

Petri 网的进程文法和进程语言^{*}

Process Grammar and Process Language of Petri Nets

吴哲辉

张继军

(山东科技大学信息学院 泰安 271019) (山东农业大学理学院 泰安 271018)

Abstract The concept of process grammar, based on the idea of vector grammar, for petri net is presented in this paper. The process language of a petri net can be generated by derivation according to its process grammar. The set of basic process section sequence represented by the process language of a petri net Σ is the same as that represented by the process expression of Σ if Σ is bounded or fair. In the case of that Σ is unbounded and unfair, the set of all process of Σ can also be represented by its process language passed through parallel processing.

Keywords Petri net, Process, Process grammar, Process expression

一、引言

Petri 网进程是分析网系统运行行为最有力的工具。Petri 网进程既考虑了网系统的状态变化又准确记录了引起这些变化的条件,清晰地反映事件(变迁)之间的并发和顺序关系,因此, Petri 网进程比其它分析方法更便于对并发现象以及系统中同并发有关的一些性质进行分析。然而, Petri 网的一个进程只能反映 Petri 网的一种可能的运行情况。一个 Petri 网往往有许多个进程,不可能一一列举,这就为进程分析带来了困难。文[1,2]中,提出了利用进程表达式描述 Petri 网的所有进程的方法,一个有界 Petri 网的进程表达式是该网的基本子进程集(有限集)为字母表的正规表达式,一个无界公平 Petri 网在可重复进程段行为等价的条件下也是一个正规表达式。对于一般的(不公平的)无界 Petri 网,文[3]已经证明其基本进程段的集合是一个有限集,并给出了一个求解基本进程段集合的算法。然而由于进程段之间的衔接关系的复杂性,不可能用正规表达式来表示出来。

本文根据向量文法的基本思想提出了 Petri 网进程文法的概念。一个 Petri 网 Σ 的进程文法是根据 Σ 的基本进程段集和 Σ 的初始标识定义的一个向量文法。根据向量文法的推导规则,由 Σ 的进程文法产生的语言称为 Σ 的进程语言。我们证明了当 Σ 为一个有界 Petri 网或无界公平 Petri 网时, Σ 的进程语言 PL(Σ)同文[1,2]定义的 Σ 的进程表达式 EXP($P(\Sigma)$)所表示的 Σ 的基本进程段序列的集合是相同的,它们都可以表示出 Σ 的全部进程的集合。当 Σ 不是有界 Petri 网也不是无界公平 Petri 网时,通过 PL(Σ)进行并行化处理也可以表示出 Σ 的全部进程的集合。这显示出 Petri 网的进程文法和进程语言的概念比 Petri 网的进程表达式的概念有更强的表示能力。

二、Petri 网进程的基本概念及有关结论

假设读者对 Petri 网的基本概念已有所了解。这里只给出同 Petri 网的进程和进程表达式的有关概念、术语和记号,以便于后面的讨论。

定义 2.1(Petri 网进程)^[1] 设 $\Sigma = (S, T; F, M_0)$ 为一个

Petri 网, $N = (B, E; G)$ 为一个出现网,若映射 $\varphi: B \cup E \rightarrow S \cup T$ 满足条件:

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

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

$$\forall e \in E; \varphi(e') = \varphi(e) \wedge \varphi(e'') = \varphi(e)';$$

则称 φ 定义了从 N 到 Σ 的一个映射,记为 $\varphi: N \rightarrow \Sigma$ 。如果这个网映射满足条件:

$$1) \forall b_1, b_2 \in B (b_1 \neq b_2);$$

$$\varphi(b_1) = \varphi(b_2) \rightarrow b_1' \neq b_2' \wedge b_1'' \neq b_2'';$$

$$2) \forall s \in S; |\{b | \varphi(b) = s \wedge b' = \phi\}| \leq M_0(s),$$

则称 (N, φ) 为 Petri 网的一个进程,记为 $P = (N, \varphi)$ 。□

Petri 网 $\Sigma = (S, T; F, M_0)$ 的一个进程,表示网的一种可能的运行情况,而网的所有的运行情况用 $P(\Sigma)$ 表示,称为网 Σ 的进程集。

定义 2.2(进程段)^[2] 设 $P = (N, \varphi)$ 为 Σ 的一个满进程,其中 $\Sigma = (S, T; F, M_0)$, $N = (B, E; G)$, u_1, u_2 为 N 的两个 S-切, $u_1 \leq u_2$ 。记:

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

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

$$G_1 = G \cap ((B_1 \times E_1) \cup (E_1 \times B_1)).$$

令 $N_1 = (B_1, E_1; G_1)$, $\varphi_1: N_1 \rightarrow \Sigma$ 满足 $\forall x \in B_1 \cup E_1; \varphi_1(x) = \varphi(x)$, 则称 (N, φ_1) 为进程 P 界于 u_1 和 u_2 之间的进程段。记作 $P_1 = (N[u_1, u_2], \varphi)$ 。□

定义 2.3(基本进程段)^[3] 设 $P = (N, \varphi)$ 为 Σ 的一个满进程,设 $P_1 = (N[u_1, u_2], \varphi)$ 为 Petri 网 Σ 的一个进程段,如果 $N[u_1, u_2]$ 中的任意两个 S-切 u_i 和 $u_j (i, j \neq 1, 2)$ 都有: $u_i \neq u_j \rightarrow (\varphi(u_i) \neq \varphi(u_j)) \wedge (\varphi(u_i) \leq \varphi(u_j)) \wedge (\varphi(u_i) \not\leq \varphi(u_j))$, 则称 $P_1 = (N[u_1, u_2], \varphi)$ 是 Σ 的一个基本进程段。□

定义 2.4 设 $P_1 = (N[u_1, u_2], \varphi)$ 为 Σ 的一个基本进程段,令 $\varphi(u_1) = M_{u_1}, \varphi(u_2) = M_{u_2}$,

(1) 若 $\varphi(u_1) = \varphi(u_2)$, 则称 P_1 为 Σ 的基本不变进程段;

(2) 若 $\varphi(u_1) < \varphi(u_2)$, 则称 P_1 为 Σ 的基本增进进程段; 若 $M_{u_2}(s_i) > M_{u_1}(s_i)$, 则 s_i 称为 P_1 的输出库所;

*)本课题的工作得到国家自然科学基金(60173053, 69573018)的资助。吴哲辉 教授,博士生导师,主要研究方向为 Petri 网理论及应用、算法设计与分析等。张继军 副教授,主要研究方向为 Petri 网理论及应用、软件工程。

(3) 若 $\varphi(u_1) > \varphi(u_2)$, 则称 P_1 为 Σ 的基本减进程段; 若 $M_{u_2}(s_i) < M_{u_1}(s_i)$, 则 s_i 称为 P_1 的驱动库所;

(4) 若 $(\varphi(u_1) \neq \varphi(u_2)) \wedge (\varphi(u_1) \leq \varphi(u_2)) \wedge (\varphi(u_1) \not\geq \varphi(u_2))$, 则称 P_1 为 Σ 的基本传递进程段; 若 $M_{u_2}(s_i) > M_{u_1}(s_i)$, 则 s_i 称为 P_1 的输出库所; 若 $M_{u_2}(s_i) < M_{u_1}(s_i)$, 则 s_i 称为 P_1 的驱动库所。□

将上面的四种类型的进程段统称基本进程段; 其中将基本不变进程段和基本增进程段称为 Σ 的基本可重复进程段; 基本减进程段和基本传递进程段称为 Σ 的基本受控可重复进程段。一个 Petri 网 Σ 的所有的基本进程段构成的集合称为基本进程段集合。

在文[2]定义了进程段的行为等价, 而在文[3]中证明了: 一个 Petri 网 Σ 在进程段的行为等价的意义上, 基本进程段集是一个有限集, 并给出了求 Petri 网 Σ 基本进程段集的一个有效算法。(详细的算法过程见文[3])。

三、Petri 网的进程文法和进程语言

文[4]提出了向量文法的概念, 并通过向量文法来定义 Petri 网语言。这里我们借助于向量文法的思想方法定义 Petri 网的进程文法, 通过向量文法的推导规则可以得到 Petri 网的进程语言。

定义 3.1 设 $\Sigma = (S, T, F, M_0)$ 为一个 Petri 网, $|S| = m$, Σ 的基本进程段集合为 $BP(\Sigma) = \{P_1, P_2, \dots, P_n\}$, 其中 $P_i = (N[u_{i1}, u_{i2}], \varphi)$ (记 $\varphi(u_{ij}) = M_{ij}$, $i=1, 2, \dots, n; j=1, 2$)。 Σ 的进程文法定义为:

$$PG(\Sigma) = (PV(\Sigma), BP(\Sigma), PP(\Sigma), M_0) \quad (3.1)$$

其中:

1) $PV(\Sigma)$ 是一个非负整数 m 维向量的集合:

$$PV(\Sigma) = \bigcup_{i=1}^n \{\varphi(u_{i1})\} \cup \bigcup_{i=1}^n \{\varphi(u_{i2})\} = \bigcup_{i=1}^n \{M_{i1}\} \cup \bigcup_{i=1}^n \{M_{i2}\} \quad (3.2)$$

2) $M_0 \geq \varphi(u_{11}) = M_{11}$ 为 Σ 的初始标识向量;

3) $PP(\Sigma)$ 是有 n 个产生式组成的集合, 这些产生式为:

$$\varphi(u_{i1}) \rightarrow P_i \varphi(u_{i2}) \quad i=1, 2, \dots, n \quad (3.3)$$

(或记为 $M_{i1} \rightarrow P_i M_{i2}$) □

定义 3.2 设 Σ 为一个 Petri 网, $PG(\Sigma)$ 的推导规则为:

1) 若 $M_{i1} \rightarrow P_i M_{i2}$ 为 $PG(\Sigma)$ 的一个产生式, 则对任意 $M \geq M_{i1}$ 和任意 $\alpha \in BP(\Sigma)^*$ 都有

$$\alpha M \Rightarrow \alpha P_i (M_{i2} + M - M_{i1}) \quad (3.4)$$

2) 设 $\alpha_1, \alpha_2, \dots, \alpha_k \in BP(\Sigma)^*$, $M_1, M_2, \dots, M_k$ 为 m 维非负整数向量 ($k > 0$), 若 $\alpha_1 M_1 \Rightarrow \alpha_2 M_2 \Rightarrow \dots \Rightarrow \alpha_{k-1} M_{k-1} \Rightarrow \alpha_k M_k$

$$\text{则 } \alpha_1 M_1 \Rightarrow \alpha_k M_k \quad \square$$

定义 3.3 设 Σ 为一个 Petri 网, $PG(\Sigma) = (PV(\Sigma), BP(\Sigma), PP(\Sigma), \varphi(u_{11}))$ 为 Σ 的进程文法, 则 Σ 的进程语言定义为:

$$PL(\Sigma) = \{w \in BP(\Sigma)^* \mid \varphi(u_{11}) \xrightarrow{w} M, M \text{ 为任意非负整数向量}\} \quad (3.5) \square$$

对任意给定的一个 Petri 网, 都可以通过文[3]给出的算法求出 Σ 的基本进程段的集合 $BP(\Sigma)$, 而且 $BP(\Sigma)$ 为一个有限集。设 $|BP(\Sigma)| = n$, 则由 (3.2) 和 (3.3) 知 $|PV(\Sigma)| \leq 2n$, $|PP(\Sigma)| = n$ 。这表明, 对任意一个 Petri 网 Σ , 其文法的终极符集、变量集和产生式集都是有限集, 而且只要求出了 Σ 的基本进程段的集合 (按文[3]给出的算法即可求出) $BP(\Sigma)$, 那么 $PV(\Sigma)$ 和 $PP(\Sigma)$ 都是容易得到的。通过定义 3.2 给出的推导

规则就可以确定 Σ 的进程语言 $PL(\Sigma)$ 。这表明, Petri 网的进程语言的概念适用于所有的 Petri 网 (不论是有界的还是无界的, 公平的还是不公平的)。这是用 Petri 网进程语言比用 Petri 网的进程表达式来描述一个 Petri 网的全部进程的集合的优越之处, 因为 Petri 网的进程表达式 (以基本进程段集合为字母的正规表达式) 的概念只适用于有界 Petri 网和无界公平 Petri 网^[1,2]。

然而, 当一个 Petri 网 Σ 不是有界 Petri 网也不是无界公平 Petri 网时, Σ 的进程语言 $PL(\Sigma)$ 中的句子同 Σ 的进程之间并不是直接对应的。为了说明这一点, 我们先对 Petri 网的进程语言的性质做一点研究。

引理 3.1 设 Σ 为一个 Petri 网, $PL(\Sigma)$ 为 Σ 的进程语言, $w_1, w_2 \in PL(\Sigma)$, 若

$$w_1 = w_{11} w_{12} \quad (3.6)$$

$$w_2 = w_{11} w_{22} \quad (3.7)$$

且 w_{12} 和 w_{22} 都是 Σ 的可重复进程段, 则

$$w = w_{11} (w_{12} + w_{22})^* \subseteq PL(\Sigma)$$

证明: 先用数学归纳法证明对任意非负整数 k 都有:

$$w_{11} (w_{12} + w_{22})^k \subseteq PL(\Sigma) \quad (3.8)$$

显然, $w_{11} \in PL(\Sigma)$, $w_{11} (w_{12} + w_{22}) \subseteq PL(\Sigma)$, 即当 $k=0$, 1 时, (3.8) 成立。

假设 (3.8) 对 k 成立, 则对任意 $w_k \in (w_{12} + w_{22})^k$, 由于 w_{12} 和 w_{22} 是可重复进程段, w_k 也是 Σ 的可重复进程段, 记

$$M_0 \xrightarrow{w_{11}} M_1 \quad M_1 \xrightarrow{w_k} M_2$$

由 w_k 是可重复进程段知 $M_2 \geq M_1$, 从而有

$$M_2 \xrightarrow{w_{12}} M_{12} \quad M_2 \xrightarrow{w_{22}} M_{22}$$

这表明 $w_k w_{12} \in PL(\Sigma)$ 且 $w_k w_{22} \in PL(\Sigma)$, 注意到

$$w_{11} (w_{12} + w_{22})^{k+1} = \{w_k w_{12} + w_k w_{22} \mid w_k \in (w_{12} + w_{22})^k\}$$

就有:

$$w_{11} (w_{12} + w_{22})^{k+1} \subseteq PL(\Sigma) \quad \square$$

推论 3.1 设 Σ 为一个 Petri 网, $PL(\Sigma)$ 为 Σ 的进程语言, $w_1, w_2 \in PL(\Sigma)$, 若 $w_1 = w_{11} w_{12} w_{13}$; $w_2 = w_{11} w_{22} w_{23}$ 且 w_{12} 和 w_{22} 都是 Σ 的可重复进程段, 则有

$$w_{11} (w_{12} + w_{22})^* w_{13} \subseteq PL(\Sigma)$$

$$w_{11} (w_{12} + w_{22})^* w_{23} \subseteq PL(\Sigma)$$

证明: (由引理 3.1 可直接推出。) □

定理 3.1 若 Σ 为一个有界 Petri 网或者一个无界公平 Petri 网, 则

$$PL(\Sigma) = \text{EXP}(P(\Sigma)) \quad (3.9)$$

其中 $\text{EXP}(P(\Sigma))$ 为 Σ 的进程表达式。

证明: 由文[1,2]知, 当 Σ 为一个有界 Petri 网或无界公平 Petri 网时, Σ 的基本进程段之间只存在连接、冲突关系, 对可重复的基本进程段来说, 可以进行多重连接 (即幂运算) 和 Kleene 闭包运算, 这表明任意 $w \in \text{EXP}(P(\Sigma))$ 都可以从 Σ 的进程文法通过推导规则推出。

反之, 任意 $w \in PL(\Sigma)$, 由于 w 是从进程文法的推导规则推导得到, 这些推导规则对应着 (且只对应着) 进程段之间连接或选择, 因此 w 都是 $\text{EXP}(P(\Sigma))$ 中的一个元素。 □

推论 3.2 若 Σ 为一个有界 Petri 网或者一个无界公平 Petri 网, 则

1) 每个 $w \in PL(\Sigma)$ 都对应着 Σ 的一个进程;

2) p 是 Σ 的一个进程, 当且仅当存在一个 $w \in PL(\Sigma)$ 使得

$p \in \text{Pref}(w)$

其中 $\text{Pref}(w)$ 表示 w 的全部前缀的集合。

证明: 由文[1,2]知, 一个有界或无界公平 Petri 网 Σ 的全部进程的集合等于 Σ 的进程表达式表示的每个元素的前缀的集合, 即

$$(P(\Sigma) = \text{Pref}(\text{EXP}(P(\Sigma)))) \quad (3.10)$$

若 $p \in P(\Sigma)$, 则存在 $w \in \text{EXP}(P(\Sigma))$ 使得

$p \in \text{Pref}(w)$

从而, 存在 $w \in \text{PL}(\Sigma)$ 使得

$p \in \text{Pref}(w)$

反之, 如果 $w \in \text{PL}(\Sigma)$, 即 $w \in \text{EXP}(P(\Sigma))$ 使得任意 $p \in \text{Pref}(w)$ 都是 Σ 的一个进程, 从而 w 也是 Σ 的一个进程。□

四、Petri 网的并发进程语言

定义 4.1 设 Σ 为一个 Petri 网, $\text{PL}(\Sigma)$ 为 Σ 的进程语言, $P_i, P_j \in \text{BP}(\Sigma)$, 如果存在 $w_1, w_2 \in \text{PL}(\Sigma)$ 使得

$$1) w_1 = w_{11} P_i P_j w_{12}, w_2 = w_{11} P_j P_i w_{12}$$

$$2) M_0 \xrightarrow{*} w_{11} M_1$$

则称基本进程段 P_i 和 P_j 在 M_1 处于并发关系, 或者说在 w_1 (和 w_2) 中, P_i 和 P_j 是并发进程段。□

引理 4.1 若 Σ 为一个有界 Petri 网或者一个无界公平 Petri 网, 则对任意 $w \in \text{PL}(\Sigma)$, w 中都不存在并发进程段。□

证明: (由定理 3.1 可直接推出)

引理 4.2 设 Σ 为一个 Petri 网, $P_i, P_j \in \text{BP}(\Sigma)$, 如果存在 $w_1 \in \text{PL}(\Sigma)$ 使得 P_i 和 P_j 是 w_1 中的并发进程段, 则必有另一个元素 $w_2 \in \text{PL}(\Sigma)$, 使得 P_i 和 P_j 也是 w_2 中的并发进程段。

证明: 若 P_i 和 P_j 是 w_1 中的并发进程段, 则 w_1 必有下列形式:

$$w_1 = w_{11} P_i P_j w_{12}$$

$$\text{令 } w_2 = w_{11} P_j P_i w_{12}$$

则显然, $w_2 \in \text{PL}(\Sigma)$, 且 P_i 和 P_j 也是 w_2 中的并发进程段。□

定义 4.2 设 Σ 为一个 Petri 网, $w_1 \in \text{PL}(\Sigma)$, 若 w_1 中存在并发进程段 P_i 和 $P_j \in \text{BP}(\Sigma)$ 则对

$$w_1 = w_{11} P_i P_j w_{12} \quad (4.1)$$

其中, P_i 和 P_j 是并发进程段, 我们用并发序列(步序列)

$$w_1 = w_{11} (P_i \parallel P_j) w_{12} \quad (4.2)$$

替换 w_1 (以及 $w_2 = w_{11} P_j P_i w_{12}$), 并称这种替换为 w_1 (和 w_2) 的并行化处理。□

定义 4.3 设 Σ 为一个 Petri 网, $\text{PL}(\Sigma)$ 为 Σ 的进程语言, 如果对任意 $w \in \text{PL}(\Sigma)$, w 中若存在并发进程段, 则对 w 都进行并行化处理, 称得到的步序列的集合为 Σ 的并发进程语言, 记为 $\text{PPL}(\Sigma)$ 。□

根据定理 3.1 和推论 3.2 以及定义 4.2, 我们容易得到下列结论:

定理 4.1 对任意一个 Petri 网 Σ , 都有: 1) 若 $w \in \text{PPL}(\Sigma)$, 则 w 对应着 Σ 的一个进程; 2) 若 p 是 Σ 的一个进程, 则存在 $w \in \text{PPL}(\Sigma)$, 使得

$$p \in \text{Pref}(w) \quad (4.3)$$

例 4.1 对于图 1 所示的 Petri 网 Σ , 根据文[3]的算法, 可以求出它的基本进程段有 3 个: P_1 、 P_2 和 P_3 , (如图 2 所示), 其中 P_1 和 P_2 是可重复基本进程段。

根据定义 3.1, 我们可以写出 Σ 的进程文法 $\text{PG}(\Sigma) =$

$(\text{PV}(\Sigma), \text{BP}(\Sigma), \text{PP}(\Sigma), M_0)$, 其中:

$$\text{PV}(\Sigma) = \{M_{11}, M_{12}, M_{21}, M_{22}, M_{31}, M_{32}\}$$

$$M_{11} = \{1, 1, 0, 0, 0, 0, 0, 0\}^T$$

$$M_{12} = \{1, 1, 0, 0, 0, 0, 0, 1, 0\}^T$$

$$M_{21} = \{0, 1, 1, 0, 0, 0, 0, 0, 0\}^T$$

$$M_{22} = \{0, 1, 1, 0, 0, 0, 0, 0, 1\}^T$$

$$M_{31} = \{0, 0, 0, 0, 0, 0, 0, 1, 1\}^T$$

$$M_{32} = \{0, 1, 0, 0, 0, 0, 0, 0, 0\}^T$$

$$\text{BP}(\Sigma) = \{P_1, P_2, P_3\}$$

$\text{PP}(\Sigma)$:

$$M_{11} \rightarrow P_1 M_{12}$$

$$M_{21} \rightarrow P_2 M_{22}$$

$$M_{31} \rightarrow P_3 M_{32}$$

$$M_0 = \{1, 1, 1, 0, 0, 0, 0, 0\}^T$$

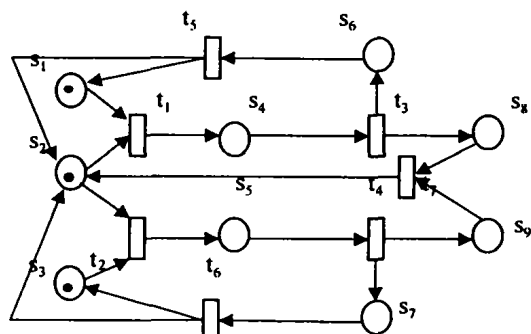


图 1 一个 Petri 网 Σ

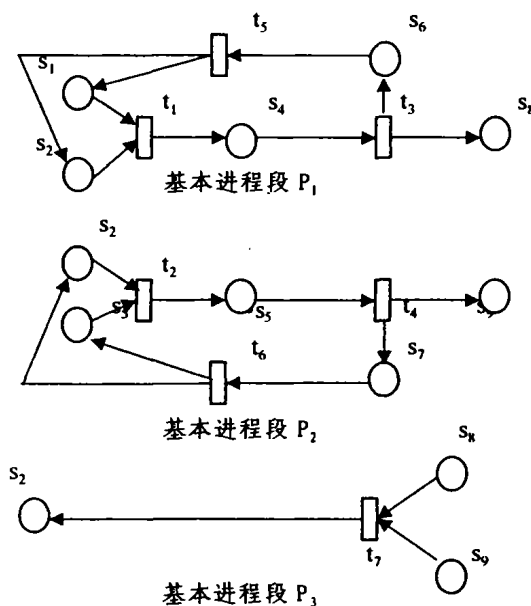


图 2 Σ 的 3 个基本进程段 P_1 、 P_2 和 P_3

通过推导规则, 可以证明

$$\text{PPL}(\Sigma) = (P_1 + P_2)^* + (P_1 P_2 + P_2 P_1) P_3 ((P_1 \parallel P_2) P_3)^* \quad (4.4)$$

其中 $((P_1 \parallel P_2) P_3)^* = \{w \mid w \in ((P_1 \parallel P_2) P_3)^* \text{ 且 } w \text{ 的任意前缀中 } (P_1 \parallel P_2) \text{ 出现的次数不小于 } P_3 \text{ 的出现的次数}\}$

容易证明, 任意进程段并发序列 $w \in \text{PPL}(\Sigma)$, w 都是 Σ 的一个进程。反之, 若 P 是 Σ 的一个进程, 则存在 $w \in \text{PPL}(\Sigma)$, 使得 $P \in \text{Pref}(w)$ 。

结束语 本文借助向量文法的思想定义了 Petri 网 Σ 的

第二十三届国际 Petri 网应用与理论会议散记

陆维明

(中国科学院数学与系统科学研究院 北京 100080)

在国家自然科学基金委员会(在研项目 60073013)与国家重大基础研究九七三(在研项目 G1998030416)资助下,我参加了第二十三届国际 Petri 网应用与理论会,作会议散记如下。它可以看作是文[1~4]的续篇。

1

2002 国际 Petri 网年会为第二十三届,在澳大利亚的阿德莱德(Adelaide)举行,由南澳大利亚大学组织,时间为 6 月的最后一周,6.24-6.30。与以往若干届会议相仿,会议参加者约 150 人,代表 30 多个国家;中国代表 6 名,1 位从香港来,2 位代表北京,3 位是在南澳大利亚大学攻读博士的大陆学生;其中三分之一来自工业界,其余来自大学和研究机构,没有政治问题、也从未发生过政治纠纷。

程序委员会以“一文两审”,“小同行评议”指导组织审稿,论文评审非常严格(录用率约 10%),保证了质量。本届年会的论文集由斯普林格(Springer)出版社出版,列为该社的灰皮系列丛书,计算机科学丛书(Lecture Notes in Computer Science, Vol. 2360)第 2360 卷。

2

本届会议受到各方重视,特别是澳大利亚国防科学与技术组织及阿德莱德市政府的支持。比如,会议的招待会,专门安排在阿德莱德市政大楼中最大一个房间(皇后厅)举行,由市长与南澳大利亚大学校长欢迎所有会议出席者及其亲属,他们在入口处与我们一一亲切握手并简短交谈;招待会时间不长,约一个半小时,但市长、校长热情的欢迎词,以及欧洲计算机科学协会主席(兼会议指导委员会主席)代表会议的幽默答词,使宾主在欣赏南澳著名葡萄酒与澳洲美味奶酪小点心的同时通过交流增加了相互了解、增进了友谊。

3

国际 Petri 网年会已举行了廿二届,取得了成功、积累了经验。本届会议在指导委员会的原则指导下,在组织委员会的精心安排下,除会议本身邀请了 6 个大会报告(均为具前瞻

性、前沿性的 Petri 网成果报告)外,还举办了 4 个附属学术活动:一为 Petri 网高级讲习班;一为形式化软件(特别是并发软件)工程方法专题研讨会;一为国防系统形式模型专题研讨会;一为 Petri 网软件工具(各类 Petri 网系统建模和分析的软件)展示、交流会。由于这些附属的学术活动是并行举行的,我挑选了第三项(参见文[6]),一天时间内接触到几个国家的国防系统形式模型,很有意思,很有兴趣,也很有收获。

4

年会共分为十二个分会,在南澳大利亚大学阿德莱德分部的教学区进行,学生已放假,省钱、清静、无干扰,效果不错。这些分会的论文均已收入文集,反映了当今国际上 Petri 网研究的水平;其中诺基亚(Nokia)公司研究中心的一项成果引人注目,关于移动通讯系统软件支持的研究。目前运行的这类通讯系统因用户骤增、不堪承受,过于繁忙、质量下降,甚至断线不通;他们利用着色 Petri 网设计了新的软件支持系统(手机通讯是典型的分布式并发系统,利用 Petri 网非常合适),不但通过设计考核验证,而且有的设计者据此获得提升。比如,论文作者之一曾是我的年青同事,他在诺基亚工作若干年,由于其它工作成绩和这项业绩,被提升为有 150 研究人员的 Nokia 研究中心仅有 3 名首席科学家之一。我们的论文宣读安排在最后一个分会的最后一个(参见文[5]),压轴,正在闭幕式之前,由于我们的工作基于 Petri 网结构理论,针对多项式实用算法,在网系统分析中得到一系列结果,反应很好。我们在会后拟增加更多结果,形成更有分量的长论文拟投国际上最好学术刊物之一,理论计算机科学(Theoretical Computer Science)。

5

我是第五次参加这一系列年会,觉得其架构比较理想:指导委员会只管原则,程序委员会严格审稿,且已有一批相对稳定小同行审稿人,能保证论文质量;组织委员会可以在安排好年会的前提下,合理组织若干项有关的学术活动,增加会议的吸引力,扩大交流范围,增强交叉学科的派生。比如,从这一

进程文法和 Petri 网 Σ 的进程语言。当一个 Petri 网为有界 Petri 网或无界公平 Petri 网时,根据 Σ 的进程文法推导得到的进程语言 $PL(\Sigma)$ 同文[1,2]中定义的进程表达式 $EXP(P(\Sigma))$ 是相等的,即它们所表示的 Σ 的基本进程段序列的集合是相同的。当 Σ 为无界不公平的 Petri 网时,我们可以通过对 $PL(\Sigma)$ 中的进程段序列进行并行化处理得到 Σ 的并发进程语言 $PPL(\Sigma)$ 。 $PPL(\Sigma)$ 的前缀集等于 Σ 的全部进程的集合。

参考文献

1 吴哲辉. 有界 Petri 网的进程表达式. 中国科学 A 辑, 1995, 25(12): 1332~1339

- 2 吴哲辉, 王培良, 赵茂先. 无界公平 Petri 网的进程表达式. 计算机学报, 2000, 23(4): 337~344
- 3 曾庆田, 吴哲辉. Petri 网的基本进程段的一个求解算法. 计算机科学, 28(7. 增刊): 5~11
- 4 蒋昌俊. 向量文法与 PN 机. 中国科学 A 辑, 1995, 25(12): 1315~1322
- 5 吴哲辉. Pumping 引理的 Petri 网描述——Petri 网语言属型的一组判定条件. 计算机学报, 1994, 17(11): 853~858
- 6 吴哲辉译. Petri 网理论与系统模拟. 徐州: 中国矿业大学出版社, 1989
- 7 Murata T. Petri nets: Properties Analysis and Applications. In: Proc. of The IEEE. 1989
- 8 Hopcroft J E, Ullman J D. Formal Languages and Their Relation to Automata. Addison-Wesley Reading M A. 1969