基于时间动态下推网络可达性分析

钱俊彦,徐 力,古天龙,赵岭忠,蔡国永
(桂林电子科技大学广西可信软件重点实验室,广西桂林 541004)

摘 要: 动态下推网络(DPN,Dynamic Pushdown Networks)由一组能刻画动态创建线程的动态下推系统(DPDS,Dynamic PushDown Systems)组成.本文首先将描述连续时间的实时时钟引入DPN,提出了时间动态下推网络(TDPN,Timed Dynamic Pushdown Networks),能对动态创建线程的实时并发递归系统建模;然后基于时钟关键点的时钟等价优化方法,并采用on-the-fly技术,仅关心栈顶及下一层的域状态转换,动态的将连续时间模型TDPN转换为时间域表示的离散模型DPN,同时给出TDPN到DPN的转换算法;最后证明在TDPN中的可达状态当且仅当其转换状态在DPN中可达,从而可解决带动态线程创建的实时并发系统的可达性分析.

关键词: 动态下推网络; 时钟等价; 实时并发递归系统; 时间动态下推网络
中图分类号: TP311 **文献标识码:** A **文章编号:** 0372-2112 (2017)09-2241-09
电子学报URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2017.09.027

Reachability Analysis for Time Dynamic Pushdown Networks

QIAN Jun-yan,XU Li,GU Tian-long,ZHAO Ling-zhong,CAI Guo-yong
(*Guangxi Key Laboratory of Trusted Software,Guilin University of Electronic Technology,Guilin,Guangxi* 541004,*China*)

Abstract: Dynamic pushdown networks (DPN) consist of a group of dynamic pushdown systems (DPNS). And each DPNS model can dynamically demonstrate as new threads. Firstly,this paper introduces the real numeric clock to DPN to verify the real-time concurrent recursive system created by dynamic threads,and demonstrate it as time dynamic pushdown networks (TDPN). Secondly,this paper uses the clock equivalent optimization technique based on time key to convert a continuous TDPN models into a series of discrete TDPN models. Besides,this paper proposes the equivalence algorithm of TDPN into DPN. At last,we prove the status in TDPN is reachable if and only if its transition status in DPN is reachable. At the same time,this paper solves reachability problem of the real-time concurrent system created by dynamic threads.

Key words: DPN;clock equivalent;real-time concurrent recursive system;TDPN

1 引言

随着多核技术的发展,并发程序[1]已成为当前程序设计的研究热点.由于并发执行存在不确定性[2],故传统测试方法难以发现程序中隐匿的错误和漏洞.作为测试技术的补充,模型检验[3]已成为保证程序安全和可靠的重要手段,其核心技术的可达性分析,主要原理就是分析某一状态是否可达,来保证模型的安全可靠.

近年来,研究人员基于自动机模型,引入实时时钟,用于实时系统建模及其验证.1994年Alur提出的时间自动机[4],是在自动机的基础上引入描述连续时间的时钟,并通过时钟等价技术把时钟离散化,从而实现模型检验时间自动机[5].为了对含有递归的实时系统进行建模,Trivedi[6]提出时间下推自动机(TPDA,Timed PushDown Automaton),并通过时钟等价技术把连续时间下推自动机模型转换为时间域表示的离散下推自动机模型,解决最小时间花费的可达性问题.2013年Abdulla[7]将时间下推自动机的研究对象扩展到无限迁移的实时系统上,同时证明了该情况下的实时系统可达性分析为EXPTIME完全问题.2013年Li[8]提出了嵌套时间自动机,其整体是一个下推自动机,栈的每一层为时间自动机,利用嵌套的思想来解决实时系统中的递归问题.上述模型能描述实时系统的并发及递归问题,

收稿日期:2015-11-17;修回日期:2016-07-18;责任编辑:覃怀银
基金项目:国家自然科学基金(No.61262008,No.61562015,No.61572146,No.U1501252);广西高等学校高水平创新团队及卓越学者计划;广西自然科学基金(No.2014GXNSFAA118365,No.2015GXNSFDA139038);广西可信软件重点实验室重点基金;桂林电子科技大学创新团队

然而无法描述实时并发系统的动态线程创建.

2005 年 Bouajjani[9] 提出一种并发下推系统[10] 的扩展模型——动态下推网络(DPN),解决并发系统中新线程的动态创建,该模型适用于含有递归且带有动态线程创建的并发系统建模. 基于 DPN,Lammich[11] 提出了上下文锁的技术,解决进程之间递归的同步问题,并进行了逆向可达分析. Wenner[12] 在 DPN 中引入权值,用来求解最短路径的可达性. DPN 虽能描述并发系统中新线程的动态创建,但无有描述连续时间的时钟,故无法刻画实时性.

鉴于此,在 DPN 中引入描述连续时间的时钟,包括全局时钟、与全局变量相关的时钟、以及栈字符"年龄"的实数时钟,提出了时间动态下推网络(TDPN),能对含有递归、动态创建线程的实时并发系统进行建模. 由于 TDPN 含有连续时间的时钟约束,当求解可达状态时,可能产生状态空间爆炸问题,故难以直接对 TDPN 进行可达性分析. 为了能对 TDPN 进行可达性分析,把 TDPN 转换为 DPN,利用 DPN 的可达技术解决 TDPN 的可达性问题,也就是基于文献[4,13]的时钟等价思想,提出一种基于时钟关键点的时钟等价优化方法,通过各自时钟的关键点,优化时钟域划分,缩减时钟域及转换后的状态空间. 为了进一步大幅度缩减状态空间,采用动态转换方法,不关心整个项集,而仅关注栈顶及其下一层的项集. 然后通过上述方法,把描述连续时间的 TDPN 模型转换成相应离散模型 DPN,并给出了相应的转换算法,同时分析并证明了 TDPN 转换为 DPN 的正确性,以及 TDPN 模型可达问题与转换后 DPN 模型可达问题是一致的. 最后通过实例,详细阐述整个转换过程.

2 动态下推网络(DPN)和时间动态下推网络(TDPN)

2.1 动态下推网络(DPN)

DPN 是由一组动态下推系统(DPDS)组成的并发动态网络系统,能并行执行各自顺序序列的迁移系统,能对含有动态创建线程的并发程序建模,也就是说每个 DPNS 模型都能执行下推操作,刻画动态创建的新线程. DPN $M=\{D_1,\cdots,D_i,\cdots,D_n\}$,其中 $1\leqslant i\leqslant n$,$D_i=(P_i,\Gamma_i,\Delta_i)$ 是一个动态下推系统,P_i 表示状态集,Γ_i 表示栈字符集,Δ_i 表示具有下列形式的迁移规则集合:

(1)$p_i a\rightarrow p_j\omega_j$;

(2)$p_i a\rightarrow p'_i\omega_i\triangleright p_j\omega_j$, s. t. $p_i,p'_i\in P_i$, $a\in\Gamma_i$, $\omega_i\in\Gamma_i^*$,其中 $p_j\,\omega_j\in P_j\times\Gamma_j^*$ 为动态创建的新线程.

假定 G 为全局变量集,DPN 格局可表示为 $(g,\langle p,\omega\rangle)\in G\times(P\times\Gamma^*)$,其中 $g\in G$ 为全局变量,元组 $\langle p,\omega\rangle=\langle p_1,\omega_1\rangle\cdots\langle p_i,\omega_i\rangle\cdots\langle p_n,\omega_n\rangle$,每个子项 $\langle p_i,\omega_i\rangle\in P\times\Gamma^*$ 表示一个下推线程的局部格局.

2.2 时间动态下推网络(TDPN)

TDPN 模型是 DPN 的一个扩展,在其基础上引入了描述连续时间的实时时钟,用来对动态创建线程的实时并发递归系统建模. TDPN 模型为一个四元组 $T=(P,\Gamma,\Delta,X)$,其中:P 是状态集;Γ 是栈字符集;$\Delta=\Delta_{\text{nop}}\cup\Delta_{\vdash}=\Delta_{\vdash}\cup\Delta_{\text{push}}\cup\Delta_{\text{pop}}\cup\Delta_{\text{dc}}$ 是迁移集合,其中 Δ_{nop} 表示空操作,$\Delta_{=}$ 表示时钟赋值,$\Delta_{\vdash}$ 表示时间流逝,Δ_{push} 表示入栈,Δ_{pop} 表示出栈,Δ_{dc} 表示动态线程创建;X 是时钟集,设 $\mathbb{R}^+$ 为正实数,其时钟解释函数 θ: $X\rightarrow\mathbb{R}^+$ 表示时钟 x(X 在当前的取值为 $\theta(x)\in\mathbb{R}^+$,与时间相关的全局变量和栈字符"年龄"取值亦与之类似.

TDPN 可用于描述含动态线程创建的实时并发递归程序,当模型迁移时,多个下推系统并发执行,也就是说在同一时刻有多个栈进行迁移操作. 假定 G 为全局变量集,TDPN 格局可表示为:$\gamma=\langle\bar{g},\langle p,\bar{\omega}\rangle,\bar{x}\rangle$,其中 $\bar{g}$ 表示当前全局变量 g 和其"年龄" $\theta(g)$ 的二元组 $\langle g,\theta(g)\rangle$;$\langle p,\bar{\omega}\rangle=\langle p_1,\bar{\omega}_1\rangle\cdots\langle p_i,\bar{\omega}_i\rangle\cdots\langle p_n,\bar{\omega}_n\rangle)$,其中 $p_i\in P$ 表示局部状态节点,$\bar{\omega}_i$ 表示栈序列为 i 的栈内容 ω_i 和其"年龄" $\theta(\omega_i)$ 的二元组 $\langle\omega_i,\theta(\omega_i)\rangle$;$\bar{x}$ 表示时钟 x 和其取值 $\theta(x)$ 的二元组 $\langle x,\theta(x)\rangle$. TDPN 的迁移动作集 op 包含:

(1)空操作 nop:表示格局内元素未发生变化;

(2)时钟重置 $x\leftarrow I$:表示把任意值 $c\in I$ 重新赋值给时钟 $x\in X$;

(3)时间流逝 Time$\leftarrow v$:表示格局内所有时钟增加 v,v 为具体流逝的实值,在没有歧义情况下本文用 $\theta+v$ 表示时钟值增加 v;

(4)入栈操作 push(a,I):表示入栈字符 a,并设定相应年龄时钟赋值属于区间 I 的任意值 $c\in I$;

(5)出栈操作 push(a,I):判断栈符 a 的"年龄"是否满足区间 I,如果满足则进行出栈操作,否则不执行操作;

(6)动态线程创建操作 $\triangleright$:表示动态创建一个新线程.

为了描述方便,描述格局迁移时,只关注一个下推系统的迁移执行,其余下推系统的格局状态保持不变. 若描述多个下推系统并发执行时,其单个下推系统的情况亦与之类似. 在此约定的基础上,TDPN 格局迁移关系的操作语义如图 1 所示,其中 $\bar{g}^{+v}$ 表示其"年龄" $\theta(g)$ 加上时间流逝 v,$\bar{\omega}_i^{+v}$ 和 $\bar{x}^{+v}$ 亦与之类似;$\bar{\omega}_j\cdot(a,v)$ 表示 j 号栈内容加上栈字符 a,其"年龄"为 v.

给定 TDPN T,若存在一条迁移序列 $\gamma_0\xrightarrow{\text{op}_1}\gamma_1\cdots\xrightarrow{\text{op}_{n-1}}\gamma_{n-1}\xrightarrow{\text{op}_n}\gamma_n$,其中初始格局 $\gamma_0=\langle\bar{g}_{\text{init}},(\langle p_1,\varepsilon_1\rangle,\cdots,\langle p_n,\varepsilon_n\rangle),\bar{x}_{init}\rangle$,栈为空,目标格局 $\gamma_n=\langle\bar{g}',(\langle p'_1,$

$\overline{\omega}_1\rangle,\cdots,\langle p'_n,\overline{\omega}_n\rangle),\overline{x}'\rangle$,则称状态 $p'_i\in P$ 在 TDPN 中可达. TDPN 的可达性问题描述为:给定描述实时并发递归系统的 TDPN 和目标状态集合,确定经过某次运行,状态是否可达.

$$\frac{\langle\overline{g},(\langle p_1,\overline{\omega}_1\rangle\cdots\langle p_n,\overline{\omega}_n\rangle),\overline{x}\rangle\xrightarrow{op=nop}\langle\overline{g}',(\langle p_1',\overline{\omega}_1'\rangle\cdots\langle p_n',\overline{\omega}_n'\rangle),\overline{x}'\rangle}{\overline{g}'=\overline{g}\quad(\quad\overline{\omega}_i'=\overline{\omega}_i\quad s.t.\forall i\quad 1\le i\le n\quad)\quad\overline{x}'=\overline{x}}\quad[\text{nop}]$$

$$\frac{\langle\overline{g},(\langle p_1,\overline{\omega}_1\rangle\cdots\langle p_n,\overline{\omega}_n\rangle),\overline{x}\rangle\xrightarrow{op=(x\leftarrow I)}\langle\overline{g}',(\langle p_1',\overline{\omega}_1'\rangle\cdots\langle p_n',\overline{\omega}_n'\rangle),\overline{x}'\rangle}{\overline{g}'=\overline{g}\quad(\quad\overline{\omega}_i'=\overline{\omega}_i\quad s.t.\forall i\quad 1\le i\le n\quad)\quad\overline{x}'=\overline{x}[x\leftarrow v]v\in I}\quad[=]$$

$$\frac{\langle\overline{g},(\langle p_1,\overline{\omega}_1\rangle\cdots\langle p_n,\overline{\omega}_n\rangle),\overline{x}\rangle\xrightarrow{op=(Time\leftarrow v)}\langle\overline{g}',(\langle p_1',\overline{\omega}_1'\rangle\cdots\langle p_n',\overline{\omega}_n'\rangle),\overline{x}'\rangle}{\overline{g}'=\overline{g}^{+v}\quad(\quad\overline{\omega}_i'=\overline{\omega}_i^{+v}\quad s.t.\forall i\quad 1\le i\le n\quad)\quad\overline{x}'=\overline{x}^{+v}}\quad[\vdash]$$

$$\frac{\langle\overline{g},(\langle p_1,\overline{\omega}_1\rangle\cdots\langle p_j,\overline{\omega}_j\rangle\cdots\langle p_n,\overline{\omega}_n\rangle),\overline{x}\rangle\xrightarrow{op=push(a,I)}\langle\overline{g}',(\langle p_1',\overline{\omega}_1'\rangle\cdots\langle p_j',\overline{\omega}_j'\rangle\cdots\langle p_n',\overline{\omega}_n'\rangle),\overline{x}'\rangle}{\overline{g}'=\overline{g}\quad(\quad\overline{\omega}_i'=\overline{\omega}_i\quad s.t.\forall i\ne j\quad 1\le i\le n\qquad\overline{\omega}_j'=\overline{\omega}_j\bullet(a,v)\quad)\quad\overline{x}'=\overline{x}}\quad[\text{push}(a,I)]$$

$$\frac{\langle\overline{g},(\langle p_1,\overline{\omega}_1\rangle\cdots\langle p_j,\overline{\omega}_j\rangle\cdots\langle p_n,\overline{\omega}_n\rangle),\overline{x}\rangle\xrightarrow{op=pop(a,I)}\langle\overline{g}',(\langle p_1',\overline{\omega}_1'\rangle\cdots\langle p_j',\overline{\omega}_j'\rangle\cdots\langle p_n',\overline{\omega}_n'\rangle),\overline{x}'\rangle}{\overline{g}'=\overline{g}\quad(\quad\overline{\omega}_i'=\overline{\omega}_i\quad s.t.\forall i\ne j\quad 1\le i\le n\qquad\overline{\omega}_j'\bullet(a,v)=\overline{\omega}_j\quad)\quad\overline{x}'=\overline{x}}\quad[\text{pop}(a,I)]$$

$$\frac{\langle\overline{g},(\langle p_1,\overline{\omega}_1\rangle\cdots\langle p_n,\overline{\omega}_n\rangle),\overline{x}\rangle\xrightarrow{op=dc}\langle\overline{g}',(\langle p_1',\overline{\omega}_1'\rangle\cdots\langle p_n',\overline{\omega}_n'\rangle\langle p_{n+1}',\overline{\omega}_{n+1}'\rangle),\overline{x}'\rangle}{\overline{g}'=\overline{g}\quad(\quad\overline{\omega}_i'=\overline{\omega}_i\quad s.t.\forall i\quad 1\le i\le n\quad)\quad\overline{x}'=\overline{x}}\quad[\triangleright]$$

图1　TDPN T的操作语义

3　TDPN 转换成 DPN

当前验证实时系统的主要方法是将连续模型转换为不含时间约束的离散模型,通常用等价域描述离散模型状态. 时钟等价主要有两种方法:域图(region graph)和带图(zone graph)[14],与域图相比,带图是一种更紧凑的符号状态表示,已应用于诸如模型验证工具 UPPAAL 等,但带图的方法可能会产生状态无限的情况[15];且带图生成的符号状态图可能具有不稳定性,符号状态迁移序列不一定可行[16];以及存在构造太抽象而不能直接判断时间进程,从而产生 non-Zenoness 问题[17]. 故本文采用域图方法.

采用时钟域等价技术,将连续时钟转换为离散时钟,文献[4,13]已给出其可行性证明. 为了缩减转换后的状态空间,基于文献[4,13]的思想,优化时钟区域的划分,3.1 节将给出基于关键点的时钟等价优化技术,把 TDPN 模型中与连续时间相关变量-全局变量及其“年龄”、栈字符及其“年龄”、时钟变量及其取值,转换为 DPN 模型中描述 DPN 的域,该方法与文献[13]提出的整数划分区域相比,转换后的状态空间指数级缩减. 为了进一步缩减状态空间,采用 on-the-fly 技术和动态转换思想,仅关注栈顶及下一层的域转换,而无需关心栈其它部分. 根据不同迁移动作,给出不同转换规则,从而可将复杂的 TDPN 可达问题转换为 DPN 可达问题. 最后给出了动态转换算法的伪代码.

3.1　时钟域

为了把描述连续时间的 TDPN 转换为离散的 DPN,引入域 R 的概念,主要是把 TDPN 中的栈符及其时钟抽象为 DPN 的栈符集,也就是说把时钟变量离散化. 域 R 是由一组项 r 组成,而项 r 是由字符集 Z 和关键点集 key 组成. 下文首先定义字符集,然后描述时钟关键点及时钟等价.

字符集 Z 包含普通字符集 Y 和记录字符集 $Y^{\cdot}$,描述为 $Z=Y\cup Y^{\cdot}$. 其中普通字符集 Y 包括:(1)时钟集 X;(2)栈字符 Γ;(3)全局变量集 G;(4)参考时钟字符 $\vdash$,用来记录时间流逝,除非进行出栈操作,否则始终为 0,故普通字符集 Y 可描述为 $Y=X\cup\Gamma\cup G\cup\{\vdash\}$. 记录字符集 $Y^{\cdot}$ 表示普通字符 Y 的时间流逝,设 $X^{\cdot}=\{x^{\cdot}\mid x\in X\}$ 表示记录全局时钟;$\Gamma^{\cdot}=\{a^{\cdot}\mid a\in\Gamma\}$ 表示记录栈字符;$G^{\cdot}=\{g^{\cdot}\mid g\in G\}$ 表示记录全局变量;$\{\vdash^{\cdot}\}$ 表示记录参考时钟字符集,主要来记录时间流逝,以便出栈操作时更新时间,故记录字符集描述为 $Y^{\cdot}=X^{\cdot}\cup\Gamma^{\cdot}\cup G^{\cdot}\cup\{\vdash^{\cdot}\}$.

时间离散化的主要思想是把连续正整数作为时钟区间划分的界限,即实数取整、小数部分大小比较来划分等价域. 然而实时系统的动作迁移及事件通常由某些时钟关键点所确定,也就是说时钟关键点是系统迁移及其他事件的时间触发点. 时钟关键点是非连续的正数,使用时钟关键点作为时钟区间划分界限,与原时钟等价技术相比,等价域减少一、两个数量级,并在多个时钟的情况下,划分的等价区域将减少几个数量级,转换后的等价状态空间将指数级缩减.

对任意与时间相关的字符时钟,时钟关键点都有各自的上限阀值,用 k_{Max} 表示,对所有大于 k_{Max} 的时钟值用符号 ∞ 表示. 对任意与时间相关的字符 $z_i\in Z$,其时间关键点集 $\text{key}_i=\{0,k_{i1},\cdots,k_{il},\cdots,k_{im},k_{i\text{Max}},\infty\}$,其中 $1<l<m,1<i\leqslant|Z|$. 根据上述字符集和关键点集,可得域 $R=r_1\cdots r_n\in(2^{\sum_{i=1}^{|Z|}z_i\times\text{key}_i})^{+}$.

基于时钟关键点的时钟等价技术把实数时钟值 $x=\lfloor x\rfloor+\mathrm{re}(x)$ 划分为两个部分:(1)关键点部分 $\lfloor x\rfloor$,表示对实数值 x 向下取关键点;(2)剩余部分 $\mathrm{re}(x)=x-\lfloor x\rfloor$.

假设 $x\in X$ 为 TDPN 中的任意时钟,用 $\theta(x)$ 表示时钟 x 的取值,$\lfloor\theta(x)\rfloor$ 表示 $\theta(x)$ 向下取关键点部分. 给定任意两个时钟 $x_i,x_j\in X$,两者时钟值等价当且仅当满足如下规则:

(1)$\theta(x_i)>k_{i\mathrm{Max}}$ 当且仅当 $\lfloor\theta(x_i)\rfloor=\infty$,即 x_i 的时钟值大于最大值,x_i 取无穷大 ∞.

(2)$k_{il}\leqslant\theta(x_i)<k_{i(l+1)}$ 当且仅当 $\lfloor\theta(x_i)\rfloor=k_{il}$,即 x_i 的时钟值小于关键点 $k_{i(l+1)}$ 且大于等于 k_{il} 时,x_i 取关键点 k_{il}.

(3)假定 $\lfloor\theta(x_i)\rfloor=k_{il}$,$\lfloor\theta(x_j)\rfloor=k_{jl}$,区域 $\mathrm{re}(\theta(x_i))<\mathrm{re}(\theta(x_j))$ 为等价域当且仅当 $\theta(x_i)-k_{il}<\theta(x_j)-k_{jl}$,即 x_i 取关键点 k_{il},x_j 取关键点 k_{jl},x_i 时钟值与关键点 k_{il} 的差值小于 x_j 时钟值与关键点 k_{jl} 的差值的这部分区域,为了描述简洁,记作 $\mathrm{re}(x_i)<\mathrm{re}(x_j)$. 同理区域 $\mathrm{re}(\theta(x_i))=\mathrm{re}(\theta(x_j))$、$\mathrm{re}(\theta(x_i))>\mathrm{re}(\theta(x_j))$ 亦为等价域.

与文献[4,13]提出时钟等价技术区别就是增加了区间划分跨度,故上述时钟等价规则能保证给出的时钟区域是等价的. 假设对于任意两个时钟 $x_i,x_j\in X$ 和 $\mathrm{key}=\{\mathrm{key}_1,\cdots,\mathrm{key}_n\}$,基于时钟关键点的时钟域等价规则. 譬如 $(0\leqslant\theta(x_i)<k_{i1},0\leqslant\theta(x_j)<k_{j1})$ 区域,$\lfloor\theta(x_i)\rfloor=0$,$\lfloor\theta(x_j)\rfloor=0$,且 $\mathrm{re}(\theta(x_i))>\mathrm{re}(\theta(x_j))$;在 $(k_{i1}\leqslant\theta(x_i)<k_{i2},k_{j1}\leqslant\theta(x_j)<k_{j2})$ 区域,$\lfloor\theta(x_i)\rfloor=k_{i1}$,$\lfloor\theta(x_j)\rfloor=k_{j1}$,且 $\mathrm{re}(\theta(x_i))<\mathrm{re}(\theta(x_j))$,则区间内的时钟值为等价时钟,如图 2 所示的阴影部分.

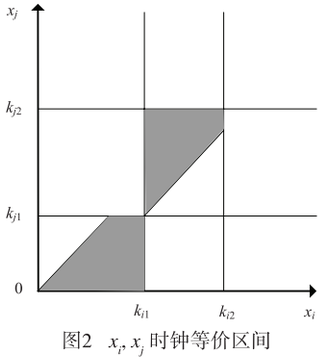

图2 x_i,x_j 时钟等价区间

3.2 构造规则

为了进一步缩减转换后的状态空间,采用 on-the-fly 的思想,动态地将连续模型 TDPN 抽象转换成离散 DPN 模型. 假设给定一个 TDPN $T=(P,\Gamma,\Delta,X)$,将 T 动态转换为 DPN $M=(P^M,\Gamma^M,\Delta^M)$,下文将描述 P^M,Γ^M,Δ^M 的构造:

(I)状态集 P^M:$p\in P^M\Leftrightarrow p\in P$,即 $P^M=P$.

(II)栈字符集 $\Gamma^M\subseteq 2^{\sum_{i=1}^{|Z|}z_i\times\mathrm{key}_i}$:若 $a\in\{\Gamma,\vdash\}$,则 $(a,\lfloor\theta(a)\rfloor)\in\Gamma^M$,且 $(a^{\cdot},\lfloor\theta(a^{\cdot})\rfloor)\in\Gamma^M$.

(III)迁移关系 Δ^M 的构造:

假设 TDPN 含有 n 个栈,为了描述方便,仅描述序列号为 i 的下推系统执行入栈和出栈操作,其它栈操作与之类似. 假定 TDPN 含有全局变量 g,时钟变量 x,栈内容 $\omega=\{\omega_1\cdots\omega_i\cdots\omega_n\}$,其中 ω_i 表示 i 号下推系统的栈内容,且栈深度为 l,栈底编号为 1,栈顶编号为 l,用 $\omega_{il}|_\Gamma$ 表示 ω_i 投影在 Γ 的栈顶字符. $\lfloor\theta(g)\rfloor$、$\lfloor\theta(x)\rfloor$、$\lfloor\theta(\omega_{il}|_\Gamma)\rfloor$ 表示各自"年龄"值在时钟等价下的关键点,从而可知对应在 M 的当前时钟等价域 $R_l=(g,\lfloor\theta(g)\rfloor)\cup(g^{\cdot},\lfloor\theta(g^{\cdot})\rfloor)\cup\sum_{i=1}^{n}(\omega_{il}|_\Gamma\lfloor\theta(\omega_{il}|_\Gamma)\rfloor)\cup\sum_{i=1}^{n}((\omega_{i(l-1)}|_\Gamma)^{\cdot},\lfloor\theta((\omega_{i(l-1)}|_\Gamma)^{\cdot})\rfloor)\cup(x,\lfloor\theta(x)\rfloor)\cup(x^{\cdot},\lfloor\theta(x^{\cdot})\rfloor)\cup(\vdash,0)\cup(\vdash^{\cdot},\lfloor\theta(\vdash^{\cdot})\rfloor)$,其中 $\sum_{i=1}^{n}(\omega_{il}|_\Gamma,\lfloor\theta(\omega_{il}|_\Gamma)\rfloor)$ 表示域 R_l 记录栈顶字符,$\vdash$ 表示域 R_l 的参考时钟字符,$\vdash^{\cdot}$ 表示域 R_l 对应的时间流逝字符.

给定 TDPN 格局 $\gamma=(\bar{g},\langle p,\bar{\omega}\rangle,\bar{x})$,$\phi=(\gamma,\mathrm{op},\gamma')\in\Delta$ 表示 T 的格局迁移,对应 M 的格局迁移可表示为 $\langle p,R_l\rangle\xrightarrow{\mathrm{op}'}\langle p',R'_l\rangle$,其中 p 和 p' 与 T 中状态相同,分别表示格局迁移前后的状态;$R_l=\{R_{1l}\cdots R_{il}\cdots R_{nl}\}$ 表示栈顶域,其中 R_{il} 表示 i 号下推系统的栈顶域,R_l 和 R'_l 分别表示格局迁移前后的栈顶域;动作迁移集 op′对应于 T 的 op,下面根据不同的动作 op 构造 R'_l:

(1)op = nop:若 $(\gamma,\mathrm{nop},\gamma')\in\Delta$,表示在 T 中空操作只改变状态结点,不改变其他变量值,则 M 中应存在对应迁移 $(\langle p,R_l\rangle\xrightarrow{\mathrm{nop}}\langle p',R_l\rangle)\in\Delta^M$,在 M 中格局迁移也只改变状态,域 R_l 保持不变.

(2)op = $(x\leftarrow I)$:若 $\forall(\gamma,x\leftarrow I,\gamma')\in\Delta$,则 M 中存在迁移 $(\langle p,R_l\rangle\xrightarrow{=\lfloor\theta(x)'\rfloor}\langle p',R'_l\rangle)\in\Delta^M$,表示 DPN 域 R_l 中时钟为 x 的项执行重新赋值操作 $x\leftarrow\lfloor\theta(x)'\rfloor$,$\theta(x)'\in I$,通过构造获得域 R'_l. 具体执行过程如下:

(a)域 R_l 出栈,获得 R_l 里的项 $(x,\lfloor\theta(x)\rfloor)$,重置 $\theta(x)$ 为 $\theta(x)'$,形成新项 $(x,\lfloor\theta(x)'\rfloor)$;

(b)项 $(x,\lfloor\theta(x)'\rfloor)$ 代替域 R_l 中的项 $(x,\lfloor\theta(x)\rfloor)$,获得域 R'_l,并入栈,转换到新的状态 p'.

(3)op = $(\mathrm{Time}\leftarrow v)$:若 $(\gamma,\mathrm{Time}\leftarrow v,\gamma')\in\Delta$,当且仅当 M 中存在 $(\langle p,R_l\rangle\xrightarrow{\vdash}\langle p',R_l^+\rangle)\in\Delta^M$,表示域 R_l 中除了参考时钟项 $(\vdash,0)$,其余所有项的时钟值加上时间流逝 v,通过构造获得域 R_l^+. 具体执行步骤为:

(a)域 R_l 出栈,除了参考时钟,全部加上时间流逝

v,$(g,\lfloor\theta(g)+v\rfloor)$、$(\omega_{il}|_\Gamma,\lfloor\theta(\omega_{il}|_\Gamma)+v\rfloor)$、$(x,\lfloor\theta(x)+v\rfloor)$表示对应于$g$、$\omega_{il}$、$x$的新普通项,$(g^{\cdot},\lfloor\theta(g^{\cdot})+v\rfloor)$、$((\omega_{i(l-1)}|_\Gamma)^{\cdot},\lfloor(\theta(\omega_{l-1}|_\Gamma)^{\cdot})+v\rfloor)$、$(x^{\cdot},\lfloor\theta(x^{\cdot})+v\rfloor)$各表示相应的新记录项,$(\vdash^{\cdot},\lfloor\theta(\vdash^{\cdot})+v\rfloor)$表示参考时钟记录项;

(b)新项代替原来项,得到域R_l^+,并入栈,转换到新的状态p'.

(4)op = push(a,I):若$(\gamma,\text{push}(a,I),\gamma')\in\Delta$,当且仅当$M$存在$(\langle p,R_l\rangle\xrightarrow{\text{push}(\langle a,\lfloor\theta(a)\rfloor\rangle)}\langle p',R'_l\rangle)\in\Delta^M$,表示对$i$号下推系统进行入栈操作,将字符为$a$,时钟值为$\lfloor\theta(a)\rfloor$的项入栈域$R_{il}$,来构造域$R_{i(l+1)}$. 具体过程如下:

(a)从域R_{il}获得项$(\omega_{il}|_\Gamma,\lfloor\theta(\omega_{il}|_\Gamma)\rfloor)$和$((\omega_{i(l-1)}|_\Gamma)^{\cdot},\lfloor\theta((\omega_{i(l-1)}|_\Gamma)^{\cdot})\rfloor)$.

(b)$(a,\lfloor\theta(a)\rfloor)$、$((\omega_{il}|_\Gamma)^{\cdot},\lfloor\theta((\omega_{il}|_\Gamma)^{\cdot})\rfloor)$分别代替$((\omega_{il}|_\Gamma),\lfloor\theta(\omega_{il}|_\Gamma)\rfloor)$、$((\omega_{i(l-1)}|_\Gamma)^{\cdot},\lfloor\theta((\omega_{i(l-1)}|_\Gamma)^{\cdot})\rfloor)$,获得域$R_{i(l+1)}$,并入栈,转换到新的状态$p'$.

(5)op = pop(a,I):若$(\gamma,\text{pop}(a,I),\gamma')\in\Delta$,当且仅当$M$存在$(\langle p,R_l\rangle\xrightarrow{\text{pop}(\langle a,|\theta(a)|\rangle)}\langle p',R'_l\rangle)\in\Delta^M$,表示在域$R_{il}$中出栈栈字符为$a$,且时钟值为$\theta(a)\in I$的项,通过构造获得域$R'_{il}$. 具体步骤描述如下:

(a)出栈域R_{il}和域$R_{i(l-1)}$,获得域R_{il}里的项$(\vdash^{\cdot},\lfloor\theta(\vdash^{\cdot})\rfloor)$;

(b)域$R_{i(l-1)}$中所有项的时钟值加上$\theta(\vdash^{\cdot})$,获得域$R'_{i(l-1)}$;

(c)通过R_{il}和$R'_{i(l-1)}$获得域R'_{il},R'_{il}项分别为:普通栈字符项来自域$R'_{i(l-1)}$;普通时钟项、全局变量项来自域R_{il};记录项全部来自域$R'_{i(l-1)}$;

(d)入栈域R'_{il},转换到新的状态p'.

(6)op = $\triangleright$:若$(\gamma,\triangleright,\gamma')\in\Delta$,当且仅当$M$中存在$(\langle p,R_l\rangle\xrightarrow{\triangleright}\langle p',R'_l\rangle)\in\Delta^M$,表示创建新线程来构造域$R'_l$. 假设TDPN模型,动态创建新下推系统的栈编号为$n+1$. 具体执行步骤如下:

(a)基于关键点的时钟等价优化技术,依据域R_l可得普通字符项$(\omega_{(n+1)l}|_\Gamma,\lfloor\theta(\omega_{(n+1)l}|_\Gamma)\rfloor)$,记录项$((\omega_{(n+1)(l-1)}|_\Gamma)^{\cdot},\lfloor\theta((\omega_{(n+1)(l-1)}|_\Gamma)^{\cdot})\rfloor)$;

(b)把项$(\omega_{(n+1)l}|_\Gamma,\lfloor\theta(\omega_{(n+1)l}|_\Gamma)\rfloor)$和$((\omega_{(n+1)(l-1)}|_\Gamma)^{\cdot},\lfloor\theta((\omega_{(n+1)(l-1)}|_\Gamma)^{\cdot})\rfloor)$加入域$R_l$,得到域$R'_l$,并入栈,转换到新的状态$p'$.

3.3　转换算法

基于时钟等价优化和动态转换思想,提出TDPN $T=(P,\Gamma,\Delta,X)$转换为DPN $M=(P^M,\Gamma^M,\Delta^M)$的算法,该算法根据$T$的迁移关系$\Delta$,通过上节的转换规则,穷尽地计算在$M$对应的迁移关系$\Delta^M$. 算法的输入是连续的TDPN T,输出是离散的DPN M. 假设TDPN的初始格局为$\gamma_{\text{init}}=\langle\bar{g}_{\text{init}},\langle p_{\text{init}},\varepsilon\rangle,\bar{x}_{\text{init}}\rangle$,每个栈内容初始都为空,对应构造$M$初始域$R_{\text{init}}$.

若T的迁移关系集合存在$\phi=(\gamma,\text{op},\gamma')\in\Delta$,其格局包含全局变量$\bar{g}$,栈字符串$\bar{\omega}_i$,时钟$\bar{x}$. M的当前格局为$\beta=\langle p,R_l\rangle$,域$R_l$包含$(g,\lfloor\theta(g)\rfloor)$、$(\omega_{il}|_\Gamma,\lfloor\theta(\omega_{il}|_\Gamma)\rfloor)$、$(x,\lfloor\theta(x)\rfloor)$、$(\vdash,0)$表示$g$、$\omega_{il}$、$x$、$\vdash$对应的普通项,$(g^{\cdot},\lfloor\theta(g^{\cdot})\rfloor)$、$((\omega_{i(l-1)}|_\Gamma)^{\cdot},\lfloor\theta((\omega_{i(l-1)}|_\Gamma)^{\cdot})\rfloor)$、$(x^{\cdot},\lfloor\theta(x^{\cdot})\rfloor)$、$(\vdash^{\cdot},\lfloor\theta(\vdash^{\cdot})\rfloor)$表示相应的记录项. 根据$\phi$和$R_l$可动态构造域$R'_l$,且$\Delta^M$存在迁移关系$\langle p,R_l\rangle\xrightarrow{\text{op}'}\langle p',R'_l\rangle$,可得基于时钟等价优化技术的动态转换算法,如算法1所示.

算法1　TDPN转化为DPN算法

输入:TDPN $T=(P,\Gamma,\Delta,X)$

输出:对应的DPN $M=(P^M,\Gamma^M,\Delta^M)$

1. Let $\gamma_{\text{init}}=\langle\bar{g}_{\text{init}},\langle p_{\text{init}},\varepsilon\rangle,\bar{x}_{\text{init}}\rangle$　　//TDPN T栈内容初始为空
2. $R_{\text{init}}=\{(g,0)\mid g\in G\cup G^{\cdot}\}\cup\{(\varepsilon_i,0),(\varepsilon_i^{\cdot},0)\}\cup\{(x,0)\mid x\in X\cup X^{\cdot}\}\cup\{\vdash,\vdash^{\cdot}\}$//其中$\varepsilon_i$表示$i$号下推系统的栈为空栈
3. Move $(p_{\text{init}},R_{\text{init}})$ into Δ^M
4. for all $\phi\in\Delta$
5. while ($\Delta\neq\varnothing$)
6. begin
7. Remove ϕ from Δ
8. op form ϕ
9. case (op)
10. begin
11. op = nop:　//空操作,对应M的只改变状态,域R_l不变
12. Move $\langle p,R_l\rangle\xrightarrow{\text{nop}}\langle p',R_l\rangle$ into Δ^M; break;
13. op = (x←I):　//时间重置操作,把域R_l中全局时钟x值$\theta(x)$更新为$\theta(x)'$
14. $R'_l=R_l-(x,\lfloor\theta(x)\rfloor)+(x,\lfloor\theta(x)'\rfloor)$;
15. Move $\langle p,R_l\rangle\xrightarrow{=\lfloor\theta(x)'\rfloor}\langle p',R'_l\rangle$ into Δ^M;break;
16. op = (Time←v):　//时间流逝操作,域R_l里除(项其余都增加时间流逝值v
17. $R_l^+=R_l-\{(g,\lfloor\theta(g)\rfloor),(g^{\cdot},\lfloor\theta(g^{\cdot})\rfloor),(\omega_{il}|_\Gamma,\lfloor\theta(\omega_{il}|_\Gamma)\rfloor),((\omega_{i(l-1)}|_\Gamma)^{\cdot},\lfloor\theta((\omega_{i(l-1)}|_\Gamma)^{\cdot})\rfloor),(x,\lfloor\theta(x)\rfloor),(x^{\cdot},\lfloor\theta(x^{\cdot})\rfloor),(\vdash^{\cdot},\lfloor\theta((^{\cdot})\rfloor)\}+\{(g,\lfloor\theta(g)+v\rfloor),(g^{\cdot},\lfloor\theta(g^{\cdot})+v\rfloor),(\omega_{il}|_\Gamma,\lfloor\theta(\omega_{il}|_\Gamma)+v\rfloor),((\omega_{i(l-1)}|_\Gamma)^{\cdot},\lfloor\theta((\omega_{i(l-1)}|_\Gamma)^{\cdot})+v\rfloor),(x,\lfloor\theta(x)+v\rfloor),(x^{\cdot},\lfloor\theta(x^{\cdot})+v\rfloor),(\vdash^{\cdot},\lfloor\theta(\vdash^{\cdot})+v\rfloor)\}$;
18. Move $\langle p,R_l\rangle\xrightarrow{\vdash}\langle p',R_l^+\rangle$ into Δ^M;break;
19. op = push(a,I):　//入栈操作,i号栈入栈字符a及其“年龄”为$\theta(a)\in I$的项

20. $R'_{il}=R_{il}-\{(\omega_{il}|_\Gamma,\lfloor\theta(\omega_{il}|_\Gamma)\rfloor),((\omega_{i(l-1)}|_\Gamma)^\cdot,\lfloor\theta((\omega_{i(l-1)}|_\Gamma)^\cdot)\rfloor)\}+\{(a,\lfloor\theta(a)\rfloor),((\omega_{il}|_\Gamma)^\cdot,\lfloor\theta(\omega_{il}|_\Gamma)^\cdot)\rfloor)\}$

21. Move $\langle p,R_l\rangle\xrightarrow{\text{push}(\langle a,|\theta(a)|\rangle)}\langle p',R'_l\rangle$ into Δ^M; break;

22. op = pop(a,I): //出栈操作,i 号栈出栈字符 a 及其"年龄"为 $\theta(a)\in I$ 的项

23. $R'_{i(l-1)}=R_{i(l-1)}-\{(g,\lfloor\theta(g)\rfloor),(g^\cdot,\lfloor\theta(g^\cdot)\rfloor),(\omega_{il}|_\Gamma,\lfloor\theta(\omega_{il}|_\Gamma)\rfloor),((\omega_{i(l-1)}|_\Gamma)^\cdot,\lfloor\theta((\omega_{i(l-1)}|_\Gamma)^\cdot)\rfloor),(x,\lfloor\theta(x)\rfloor),(x^\cdot,\lfloor\theta(x^\cdot)\rfloor),(\vdash,0),(\vdash^\cdot,\lfloor\theta(\vdash^\cdot)\rfloor)\}+\{(g,\lfloor\theta(g)+\theta(\vdash^\cdot)\rfloor),(g^\cdot,\lfloor\theta(g^\cdot)+\theta(\vdash^\cdot)\rfloor),(\omega_{i(l-1)}|_\Gamma,\lfloor\theta(\omega_{i(l-1)}|_\Gamma)+\theta(\vdash^\cdot)\rfloor),(\omega_{i(l-2)}|_\Gamma)^\cdot,\lfloor\theta(\omega_{i(l-2)}|_\Gamma)^\cdot+\theta(\vdash^\cdot)\rfloor),(x,\lfloor\theta(x)+\theta(\vdash^\cdot)\rfloor),(x^\cdot,\lfloor\theta(x^\cdot)+\theta(\vdash^\cdot)\rfloor),(\vdash,\theta(\vdash^\cdot)),\vdash^\cdot,\lfloor\theta(\vdash^\cdot)+\theta(\vdash^\cdot)\rfloor)\}$

24. $R'_{il}=\{R_{il}-\{(g^\cdot,\lfloor\theta(g^\cdot)\rfloor),(\omega_{il}|_\Gamma,\lfloor\theta(\omega_{il}|_\Gamma)\rfloor),((\omega_{i(l-1)}|_\Gamma)^\cdot,\lfloor\theta((\omega_{i(l-1)}|_\Gamma)^\cdot)\rfloor),(x^\cdot,\lfloor\theta(x^\cdot)\rfloor),(\vdash^\cdot,\lfloor\theta(\vdash^\cdot)\rfloor)\}\}+\{R'_{i(l-1)}-\{(g,\lfloor((g)+\theta(\vdash^\cdot)\rfloor),(\omega_{i(l-1)}|_\Gamma,\lfloor\theta(\omega_{i(l-1)}|_\Gamma)+\theta(\vdash^\cdot)\rfloor),(x,\lfloor\theta(x)+\theta(\vdash^\cdot)\rfloor),(\vdash,\theta(\vdash^\cdot))\}\}+\{(\omega_{i(l-1)}|_\Gamma,\lfloor\theta(\omega_{i(l-1)}|_\Gamma)\rfloor)\}$;

25. Move $\langle p,R_l\rangle\xrightarrow{\text{pop}(\langle a,|\theta(a)|\rangle)}\langle p',R_l\rangle$ into Δ^M; break;

26. op = $\triangleright$: //动态创建栈编号为 $n+1$ 的新线程

27. $R'_l=R_l+\{(\omega_{(n+1)l}|_\Gamma,\lfloor\theta(\omega_{(n+1)l}|_\Gamma)\rfloor),((\omega_{(n+1)(l-1)}|_\Gamma)^\cdot,\lfloor\theta((\omega_{(n+1)(l-1)}|_\Gamma)^\cdot)\rfloor)\}$;

28. Move $\langle p,R_l\rangle\xrightarrow{\triangleright}\langle p',R'_l\rangle$ into Δ^M; break;

29. endcase;

30. endwhile;

转换算法中第 1 和 2 行分别表示对工作线程的格局和域的初始化,从第 4 行开始,针对 T 的格局迁移关系 Δ,穷尽计算 M 中用域表示的迁移关系 Δ^M. 其中第 11 和 12 行表示空操作迁移,对应 M 只改变状态,域不变. 第 13 至 15 行描述时钟重置操作迁移,域 R_l 中的时钟 x 的值$\lfloor\theta(x)\rfloor$重置为$\lfloor\theta(x)'\rfloor$. 第 16 至 18 行描述时间流逝迁移,域 R_l 中除参考时钟项($\vdash$,0),其余项的时钟值,全部加上时间流逝 v. 第 19 至 21 行描述入栈操作迁移,压入字符 a. 第 22 至 25 行描述出栈操作迁移,其中 $R'_{i(l-1)}$ 表示域 $R_{i(l-1)}$ 所有项都加上域 R_{il} 的时间流逝 $\theta(\vdash^\cdot)$. 第 26 至 28 行描述动态创建线程迁移,创建的新线程为 $n+1$. 对于 TDPN T,该算法是可终止的,且该算法的时间复杂度,与项字符集和关键点集的笛卡尔积呈指数关系,与程序的大小呈指数关系.

4 转换正确性证明

通过上述时钟等价优化技术及动态转换方法,把 TDPN 可达性问题转化成 DPN 可达性问题. 为了描述 TDPN T 可被转换为 DPN M,需证明从 T 转化成 M 的正确性,即状态 p'_F 在 M 可达当且仅当转换前所对应的状态 p_F 在 T 可达.

定义 1(可达性) 设迁移系统 TDPN T,$\gamma_{\text{init}}=\langle\bar{g}_{\text{init}},\langle p_{\text{init}},\varepsilon\rangle,\bar{x}_{init}\rangle$ 为 T 的初始格局,其中 $\bar{g}_{\text{init}}$ 为全局变量初始值;p_{init} 为初始状态;ε 为栈初始值(表示栈空);$\bar{x}_{\text{init}}$ 为初始时钟(赋值为 0),目标格局 $\gamma_F=\langle\bar{g}_F,\langle p_F,\bar{\omega}_F\rangle,\bar{x}_F\rangle$. 如果 T 存在格局迁移 $\gamma_{\text{init}}\xrightarrow{\text{op}}\gamma_F$,那么状态 p_F 在 T 可达.

定理 1 对于 T 任意一个格局 γ,通过时钟等价转化,在 M 都存在与之对应的格局 β.

证明:设 M 的一个格局 $\theta=\langle p,R\rangle$,$T$ 的一个格局 $\gamma=\langle\bar{g},\langle p',\bar{\omega}\rangle,\bar{x}\rangle$,其中 $\bar{\omega}\in(2^{\sum_{i=1}^{|\Omega|}z_i\times\text{key}_i})^+$,假设 S 为迁移系统 T 此刻变量集合,S 经过时钟域等价转换成 M 中的域 R. 设 $\bar{\omega}=\langle a_1,v_1\rangle\cdots\langle a_n,v_n\rangle$,$R=R_0R_1\cdots R_n$ 和 S 的值 θ(即 $\theta|=S$),由于下述表达式成立:

(1) $p'=p$

(2) $(g,\lfloor\theta(g)\rfloor)\in R$ iff $\bar{g}\in S$

(3) $(\omega_{il}|_\Gamma,\lfloor\theta(\omega_{il}|_\Gamma)\rfloor)\in R$ iff $\bar{\omega}_i\in S$

故 $\gamma|=_S\beta$,即对于 T 任意一个格局 γ,通过时钟域编码转化后,在 M 都存在与之对应的格局 β.

证毕.

设 $R=R_0R_1\cdots R_n$ 是 M 栈域集合上的一组域. 对于域 R_1 和 R_2,如果 R_1 是 R_2 严格偏序关系,记为 $R_1\prec R_2$;如果 R_1 是 R_2 非严格偏序关系,记为 $R_1\preccurlyeq R_2$. 对于域集 R,如果 $R_i\prec R_{i+1}$ s. t. $\forall i;0\leqslant i\leqslant n$,则称 R 为相关域,如果 $R_i\preccurlyeq R_{i+1}$,则称 R 为弱相关域. 如果 R 为(弱)相关域,则格局 $\beta=\langle p,R\rangle$ 为(弱)相关格局. 对于弱相关域 $R=R_0R_1\cdots R_n$ 和域 $R'=R'_0\,R'_1\cdots\,R'_n$,如果存在 $R'_n=R_n$,$R'_i\in R_i^+$(其中 R_i^+ 是 R_i 的时间迁移域),且 $R'_i\prec R_{i+1}$ s. t. $\forall i;0\leqslant i\leqslant n$,则域 R'是域 R 的强相关域. 给定一个在 M 的相关格局 $\beta=\langle p,R\rangle$,如果域 R'是域 R 的强相关域,则格局 $\beta=\langle p,R'\rangle$ 就是 β 的强相关格局.

为了证明转换前后的可达性等价,由于 TDPN 中的单个时间下推系统转换类似于 TPDA,故引入文献[13]的引理用于定理 2 证明,具体两个引理如下:

引理 1 对于属于 M 的任意一个正则可达格局 β,β 的强相关格局 $\beta'=\langle p,R'\rangle$,$S$ 为 T 时刻变量集合,在 T 里必定存在与之对应格局 γ,存在 $\gamma|=_S\beta$ 并且 $\gamma_{\text{init}}\xrightarrow{\text{op}}\beta$.

引理 2 对于属于 T 的任意一个格局 γ,在 M 里必定存在对应格局 β,至少存在一个 β 的强相关格局 $\beta'=\langle p,R'\rangle$,并且存在域 R'的转换集合 S,那么存在 $\gamma|=_S\beta$ 并且 $\beta_{\text{init}}\xrightarrow{\text{op}'}\beta$.

定理 2 p'_F 在 DPN M 可达当且当仅状态 p_F 在

TDPN T 可达.

证明:先证充分性:状态 p_F 在 TDPN 可达⇐其转换状态 p'_F 在 DPN 可达.

如果目标状态 p'_F 在 M 是可达,那么就存在一个正则可达格局 $\beta(p'_F$ 为格局 β 的状态). 由于 DPN M 所有可达格局都是弱相关,即可达格局 β 为弱相关格局,因此至少存在一个对应的强相关格局 $\beta'=\langle p,R'\rangle$. 由定律 1 可知在迁移系统 M 的一个正则格局 β,存在一个强相关格局 β 和转换成 R' 的集合 S,在 T 里必定存在与之对应格局 γ,存在 $\gamma \models_S \beta$ 并且 $\gamma_{\text{init}} \xrightarrow{\text{op}} \beta$,即状态 $p_F(p_F$ 为格局 γ 的状态)在 T 可达.

再证必要性:状态 p_F 在 TDPN 可达(其转换状态 p'_F 在 DPN 可达.

如果目标状态 p'_F 在 T 是可达,由定理 1 可知在 M 里必定存在与之对于格局 $\beta(p'_F$ 为格局 β 的状态),因此至少存在一个强相关格局 $\beta'=\langle p,R'\rangle$. 由定律 2 可知在迁移系统 T 的一个格局 γ,存在一个强相关格局 β' 和转换成 R' 的域集合 S,在 M 里必定存在与之对应格局 β,存在 $\gamma \models_S \beta$ 并且 $\beta_{\text{init}} \xrightarrow{\text{op}'} \beta$,即状态 $p'_F(p'_F$ 为格局 β 的状态)在 M 可达.

因此,状态 p_F 在 TDPN 可达当且仅当其转换状态 p'_F 在 DPN 可达.

证毕.

5　实例分析

本节首先给出 TDPN 模型根据不同类型格局迁移的实例,然后详细阐述如何根据时钟等价优化及动态转换技术,将其转换为易求解可达性的 DPN 模型.

5.1　TDPN 实例描述

假定一个时间动态下推系统 $T=(P,\Gamma,\Delta,X)$,其中状态集合 $P=\{p_0,\cdots,p_5\}$,时钟变量集合 $X=\{x_1,x_2\}$,栈字符集合 $\Gamma=\{a_1,b_1,a_2,b_2,c_2,a_3,\varepsilon_3\}$,格局迁移集合 $\Delta=\{\Delta_1,\cdots,\Delta_5\}$,预先定义的全局变量集合 $G=\{g\}$,TDPN 模型的格局迁移如图 3 所示.

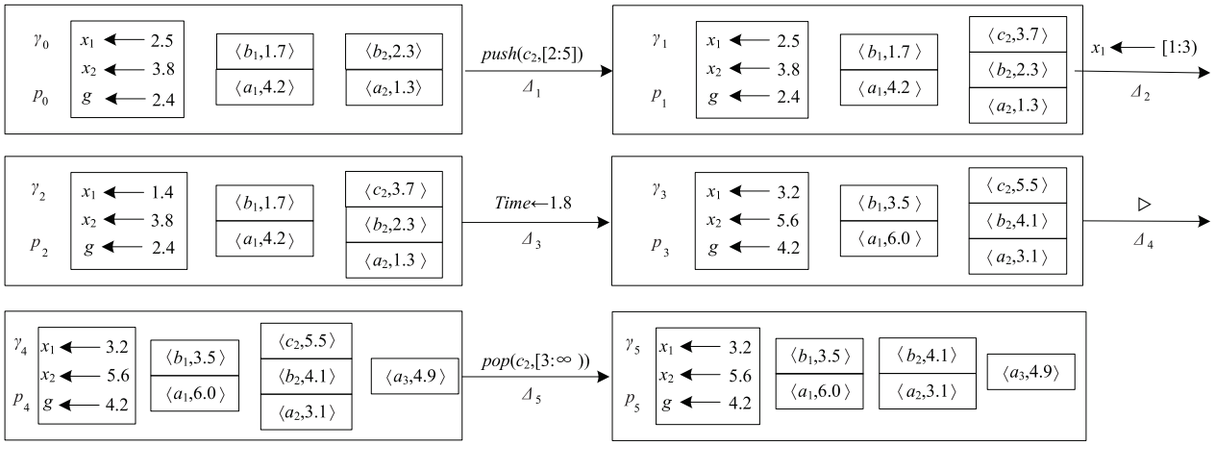

图3　TDPN 的格局迁移

图中格局之间迁移具体表示如下:

(1)入栈操作 push(c_2,[2:5]):从格局 γ_0 到格局 γ_1,2 号栈入栈"年龄"为 3.7∈[2:5]的字符 c_2.

(2)时钟重置操作 x_1←[1:3):从格局 γ_1 到格局 γ_2,全局时钟 x_1 的值更新为 1.4∈[1:3).

(3)时间流逝操作 Time←1.8:从格局 γ_2 到格局 γ_3,所有时钟值、全局变量的"年龄"、栈字符的"年龄"都增加实数 1.8.

(4)动态创建操作▷:从格局 γ_3 到格局 γ_4,TDPN 系统执行动态线程创建操作,创建的新下推系统入栈"年龄"为 4.9 的字符 a_3 到该系统的栈中.

(5)出栈操作 pop(c_2,[3:∞)):从格局 γ_4 到格局 γ_5,执行出栈操作前,先判断出栈条件,即栈顶字符 c_2 的值是否属于区间[3:∞),图 3 中 $c_2=5.5$ 满足条件,可执行出栈操作.

5.2　转换为 DPN

基于关键点的时钟等价优化技术,按照 3.2 节的动态转换构造方法,可构造出对应的 DPN M 模型. 为了描述方便,只模拟 2 号下推系统的转换. 其它下推系统相互独立并发运行,其转换与之类似,设 M 进行迁移之前域 R_1 如图 4 所示. 域 R_1 对应 TDPN 中格局 γ_0,R_1 由普通项(全局变量 g、时钟 x_1,x_2、栈字符 b_2、参考时钟字符 ⊢)和记录项(记录全局变量 g'、记录时钟 x_1',x_2'、记录栈字符 a_2'、记录参考字符 $\vdash'$)组成,其中 a_2' 表示该栈的下一个栈顶字符,$\vdash'$ 表示从入栈开始时间的迁移,记录时间流逝.

图 3 中 TDPN 的格局迁移转换成 DPN 的格局迁移如下:

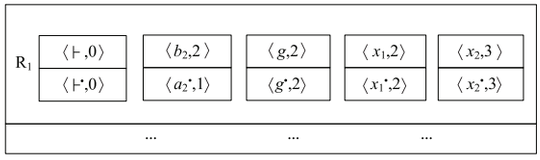

图4 DPN域R_1

(1)图3中Δ_1描述从格局γ_0到格局γ_1执行入栈操作,将$c_2=3.7$入栈2号栈. 图5(a)所示的域R_2是经过入栈操作后获得的新域. 当进行入栈操作时,域R_1的普通项都会以记录项保存在域R_2中,即域R_1中时钟x_1,x_2、g和b_2的记录项$x_1^{\cdot}$、$x_2^{\cdot}$、$g^{\cdot}$和$b_2^{\cdot}$分别保存到域R_2中. 而记录项只随着时间流逝进行更新,其值不会随赋值时钟重置、入栈、出栈和动态迁移而改变.

(2)图3中Δ_2描述从格局γ_1到格局γ_2执行时钟重置操作,将时钟$x_1=2.5$重置为$x_1=1.4$,其它变量保持不变. 相应地在域R_2中只需修改含有时钟x_1的项,其它项保持不变. 时钟x_1重置,其值改变,相应改变其关键点部分$\lfloor x_1 \rfloor$和剩余部分$\mathrm{re}(x_1)$. 根据时钟等价规则,此时x_1的剩余部分$\mathrm{re}(x_1)=0.4$,与g的剩余部分$\mathrm{re}(g)$相同,放在同一项集合里,更新后获得域R_3,如图5(b)所示.

(3)图3中Δ_3描述从格局γ_2到格局γ_3执行时间流逝操作,将会更新所有时钟值、全局变量和栈字符的"年龄". 由于动态转换仅关注当前栈顶和下一个栈顶字符,故转换过程将当前栈顶字符以普通项,下一个栈顶字符以记录项保存在域中. 执行图3所示时间流逝time←1.8操作时,栈中每个字符的"年龄"都增加1.8,在DPN中该操作只更新栈顶域中的值,即只更新图5(b)所示的栈顶域R_3中的栈字符值,并用$\vdash^{\cdot}$记录了时间流逝,在出栈操作时,可根据$\vdash^{\cdot}$对下一层栈字符值进行更新. 域R_3进行更新时,除参考时钟$\vdash$的值仍然为常数0,其它所有项的时钟值都增加时间流逝时间,更新后获得域R_4,其下层域不作更新,如图5(c)所示.

(4)图3中Δ_4描述从格局γ_3到格局γ_4执行动态创建迁移,创建一个新下推系统. 相应在DPN域中,增加新创建下推系统的栈字符信息,如图5(d)所示的域R',在域R'中增加动态创建线程的普通栈字符项a_3和记录栈字符项$\varepsilon_3^{\cdot}$,其中a_3表示新线程当前栈顶字符,由于该栈现在只有一个字符,所以用ε_3表示下一个栈顶字符.

(5)图3中Δ_5描述从格局γ_4到格局γ_5执行出栈操作. 在栈顶字符满足出栈条件的情况下,执行出栈操作,获得域R_5,如图5(e)所示. TDPN模型随着时间流逝,非栈顶字符"年龄"也需要同步更新,然而动态转换

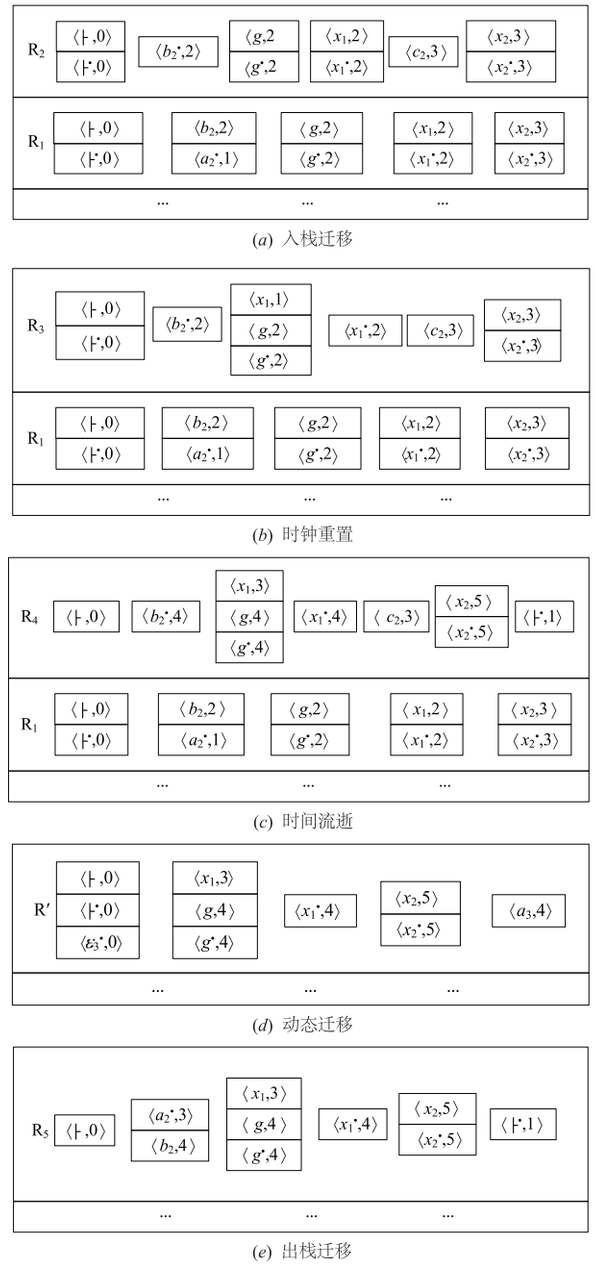

图5 TDPN的格局迁移转换成DPN的格局迁移

过程中仅关心当前栈顶字符和下一个栈顶字符,则需要结合图5(c)所示的域R_1和R_4生成新的域R_5,转换过程需要经过两步:

(a)域R_1的更新:域R_1除参考时钟项,其余所有项的值都加上域R_4中的记录参考时钟项的值.

(b)通过出栈R_1和R_4并入栈得到域R_5,其中域R_5分成三个部分:普通时钟项$\{(x_1,3),(x_2,5)\}$、普通全局变量项$\{(g,4)\}$和参考时钟项$\{(\vdash,0)\}$来自R_4;普通栈顶字符项$\{(b_2,4)\}$来自R_1;记录项$\{(x_1^{\cdot},4),(x_2^{\cdot},5),(g^{\cdot},4),(a_2^{\cdot},3),(\vdash^{\cdot},1)\}$全部来自$R_1$.

6 总结与未来工作

把时钟引入 DPN,给出了能对带动态创建的实时并发递归系统建模的模型——TDPN. 基于关键点的时钟等价优化技术和 on-the-fly 技术,仅关心栈顶及下一层的域状态转换,把 TDPN 可达性问题转化成 DPN 可达性问题. 下一步的主要工作:考虑使用时间带的方法对时钟抽象,和抽象解释框架更加形式化的描述转换工作,以及开发工具实现转换,并利用现有的验证工具(如 Moped)对转换后的抽象模型进行验证.

参考文献

[1] Song F, Touili T. Model checking dynamic pushdown networks[A]. Programming Languages and Systems[C]. Switzerland: Springer International Publishing, 2013. 33 – 49.

[2] Atig M F, Bouajjani A, Qadeer S. Context-bounded analysis for concurrent programs with dynamic creation of threads [A]. Tools and Algorithms for the Construction and Analysis of Systems[C]. Berlin Heidelberg: Springer, 2009. 107 – 123.

[3] Schwoon S. Model-checking Pushdown Systems[D]. Munich: Technical University of Munich, 2002.

[4] Alur R, Dill D. A theory of timed automata[J]. Theoretical Computer Science, 1994, 126(2): 183 – 235.

[5] Bengtsson J, Yi W. Timed automata: Semantics, algorithms and tools[A]. Lectures on Concurrency and Petri Nets [C]. Berlin Heidelberg: Springer, 2004. 87 – 124.

[6] Trivedi A, Wojtczak D. Recursive timed automata[A]. Automated Technology for Verification and Analysis[C]. Berlin Heidelberg: Springer, 2010. 306 – 324.

[7] Abdulla P A, Atig M F, Stenman J. Zenoness for timed pushdown automata[J]. Electronic Proceedings in Theoretical Computer Science, 2013, 140: 35 – 47.

[8] Li G Q, Cai X J, Mizuhito O, Yuen S J. Nested timed automata[A]. Formal Modeling and Analysis of Timed Systems [C]. Berlin Heidelberg: Springer, 2013. 168 – 182.

[9] Bouajjani A, Olm M M, Touili T. Regular symbolic analysis of dynamic networks of pushdown systems[A]. CONCUR 2005—Concurrency Theory[C]. Berlin Heidelberg: Springer, 2005. 473 – 487.

[10] Bollig B, Kuske D, Mennicke R. The complexity of model checking multi-stack systems[A]. Logic in Computer Science[C]. Washington: IEEE Computer Society, 2013. 163 – 172.

[11] Lammich P, Olm M M, Seidl H. Contextual locking for dynamic pushdown networks[A]. Static Analysis[C]. Berlin Heidelberg: Springer, 2013. 477 – 498.

[12] Wenner A. Weighted dynamic pushdown networks[A]. Programming Languages and Systems[C]. Berlin Heidelberg: Springer, 2010. 590 – 609.

[13] Abdulla P A, Atig M F, Stenman J. Dense-timed pushdown automata[A]. Logic in Computer Science [C]. Los Alamitos: IEEE Computer Society, 2012. 35 – 44.

[14] Daws C, Tripakis S. Model checking of real-time reachability properties using abstractions[A]. Tools and Algorithms for the Construction and Analysis of Systems[C]. Berlin Heidelberg: Springer, 1998. 313 – 329.

[15] Johan B, Wang Y. Timed automata: semantics, algorithms and tools[A]. Lectures on Concurrency and Petri Nets [C]. Berlin Heidelberg: Springer, 2003. 87 – 124.

[16] 陈伟,薛云志,赵琛,李明树. 一种基于时间自动机的实时系统测试方法[J]. 软件学报,2007,18(1): 62 – 73. CHEN Wei, XUE Yun-zhi, ZHAO Chen, LI Ming-shu. A method for testing real-time system based on timed automata[J]. Journal of Software, 2007, 18(1): 62 – 73. (in Chinese)

[17] Si Y, Sun J, Liu Y, Wang T. Improving model checking stateful timed CSP with non-zenoness through clock-symmetry reduction[A]. Formal Methods and Software Engineering [C]. Berlin Heidelberg: Springer, 2013. 182 – 198.

作者简介

钱俊彦　男,1973 年 7 月出生于浙江省绍兴市. 博士,现为桂林电子科技大学教授. 主要研究领域为软件工程、软件分析与验证、模型检验和 VLSI 容错.
E-mail: qjy2000@gmail.com

徐　力　男,1989 年 8 月出生于湖北省天门市. 硕士,主要研究方向为软件工程、软件分析与验证.
E-mail: xuli890808@hotmail.com

古天龙　男,1964 年 11 月出生于山西省芮城市. 博士,现为桂林电子科技大学教授、博士生导师. 主要研究领域为形式化方法、符号计算.
E-mail: cctlgu@guet.edu.cn

赵岭忠(通信作者)　男,1977 年 12 月出生于河南省南阳市. 博士,现为桂林电子科技大学教授. 主要研究领域为形式化技术与软件验证.
E-mail: zhaolingzhong163@163.com

蔡国永　男,1971 年 7 月出生于广西区河池市. 博士,现为桂林电子科技大学教授. 主要研究方向为社交媒体数据挖掘、机器学习、可信软件理论.
E-mail: ccgycai@guet.edu.cn