[12−16]提出了高效的编码方法.

另一类研究[17−19]重点关注实时系统中任务的可调度性分析,扩展的时间自动机的每条变迁对应触发新的任务,充分利用时间自动机的时钟守卫及时钟重置,非周期性的任务触发行为可被灵活地描述.任务为元组 (C,D)所刻画,其中,C 表示任务运行的时间,D 表示任务的相对截止时间.所有的任务被缓存在一个队列,新任务通过调度算法插入到队列.任务可调度性问题被规约到时间自动机的可达性问题,是可判定的.

复杂任务的 C 很难得到,时间正则任务自动机对此进行了改进,用时间自动机来刻画任务.通过规约到嵌套时间自动机,将其可调度性问题转化为嵌套时间自动机的可达性问题,从而证明其可调度性问题可判定.

6 总结以及未来的工作

本文提出了异步多进程时间自动机模型,用于建模运行时动态触发其他进程的实时系统,并且通过编码至可读边时间 Petri 网证明了其可覆盖性问题可判定.

为了能够规约到可读边时间 Petri 网,对异步多进程时间自动机的语义做了限制,一个进程时间自动机的多个实例中,最多只能有 1 个处于激活状态.未来希望能够放松此限制,允许无条件地激活变迁,但该语义下的可覆盖性问题是否可判定还不得而知.

References:

[1] Rajeev A, Dill DL. A theory of timed automata. Theoretical Computer Science, 1994,126(2):183−235. [doi: 10.1016/0304-3975(94)90010-8]

[2] Johan B, Wang Y. Timed automata: Semantics, algorithms and tools. In: Jorg D, Wolfgang R, Grzegorz R, eds. Proc. of the Lectures on Concurrency and Petri Nets. Berlin, Heidelberg: Springer-Verlag, 2004. 87−124. [doi: 10.1007/978-3-540-27755-2_3]

[3] Gerd B, David A, Larsen KG. A tutorial on UPPAAL. In: Marco B, Flavio C, eds. Proc. of the Formal Methods for the Design of Real-Time Systems. Berlin, Heidelberg: Springer-Verlag, 2004. 200−236. [doi: 10.1007/978-3-540-30080-9_7]

[4] Larsen Kim G, Paul P, Wang Y. UPPAAL in a nutshell. Int'l Journal on Software Tools for Technology Transfer, 1997,1(1):134−152. [doi: 10.1007/s100090050010]

[5] Li GQ, Cai XJ, Mizuhito O, Shoji Y. Nested timed automata. In: Victor B, Laurent F, eds. Proc. of the Int'l Conf. on Formal Modeling and Analysis of Timed Systems. Berlin, Heidelberg: Springer-Verlag, 2013. 168−183. [doi: 10.1007/978-3-642-40229-6_12]

[6] Bernard B, Menasche M. An enumerative approach for analyzing time Petri nets. In: Proc. of the IFIP. Elsevier Science Publishers, 1983. 41−46.

[7] Bernard B, Michel D. Modeling and verification of time dependent systems using time Petri nets. IEEE Trans. on Software Engineering, 1991,17(3):259−273. [doi: 10.1109/32.75415]

[8] Abdulla PA, Nylén A. Timed Petri nets and BQOs. In: Jose-Manuel C, Maciej K, eds. Proc. of the Int'l Conf. on Application and Theory of Petri Nets. Berlin, Heidelberg: Springer-Verlag, 2001. 53−70. [doi: 10.1007/3-540-45740-2_5]

[9] Ruiz VV, Gomez FC, de Frutos Escrig D. On non-decidability of reachability for timed-arc Petri nets. In: Proc. of the 8th Int'l Workshop on Petri Nets and Performance Models (PNPM'99). Washington: IEEE Computer Society, 1999. 188−196. [doi: 10.1109/PNPM.1999.796565]

[10] Bouyer P, Haddad S, Reynier PA. Timed Petri nets and timed automata: On the discriminating power of zeno sequences. Information and Computation, 2008,206(1):73−107. [doi: 10.1016/j.ic.2007.10.004]

[11] Jalil B, Frits V, Jean-Paul B. Extending UPPAAL for the modeling and verification of dynamic real-time systems. In: Arbab F, Sirjani M, eds. Proc. of the Int'l Conf. on Fundamentals of Software Engineering. Berlin, Heidelberg: Springer-Verlag, 2013. 111−132. [doi: 10.1007/978-3-642-40213-5_8]

[12] Jiří S. Timed-Arc Petri nets vs. networks of timed automata. In: Gianfranco C, Philippe D, eds. Proc. of the Int'l Conf. on Application and Theory of Petri Nets. Berlin, Heidelberg: Springer-Verlag, 2005. 385−402. [doi: 10.1007/11494744_22]

[13] Kenneth JB, Jørgensen Y, Jiří S. An efficient translation of timed-arc Petri nets to networks of timed automata. In: Karin B, Ana C, eds. Proc. of the Int'l Conf. on Formal Engineering Methods. Berlin, Heidelberg: Springer-Verlag, 2009. 698−716. [doi: 10.1007/978-3-642-10373-5_36]

[14] Cassez F, Roux OH. Structural translation from time Petri nets to timed automata. The Journal of Systems and Software, 2006, 79(10):1456−1468. [doi: 10.1016/j.jss.2005.12.021]

[15] Bouyer P, Haddad S, Reynier PA. Extended timed automata and time Petri nets. In: Proc. of the 6th Int'l Conf. on Application of Concurrency to System Design. Washington: IEEE Computer Society, 2006. 91−100. [doi: 10.1109/ACSD.2006.6]

[16] Bérard B, Cassez F, Haddad S, Lime D, Roux OH. Comparison of the expressiveness of timed automata and time Petri nets. In: Paul P, Wang Y, eds. Proc. of the Int'l Conf. on Formal Modeling and Analysis of Timed Systems. Berlin, Heidelberg: Springer-Verlag, 2005. 211−225. [doi: 10.1007/11603009_17]

[17] Fersman E, Krcal P, Pettersson P, Wang Y. Task automata: Schedulability, decidability and undecidability. Information and Computation, 2007,205(8):1149−1172. [doi: 10.1016/j.ic.2007.01.009]

[18] Ericsson C, Wall A, Wang Y. Timed automata as task models for event-driven systems. In: Proc. of the 6th Int'l Conf. on Real-Time Computing Systems and Applications (RTCSA'99). Washington: IEEE Computer Society, 1999. 182−189. [doi: 10.1109/RTCSA.1999.811218]

[19] Elena F, Paul P, Wang Y. Timed automata with asynchronous processes: Schedulability and decidability. In: Joost-Pieter K, Perdita S, eds. Proc. of the 8th Int'l Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2002). Berlin, Heidelberg: Springer-Verlag, 2002. 67−82. [doi: 10.1007/3-540-46002-0_6]

刘立(1992－),男,湖北孝感人,硕士,主要研究领域为形式化方法.

李国强(1979－),男,博士,副教授,CCF 专业会员,主要研究领域为形式化方法,程序语言理论,可计算学习理论.