

# 一种基于进程验证的 Petri 网可达性判定方法

于 枫<sup>1),2)</sup> 罗军舟<sup>1)</sup> 李 伟<sup>1)</sup> 王 鹏<sup>1)</sup>

<sup>1)</sup>(东南大学计算机科学与工程学院 南京 210096)

<sup>2)</sup>(江苏科技大学计算机科学与工程学院 江苏 镇江 212003)

**摘 要** 为了解决 Petri 网的可达性判定问题,提出了一种基于进程验证的可达性判定方法.通过分析 Petri 网基本进程段子段间偏序关系,该方法提出原子进程段概念,求得原子进程段集及其偏序关系集作为可达性判定的验证对象.基于原网状态方程解向量、原子进程段  $T$ -向量的线性纯整数规划问题解,选取该解中非零分量对应的原子进程段子集作为待验进程段集,选取待验进程段间的偏序关系子集作为待验推理规则集,通过验证是否存在这两个集合上的以初始标识  $M_0$  为推导初态、目的标识  $M_d$  为推导终态的序列而判定  $(M_0, M_d)$  是否可达.分析表明该可达性判定算法的时间复杂度是依赖于原子进程段集规模的,最坏情况下不超过变迁集规模.

**关键词** Petri 网;可达性判定;Petri 网进程;进程验证

中图法分类号 TP301 DOI号: 10.3724/SP.J.1016.2010.00288

## A Reachability Determining Algorithm of Petri Net Based on Process Verification

YU Feng<sup>1),2)</sup> LUO Jun-Zhou<sup>1)</sup> LI Wei<sup>1)</sup> WANG Peng<sup>1)</sup>

<sup>1)</sup>(School of Computer Science and Engineering, Southeast University, Nanjing 210096)

<sup>2)</sup>(School of Computer Science and Engineering, Jiangsu University of Science and Technology, Zhenjiang, Jiangsu 212003)

**Abstract** In this paper, a process verification based algorithm is proposed to solve the reachability determining problem of Petri net. At first, the definition of atomic process section is introduced based on the partial relation decomposition, and the atomic process section set and the partial relation set are sought to act as verified objects in reachability determining. After the solution vector to pure linear nonequivalent expression of state equation solution vector and  $T$ -vectors of atomic process sections is calculated, a subset is selected from the atomic process section set with the standard that each element of the subset corresponds to a nonzero component in the solution vector. The partial relation is picked out to form the partial relation subset if it just describes partial relation between two elements of the atomic process section subset. If there is a reductive sequence which explores source marking as start state and target marking as end state, the reachability of Petri net could be concluded. The analysis shows that the time complexity of this determining algorithm depends on the scale of atomic process section set, and is no more than the scale of transition set under the worst condition.

**Keywords** Petri net; reachability determining; process of Petri net; process verification

收稿日期:2009-05-08;最终修改稿收到日期:2009-11-06. 本课题得到国家自然科学基金项目(60773103,60903161,60903162)、国家“九七三”重点基础研究发展规划项目基金(2010CB328104)、高等学校博士学科点专项科研基金(200802860031)、江苏省自然科学基金重点项目(BK2008030)、江苏省网络与信息安全重点实验室(BM2003201)和计算机网络和信息集成教育部重点实验室(93K-9)资助. 于 枫,女,1974 年生,博士研究生,研究方向为 Petri 网理论及应用、下一代互联网. E-mail: bo20007yf@seu.edu.cn. 罗军舟,男,1960 年生,博士,教授,博士生导师,研究领域为下一代互联网、协议工程、网络安全与管理、网络计算. 李 伟,男,1978 年生,博士,研究方向为下一代互联网、服务计算. 王 鹏,男,1983 年生,博士研究生,研究方向为下一代互联网、网络管理.

## 1 引言

Petri 网是适用于具有离散、并发、异步等特性的分布式系统建模分析的工具,在协议工程、逻辑推理、 workflow 建模等多领域得到广泛应用. Petri 网可达性判定理论与方法是领域研究的一项重要内容,某些动态性质(如活性)判定在一定程度上要以它为基础.

通常, Petri 网可达性判定方法可分为两类,一类基于 Petri 网的可达图或可覆盖图判定标识可达性,另一类通过求证状态方程解向量的合法变迁引发序列(Legal Transition Firing Sequence, LFS)存在来判定可达性. 基于可达图的有界 Petri 网可达性判定算法在理论上是可行的,但实际应用中,算法会因“状态组合爆炸”带来可达图线性时间内不可搜索而失效. 无界 Petri 网的可覆盖图中,“无穷量  $\omega$ ”的引入使其不再具备可达性判定能力. 因此,可达图或可覆盖图并不能真正解决 Petri 网的可达性判定问题. 基于状态方程的可达性判定方法中,状态方程求解和 LFS 验证均已证明是 NP-难问题,想要找到多项式时间复杂度的算法也是很困难的.

基于 LFS 存在判定可达性问题,文献[1]针对  $T$ -invariant-less Petri 网提出了通过添加独特的补变迁把原 Petri 网的 LFS 判定转化为补 Petri 网的  $T$ -不变量线性表出系数求解问题的可达性分析方法;文献[2]扩展了文献[1]思想到一般 Petri 网,进一步研究补 Petri 网的  $T$ -不变量含义;文献[3]给出一种一般 Petri 网的 LFS 判定方法;文献[4]基于 deadlock-avoidance 思想提出 LFS 判定问题的启发式算法 FSDC,此算法以先求得的 MAX LFS 为基础求解目标 LFS. 这些方法却各有局限:文献[1-2]的算法为保证补变迁只引发一次使得  $T$ -不变量的线性表出系数取值于实数域,难以求解且常常不符合 Petri 网应用于现实系统建模的物理特性;文献[3]算法虽对可达树进行缩减,本质上仍是完全搜索策略;文献[4]的迭代算法运算量巨大. 因此,继续研究 Petri 网可达性判定的有效算法仍是十分有意义的.

可达性是 Petri 网动态行为研究的基础,标识间可达必对应于网的某个运行轨迹,而 Petri 网进程是记录网运行轨迹的优良工具,因此,从进程角度研究可达性判定问题不失为一种有益尝试. 由于一个进程(段)仅反映 Petri 网的一次运行(片段),为记录 Petri 网的全部可能运行过程,前人提出并研

究了 Petri 网的进程表达式问题. 文献[5]给出了有界 Petri 网的进程表达式,并证明有界 Petri 网的进程集是正规集. 文献[6]提出进程段划分的行为等价准则从而给出无界公平 Petri 网的进程表达式. 文献[7]设计无界 Petri 的进程网系统构造算法. 文献[8]阐述基于库所指标分解的进程表达式求取方法. 文献[9]首先求解  $S$ -网的进程表达式,随后证明任一 Petri 网可由一组  $S$ -网的拓展同步合成运算得到,进而得出进程表达式.

然而,进程表达式是以极小基础进程段为基本元素构成的,而极小基本进程段并不能直接作为可达性判定的基本元素,主要原因在于:

(1) 粒度不等. 进程表达式中,基本研究对象是极小基本进程段;可达性判定中,要分析至极小进程段的每个  $s$ -切. 在进程网系统中,变迁引发(不引发)意味着其对应的极小基本进程段中的全部  $s$ -切都可(都不可)达,这不符合可达性判定的需求.

(2) 范围不等. 进程表达式所确定的集合要小于可达性分析中须判定的合法变迁引发序列集. 后文中,我们通过证明等价于进程表达式语言集的无标注 Petri 网进程网语言是标识对间合法变迁引发序列集的真子集说明这种不等价性.

本文中,我们提出一种基于进程验证的可达性判定. 首先,基于 Petri 网基本进程段子段的偏序关系分解,提出原子进程段概念(即不可偏序再分),求得该网的原子进程段集及其偏序关系集(推理规则集). 然后,求原网状态方程解向量  $\mathbf{X}$ 、原子进程段  $T$ -向量的线性纯整数规划问题解,选取该解中非零分量对应的原子进程段作为待验证进程段集,选取待验进程段间的偏序关系作为待验推理规则子集. 最后,通过判定是否存在以初始标识  $M_0$  为推导初态、目的标识  $M_d$  为推导终态的序列存在来判定标识对  $(M_0, M_d)$  间的可达性.

文中第 2 节介绍相关的 Petri 网基础知识;第 3 节提出原子进程段概念,设计 Petri 网的原子进程段集、原子进程段间偏序关系集求解算法;第 4 节首先证明原子进程段  $T$ -向量的线性纯整数规划问题解是可达性的必要非充分条件,然后阐述基于进程验证的可达性判定算法,并以示例演示和分析;最后一节总结全文,展望后续工作.

## 2 相关的 Petri 网基础知识

本节给出的基本概念、术语等来自参考文献[7-8,

10-11]. 不失一般性, 假定 Petri 网中  $K=\omega, W=1$ .

**定义 1**<sup>[10]</sup>. 设  $N=(B, E; G)$  为一个网, 如果,

$$(1) \forall b \in B: | \cdot b | \leq 1 \wedge | b \cdot | \leq 1.$$

$$(2) \forall x, y \in B \cup E: (x, y) \in G^+ \rightarrow (y, x) \notin G^+,$$

则称  $N$  为一个出现网, 其中  $G^+$  表示流关系  $G$  的传递闭包.

**定义 2**<sup>[10]</sup>. 若  $N_1=(S, T; F)$  为一个网,  $N_2=(B, E; G)$  为一个出现网, 若映射  $\varphi: B \cup E \rightarrow S \cup T$  满足:

$$(1) \varphi(B) \subseteq S, \varphi(E) \subseteq T;$$

$$(2) \forall x, y \in B \cup E: (x, y) \in G \rightarrow (\varphi(x), \varphi(y)) \in F;$$

$$(3) \forall e \in E: \varphi(e) = \cdot \varphi(e), \varphi(e) = \varphi(e) \cdot,$$

则称  $\varphi$  定义了从  $N_2$  到  $N_1$  的一个映射, 记为  $\varphi: N_2 \rightarrow N_1$ .

**定义 3**<sup>[10]</sup>. 设  $\Sigma=(N_1, M_0)=(S, T, F; M_0)$  为一个 Petri 网,  $N=(B, E; G)$  为一个出现网, 如果  $\varphi: N \rightarrow N_1$  满足:

$$(1) \forall b_1, b_2 \in B: b_1 \neq b_2, \text{ 若 } \varphi(b_1) = \varphi(b_2), \text{ 则 } \cdot b_1 \neq \cdot b_2 \text{ 且 } b_1 \neq b_2;$$

$$(2) \forall s \in S: | \{ b | \varphi(b) = s \wedge \cdot b = \emptyset \} | \leq M_0(s),$$

则称  $(N, \varphi)$  为  $\Sigma$  的一个进程.

**定义 4**<sup>[10]</sup>. 设  $\varphi$  为出现网  $N=(B, E; G)$  到 Petri 网  $\Sigma=(S, T, F; M_0)$  的一个网映射, 如果:

$$(1) \forall b_1, b_2 \in B (b_1 \neq b_2): \varphi(b_1) = \varphi(b_2) \rightarrow (\cdot b_1 \neq \cdot b_2 \vee \cdot b_1 = \cdot b_2 = \emptyset) \wedge (b_1 \neq b_2 \vee b_1 = b_2 = \emptyset),$$

$$(2) | \{ b | \varphi(b) = s \wedge \cdot b = \emptyset \} | = M_0(s),$$

则称  $P=(N, \varphi)$  为  $\Sigma$  的一个满进程.

**定义 5**<sup>[7]</sup>. 设  $P=(N, \varphi)$  为  $\Sigma$  的一个满进程, 其中  $N=(B, E; G), u_1 \leq u_2$ , 记

$$(1) B_1 = \{ x \in B | \exists b_1 \in u_1, b_2 \in u_2: (b_1, x) \in G^* \wedge (x, b_2) \in G^* \},$$

$$(2) E_1 = \{ y \in E | \exists b_1 \in u_1, b_2 \in u_2: (b_1, y) \in G^+ \wedge (y, b_2) \in G^+ \},$$

$$(3) G_1 = G \cap (\{ (B_1 \times E_1) \cap (E_1 \times B_1) \}).$$

令  $N_1=(B_1, E_1; G_1), \varphi_1: N_1 \rightarrow \Sigma$  满足:  $\forall x \in B \cup E: \varphi_1(x) = \varphi(x)$ , 则称  $(N, \varphi_1)$  为进程  $P$  的(界于  $u_1$  和  $u_2$  之间的)一段, 也称作  $\Sigma$  的一个进程段, 记为  $(N[u_1, u_2], \varphi)$ .

**定义 6**<sup>[7]</sup>. 设  $P=(N, \varphi)$  为  $\Sigma$  的一个满进程,  $P_1=(N[u_1, u_2], \varphi)$  为  $\Sigma$  的一个进程段, 如果  $N[u_1, u_2]$  中的任意两个  $s$ -切  $u_i$  和  $u_j, i, j \neq 1, 2$ , 有

$$u_i \neq u_j \rightarrow ((\varphi(u_i) \neq \varphi(u_j)) \wedge ((\varphi(u_i) \triangleleft \varphi(u_j)) \wedge ((\varphi(u_i) \triangleright \varphi(u_j))),$$

则称  $P_1=(N[u_1, u_2], \varphi)$  是  $\Sigma$  的一个基本进程段.

**定义 7**<sup>[7]</sup>. 给定一个 Petri 网  $\Sigma=(S, T, F; M_0), C_{m \times n}$  为网的关联矩阵,  $|P|=m, |T|=n$ , 若一个  $n$  维非负整数向量  $Y$  满足

(1)  $CY^T=0$ , 则称  $Y$  为  $\Sigma$  的一个  $T$ -不变量. 若对  $\forall 0 < Y' < Y$  有  $C(Y')^T \neq 0$ , 则称  $Y$  为  $\Sigma$  的一个极小  $T$ -不变量;

(2)  $CY^T > 0$ , 则称  $Y$  为  $\Sigma$  的一个增向量. 若对  $\forall 0 < Y' < Y$  不是  $\Sigma$  的增向量, 则称  $Y$  为  $\Sigma$  的一个极小增向量;

(3)  $CY^T < 0$ , 则称  $Y$  为  $\Sigma$  的一个减向量. 若对  $\forall 0 < Y' < Y$  不是  $\Sigma$  的减向量, 则称  $Y$  为  $\Sigma$  的一个极小减向量;

(4)  $(CY^T \triangleleft 0) \wedge (CY^T \triangleright 0)$ , 则称  $Y$  为  $\Sigma$  的一个传递向量. 若对  $\forall 0 < Y' < Y$  不是  $\Sigma$  的传递向量, 则称  $Y$  为  $\Sigma$  的一个极小传递向量.

极小  $T$ -不变量和极小增向量被称为  $\Sigma$  的极小可重复向量, 极小减向量和极小传递向量被称为  $\Sigma$  的极小受控可重复向量.

**定义 8**<sup>[7]</sup>. 给定 Petri 网  $\Sigma=(S, T, F; M_0), X$  是  $\Sigma$  的极小可重复向量, 若存在  $M \in R(M_0)$  和变迁序列  $\sigma(\#(t_i/\sigma) = X(i), i=1, 2, \dots, n)$ , 使得对任意正整数  $k$ , 都有  $M[\sigma^k >$ , 则称  $X$  为  $\Sigma$  的有效极小可重复向量.

**定义 9**<sup>[7]</sup>. 给定  $\Sigma=(S, T, F; M_0), X$  是  $\Sigma$  的极小受控可重复向量, 若任意正整数  $k$ , 都存在  $M \in R(M_0)$  和变迁序列  $\sigma(\#(t_i/\sigma) = X(i), i=1, 2, \dots, n)$ , 使得  $M[\sigma^k >$ , 则称  $X$  为  $\Sigma$  的有效极小受控可重复向量.

**定义 10**<sup>[7]</sup>. 设  $P_1=(N[u_1, u_2], \varphi)$  为  $\Sigma$  的一个进程段,  $N=(B, E; G), \forall t_i \in T | \{ e \in E | \varphi(e) = t_i \wedge (\exists b_1 \in u_1, b_2 \in u_2: (b_1, e) \in G^+ \wedge (e, b_2) \in G^+) \}$  称作变迁  $t_i$  在  $P_1$  中出现的次数, 记作  $\#(t_i, P_1)$ ;  $n$  维非负整数向量  $X_{P_1}(i) = \#(t_i, P_1), i=1, 2, \dots, n$ , 称为进程段  $P_1$  的  $T$ -向量.

**定义 11**<sup>[7]</sup>. 设  $P_1=(N[u_1, u_2], \varphi)$  为  $\Sigma$  的一个进程段,  $X_P$  为其  $T$ -向量,

(1) 称  $P_1$  为  $\Sigma$  的一个(基本)不变进程段, 当且仅当  $X_P$  为  $\Sigma$  的一个有效(极小)  $T$ -不变量;

(2) 称  $P_1$  为  $\Sigma$  的一个(基本)增进程段, 当且仅当  $X_P$  为  $\Sigma$  的一个有效(极小)增向量;

(3) 称  $P_1$  为  $\Sigma$  的一个(基本)减进程段, 当且仅当  $X_P$  为  $\Sigma$  的一个有效(极小)减向量;

(4) 称  $P_1$  为  $\Sigma$  的一个(基本)传递进程段, 当且仅当  $X_P$  为  $\Sigma$  的一个有效(极小)传递向量.

给定  $\Sigma=(P, T, F; M_0)$  的一个有限进程段  $P_1=(N[u_1, u_2], \varphi)$ ,  $\varphi(u_1)=M_{u_1}$ ,  $\varphi(u_2)=M_{u_2}$ . 若  $P_1$  是  $\Sigma$  的

- (1) 不变进程段, 则  $\varphi(u_1)=\varphi(u_2)$ ;
- (2) 增进程段, 则  $\varphi(u_1)<\varphi(u_2)$ , 满足  $M_{u_2}(s_i)>M_{u_1}(s_i)$  的库所  $s_i$  称为  $P_1$  的输出库所;
- (3) 减进程段, 则  $\varphi(u_1)>\varphi(u_2)$ , 满足  $M_{u_2}(s_i)<M_{u_1}(s_i)$  的库所  $s_i$  称为  $P_1$  的驱动库所;
- (4) 传递进程段,  $\varphi(u_1)\neq\varphi(u_2)\wedge\varphi(u_1)\not\prec\varphi(u_2)\wedge\varphi(u_1)\not\succeq\varphi(u_2)$ .

满足  $M_{u_2}(s_i)>M_{u_1}(s_i)$  的库所  $s_i$  称为  $P_1$  的输出库所; 满足  $M_{u_2}(s_i)<M_{u_1}(s_i)$  的库所  $s_i$  称为  $P_1$  的驱动库所.

**定义 12**<sup>[7]</sup>. 给定 Petri 网  $\Sigma=(S, T, F; M_0)$ ,  $CT(\Sigma)$  为其可覆盖树, 满足以下条件的树  $CRT(\Sigma)$  称为  $\Sigma$  的特征可达树:

- (1)  $CRT(\Sigma)$  与  $CT(\Sigma)$  同构;
- (2)  $CRT(\Sigma)$  的根结点标注满足

$$M_r(p_i)=\begin{cases} M_0(p_i), & p_i \text{ 有界} \\ k_i, & p_i \text{ 无界} \end{cases};$$

- (3) 对  $CRT(\Sigma)$  中的任意两个结点  $M_j$  和  $M_k$ , 若  $M_j$  到  $M_k$  有一条有向边标注为  $t_i$ , 则

$$M_k(p_i)=\begin{cases} M_j(p_i)-1, & p_i \in t_i - t_i \\ M_j(p_i)+1, & p_i \in t_i - t_i \\ M_j(p_i), & \text{其它} \end{cases}.$$

**定义 13**<sup>[7]</sup>. 设  $P_1=(N[u_1, u_2], \varphi)$  为  $\Sigma=(P, T, F; M_0)$  的一个进程段,  $P_1$  对应的可达树是指:

- (1) 根结点标注为  $M_1=\varphi(u_1)$ ;
- (2) 树中一结点标注为  $M_j$  当且仅当存在  $s$ -切  $u$  满足:  $u_1 \leq u \leq u_2 \wedge \varphi(u)=M_j$ ;
- (3) 树中两结点  $M_j, M_k$  间存在标注变迁  $t$  的有向弧当且仅当  $M_j[t>M_k$ .

由定义 12、13 知, 进程段对应的可达树是 Petri 网的可覆盖树同构意义下的子树, 也同构于其特征可达树上的相应子树.

**定义 14**. 设  $P_1=(N[u_{11}, u_{12}], \varphi)$  和  $P_2=(N[u_{21}, u_{22}], \varphi)$  为  $\Sigma$  的两个基本进程段, 如果  $P_1$  和  $P_2$  在  $\Sigma$  的可达树满足:

- (1) 同构于相同的可覆盖子树,
- (2) 可覆盖子树中标注的变迁相同,

则说  $P_1$  和  $P_2$  是功能等效的, 记作  $P_1 \cong P_2$ .

**定理 1**<sup>[7]</sup>. 给定 Petri 网  $\Sigma=(S, T, F; M_0)$ , 则  $\Sigma$  的基本进程段集在功能等效的意义下是有限集.

**推论 1**<sup>[7]</sup>. 在行为等价意义下, Petri 网的基本

进程段集是有限集.

**定义 15**. 给定 Petri 网  $\Sigma=(S, T, F; M_0)$ , 基本进程段集  $BP(\Sigma)$ , 记  $+、|、*、\alpha$ -闭包、 $\alpha$ -闭包分别表示进程的连接、选择和闭包运算,  $E(P_j, P_k, \dots, +, |, *, \alpha)$  表示基本进程段  $P_j, P_k, \dots$ , 经由上述运算所得表达式. 若  $P_i \in BP(\Sigma); P_i \neq E(P_j, P_k, \dots, +, |, *)$ ,  $P_j, P_k \in BP(\Sigma)$ , 则称  $P_i$  为一个极小进程段. 极小进程段的集合称为极小进程段集, 记为  $BP_e(\Sigma)$ .

按文献[7]描述, Petri 网  $\Sigma$  的基本进程段集是基本不变进程段集  $BP^=(\Sigma)$ 、基本增进程段集  $BP^+(\Sigma)$ 、基本减进程段集  $BP^-(\Sigma)$ 、基本传递进程段集  $BP^+(\Sigma)$  和基本开进程段集  $NBP(\Sigma)$  的并  $BP(\Sigma)=BP^=(\Sigma) \cup BP^+(\Sigma) \cup BP^-(\Sigma) \cup BP^+(\Sigma) \cup NBP(\Sigma)$ . 由于基本开进程段并不对应一个有效的传递向量, 非有效进程段, 故后文中, 默认极小进程段集  $BP_e(\Sigma)=BP_e(\Sigma)-NBP(\Sigma)$ .

**定义 16**. 设  $P_1=(N[u_{11}, u_{12}], \varphi)$  和  $P_2=(N[u_{21}, u_{22}], \varphi)$  为  $\Sigma$  的两个进程段,  $P_1$  和  $P_2$  等价当且仅当  $P_1$  和  $P_2$  对应的可达树等价.

### 3 Petri 网的原子进程段

#### 3.1 极小基本进程段分析

文献[7-8]中, Petri 网的进程网系统  $\Sigma_P$  是将原网的基本进程段抽象为变迁, 基本进程段的输入、输出库所集作为该变迁输入、输出库所集, 建立流关系, 将原网的初始标识向进程网系统的库所集上做投影而建立起来的. 这样, 变迁引发的原子特性就迫使极小基本进程段中  $s$ -切的可达性判定带上原子特征, 即极小进程段  $P_1=(N[u_1, u_2], \varphi)$  抽象成的变迁  $t$  引发, 则全部  $s$ -切可达; 若  $t$  不引发则任意  $u_1 < u \leq u_2, M_u=\varphi(u)$  失去可达性判定机会. 事实上, 极小进程段  $P_1$  的第一个  $s$ -切可达, 并不能说明其后任意  $s$ -切是否可达. 另一方面, 极小进程段以是否存在有效极小(受控)可重复向量  $\mathbf{Y}$  为判定标准, 而  $\mathbf{Y}$  的判定条件有两点: (1)  $\mathbf{Y}$  左乘关联矩阵  $(\mathbf{C}\mathbf{Y}^T)$  后积向量与零向量的大小比较; (2) 存在可达标识和合法引发序列, 使得该序列可引发任意正整数  $k$  次(或对任意正整数  $k$  存在可达标识和合法引发序, 该序列可引发  $k$  次). 后一条件本身就是可达性判定问题.

由上述知, 极小基本进程段粒度上并不匹配于可达性判定的基本研究对象. 另一方面, 借助进程网语言, 我们可证明, 以极小基本进程段为基本元素构

成的进程表达式所确定的集合仅是 Petri 网可达标识对间合法变迁引发序列集的真子集。

**定义 17**<sup>[10]</sup>. 称  $PN=(S, T, F; M_0, \Sigma, h, F)$  为一个标注 Petri 网, 当且仅当

- (1)  $S, T, F; M_0$  含义同前,  $\Sigma$  为字母表;
- (2)  $h: T \rightarrow \Sigma$  为标注函数;
- (3)  $F \subseteq R(M_0)$  为终止标识集,  $R(M_0)$  为可达集.

**定义 18**<sup>[10]</sup>. 设  $PN=(S, T, F; M_0, \Sigma, h, F)$  为一个标注 Petri 网, 令  $L = \{\alpha \in \Sigma^* \mid \exists \sigma \in T^* \wedge M_0[\sigma > M \wedge M \in F \wedge h(\sigma) = \alpha], F = R(M_0) \Sigma = T\}$ , 则称  $L$  为  $PN$  产生的  $P$ -型无标注语言, 记为  $P^f$ .

Petri 网中相异标识对间可达必定对应网的一次运行, 由 Petri 进程的定义知, 若一对标识  $(M_0, M_d)$  间有可达关系, 则至少存在一个进程  $P_{0d} = (N(u_0, u_d), \varphi)$ , 有  $\Gamma(P_{0d}) = \sigma_{0d}$  满足  $M_0[\sigma_{0d} > M_d$ .

**引理 1.** 给定 Petri 网  $\Sigma=(S, T, F; M_0)$ , 目标标识  $M_d$ , 记  $(M_0, M_d)$  间可达变迁引发序列集合为  $\sigma^R = \{\sigma \mid (M_0[\sigma > M_d])\}$ , 则  $\sigma^R \subset P^f$ .

证明. 由定义 18 可知, 有  $\sigma^R \subseteq P^f$ .

再证  $\sigma^R \neq P^f$ . 取  $\forall M_\omega \in R(M_0)$ ,  $M_0[\omega > M_\omega$ ,  $M_\omega \neq M_d$ , 由定义 18 知有  $\omega \in P^f$ , 记  $\omega$  对应的  $T$ -向量  $\mathbf{X}_\omega = \#(t_i/\omega)$ , 序列  $\omega$  引发后的标识  $M_\omega = M_0 + \mathbf{CX}_\omega^T$ ,

取  $\sigma \in \sigma^R$ ,  $M_0[\sigma > M_d, M_d \in R(M_0)$ ,  $\sigma$  对应  $T$ -向量  $\mathbf{X}_\sigma = \#(t_i/\sigma_{0d})$ , 必有  $M_\sigma = M_0 + \mathbf{CX}_\sigma^T$ .

若  $\omega \in \sigma^R$ ,

则有  $M_0 + \mathbf{CX}_\sigma^T = M_0 + \mathbf{CX}_\omega^T$ ,

则与  $M_\omega \neq M_d$  矛盾,

故  $\omega \notin \sigma^R$ ,

综上, 有  $\sigma^R \subset P^f$ . 证毕.

**推论 2.** 给定 Petri 网  $\Sigma=(S, T, F; M_0)$ , 任意标识  $M' \in R(M_0)$ , 标识对  $(M_0, M')$  间可达变迁引发序列集合为  $\sigma_R = \{\sigma \mid (M_0[\sigma > M_d])\}$ , 则

$$\bigcup_{M' \in R(M_0)} \sigma^R = P^f.$$

证明. 由定义 18、引理 1 可知, 该命题成立.

**引理 2.** 给定 Petri 网  $\Sigma=(S, T, F; M_0)$ , 极小进程段集  $BP(\Sigma)$ ,  $\Sigma_P$  为其进程网系统,  $L(\Sigma_P)$  为  $\Sigma_P$  所确定的语言,  $P^f$  为  $\Sigma$  产生的  $P$ -型无标注语言, 则  $L(\Sigma_P) \subset P^f$ .

证明. 由  $\Sigma_P, L(\Sigma_P), P^f$  的定义知, 显然有  $L(\Sigma_P) \subseteq P^f$ .

再证  $L(\Sigma_P) \neq P^f$ . 设  $BP(\Sigma) = \{P_1, \dots, P_k\}$ ,  $\Gamma: BP(\Sigma) \rightarrow T^*$ .

取  $\omega \in P^f$ , 有  $M_0[\omega > M_\omega, P_{0\omega} = (N(u_0, u_\omega), \varphi)$  为满进程,

不妨设  $\omega = \Gamma(P_{0\omega}) = t_1 t_2 \dots t_l$ ,

取  $(\prod \sigma_j = \omega) \wedge (\Gamma(\text{sub}(P_i)) = \sigma_j), i \in \{1 \dots k\}$ ,

$$P_{0\omega} = \prod \text{sub}(P_i),$$

若任一  $\text{sub}(P_i)$  有  $\text{sub}(P_i) \subset P_i, \Sigma_P$  中无与  $\text{sub}(P_i)$  对应的变迁可引发, 则  $\prod \text{sub}(P_i) \notin L(\Sigma_P)$ , 即  $\omega \notin L(\Sigma_P)$ . 证毕.

**定理 2.** 给定 Petri 网  $\Sigma=(S, T, F; M_0)$ , 极小进程段集  $BP(\Sigma)$ ,  $\Sigma_P$  为其进程网系统,  $L(\Sigma_P)$  为  $\Sigma_P$  所确定的语言, 标识对  $(M_0, M')$  的可达变迁引发序列集  $\sigma^R$ , 则

$$L(\Sigma_P) \subset \bigcup_{M' \in R(M_0)} \sigma^R.$$

证明. 由引理 1、推论 2、引理 2 显然可证.

综上所述, 并不能通过直接应用进程表达式来确定是否存在其所确定的集合中有对应于 Petri 网的一对标识间合法变迁引发序列存在, 即得出 Petri 网可达性判定结论.

### 3.2 原子进程段概念

由以上知, 极小基本进程段粒度相对于可达性判定来说偏大, 因此, 通过对其进行偏序再分, 我们提出原子进程段概念.

**定义 19.** 给定 Petri 网  $\Sigma=(S, T, F; M_0)$ , 变迁  $t_i, t_j$  和  $t_k$ ,

(1) 若  $t_j$  引发则必有  $t_i$  先引发, 则称  $t_i, t_j$  有顺序关系,  $t_i$  为前驱变迁,  $t_j$  为后继变迁, 记为  $t_i \rightarrow t_j$ ;

(2) 若标识  $M$  下, 有

$$((M[t_j > M']) \wedge (M'[t_i >]) \wedge (\neg M'[t_j])) \vee$$

$$((M[t_i > M'']) \wedge (M''[t_j >]) \wedge (\neg M''[t_i >]))$$

成立, 则称  $t_i, t_j$  在标识  $M$  并发, 记为  $t_i \parallel_M t_j$ ;

(3) 若标识  $M(M[t_j > M' \wedge \neg M'[t_i >]) \vee (M[t_i > M'' \wedge \neg M''[t_j >])$  成立, 则称  $t_i, t_j$  在标识  $M$  下冲突选择, 记为  $t_i \leftrightarrow_M t_j$ ;

(4) 若标识  $M$  下, 有

$$((M[t_j > M']) \wedge (t_i \leftrightarrow_{M'} t_j)) \vee$$

$$((M[t_i > M'']) \wedge (t_i \leftrightarrow_{M'} t_j))$$

成立, 则称  $t_i, t_j$  在标识  $M$  下自由选择, 记为  $t_i \Leftrightarrow_M t_j$ ;

(5) 若  $(t_k \rightarrow t_i) \wedge (t_k \rightarrow t_j) \wedge (t_i \parallel_M t_j)$  成立, 则称  $t_i, t_j$  在  $t_k$  处有分支关系, 记为  $t_i \uparrow t_j$ ;

(6) 若  $(t_i \rightarrow t_k) \wedge (t_j \rightarrow t_k) \wedge (t_i \parallel_M t_j)$  成立, 则称  $t_i, t_j$  在  $t_k$  处有汇合关系, 记为  $t_i \downarrow t_j$ .

定义 19 给出的 Petri 网变迁间偏序关系如图 1 所示.

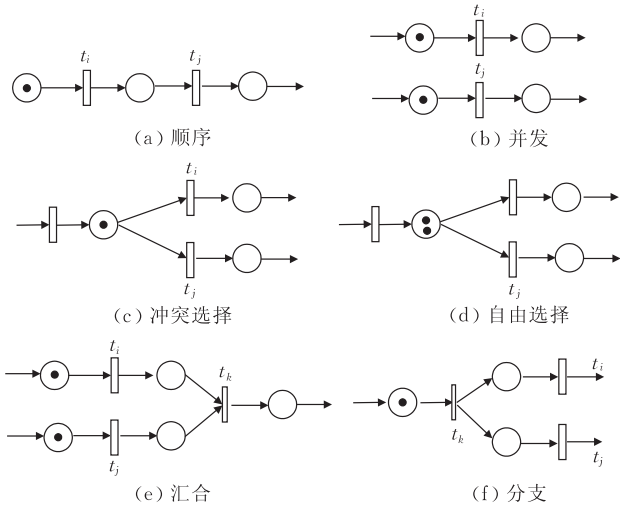


图 1 变迁间偏序关系图

**定义 20.** 给定 Petri 网  $\Sigma = (S, T, F; M_0)$ , 进程段集  $P(\Sigma)$ , 定义  $\Gamma: P(\Sigma) \rightarrow T^*$  为进程段向变迁序列的映射关系.

类似, 可分析 Petri 网的进程段间偏序关系, 有定义 21.

**定义 21.** 给定 Petri 网  $\Sigma = (S, T, F; M_0)$ ,  $BP_e(\Sigma)$  含  $k$  个极小进程段  $P_1 \cdots P_k$ , 以  $P^- = (N(u_{-1}, u_{-2}), \varphi)$ ,  $P^+ = (N(u_{+1}, u_{+2}), \varphi)$ ,  $P^- = (N(u_{-1}, u_{-2}), \varphi)$ ,  $P^+ = (N(u_{+1}, u_{+2}), \varphi)$  分别表示极小不变进程段、极小增进程段、极小减进程段、极小传递进程段, 记  $\%, \& \in \{=, +, -, \rightarrow\}$ .  $\forall P \in BP_e(\Sigma)$ , 若  $M_{\%1}[\sigma > M_{\%2}]$ , 则  $\Gamma_{P_{\%}} = \sigma_{\%}$ ,  $M_{\%i} = \varphi(u_{\%i}) \mid_{u_{\%i} \in P_{\%}}$ . 类似,  $M_{\&j} = \varphi(u_{\&j}) \mid_{u_{\&j} \in P_{\&}}$ ,  $\Gamma_{P_{\&}} = \sigma_{\&}$ .

(1) 若  $\Gamma_{P_{\%}} \cap \Gamma_{P_{\&}} = \emptyset, M_{\%2} \geq M_{\&1}$ , 有

$$(N(u_{\%1}, u_{\%2}), \varphi) \rightarrow (N(u_{\&1}, u_{\&2}), \varphi),$$

则称  $(N(u_{\%1}, u_{\%2}), \varphi)$  与  $(N(u_{\&1}, u_{\&2}), \varphi)$  在标识  $M_{\%2}$  顺序;

(2) 若  $\Gamma_{P_{\%}} \cap \Gamma_{P_{\&}} = \emptyset, \exists M M_{\%1} \leq M \wedge M_{\&1} \leq M$ , 有

$$(N(u_{\%1}, u_{\%2}), \varphi) \parallel_M (N(u_{\&1}, u_{\&2}), \varphi),$$

则称  $(N(u_{\%1}, u_{\%2}), \varphi)$  与  $(N(u_{\&1}, u_{\&2}), \varphi)$  在标识  $M$  下并发;

(3) 若  $\Gamma_{P_{\%}} \cap \Gamma_{P_{\&}} = \emptyset, \exists M M_{\%1} \leq M \wedge M_{\&1} \leq M$ , 有

$$(N(u_{\%1}, u_{\%2}), \varphi) \leftrightarrow_M (N(u_{\&1}, u_{\&2}), \varphi),$$

则称  $(N(u_{\%1}, u_{\%2}), \varphi)$  与  $(N(u_{\&1}, u_{\&2}), \varphi)$  在标识  $M$  冲突选择;

(4) 若  $\Gamma_{P_{\%}} \cap \Gamma_{P_{\&}} = \emptyset, \exists M M_{\%1} \leq M \wedge M_{\&1} \leq M$ , 有

$$(N(u_{\%1}, u_{\%2}), \varphi) \Leftrightarrow_M (N(u_{\&1}, u_{\&2}), \varphi),$$

则称  $(N(u_{\%1}, u_{\%2}), \varphi)$  与  $(N(u_{\&1}, u_{\&2}), \varphi)$  在标识  $M$

自由选择;

(5) 若  $\Gamma_{P_{\%}} \cap \Gamma_{P_{\&}} = \emptyset, \exists M M_{\%2} \leq M \wedge M_{\&2} \leq M$ , 有

$$(N(u_{\%1}, u_{\%2}), \varphi) \downarrow_M (N(u_{\&1}, u_{\&2}), \varphi),$$

则称  $(N(u_{\%1}, u_{\%2}), \varphi)$  与  $(N(u_{\&1}, u_{\&2}), \varphi)$  汇结于标识  $M$ ;

(6) 若  $\Gamma_{P_{\%}} \cap \Gamma_{P_{\&}} = \emptyset, \exists M = \varphi(u) \mid_{u \in P_{\%} \wedge u \in P_{\&}}$ , 有

$$(N(u_{\%1}, u), \varphi) \downarrow_M (N(u_{\&1}, u), \varphi);$$

$$(N(u, u_{\%2}), \varphi) \leftrightarrow_M (N(u, u_{\&2}), \varphi),$$

则称  $(N(u_{\%1}, u_{\%2}), \varphi)$  与  $(N(u_{\&1}, u_{\&2}), \varphi)$  汇合于  $M$  后冲突选择;

(7) 若  $\Gamma_{P_{\%}} \cap \Gamma_{P_{\&}} = \emptyset, \exists M = \varphi(u) \mid_{u \in P_{\%} \wedge u \in P_{\&}}$ , 有

$$(N(u_{\%1}, u), \varphi) \downarrow_M (N(u_{\&1}, u), \varphi);$$

$$(N(u, u_{\%2}), \varphi) \Leftrightarrow_M (N(u, u_{\&2}), \varphi),$$

则称  $(N(u_{\%1}, u_{\%2}), \varphi)$  与  $(N(u_{\&1}, u_{\&2}), \varphi)$  汇合于  $M$  后自由选择;

(8) 若  $\exists M M \geq M_{\%1} \wedge M \geq M_{\&1}, \exists M' M' \geq M_{\%2} \wedge M' \geq M_{\&2}, \Gamma_{P_{\%}} \cap \Gamma_{P_{\&}} = \emptyset$ , 有

$$(N(u_{\%1}, u_{\%2}), \varphi) \leftrightarrow_M (N(u_{\&1}, u_{\&2}), \varphi);$$

$$(N(u_{\%1}, u_{\%2}), \varphi) \downarrow_{M'} (N(u_{\&1}, u_{\&2}), \varphi),$$

则称  $(N(u_{\%1}, u_{\%2}), \varphi)$  与  $(N(u_{\&1}, u_{\&2}), \varphi)$  冲突汇结于标识  $M'$ ;

(9) 若  $\exists M M \geq M_{\%1} \wedge M \geq M_{\&1}, \exists M' M' \geq M_{\%2} \wedge M' \geq M_{\&2}, \Gamma_{P_{\%}} \cap \Gamma_{P_{\&}} = \emptyset$ , 有

$$(N(u_{\%1}, u_{\%2}), \varphi) \Leftrightarrow_M (N(u_{\&1}, u_{\&2}), \varphi);$$

$$(N(u_{\%1}, u_{\%2}), \varphi) \downarrow_{M'} (N(u_{\&1}, u_{\&2}), \varphi),$$

则称  $(N(u_{\%1}, u_{\%2}), \varphi)$  与  $(N(u_{\&1}, u_{\&2}), \varphi)$  自由汇结于标识  $M'$ ;

(10) 若  $\exists M M \geq M_{\%1} \wedge M \geq M_{\&1}, u_{\%1} < u < u_{\%2}, u_{\&1} < u < u_{\&2}, \exists M' M' = \varphi(u), \Gamma_{P_{\%}} \cap \Gamma_{P_{\&}} = \emptyset$ , 有

$$(N(u_{\%1}, u), \varphi) \Leftrightarrow_{M'} (N(u_{\&1}, u), \varphi);$$

$$(N(u, u_{\%2}), \varphi) \leftrightarrow_{M'} (N(u, u_{\&2}), \varphi),$$

则称  $(N(u_{\%1}, u_{\%2}), \varphi)$  与  $(N(u_{\&1}, u_{\&2}), \varphi)$  自由汇合后冲突选择;

(11) 若  $\exists M M \geq M_{\%1} \wedge M \geq M_{\&1}, u_{\%1} < u < u_{\%2}, u_{\&1} < u < u_{\&2}, \exists M' M' = \varphi(u), \Gamma_{P_{\%}} \cap \Gamma_{P_{\&}} = \emptyset$ , 有

$$(N(u_{\%1}, u), \varphi) \leftrightarrow_{M'} (N(u_{\&1}, u), \varphi);$$

$$(N(u, u_{\%2}), \varphi) \Leftrightarrow_{M'} (N(u, u_{\&2}), \varphi),$$

则称  $(N(u_{\%1}, u_{\%2}), \varphi)$  与  $(N(u_{\&1}, u_{\&2}), \varphi)$  冲突汇合后自由选择;

(12) 若  $\Gamma_{P_{\%}} \cap \Gamma_{P_{\&}} = \sigma, \{M_{\%i}\} \cap \{M_{\&j}\} = \{M' \cdots M_{\%2}\}, M'[\sigma > M_{\%2} M' = \varphi(u'), M_{\%2} = \varphi(u_{\%2})$ , 有

$$(N(u_{\%1}, u'), \varphi) \downarrow_{(N(u', u_{\&2}), \varphi)} (N(u_{\&1}, u'), \varphi),$$

则称  $(N(u_{\%1}, u_{\%2}), \varphi)$  与  $(N(u_{\&1}, u_{\&2}), \varphi)$  汇并于  $(N(u', u_{\&2}), \varphi)$ ;

(13) 若  $\Gamma_{P_{\%}} \cap \Gamma_{P_{\&}} = \sigma, \{M_{\%i}\} \cap \{M_{\&j}\} = \{M \dots M'\}, M[\sigma] > M', M = \varphi(u), M' = \varphi(u') \mid u' < u_{\%2} \wedge u' < u_{\&2}$ , 有

$$(N(u_{\%1}, u), \varphi) \downarrow_{(N(u, u'), \varphi)} (N(u_{\&1}, u), \varphi),$$

则称  $(N(u_{\%1}, u_{\%2}), \varphi)$  与  $(N(u_{\&1}, u_{\&2}), \varphi)$  汇并于  $(N(u', u_{\&2}), \varphi)$ ;

(14) 若  $\Gamma_{P_{\%}} \cap \Gamma_{P_{\&}} = \sigma, \{M_{\%i}\} \cap \{M_{\&j}\} = \{M \dots M'\}, M[\sigma] > M', M = \varphi(u), M' = \varphi(u') \mid u' = u_{\%2} \wedge u' < u_{\&2}$ , 有

$$(N(u_{\%1}, u), \varphi) \downarrow_{(N(u, u'), \varphi)} (N(u_{\&1}, u), \varphi);$$

$$(N(u', u_{\%2}), \varphi) \Leftrightarrow_{M'} (N(u', u_{\&2}), \varphi),$$

则称  $(N(u_{\%1}, u_{\%2}), \varphi)$  与  $(N(u_{\&1}, u_{\&2}), \varphi)$  汇并于

$(N(u, u'), \varphi)$  后自由选择;

(15) 若  $\Gamma_{P_{\%}} \cap \Gamma_{P_{\&}} = \sigma, \{M_{\%i}\} \cap \{M_{\&j}\} = \{M \dots M'\}, M[\sigma] > M', M = \varphi(u), M' = \varphi(u') \mid u' = u_{\%2} \wedge u' < u_{\&2}$ , 有

$$(N(u_{\%1}, u'), \varphi) \downarrow_{(N(u', u_{\&2}), \varphi)} (N(u_{\&1}, u'), \varphi);$$

$$(N(u', u_{\%2}), \varphi) \Leftrightarrow_{M'} (N(u', u_{\&2}), \varphi),$$

则称  $(N(u_{\%1}, u_{\%2}), \varphi)$  与  $(N(u_{\&1}, u_{\&2}), \varphi)$  汇并于  $(N(u, u'), \varphi)$  后冲突选择.

定义 21 中, 顺序关系表达进程段间的线性推进, 冲突选择刻画进程段间的竞争, 汇合关系定义进程推进的同步标识点(同步子进程段). 这些关系可图示化如图 2(顺序关系未表示). 图中圆圈示意进程段的始末  $s$ -切.

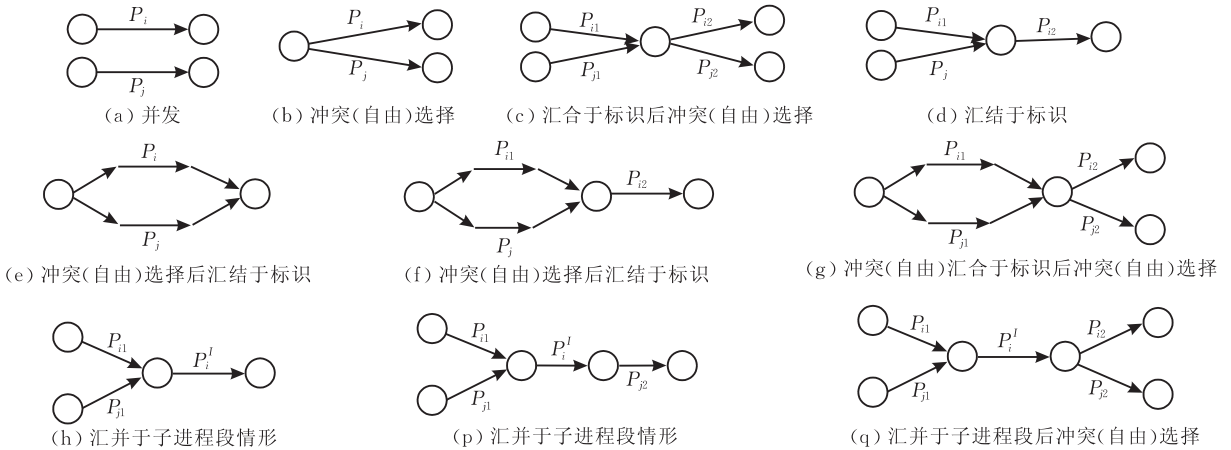


图 2 进程段间偏序关系图

定义 22. 给定 Petri 网  $\Sigma = (S, T, F; M_0)$ ,  $P$  为  $\Sigma$  的一个原子进程段当且仅为  $P$  在定义 21 下不可再分.

### 3.3 原子进程段及其偏序关系求解算法

给定 Petri 网  $\Sigma = (S, T, F; M_0)$ , 遵照定义 21, 分解其极小进程段, 求原子进程段及其间推理规则方法如附图 1 所示.

为了易于理解算法内容, 此处以一个无界 Petri 网为例演示上述内容. 该无界 Petri 网如图 3 所示, 其中  $s_3$  和  $s_5$  为无界库所.

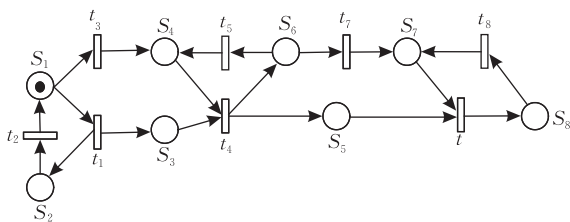


图 3 无界 Petri 网系统  $\Sigma$

按文献[7]方法求得极小进程段集  $BP_e(\Sigma) = \{P_1, P_2, P_3, P_4, P_5\}$ , 如图 4 所示.

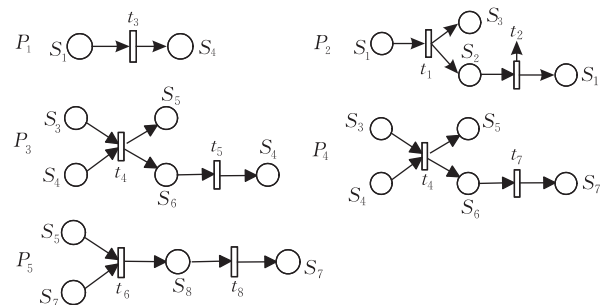


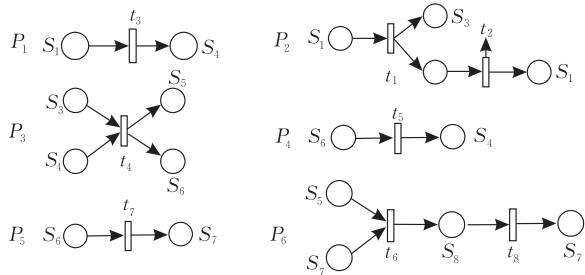
图 4 无界 Petri  $\Sigma$  的极小进程段

对图 4 给出的极小进程段, 按算法 1 求得其原子进程段集为  $AP(\Sigma) = \{P_1, P_2, P_3, P_4, P_5, P_6\}$ , 如图 5 所示.

其推理规则集为

$$R(\Sigma): \{P_1 \rightarrow P_2, P_1 \rightarrow P_3, P_1 \downarrow P_4, P_1 \leftrightarrow P_2, P_2 \rightarrow P_1, P_2 \rightarrow P_3, P_3 \rightarrow P_4, P_3 \rightarrow P_5, P_3 \rightarrow P_6, P_4 \leftrightarrow P_5, P_5 \rightarrow P_6, P_5 \downarrow P_6, P_6 \rightarrow P_6\}.$$

原子进程段及其偏序关系是本文可达性判定的基础.

图 5 无界 Petri  $\Sigma$  的原子进程段

## 4 基于进程验证的可达性判定算法

### 4.1 算法思想

通常, Petri 网可达性问题形式化表述为: 给定 Petri 网  $PN = (S, T, F; M_0)$ ,  $M_d$  由  $M_0$  可达当且仅当

- (1) 存在  $n$ -维非负整数向量  $\mathbf{X}$  满足  $M_d = M_0 + \mathbf{C}\mathbf{X}^T$ ;
- (2) 存在变迁序列  $\sigma$ , 使得  $M_0[\sigma > M_d$  且  $\#(t_i/\sigma) = X(i)$ .

其中,  $\sigma$  是合法变迁引发序列, 求证其存在是 Petri 网可达性判定的重要内容.

基于进程验证的可达性判定方法中, 原子进程段及其偏序关系集是可达性判定的验证对象, 但并不一定是最小对象集. 算法中, 我们通过求解原网状态方程解向量  $\mathbf{X}$ 、原子进程段  $T$ -向量的线性纯整数规划问题解来选取最小验证对象集. 这种最小对象验证集的选择依据是: 该线性纯整数规划问题有解是可达性判定的必要非充分条件. 这样, 算法只需对应用纯整数规划问题求解方法(如分枝定界法)或软件(如 LINGO)求得的解设计验证方法即可.

以下, 给出证明过程.

**定义 23.** 设  $P_1 = (N[u_1, u_2], \varphi)$  为  $\Sigma$  的原子进程段,  $N = (B, E; G)$ ,  $\forall t_i \in T \{ \{ e \in E \mid \varphi(e) = t_i \wedge (\exists b_1 \in u_1, b_2 \in u_2: (b_1, e) \in G^+ \wedge (e, b_2) \in G^+) \} \}$  称作变迁  $t_i$  在  $P_1$  中出现的次数, 记作  $\#(t_i, P_1)$ ;  $n$ -维非负整数向量  $\mathbf{X}_P(i) = \#(t_i, P_1)$ ,  $i = 1, 2, \dots, n$ , 称为原子进程段  $P_1$  的  $T$ -向量.

**定义 24.** 两个同维数向量  $\mathbf{X}$  和  $\mathbf{Y}$ , 关系  $\mathbf{X} < \mathbf{Y}$  成立当且仅当  $\mathbf{X}(i) \leq \mathbf{Y}(i) \wedge \mathbf{X} \neq \mathbf{Y}$ ; 关系  $\mathbf{X} \leq \mathbf{Y}$  成立当且仅当  $\mathbf{X} < \mathbf{Y} \vee \mathbf{X} = \mathbf{Y}$ .

**推论 3.** Petri 网  $\Sigma$  的一个极小进程段  $P_1$  对应的  $T$ -向量为  $\mathbf{X}_{P_1}$ ,  $P_1$  的真子进程段记为  $sub(P_1)$ , 对应的  $T$ -向量为  $\mathbf{X}_{sub(P_1)}$ , 则  $\mathbf{X}_{sub(P_1)} < \mathbf{X}_{P_1}$ .

证明. 设  $\Gamma(P_1) = t_1 \dots t_l, \Gamma(sub(P_1)) = t_1 \dots$

$t_j, \Gamma(sub(P_1)) \subset \Gamma(P_1)$ ,  $P_1, sub(P_1)$  的  $T$ -向量分别为  $\mathbf{X}_{P_1}$  和  $\mathbf{X}_{sub(P_1)}$ , 若有  $\neg(\mathbf{X}_{sub(P_1)} < \mathbf{X}_{P_1})$ , 则可能有以下 3 种情况:

- (1)  $\mathbf{X}_{sub(P_1)} = \mathbf{X}_{P_1}$ , 与  $\Gamma(sub(P_1)) \subset \Gamma(P_1)$  矛盾.
- (2)  $\mathbf{X}_{pref(P_1)} > \mathbf{X}_{P_1}$ , 不妨设分量  $k$  有  $\mathbf{X}_{pref(P_1)}(k) > \mathbf{X}_{P_1}(k)$ , 即变迁  $t_k \in T$  在  $\Gamma(sub(P_1))$  中出现次数多于  $\Gamma(P_1)$ , 则  $\Gamma(sub(P_1)) \not\subset \Gamma(P_1)$ , 与已知条件矛盾.
- (3) 存在分量  $i$  和  $j$ , 有  $\mathbf{X}_{pref(P_1)}(i) > \mathbf{X}_{P_1}(i)$  且  $\mathbf{X}_{pref(P_1)}(j) \leq \mathbf{X}_{P_1}(j)$ , 使得  $\mathbf{X}_{P_1}$  和  $\mathbf{X}_{sub(P_1)}$  不可比较. 类似(2)可证.

综上所述, 知命题成立.

证毕.

**引理 3.** 给定 Petri 网  $\Sigma = (S, T, F; M_0)$ , 原子进程段集  $AP(\Sigma)$ , 其  $T$ -向量集  $\mathbf{Y}_{AP} = \{\mathbf{Y}_1, \dots, \mathbf{Y}_l\}$ , 则  $\mathbf{Y}_1, \dots, \mathbf{Y}_l$  在正整数域内线性无关.

证明. 设  $AP(\Sigma)$  中原子进程段为  $P_1, P_2, \dots, P_l$ . 由定义 21、定义 23 知,  $\forall P_i, P_j \in AP(\Sigma): P_i \neq P_j$ , 其  $T$ -向量  $\mathbf{Y}_i, \mathbf{Y}_j$  在正整数域内线性不相关.

若存在  $\mathbf{Y}_l \subseteq \mathbf{Y}_{AP}, \mathbf{Y}_l = \{\mathbf{Y}_i \mid \mathbf{Y}_i \in \mathbf{Y}_{AP}\}, k = |\mathbf{Y}_l|$  ( $2 < k \leq l$ ), 由定义 23 知, 有  $\mathbf{Y}_1 > \mathbf{0}, \dots, \mathbf{Y}_k > \mathbf{0}$ . 若存在不全为 0 的系数  $a_1, a_2, \dots, a_k$ , 使得  $a_1\mathbf{Y}_1 + a_2\mathbf{Y}_2 + \dots + a_k\mathbf{Y}_k = \mathbf{0}$  成立, 必有某个系数  $a_i < 0$ , 即不存在正整数域内的一组系数  $a_1, a_2, \dots, a_k$  使得  $a_1\mathbf{Y}_1 + a_2\mathbf{Y}_2 + \dots + a_k\mathbf{Y}_k = \mathbf{0}$  成立. 命题得证.

证毕.

**推论 4.** 给定 Petri 网  $\Sigma = (S, T, F; M_0)$ , 原子进程段集  $AP(\Sigma)$ , 其  $T$ -向量集  $\mathbf{Y}_{AP} = \{\mathbf{Y}_1, \dots, \mathbf{Y}_l\}$ , 则  $\mathbf{Y}_1, \dots, \mathbf{Y}_l$  在实数域内线性相关.

由引理 3 的证明显然可得.

为了保证原子进程段中任一  $s$ -切都不失去可达性判定机会, 且任意标识对进程的  $T$ -向量能够由原子进程段  $T$ -向量正整数系数线性表出, 引入余量  $\mathbf{X}_b$ .

**定义 25.** 给定 Petri 网  $\Sigma = (S, T, F; M_0)$ , 目标标识  $M_d$ , 关联矩阵  $\mathbf{C}_{m \times n}$ , 状态方程  $M_d = M_0 + \mathbf{C}\mathbf{X}^T$  解向量  $\mathbf{X}$ , 原子进程段  $T$ -向量集  $\mathbf{Y}_{AP} = \{\mathbf{Y}_1, \dots, \mathbf{Y}_l\}$ , 若存在  $n$ -维非负整数向量  $\mathbf{X}_b$ :

- (1)  $(\mathbf{X}_b < \mathbf{X}) \wedge (\forall \mathbf{Y}_i \in \mathbf{Y}_{AP} \mathbf{X}_b < \mathbf{Y}_i)$ ;
- (2)  $\mathbf{X} = a_1\mathbf{Y}_1 + \dots + a_l\mathbf{Y}_l + a_{l+1}\mathbf{X}_b, a_j = 0, 1, \dots, j = 1, 2, \dots, l+1$ ,

则称  $\{\mathbf{Y}_1, \dots, \mathbf{Y}_l, \mathbf{X}_b\}$  为  $\mathbf{X}$  的一组线性表出基,  $\mathbf{X}_b$  称为余量.

**引理 4.** 给定 Petri 网  $\Sigma = (S, T, F; M_0)$ , 原子进程段  $T$ -向量集  $\mathbf{Y}_{AP} = \{\mathbf{Y}_1, \dots, \mathbf{Y}_l\}$ , 则向量组  $\mathbf{X}_b, \mathbf{Y}_1, \dots, \mathbf{Y}_l$  在正整数域内线性无关.

证明. 反证法. 由定义 25 知,  $\forall \mathbf{Y}_i \in \mathbf{Y}_{AP}, i = 1, 2, \dots, l, \mathbf{X}_b$  与  $\mathbf{Y}_i$  在正整数域内线性无关. 若向量组

$\mathbf{X}_b, \mathbf{Y}_1, \dots, \mathbf{Y}_l$  在正整数域内线性相关, 即存在一组不全为零的整数  $c_1, \dots, c_{l+1}$  有

$$c_1 \mathbf{Y}_1 + \dots + c_l \mathbf{Y}_l + c_{l+1} \mathbf{X}_b = \mathbf{0}.$$

若  $c_{l+1} = 0$ , 则必有  $c_1, \dots, c_l$  不全为零, 这与引理 3 矛盾. 故  $c_{l+1} \neq 0$ , 有

$$\frac{c_1}{c_{l+1}} \mathbf{Y}_1 + \dots + \frac{c_k}{c_{l+1}} \mathbf{Y}_k + \mathbf{X}_b = \mathbf{0}$$

又由定义 17 知,  $\mathbf{Y}_1 > \mathbf{0}, \dots, \mathbf{Y}_l > \mathbf{0}, \mathbf{X}_b > \mathbf{0}$ , 则必有  $\frac{c_i}{c_{l+1}} < 0$ , 与假设不符, 原命题得证. 证毕.

**推论 5.** 给定 Petri 网  $\Sigma = (S, T, F; M_0)$ , 原子进程段  $T$ -向量集  $\mathbf{Y}_{AP} = \{\mathbf{Y}_1, \dots, \mathbf{Y}_l\}$ , 则  $\mathbf{X}_b, \mathbf{Y}_1, \dots, \mathbf{Y}_k$  在实数域内线性相关.

证明. 由引理 4 证明显然可得.

**推论 6.** 给定 Petri 网  $\Sigma = (S, T, F; M_0)$ , 目标标识  $M_d$ , 原子进程段集  $AP(\Sigma)$ , 其  $T$ -向量集  $\mathbf{Y}_{AP} = \{\mathbf{Y}_1, \dots, \mathbf{Y}_l\}$ , 自由余量  $\mathbf{X}_b$  可为平凡向量.

证明. 若 Petri 网的标识对  $(M_0, M_d)$  间可达, 对应进程  $P_{od} = (N(u_0, u_d), \varphi)$  可分解为

$$P_{od} = (P_1)^{l_1} (P_2)^{l_2} \dots (P_k)^{l_k},$$

$$P_i \in AP(\Sigma), l = l_1 + l_2 + \dots + l_k.$$

则有  $\Gamma(P_{od}) = \sigma, \Gamma(P_1) = \sigma_1, \dots, \Gamma(P_k) = \sigma_k$ ;

表现在  $T$ -向量上, 有

$$\mathbf{X} = l_1 \mathbf{Y}_1 + l_2 \mathbf{Y}_2 + \dots + l_k \mathbf{Y}_k.$$

上式与定义 25 中(2)对比, 可知,  $\mathbf{X}_b$  取为平凡向量.

**定理 3.** 若给定 Petri 网  $\Sigma = (S, T, F; M_0)$ , 目标标识  $M_d$ , 关联矩阵  $\mathbf{C}_{m \times n}$ , 状态方程  $M_d = M_0 + \mathbf{C}\mathbf{X}^T$  的解向量  $\mathbf{X}$ , 原子进程段  $T$ -向量集  $\mathbf{Y}_{AP} = \{\mathbf{Y}_1, \dots, \mathbf{Y}_l\}$ , 原子进程段间关系集  $R_a(\Sigma)$ , 则线性不等式方程  $\mathbf{X} - a_1 \mathbf{Y}_1 - \dots - a_l \mathbf{Y}_l \geq \mathbf{0}$  的整数解是  $M_d$  由  $M_0$  可达的必要条件, 非充分条件.

证明. 设此 Petri 网的原子进程段集  $AP(\Sigma) = \{P_1, \dots, P_l\}$ , 映射关系定义如前  $\Gamma: P(\Sigma) \rightarrow T^*$ .

必要性. 若  $M_d$  由  $M_0$  可达, 则标识对  $(M_0, M_d)$  间必有满进程  $P_{od} = (N(u_0, u_d), \varphi)$ , 有  $\Gamma(P_{od}) = \sigma$ , 向量  $\mathbf{X}(i) = \#(t_i/\sigma) | t_i \in T$  满足状态方程  $M_d = M_0 + \mathbf{C}\mathbf{X}^T$ . 不妨设  $|\sigma| = k, \sigma = t_1 \dots t_k, \forall t_i \in T$ .

又,  $\forall P_j \in AP(\Sigma): \Gamma(P_j) = \sigma_j$ ;

且  $\exists P' \in AP(\Sigma): u_0 \in P' \wedge \varphi(u_0) = M_0, \Gamma(P') = \sigma'$ ;

求  $\sigma$  与  $\sigma'$  的最大匹配串, 即  $\sigma_1 = \max(\text{pref}(\sigma) \cap \text{sub}(\sigma')), \sigma'$  与  $\sigma_1$  的  $T$ -向量分别为  $\mathbf{X}_{\sigma'}, \mathbf{X}_{\sigma_1}$ , 有  $\mathbf{X}_{\sigma'} \in \mathbf{Y}_{AP}(\Sigma), \mathbf{X}_{\sigma_1} \leq \mathbf{X}_{\sigma'}$ ;

取  $\sigma = \sigma - \sigma_1$ , 重复上一步直至  $\sigma = \emptyset$ ;

记按上述方法求得的序列依次为  $\sigma_1, \sigma_2, \dots, \sigma_h$ , 它们的  $T$ -向量分别为  $\mathbf{X}_{\sigma_1}, \mathbf{X}_{\sigma_2}, \dots, \mathbf{X}_{\sigma_h}$ , 则

或者  $\mathbf{X}_{\sigma_j} = \mathbf{Y}_i, \mathbf{X}_{\sigma_j} \in \mathbf{Y}_{AP}(\Sigma)$ ;

或者  $\exists \mathbf{Y}_i \in \mathbf{Y}_{AP}(\Sigma), \mathbf{X}_{\sigma_j} < \mathbf{Y}_i$ ;

有  $\mathbf{X} = \sum_{j \in \{1 \dots h\}} \mathbf{X}_{\sigma_j}$ ,

对任意满足  $\mathbf{X}_{\sigma_j} = \mathbf{Y}_i$  计数, 取得向量  $\mathbf{Y}_i$  的系数  $a_i$ ,

对任意满足  $\mathbf{X}_{\sigma_j} < \mathbf{Y}_i$ , 取  $a_i = 0$ ,

则  $\mathbf{X} \geq \sum_{j \in \{1 \dots h\}} a_i \mathbf{X}_{\sigma_j}$  有一组整数解  $a_1, a_2, \dots, a_l$ .

非充分性. 若  $\mathbf{X} - a_1 \mathbf{Y}_1 - \dots - a_k \mathbf{Y}_k \geq \mathbf{0}$  有惟一的一组整数解  $a_1, a_2, \dots, a_l$ , 代入线性不等式方程得余量  $\mathbf{X}_b$ , 即

$$\mathbf{X} = a_1 \mathbf{Y}_1 + \dots + a_l \mathbf{Y}_l + \mathbf{X}_b.$$

取  $A = (a_1, a_2, \dots, a_l)$  中非零系数对应的原子进程段  $P_i$  形成待验证进程段集  $P_X(\Sigma), P_X(\Sigma) \subseteq AP(\Sigma)$ ,

若  $P', P'' \in AP(\Sigma) \exists u \in P': \varphi(u) = M_0, \exists u' \in P'': \varphi(u') = M_d$ , 且  $P' \notin P_X(\Sigma), P'' \notin P_X(\Sigma)$ , 则  $P_X(\Sigma) = P_X(\Sigma) \cup \{P', P''\}$ .

取  $R_X(\Sigma) \subseteq R_a(\Sigma): \forall R(P_i, P_j) \in R_X(\Sigma) \rightarrow (P_i, P_j \in P_X(\Sigma))$ ; 若不存在一个  $R_X(\Sigma)$  上的推导序列  $P' \Rightarrow P''$ , 则  $M_d$  由  $M_0$  不可达. 证毕.

## 4.2 算法描述与示例

有了原子进程段集及其偏序关系集, 求出线性纯整数规划问题的解, 基于进程验证的可达性判定算法如图 6 所示.

基于进程验证的可达性判定算法.

输入: 关联矩阵  $\mathbf{C}$ , 初始标识  $M_0$ , 目的标识  $M_d$

输出: 可达或不可达结论

1. 求  $M_d = M_0 + \mathbf{C}\mathbf{X}^T$  的解向量  $\mathbf{X}$ ;
2. 调用算法 1, 求  $AP(\Sigma) = \{P_1 \dots P_l\}, R_a(\Sigma)$ , 得  $T$ -向量集  $\mathbf{Y}_{AP}(\Sigma) = \{\mathbf{Y}_1 \dots \mathbf{Y}_l\}$ ;
3. 应用 LINGO 求解  $\mathbf{X}, \mathbf{Y}_1, \dots, \mathbf{Y}_l$  的线性不等式方程组, 得  $A = \{A_1 \dots A_h\}$ , 每一  $A_i$  为  $l$  维正整数向量;
4. If  $A < \emptyset$  null, Then {取  $A_i, A = A - \{A_i\}$  goto 5} Else goto 9;
5. If  $A_i(j) \neq 0 | (i = 1 \dots h) \wedge (j = 1 \dots l)$  Then  
取  $P_j$  形成一组待判进程集  $P_X(\Sigma) = \{(N(u_{j1}, u_{j2}), \varphi)\}$ ;
6. If  $P', P'' \in AP(\Sigma)$  且  $P', P'' \in P_X(\Sigma)$  Then  
if  $u \in P', u' \in P'', \varphi(u) = M_0, \varphi(u') = M_d$  then goto 7  
else 添加进程  $P'$  和 (或)  $P''$  入待判定进程集  $P_X(\Sigma)$ ;
7.  $R_a(\Sigma)$  的任一规则: 若其前件、后件都属于  $P_X(\Sigma)$ , 则选入验证规则集  $R_X(\Sigma)$ ;
8. 若存在  $R_X(\Sigma)$  的推导序列  $P' \Rightarrow P''$ , 则合法引发行列存在, 输出可达结论, 停机; 否则, 转步 3;
9. 输出不可达结论, 停机.

图 6 基于进程验证的可达性判定算法

算法的正确性由前面分析保证, 终止性由解向量与原子进程段  $T$ -向量的线性纯整数问题解的有限性而保证.

下面, 以图 3 所示的无界 Petri 网为例, 演示此

算法.

已知: 给定 Petri 网  $\Sigma = (S, T, F; M_0)$  如图 2 所示, 初始标识  $M_0 = [10000000]$ , 目的标识  $M_d = [00000001]$ .

问题: 判定  $M_d$  是否由  $M_0$  可达.

算法判定过程:

第 1 步. 求得满足状态方程的解向量  $\mathbf{X} = [11110110]$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}.$$

第 2 步. 求得该网的原子进程段如图 5 所示,  $T$ -向量集为  $\mathbf{Y}_{AP}(\Sigma) = \{\mathbf{Y}_1 = [0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0], \mathbf{Y}_2 = [1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0], \mathbf{Y}_3 = [0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0], \mathbf{Y}_4 = [0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0], \mathbf{Y}_5 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0], \mathbf{Y}_6 = [0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1]\}$ .

第 3 步. 应用 LINGO 软件解线性不等式方程

$$\mathbf{X} - a_1 \mathbf{Y}_1 - a_2 \mathbf{Y}_2 - a_3 \mathbf{Y}_3 - a_4 \mathbf{Y}_4 - a_5 \mathbf{Y}_5 - a_6 \mathbf{Y}_6 \geq 0.$$

求得唯一一组整数解

$$a_1 = 1, a_2 = 1, a_3 = 1, a_4 = 0, a_5 = 1, a_6 = 0,$$

使得  $\mathbf{X} > a_1 \mathbf{Y}_1 + a_2 \mathbf{Y}_2 + a_3 \mathbf{Y}_3 + a_4 \mathbf{Y}_4 + a_5 \mathbf{Y}_5 + a_6 \mathbf{Y}_6$  成立.

第 4 步. 依据此解选取进程段集  $\{P_1, P_2, P_3, P_5, P_6\}$ , 偏序关系集  $\{P_1 \rightarrow P_2, P_1 \rightarrow P_3, P_1 \downarrow P_4, P_1 \leftrightarrow P_2, P_2 \rightarrow P_1, P_2 \rightarrow P_3, P_3 \rightarrow P_5, P_3 \rightarrow P_6, P_5 \rightarrow P_6, P_5 \downarrow P_6, P_6 \rightarrow P_6\}$ .

第 5 步. 验证存在推导序列  $P_2 \Rightarrow P_1 \Rightarrow P_3 \Rightarrow P_5 \Rightarrow P_6$  且  $u_0 \in P_2, \varphi(u_0) = M_0, u_d \in P_6, \varphi(u_d) = M_d$ , 则

$$\Gamma(P_2 P_1 P_3 P_5 \text{ sub}(P_6)) = t_1 t_2 t_3 t_4 t_7 t_6.$$

以解向量  $\mathbf{X} = [1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0]$  为发生数向量, 故判定  $M_d$  由  $M_0$  可达.

### 4.3 时间复杂性分析

算法中, 可达性判定不再以单个标识作为判定对象, 而是通过原子进程段形式, 把若干带原子特性的  $s$ -切模块化成一个整体, 进行可达性判定判断. 直观上理解, 这种类似状态聚合的判定思想跳过模块中若干状态判定, 将带来工作量缩减. 分析算法知, 它的判定效率取决于原子进程段集、推理规则集求取和基于规则的推导序列验证两部分. 其时间复杂度是依赖于原子进程段集规模而不是网规模. 分析原子进程段集规模如下:

设极小基本进程段集为  $P(\Sigma) = \{p_1, p_2, \dots, p_k\}$ , 记极小基本进程段长度  $l$  为其包含的变迁个数, 则  $l(P(\Sigma)) = \{l_1, l_2, \dots, l_k\}$ . 由文献[5-8]知, 在功效等价意义下, 任意 Petri 网的极小基本进程段是有限的, 设为  $\max(l_1, l_2, \dots, l_k) = h$ , 则有  $1 \leq h \leq n = |T|$ . 若第  $i$  个极小基本进程段划分得到  $v_i$  个原子进程段,

则有

$$p_i = \{ap_{i_1}, ap_{i_2}, \dots, ap_{i_{v_i}}\}, l_i = l_{i_1} + l_{i_2} + \dots + l_{i_{v_i}}.$$

设 Petri 网原子进程段集  $AP(\Sigma) = \{ap_1, ap_2, \dots, ap_q\}$ , 原子进程段个数  $q$  为  $\sum_{i=1}^k v_i$ , 则有  $k \leq \sum_{i=1}^k v_i \leq n$ .

即原子进程段集规模介于极小基本进程段集规模与网变迁集规模之间, 最坏情况不会超过网中变迁集规模.

## 5 总结与展望

Petri 网的可达性判定问题是本领域研究的重点之一. 本文分析极小基本进程段不能直接应用于可达性判定的原因, 基于进程子段偏序关系分解引入原子进程段概念, 并给出求取 Petri 网的原子进程段集、原子进程段间偏序关系集方法. 其后, 算法借鉴形式语言(以原子进程段间偏序关系为推理规则)中句型推导思想, 以状态方程解向量、原子进程段  $T$ -向量的线性纯整数规划解为标准选取待验证原子进程段子集、偏序关系子集, 验证是否存在从初始标识到目的标识的推导序列而判定可达性. 本文的主要贡献在于:

- (1) 提出原子进程段概念并给出原子进程段集、原子进程段间偏序关系集的求取方法;
- (2) 证明 Petri 网的进程网系统产生的语言是 Petri 网可达标识对间合法变迁引发序列集的真子集;
- (3) 证明状态方程解向量与原子进程段  $T$ -向量的纯整数规划问题解是可达性判定的必要非充分条件.

分析表明, 算法的复杂性依赖于原子进程段集规模, 最坏情况不会超过网中变迁集规模. 相比于文献[3]的判定算法, 更具模块化. 在一些状态可达性模块化特征明显的 Petri 网(如分层 CPN)进行判定时, 本算法能够得到较好效果.

然而, 本文算法未能对其时间复杂度给出严格的数学证明, 序列验证中的推导方法简单. 针对以上问题, 今后不妨从两方面深入: (1) 研究 Petri 网的状态空间组合问题与网规模间的数学关系; (2) 研究如何应用形式文法与自动机理论中的成熟方法, 设计高效的规则推理算法, 提高序列验证效率.

## 参 考 文 献

- [1] Kostin Alexander E. Reachability analysis in  $T$ -invariant-less Petri nets. IEEE Transactions on Automatic Control, 2003, 48(6): 1019-1024

- [2] Kostin Alexander E. A reachability algorithm for general Petri nets based on transition invariants//Kralovic R, Urzyczyn P eds. Proceedings of the MFCS. LNCS 4162. Berlin Heidelberg: Springer-Verlag, 2006; 608-621
- [3] Yu Feng, Luo Jun-Zhou, Li-Wei. Determining algorithm for legal firing transition sequence of Petri net based on  $T$ -invariant eliminating. Journal of PLA University of Science and Technology(Natural Science Edition), 2008, 9(5): 522-527 (in Chinese)  
(于枫, 罗军舟, 李伟. 基于  $T$ -不变量消除的 Petri 网合法引发行序列判定算法. 解放军理工大学学报(自然科学版), 2008, 9(5): 522-527)
- [4] Satoshi Taoka, Shinji Furusato, Toshimasa Watanabe. A heuristic algorithm FSDC based on avoidance of deadlock components in finding legal firing sequences of Petri nets//van der Aalst W M P, Best E eds. Proceedings of the ICAT-PN. LNCS 2679. Berlin Heidelberg: Springer-Verlag, 2003; 417-439
- [5] Wu Zhe-Hui. Process expression of bounded Petri net. Science in China (Series E), 1996, 39(1): 37-49
- [6] Wu Zhe-Hui, Wang Pei-Liang, Zhao Mao-Xian. Process expression of unbounded fair Petri net. Chinese Journal of Computers, 2000, 23(4): 337-344(in Chinese)  
(吴哲辉, 王培良, 越茂先. 无界公平 Petri 网的进程表达式. 计算机学报, 2000, 23(4): 337-344)
- [7] Zeng Qing-Tian, Wu Zhe-Hui. Process net system of Petri net. Chinese Journal of Computers, 2002, 25(12): 1308-1315(in Chinese)  
(曾庆田, 吴哲辉. Petri 网的进程网系统[J]. 计算机学报, 2002, 25(12): 1308-1315)
- [8] Zeng Qing-Tian, Wu Zhe-Hui. Process expression of unbounded Petri net. Chinese Journal of Computers, 2003, 26(12): 1629-1636(in Chinese)  
(曾庆田, 吴哲辉. 无界 Petri 网的进程表达式. 计算机学报, 2003, 26(12): 1629-1636)
- [9] Zeng Qing-Tian. A construction method for the process expression of a Petri net based on synchronization composition. Chinese Journal of Computers, 2008, 31(3): 381-390 (in Chinese)  
(曾庆田. 一种基于同步合成构造 Petri 网进程表达式的方法. 计算机学报, 2008, 31(3): 381-390)
- [10] Wu Zhe-Hui. Theory of Petri Net. Beijing: China Machine Press, 2006(in Chinese)  
(吴哲辉. Petri 网导论. 北京: 机械工业出版社, 2006)
- [11] Jiang Chang-Jun. Theory of PN Machine, in DEDS. Beijing: Science Press, 2000(in Chinese)  
(蒋昌俊. 离散事件动态系统的 PN 机理论. 北京: 科学出版社, 2000)

<p>原子进程段及其偏序关系求取算法. 输入: 极小基本进程段集 <math>BP_e(\Sigma) = (P_1, P_2, \dots, P_k)</math> 输出: 原子进程段集 <math>AP(\Sigma)</math>, 原子进程段关系集 <math>R_a(\Sigma)</math> 设 <math>P_i = (N(u_{i1}, u_{i2}), \varphi)</math>, <math>P = (N(u_{j1}, u_{j2}), \varphi)</math>, <math>\Gamma: P \rightarrow T^*</math></p> <p>Begin</p> <p>1. <math>AP(\Sigma) = BP_e(\Sigma)</math>, <math>R_a(\Sigma) = \emptyset</math>;</p> <p>2. <math>\forall P_i, P_j \in AP(\Sigma), i \neq j</math>;</p> <p><math>A_i = \{M_j \mid M_j = \varphi(u_j) \wedge u_j \in P_i\}</math>;</p> <p><math>B_i = \{sub(\sigma_j) \mid \sigma_j = \Gamma(P_j)\}</math>;</p> <p><math>A_j = \{M_i \mid M_i = \varphi(u_i) \wedge u_i \in P_j\}</math>;</p> <p><math>B_j = \{sub(\sigma_i) \mid \sigma_i = \Gamma(P_i)\}</math>;</p> <p>Swith (<math>a =  A_i \cap A_j </math>)</p> <p>Case 0</p> <p><math>\{P_i, P_j</math> 添加到 <math>AP(\Sigma)</math>, 关系 <math>P_i \parallel M P_j</math> 添加到 <math>R_a(\Sigma)</math></p> <p>Case 1</p> <p><math>\{A_i \cap A_j = M \wedge \sigma_i \cap \sigma_j = \emptyset</math>;</p> <p>If <math>M = \varphi(u_{i1}) = \varphi(u_{j1})</math> Then</p> <p><math>\{P_i, P_j</math> 添加到 <math>AP(\Sigma)</math>, 关系 <math>P_i \leftrightarrow_M P_j</math> 添加到 <math>R_a(\Sigma)</math>};</p> <p>Endif</p> <p>If <math>\exists u, u_{i1} &lt; u &lt; u_{i2}, u_{j1} &lt; u &lt; u_{j2}, M = \varphi(u)</math> Then</p> <p><math>\{P_{i1} = (N(u_{i1}, u), \varphi), P_{i2} = (N(u, u_{i2}), \varphi)</math>,</p> <p><math>P_{j1} = (N(u_{j1}, u), \varphi), P_{j2} = (N(u, u_{j2}), \varphi)</math> 添加到 <math>AP(\Sigma)</math>,</p> <p>关系 <math>P_{i1} \rightarrow_M P_{i2}, P_{i1} \rightarrow_M P_{j2}, P_{j1} \rightarrow_M P_{j2}, P_{j1} \rightarrow_M P_{i2}, P_{i1} \downarrow_M P_{j1}</math>,</p> <p><math>P_{i2} \leftrightarrow_M P_{j2}</math> 添加到 <math>R_a(\Sigma)</math>};</p> <p>Endif</p> <p>If <math>\exists u, u = u_{i2}, u_{j1} &lt; u &lt; u_{j2}, M = \varphi(u)</math> Then</p> <p><math>\{P_i, P_{j1} = (N(u_{j1}, u), \varphi), P_{j2} = (N(u, u_{j2}), \varphi)</math> 添加到 <math>AP(\Sigma)</math>,</p> <p>关系 <math>P_i \rightarrow_M P_{j2}, P_i \downarrow_M P_{j1}, P_{j1} \rightarrow_M P_{j2}</math> 添加到 <math>R_a(\Sigma)</math>};</p> <p>Endif</p> <p>Case 2</p> <p><math>A_i \cap A_j = \{M, M'\} \wedge \sigma_i \cap \sigma_j = \emptyset</math>;</p> <p>If <math>M = \varphi(u_{i1}) = \varphi(u_{j1}), M' = \varphi(u_{i2}) = \varphi(u_{j2})</math> Then</p> <p><math>\{P_i, P_j</math> 添加到 <math>AP(\Sigma)</math>, 关系 <math>P_i \leftrightarrow P_j</math> 添加到 <math>R_a(\Sigma)</math>};</p> <p>Endif</p> <p>If <math>u_{j1} &lt; u_{i2} &lt; u_{j2}, M = \varphi(u_{i1}) = \varphi(u_{j1}), M' = \varphi(u_{i2})</math> Then</p>	<p><math>\{P_i, P_{j1} = (N(u_{j1}, u_{i2}), \varphi), P_{j2} = (N(u_{i2}, u_{j2}), \varphi)</math> 添加到 <math>AP(\Sigma)</math>,</p> <p>关系 <math>P_i \downarrow P_{j1}, P_{i1} \rightarrow_M P_{j2}, P_{j1} \rightarrow_M P_{j2}</math> 添加到 <math>R_a(\Sigma)</math>};</p> <p>Endif</p> <p>If <math>\exists u, u_{i1} &lt; u &lt; u_{i2}, u_{j1} &lt; u &lt; u_{j2}, M = \varphi(u_{i1}) = \varphi(u_{j1}), M' = \varphi(u)</math></p> <p>Then</p> <p><math>\{P_{i1} = (N(u_{i1}, u), \varphi), P_{i2} = (N(u, u_{i2}), \varphi)</math>,</p> <p><math>P_{j1} = (N(u_{j1}, u), \varphi), P_{j2} = (N(u, u_{j2}), \varphi)</math> 添加到 <math>AP(\Sigma)</math>,</p> <p>关系 <math>P_{i1} \downarrow P_{j1}, P_{i1} \rightarrow_M P_{i2}, P_{i1} \rightarrow_M P_{j2}, P_{j1} \rightarrow_M P_{j2}, P_{j1} \rightarrow_M P_{i2}</math> 添加到 <math>R_a(\Sigma)</math>};</p> <p>Endif</p> <p>Default</p> <p><math>(A_i \cap A_j = \{M_1 \dots M'\} \wedge (\sigma_i \cap \sigma_j = \sigma') \wedge (M[\sigma' &gt; M'])</math>;</p> <p>If <math>\exists u, u', u_{i1} &lt; u &lt; u' = u_{i2}, u_{j1} &lt; u &lt; u' = u_{j2}, M = \varphi(u)</math>,</p> <p><math>M' = \varphi(u')</math> Then</p> <p><math>\{P_{i1} = (N(u_{i1}, u), \varphi), P^\cap = (N(u, u_{i2}), \varphi), P_{j1} = (N(u_{j1}, u), \varphi)</math></p> <p>添加到 <math>AP(\Sigma)</math>,</p> <p>关系 <math>P_{i1} \downarrow P_{j1}, P_{i1} \rightarrow_M P^\cap, P_{j1} \rightarrow_M P^\cap</math> 添加到 <math>AP(\Sigma)</math>};</p> <p>Endif</p> <p>If <math>\exists u, u', u_{i1} &lt; u &lt; u' = u_{i2}, u_{j1} &lt; u &lt; u' &lt; u_{j2}, M = \varphi(u), M' = \varphi(u')</math></p> <p>Then</p> <p><math>\{P_{i1} = (N(u_{i1}, u), \varphi), P_{j1} = (N(u_{j1}, u), \varphi), P^\cap = (N(u, u'), \varphi)</math>,</p> <p><math>P_{j2} = (N(u', u_{j2}), \varphi)</math> 添加到 <math>AP(\Sigma)</math>,</p> <p>关系 <math>P_{i1} \downarrow P_{j1}, P_{i1} \rightarrow_M P^\cap, P_{j1} \rightarrow_M P^\cap, P^\cap \rightarrow P_{j2}</math> 添加到 <math>AP(\Sigma)</math>};</p> <p>Endif</p> <p>If <math>\exists u, u', u_{i1} &lt; u &lt; u' &lt; u_{i2}, u_{j1} &lt; u &lt; u' &lt; u_{j2}, M = \varphi(u), M' = \varphi(u')</math></p> <p>Then</p> <p><math>\{P_{i1} = (N(u_{i1}, u), \varphi), P_{j1} = (N(u_{j1}, u), \varphi), P^\cap = (N(u, u'), \varphi)</math>,</p> <p><math>P_{i2} = (N(u', u_{i2}), \varphi), P_{j2} = (N(u', u_{j2}), \varphi)</math> 添加到 <math>AP(\Sigma)</math>,</p> <p>关系 <math>P_{i1} \downarrow P_{j1}, P_{i1} \rightarrow_M P^\cap, P_{j1} \rightarrow_M P^\cap, P^\cap \rightarrow P_{i2}, P^\cap \rightarrow P_{j2}</math>,</p> <p><math>P_{i2} \leftrightarrow_M P_{j2}</math> 添加到 <math>AP(\Sigma)</math>};</p> <p>Endif</p> <p>3. If <math>AP(\Sigma) \cap BP(\Sigma) \neq \emptyset</math> Then</p> <p><math>AP(\Sigma) = AP(\Sigma) - AP(\Sigma) \cap BP(\Sigma)</math>;</p> <p>End</p>
--	---

附图 1 原子进程段及其偏序关系求取算法



**YU Feng**, born in 1974, Ph. D. candidate, lecturer. Her research interests include next generation Internet, network management, theory & application of Petri net.

**LUO Jun-Zhou**, born in 1960, professor, Ph. D. supervisor. His research interests mainly include next generation

network, protocol engineering, network security & management, grid computing, service computing, etc.

**LI Wei**, born in 1978, Ph. D., lecturer. His research interests include next generation network, network management, service computing, etc.

**WANG Peng**, born in 1983, Ph. D. candidate. His research interests include next generation Internet, network management.

## Background

This work is supported by National Natural Science Foundation of China under grants No. 60773103 and No. 60903161, National Basic Research Program (973 Program) of China under grant No. 2010CB328104, China Specialized Research Fund for the Doctoral Program of Higher Education (200802860031), Jiangsu Provincial Natural Science Foundation of China under Grants No. BK2007708 and No. BK2008030, Jiangsu Provincial Key Laboratory of Network and Information Security under grant No. BM2003201 and Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education under grant No. 93K-9.

In using Petri net for physical systems modeling and analyzing, properties determination of Petri net is a significant research focus. Reachability verification is a classical problem among them, but also a difficult one to design general determining algorithm for ordinary Petri net. Limited by state explosion problem, it is not easy to finish reachability verifica-

tion besides some special subclass.

Since process expression is very convenient in recording net dynamic behaviors, but the basic component of it named as minimal basic process segment is not fit for applying directly in reachability determining. So, it was decomposed into atomic process segment based on partial relation. Method for getting atomic process segment set and partial relations among them is given. With solution to pure linear nonequivalent expression of state equation solution vector and  $T$ -vectors of atomic process segments calculated, the authors design an algorithm which verifies the existence of reductive sequence by use of selected atomic process segment subset and partial relation subset according to the gotten solution. The complexity analysis shows it is depended on scale of atomic process segment set, no more than the scale of transition set under worst condition. The work the authors done have no constraint on Petri net or application condition, so it is general naturally.