无界 Petri 网的进程表达式

曾庆田[1),2),3)] 吴哲辉[2)]

[1)](中国科学院计算技术研究所 北京 100080)

[2)](山东科技大学计算机科学与技术系 泰安 271019)

[3)](中国科学院研究生院 北京 100039)

摘 要 借助进程网系统的概念,首先证明了无界 Petri 网 Σ 的进程与其进程网系统 Σ_P 的语言之间存在一一映射关系,将求取 Σ 的进程表达式的问题转化成求取 Σ_P 的语言问题. 由于 Σ_P 的结构一般比较复杂,直接求取其语言仍然比较困难. 通过定义库所的指标函数,将结构复杂的 Σ_P 分解成结构简单的子网系统,引入了语言的同步交运算,给出了利用这些子网系统求取结构复杂的 Σ_P 语言的方法,从而得到了求取无界 Petri 网 Σ 的进程表达式的算法.

关键词 无界 Petri 网;进程表达式;进程网系统;Petri 网语言;分解

中图法分类号 TP301

Process Expression of Unbounded Petri Net

ZENG Qing-Tian[1),2),3)] WU Zhe-Hui[2)]

[1)](*Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100080)

[2)](*Department of Computer Sciences and Technology, Shandong University of Science and Technology, Tai'an* 271019)

[3)](*Graduate School of Chinese Academy of Sciences, Beijing* 100039)

Abstract The one-to-one mapping relationship is proved between the process expression of an unbounded Petri net Σ and the language of its process net system Σ_P, so we can express the language of Σ_P in order to establish the process expression of Σ. However, it is not easy to describe the language behaviors of Σ_P directly because its structure is usually complex. A new decomposition method for structure-complex Petri net by assigning an index to each place is introduced, with which Σ_P can be decomposed into some structure-simple subnet systems. With these subnet systems, the language of Σ_P can be expressed based on the synchronization intersection operation of languages. By the description of the language of Σ_P, an algorithm to obtain the process expression of Σ is obtained.

Keywords unbounded Petri net; process expression; process net system; Petri net languages; decomposition

1 引 言

Petri 网的进程[1~6]是对系统行为描述和分析的有力工具,它可以很清楚地反映出网系统运行中变迁之间的顺序、并发、同步等现象. 然而,一个进程只能反映 Petri 网的一种可能运行情况. 一个 Petri 网往往有许多(可能无限多个)进程,可能无法一一列举,这就给利用进程分析 Petri 网的行为带来了许多困难. 文献[3,4]定义了 P/R 网,用以描述基本网系统的进程行为. 文献[5]定义了 Petri 网的进程网系统来刻画 Petri 网(尤其是无界 Petri 网)的进程行为. 文献[6,7]提出了进程表达式的概念,给出了有界 Petri 网和无界公平 Petri 网进程的形式刻画.

收稿日期:2002-09-23;修改稿收到日期:2003-03-31. 本课题得到国家自然科学基金(60173053,60274063)和山东省中青年科学家科研奖励基金(02BS069)资助. **曾庆田**,男,1976年生,博士研究生,研究方向为大规模知识处理与本体分析、知识系统 V&V、Petri 网理论与应用. E-mail:zqtian@ict.ac.cn. **吴哲辉**,男,1941年生,教授,博士生导师,主要研究方向为 Petri 网、算法设计与分析、形式语言等.

关于 Petri 网的进程表达式的研究首先是从有界 Petri 网开始的. 文献[6] 定义了满进程和基本子进程同构的概念,一个有界 Petri 网的进程表达式是以该网的基本子进程集为字母表的正规表达式. 证明了有界 Petri 网的进程表达式所表示的正规集是该网全体满进程的集合. 这样,对于一个有界 Petri 网,只要求出它的进程表达式,就等于给出了它的全体满进程. 文献[7] 对可重复进程段的概念进行了拓展,并以行为等价作为对进程段划分准则,以代替有界 Petri 网中按同构(包括错位同构)关系对子进程的划分,从而给出了无界公平 Petri 网的进程表达式. 这个表达式所表示的正规集,也包含了无界公平 Petri 网的全体满进程. 文献[8]对 Petri 网进程表达式的类型进行了划分,并研究了 Petri 网的语言表达式同进程表达式之间的内在关系,从网结构角度给出了判断 Petri 网进程表达式类型的代数依据.

文献[7]同时指出,要把进程表达式的概念推广到任意无界 Petri 网,会遇到两方面的困难. 一方面,由于正规表达式的字母表必须是一个有限集,是否可以(以及怎样)把任意一个无界 Petri 网的各种可能的运行情况都归结为有限个基本子进程的组装? 这是推广工作遇到的一个困难. 另外,对于任意无界 Petri 网的运行来说,当把它们表示成基本子进程的组装时,是否可以只用加法运算(表示选择)、乘法运算(表示连接)和 Kleene 闭包三种运算就可把各种组装方式表示出来? 这是第二个问题. 针对第一个问题,文献[5]对无界 Petri 网的基本进程段进行了类型的划分,将其分为增进程段、减进程段、传递进程段等,分析了各种类型的基本进程段在可达树上的分布特征,在进程段功能等效的意义下证明了无界 Petri 网的基本进程段集是一个有限集,并在文献[9]中给出了基本进程段集的具体求解算法. 为解决无界 Petri 网的进程段与进程段之间的组装方式问题,文献[5]用网结构代替表达式来描述进程段之间可能的衔接关系,从而提出了 Petri 网的进程网系统的概念. 一个 Petri 网 Σ 的进程网系统 Σ_P 就是以 Σ 的基本进程段集为变迁集,以 Σ 的基本进程段集的输入、输出库所集为库所集构成的一个新的网系统,Σ_P 能够很好地描述 Σ 的进程行为[5]. 但是同 Petri 网的进程表达式相比,进程网系统只是对进程行为的模型化描述,仍不能给出全体进程的形式化表示.

本文在先前工作[5~9]的基础上,通过进一步研究,证明了一个无界 Petri 网 Σ 的进程与其进程网系统 Σ_P 的语言之间存在着一一映射关系,从而将求取 Σ 的进程表达式的问题转化成求取 Σ_P 的语言表达式的问题. 由于 Σ_P 的结构一般比较复杂[5],所以直接写出其语言表达式仍然比较困难. 文中通过定义库所的指标函数,将 Σ_P 分解成结构简单的子网系统,引入语言的同步交运算,通过子网的语言给出了一种求取 Σ_P 的语言表达式的方法,从而得到了求取 Σ 的进程表达式的可行算法.

本文第 2 节介绍了与本文讨论有关的基本概念和术语,其中包括无界 Petri 网的满进程和基本进程段;第 3 节介绍了 Petri 网的进程网系统;第 4 节给出了无界 Petri 网的进程表达式的概念,并证明了进程表达式与进程网系统的语言表达式之间的一一映射关系;第 5 节首先给出基于库所指标的 Petri 网分解方法,通过引入语言的同步交运算给出了无界 Petri 网的进程表达式的求取算法. 最后给出了例子和全文的总结.

2 无界 Petri 网的进程与基本进程段

本文的讨论是在读者对 Petri 网及其进程的概念有所了解的假设前提下进行的,这里只对与本文讨论有关的基本概念、术语和记号做一下简述或约定.

定义 1[6]. 设 $N=(B,E;G)$ 为一个网,如果

(1) $\forall b\in B$: $|{}^\cdot b|\le 1 \wedge |b^\cdot|\le 1$;

(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[2]. 设 $N_1=(S,T;F)$ 为一个网,$N_2=(B,E;G)$ 为一个出现网,若映射 φ: $B\cup E\rightarrow S\cup T$ 满足条件:

(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) $\forall e\in E$: $\varphi({}^\cdot e)={}^\cdot\varphi(e)$, $\varphi(e^\cdot)=\varphi(e)^\cdot$.

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

定义 3[6]. 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 φ: $N\rightarrow N_1$ 满足条件

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

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

则称(N,φ)为Σ的一个进程.

定义 4[6]. 　设φ为出现网$N=(B,E;G)$到网$\Sigma=(S,T;F,M_0)$的一个网映射，如果

(1) $\forall b_1,b_2\in B(b_1\neq b_2):\varphi(b_1)=\varphi(b_2)\rightarrow({}^{\cdot}b_1\neq{}^{\cdot}b_2\vee{}^{\cdot}b_1={}^{\cdot}b_2=\Phi)\wedge(b_1^{\cdot}\neq b_2^{\cdot}\vee b_1^{\cdot}=b_2^{\cdot}=\Phi)$，

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

称$P=(N,\varphi)$为Σ的一个满进程.

定义 5[6]. 　设$P=(N,\varphi)$为Σ的一个满进程，其中，$N=(B,E;G)$，u_1,u_2为N的两个S-切，$u_1\le u_2$. 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$\forall x\in B_1\cup E_1:\varphi_1(x)=\varphi(x)$，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[5]. 设$P=(N,\varphi)$为Σ的一个满进程，$P_1=(N[u_1,u_2],\varphi)$是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\neq1,2)$都有$u_i\neq u_j\rightarrow(\varphi(u_i)\neq\varphi(u_j))\wedge(\varphi(u_i)\nless\varphi(u_j))\wedge(\varphi(u_i)\ngtr\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[5]. 设$P=(N[u_1,u_2],\varphi)$为 Petri 网$\Sigma=(S,T;F,M_0)$的一个进程段，P所对应的可达树是指这样的一棵树$Tree(P)$：

(1) $Tree(P)$根结点的标注为$M_1=\varphi(u_1)$；

(2) $Tree(P)$中有一个结点标注为M，当且仅当存在S-切$u:u_1\le u\le u_2$且$\varphi(u)=M$；

(3) $Tree(P)$中有两个结点M_j和M_k有一条有向边且其标注为t_r，当且仅当$M_j[t_r\rangle M_k$.

定义 8[5]. 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$\Sigma$的可覆盖树中对应的可达树子段相同，则说$P_1$和$P_2$是功能等效的.

定理 1[5]. 　设$\Sigma=(S,T;F,M_0)$为一个 Petri 网，则Σ的基本进程段集在功能等效的意义下是个有限集.

文献[6]定义了进程段的行为等价，易知在功能等效的进程段中，按行为等价的意义划分，只有有限多个的进程段. 因此，可以得到下面的结论.

推论 1. 　一个 Petri 网在行为等价意义下的基本进程段集是一个有限集.

在进程段功能等效和行为等价的意义下，无界 Petri 网的基本进程段集是一个有限集合. 由可覆盖树可以求得无界 Petri 网的全体基本进程段，具体方法可参见文献[9]. 图 1 给出了一个无界 Petri 网Σ（其中，S_3和S_5都是无界库所），由文献[9]中基本进程段的求取算法求得Σ的 5 个基本进程段如图 2 所示.

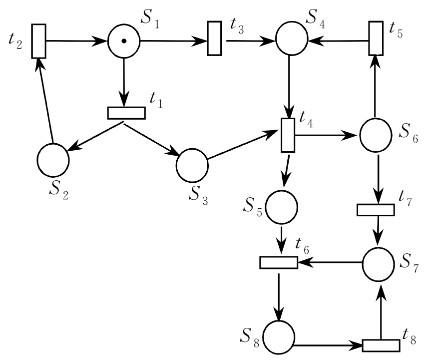


图 1　一个无界 Petri 网 Σ

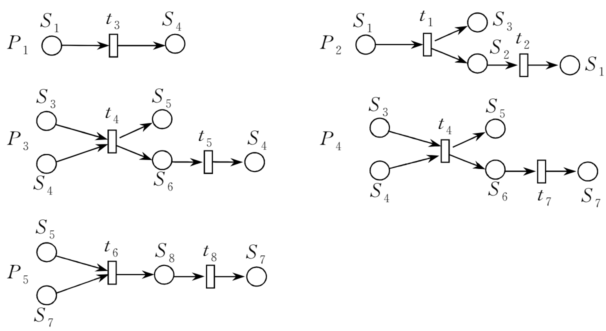


图 2　Σ 的5个基本进程段

3　无界 Petri 网的进程网系统

对于有界 Petri 网和无界公平 Petri 网而言，求得其基本进程段的集合，由文献[6,7]所给的方法便可构造出其进程表达式来刻画其进程行为. 对于任意无界 Petri 网，由于基本进程段之间衔接关系的复杂性，直接构造其进程表达式不像有界和无界公平 Petri 网那么简单[7]. 为此，文献[5]引入一个新的 Petri 网用来描述原网系统中各个基本进程段之间的衔接关系，将其定义为进程网系统.

为给出进程网系统的定义，首先给出几个概念和记号.

定义 9[5]. 　设$P=([u_1,u_2],\varphi)$为$\Sigma=(S,T;F,M_0)$的一个基本进程段，φ为出现网$N=(B,E;G)$到Σ的一个网映射，定义

${}^0P=\{s\,|\,s\in S\wedge\varphi^{-1}(s)\in u_1\}$为$P$的输入库所集；

$P^0=\{s|s\in S\wedge\varphi^{-1}(s)\in u_2\}$为 P 的输出库所集.

分别定义：

袋集 $B(^0P)=\{s|s\in S\wedge\varphi^{-1}(s)\in u_1\}$且$\forall s\in B(^0P)$，$\#(s,B(^0P))=|\{b|b\in u_1\wedge\varphi(b)=s\}|$(即 b 在 u 中出现的次数)为 P 的输入库所袋集；

袋集 $B(P^0)=\{s|s\in S\wedge\varphi^{-1}(s)\in u_2\}$且$\forall s\in B(P^0)$，$\#(s,B(P^0))=|\{b|b\in u_2\wedge\varphi(b)=s\}|$为 P 的输出库所袋集.

定义 10[5]. 设 $BP(\Sigma)$为 Σ 的基本进程段集，$S(P)\subseteq BP(\Sigma)$，定义

(1) $^0S(P)=\bigcup\limits_{\forall P\in S(P)}{^0P}$；

(2) $S(P)^0=\bigcup\limits_{\forall P\in S(P)}P^0$；

(3) $B(^0S(P))=\sum\limits_{\forall P\in S(P)}B(^0P)$①；

(4) $B(S(P)^0)=\sum\limits_{\forall P\in S(P)}B(P^0)$.

定义 11[5]. 设 φ 为出现网 $N=(B,E;G)$到 $\Sigma=(S,T;F,M_0)$的一个网映射，$BP(\Sigma)$为 Σ 的基本进程段集，定义 $\Sigma_P=(S_P,T_P;F_P,M_{0P})$为 Σ 进程 Petri 网. 其中，

(1) $S_P={^0BP(\Sigma)}\cup BP(\Sigma)^0$；

(2) 存在一一映射 $\xi:T_P\rightarrow BP(\Sigma)$：$\forall t\in T_P$，$\xi(t)\in BP(\Sigma)$；

(3) $F_P=\bigcup\limits_{\forall t\in T_P}(\{t\}\times B(\xi(t)^0)\cup B(^0\xi(t))\times\{t\})$；

(4) $M_{0P}=\Gamma_{S\rightarrow S_P}(M_0)$(其中 $\Gamma_{S\rightarrow S_P}$表示 M_0 在 S 到 S_P上的投影).

定义 11 实际上给出了由 Petri 网的基本进程段集求其进程网系统的方法. 简言之，进程网系统实际上就是将原 Petri 网的每个基本进程段浓缩为一个变迁，每个基本进程段的输入、输出库所集作为对应的变迁的输入、输出库所集，建立相应的流关系，将原网系统的初始标识向进程网系统的库所集上做投影.

图 1 的 Petri 网 Σ 对应的进程网系统 Σ_P 如图 3 所示，图中直接用基本进程段的名字 P_i 作为进程网系统 Σ_P 的变迁名.

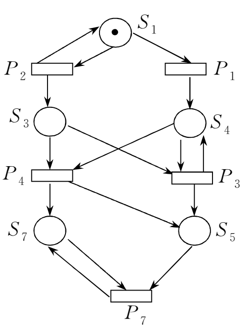


图 3 Σ 的进程网系统 Σ_P

4 无界 Petri 网的进程表达式

Petri 网的进程网系统能够很好地描述原网系统的进程行为[5]. 进程网系统的每一个变迁发生序列对应着原 Petri 网的一个满进程. 反之，如果原 Petri 网的一个满进程中最后的一段是完整的进程段，也有进程网系统的一个变迁发生序列与之对应. 就此而言，Petri 网的进程网系统也是描述进程行为的有力工具. 但是，如引言中所述，进程网系统给出的只是进程行为的 Petri 网模型，还不是完全形式化的刻画. 通过进一步研究证明，一个无界 Petri 网的进程同其进程网系统的语言之间存在着映射关系.

定义 12. 设 $\Sigma=(S,T;F,M_0)$为一个 Petri 网，称 $L(\Sigma)$为 Σ 的语言当且仅当 $L(\Sigma)=\{\sigma|(\sigma\in T^*)\wedge(M_0[\sigma\rangle M)\wedge(M\in M_f)\}$，其中 $M_f\subseteq R(M_0)$，$\forall M\in M_f$满足 $\exists P=([u_1,u_2],\varphi)\wedge P\in BP(\Sigma)$使得 $M=\varphi(u_2)$.

定义 13. 设 $\Sigma=(S,T;F,M_0)$为一个 Petri 网，$BP(\Sigma)$为 Σ 的基本进程段集，$Exp(P(\Sigma))$是以 $BP(\Sigma)$中的元素为字母表的一个表达式，该表达式所描述的集合为 $CRE(P(\Sigma))$. 如果 Σ 的每个满进程都是集合：$Pref\{Exp(P(\Sigma))\}=\bigcup\limits_{P\in CRE(P(\Sigma))}Pref(P)$的一个元素，则称 $Exp(P(\Sigma))$为 Σ 的进程表达式.

文献[6,7]分别给出了有界 Petri 网和无界公平 Petri 网的进程表达式的定义，定义 13 是对这两个定义的拓展.

下面引入一个记号. 一一映射 $\xi:A\rightarrow B$，任取 $\sigma\in A^*$，不妨设 $\sigma=\sigma_1\sigma_2\cdots\sigma_n(n\geq 0)$，定义 $\xi(\sigma)=\xi(\sigma_1)\xi(\sigma_2)\cdots\xi(\sigma_n)$，易知 $\xi(\sigma)\in B^*$.

定理 2. 设 $\Sigma_P=(S_P,T_P;F_P,M_{0P})$为 $\Sigma=(S,T;F,M_0)$的进程网系统，$BP(\Sigma)$为 Σ 的基本进程段集，一一映射 $\xi:T_P\rightarrow BP(\Sigma)$，记 $L(\Sigma_P)$为 Σ_P所确定的语言，$Exp(P(\Sigma))$为 Σ 的进程表达式，则 $Exp(P(\Sigma))=\xi(L(\Sigma_P))$.

证明. (1)$Exp(P(\Sigma))\subseteq\xi(L(\Sigma_P))$.

$\forall\sigma\in BP(\Sigma)^*$且 $\sigma\in Exp(P(\Sigma))$，由 $Exp(P(\Sigma))$的定义知 $M_0[\sigma\rangle$；不妨设 $\sigma=\sigma_1\sigma_2\cdots\sigma_n$，则 $M_0[\sigma_1\sigma_2\cdots\sigma_n\rangle$.

设 $M_0[\sigma_1\rangle M_1[\sigma_2\rangle M_2\cdots M_i[\sigma_{i+1}\rangle M_{i+1}\cdots M_n[\sigma_n\rangle M_{n+1}$，由进程网系统的定义知

① 根据文献[2]，两个袋 A,B 的和 $A+B$，$\#(x,A+B)=\#(x,A)+\#(x,B)$.

$\Gamma_{S\to S_P}(M_0)[\xi^{-1}(\sigma_1)\rangle\Gamma_{S\to S_P}(M_1)[\xi^{-1}(\sigma_2)\rangle\Gamma_{S\to S_P}(M_2)\cdots$
$\Gamma_{S\to S_P}(M_i)[\xi^{-1}(\sigma_{i+1})\rangle\Gamma_{S\to S_P}(M_{i+1})\cdots$
$\Gamma_{S\to S_P}(M_n)[\xi^{-1}(\sigma_n)\rangle\Gamma_{S\to S_P}(M_{n+1})$.

由 $M_{0P}=\Gamma_{S\to S_P}(M_0)$知

$M_{0P}[\xi^{-1}(\sigma_1)\rangle M_{1P}[\xi^{-1}(\sigma_2)\rangle M_{2P}\cdots M_{iP}[\xi^{-1}(\sigma_{i+1})\rangle M_{(i+1)P}\cdots M_{nP}[\xi^{-1}(\sigma_n)\rangle M_{(n+1)P}$，其中 $M_{iP}=\Gamma_{S\to S_P}(M_i)$，

即

$$M_{0P}[\xi^{-1}(\sigma_1)\xi^{-1}(\sigma_2)\cdots\xi^{-1}(\sigma_{i+1})\cdots\xi^{-1}(\sigma_n)\rangle,$$

从而

$$\xi^{-1}(\sigma_1)\xi^{-1}(\sigma_2)\cdots\xi^{-1}(\sigma_{i+1})\cdots\xi^{-1}(\sigma_n)\in L(\Sigma_P).$$

由于 $\xi(\xi^{-1}(\sigma_1)\xi^{-1}(\sigma_2)\cdots\xi^{-1}(\sigma_{i+1})\cdots\xi^{-1}(\sigma_n))=\sigma_1\sigma_2\cdots\sigma_n=\sigma$，所以 $\sigma\in\xi(L(\Sigma_P))$.

(2)$\xi(L(\Sigma_P))\subseteq Exp(P(\Sigma))$.

$\forall\sigma\in T_P{}^*$ 且 $\sigma\in L(\Sigma_P)$，则 $M_{0P}[\sigma\rangle$，不妨设 $\sigma=\sigma_1\sigma_2\cdots\sigma_n$，有 $M_{0P}[\sigma_1\sigma_2\cdots\sigma_n\rangle$. 设

$$M_{0P}[\sigma_1\rangle M_{1P}[\sigma_2\rangle M_{2P}\cdots M_{iP}[\sigma_{(i+1)}\rangle M_{(i+1)P}\cdots M_{nP}[\sigma_n\rangle M_{(n+1)P} \quad (*)$$

定义$|S|$维向量 $\boldsymbol{M}'_0$：

$$\boldsymbol{M}'_0(s)=\begin{cases}M_{0P}(s), & s\in S_P\\ 0, & s\notin S_P\end{cases},$$

记作 $\boldsymbol{M}'_0=\Gamma_{S_P\to S}M_{0P}$，易知 $\boldsymbol{M}'_0=\boldsymbol{M}_0$.

由式(*)及进程网系统 Σ_P 的定义知

$\boldsymbol{M}'_0[\xi(\sigma_1)\rangle\boldsymbol{M}'_{1P}[\xi(\sigma_2)\rangle\boldsymbol{M}'_{2P}\cdots\boldsymbol{M}'_{iP}[\xi(\sigma_{(i+1)})\rangle\boldsymbol{M}'_{(i+1)P}\cdots\boldsymbol{M}'_{nP}[\xi(\sigma_n)\rangle\boldsymbol{M}'_{(n+1)P}$(其中 $\boldsymbol{M}'_{iP}=\Gamma_{S_P\to S}M_{0P}$，$i=1,2,\cdots,n+1$).

由 $\boldsymbol{M}'_0=\boldsymbol{M}_0$，知 $\boldsymbol{M}'_0[\xi(\sigma_1)\xi(\sigma_2)\cdots\xi(\sigma_i)\cdots\xi(\sigma_n)\rangle$.

由进程网系统 Σ_P 的定义知：$\xi(\sigma_i)\in BP(\Sigma)$，$\xi(\sigma_1)\xi(\sigma_2)\cdots\xi(\sigma_n)$为 Σ 的一个满进程，故 $\xi(\sigma_1)\xi(\sigma_2)\cdots\xi(\sigma_n)\in Exp(P(\Sigma))$.

又 $\xi(\sigma_1)\xi(\sigma_2)\cdots\xi(\sigma_n)=\xi(\sigma_1\sigma_2\cdots\sigma_n)=\sigma$，所以 $\xi(\sigma)\in Exp(P(\Sigma))$.

由(1)和(2)知，$Exp(P(\Sigma))=\xi(L(\Sigma_P))$. 证毕.

由定理 2 知，无界 Petri 网的进程同其进程网系统的语言之间存在着一一映射关系，这就将求取无界 Petri 网的进程表达式的问题转化成了求取其进程网系统的语言问题. 但是，由文献[5]知 Petri 网的进程网系统的结构一般是比较复杂的，直接写出其语言可能比较困难.

5 无界 Petri 网的进程表达式的求取方法

本节解决结构复杂的进程网系统的语言求取问题以便给出求取无界 Petri 网的进程表达式的算法. 首先定义库所的指标函数，将一个结构复杂的网系统分解成结构简单且语言可求的子网，引入语言的同步交运算，通过子网的语言来求取原复杂系统的语言，以达到解决求取无界 Petri 网的进程表达式的目的.

首先介绍基于库所指标的 Petri 网分解方法.

定义 14. 设 $\Sigma=(S,T;F,M_0)$为一个 Petri 网，函数 $f:S\to\{1,2,\cdots,k\}$满足 $\forall s_1,s_2\in S$，$\exists t\in T$，$\{s_1,s_2\}\subseteq{}^{\bullet}t$ 或$\{s_1,s_2\}\subseteq t^{\bullet}$，则 $f(s_1)\neq f(s_2)$，称 f 为 Σ 的库所指标函数，$f(s)$为库所 s 的指标.

定义 15. 设 $\Sigma=(S,T;F,M_0)$为一个 Petri 网，函数 $f:S\to\{1,2,\cdots,k\}$为 Σ 的库所指标函数，称 $\Sigma_i=(S_i,T_i;F_i,M_{0i})(i\in\{1,2,\cdots,k\})$为 Σ 基于库所指标的分解网. Σ_i满足：

(1) $S_i=\{s\in S|f(s)=i\}$；

(2) $T_i=\{t\in T|\exists s\in S_i,t\in{}^{\bullet}s\cup s^{\bullet}\}$；

(3) $F_i=\{(S_i\times T_i)\cup(T_i\times S_i)\}\cap F$；

(4) $M_{0i}=\Gamma_{S\to S_i}M_0$($\Gamma_{S\to S_i}M_0$表示 M_0 在 S_i上的投影).

图 4 给出了图 3 的 Petri 网 Σ_P 的基于库所指标的分解网. 定义库所指标函数 f 满足：$f(s_1)=f(s_4)=f(s_7)=1$；$f(s_3)=f(s_5)=2$. 显然，f 满足定义 15 的条件. 按 f 分解 Σ_P 得到两个子网 Σ_{P1} 和 Σ_{P2} 分别如图 4 所示.

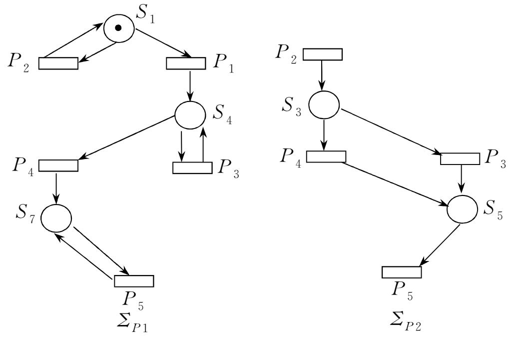


图 4 Σ_P 的两个库所指标分解子网

Petri 网的基于库所指标的分解并不是唯一的. 由于 f 选择的不同，可能出现不同的库所指标集，但无论怎样选择 f，同一个变迁的两个(或多个)不同输入(输出)位置分别分解到不同的子网中. 同时，对于 k 的取值也可能有多个，为了问题分析的方便，我们通常取满足条件的最小 k 值.

定理 3. 设 $\Sigma_i=(S_i,T_i;F_i,M_{0i})(i\in\{1,2,\cdots,k\})$为 $\Sigma=(S,T;F,M_0)$的基于库所指标的分解网，则有 $\forall\Sigma_i(i\in\{1,2,\cdots,k\})$，$\forall t\in T_i$，$|{}^{\bullet}t|\leq1$ 且$|t^{\bullet}|\leq1$.

证明. 根据定义 15 容易证明.

定理 3 说明分解后的每个子网都具有很好的结构性质：$\forall t\in T_i, |{}^{\cdot}t|\leq 1$ 且 $|t^{\cdot}|\leq 1$. 文献[10] 分析了这些网子类的语言性质，并给出了其语言的求取方法，这些网子类的语言表达式均可由正规表达式或正规表达式的 α-闭包①通过“+”,“·”和“||”(并发) 等几种运算得到. 由此看出，通过基于库所指标的 Petri 网分解方法得到的子网结构简单且语言可求.

下面分析如何通过这些结构简单的子网来求解原先结构复杂的网系统的语言. 为了便于语言的描述，我们引入语言 “同步交”运算的概念[13].

定义 16. 设 $L(\Sigma_i)$分别为 Petri 网 $\Sigma_i=(S_i,T_i;F_i,M_{0i})(i\in\{1,2\})$的语言，称 L 为 $L(\Sigma_1)$与 $L(\Sigma_2)$的同步交语言当且仅当：$T=T_1\cup T_2$, $\forall\sigma\in T^*\wedge\sigma\in L$, $\sigma_i=(\Gamma_{T\to T_i}\sigma)\in L(\Sigma_i)$($\Gamma_{T\to T_i}\sigma$ 表示 σ 在 T 到 T_i 上的投影，$i\in\{1,2\}$).

记作 $L=L(\Sigma_1)[]L(\Sigma_2)$或 $L=\overset{2}{\underset{i=1}{[]}}L(\Sigma_i)$.

同样可以定义 $k(k\geq 3)$个语言的同步交语言，记 k 个语言 $L(\Sigma_i)(i\in\{1,2,\cdots,k\})$的同步交语言 $L(\Sigma)$为

$$L(\Sigma)=(\overset{k-1}{\underset{i=1}{[]}}L(\Sigma_i))[]L(\Sigma_k)=\overset{k}{\underset{i=1}{[]}}L(\Sigma_i).$$

定理 4. 设 $L(\Sigma_i)$分别为 Petri 网 $\Sigma_i=(S_i,T_i;F_i,M_{0i})(i\in\{1,2\})$的语言，[]为语言的同步交运算，若 $T_1=T_2$，则 $L(\Sigma_1)[]L(\Sigma_2)=L(\Sigma_1)\cap L(\Sigma_2)$.

证明. 先证明 $L(\Sigma_1)[]L(\Sigma_2)\subseteq L(\Sigma_1)\cap L(\Sigma_2)$.

取 $\sigma\in L(\Sigma_1)[]L(\Sigma_2)$，因为 $T_1=T_2$，令 $T=T_1\cup T_2$，则 $T=T_1=T_2$，所以 $\sigma_i=(\Gamma_{T\to T_i}\sigma)=\sigma$，从而 $\sigma_i\in L(\Sigma_i)$，故 $\sigma\in L(\Sigma_1)\cap L(\Sigma_2)$. 证毕.

同理可证 $L(\Sigma_1)\cap L(\Sigma_2)\subseteq L(\Sigma_1)[]L(\Sigma_2)$.

由定理 4 可以看出，Petri 网语言的同步交运算不同于语言的交运算. 语言的交运算只是同步交运算的一种特殊情况(当 $T_1=T_2$时).

定理 5. 设 $\Sigma_i=(S_i,T_i;F_i,M_{0i})(i\in\{1,2,\cdots,k\})$为 $\Sigma=(S,T;F,M_0)$的基于库所指标的分解网，则 $L(\Sigma)=\overset{k}{\underset{i=1}{[]}}L(\Sigma_i)$.

证明. 设 $L(\Sigma)=\{\sigma\mid(\sigma\in T^*)\wedge(M_0[\sigma>M)\wedge(M\in M_f)\}$，其中 $M_f\subseteq R(M_0)$，$\forall M\in M_f$ 满足 $\exists P=([u_1,u_2],\varphi)\wedge P\in BP(\Sigma)$使得 $M=\varphi(u_2)$.

$M_{fi}=\{\Gamma_{S\to S_i}M\mid M\in M_f\}$，对$|\sigma|$做归纳证明该定理成立.

(1) 当 $|\sigma|=1$ 时，$\sigma_i=\Gamma_{T\to T_i}\sigma=\begin{cases}\sigma, & \sigma\in T_i\\ \varepsilon, & \sigma\notin T_i\end{cases}$ $(i\in\{1,2,\cdots,k\})$；由 $\sigma\in L(\Sigma)$当且仅当 $M_0[\sigma\rangle M_1\wedge M_1\in M_f$，当且仅当

$\Gamma_{S\to S_i}(M_0)[\sigma_i\rangle\Gamma_{S\to S_i}(M_1)\wedge\Gamma_{S\to S_i}(M_1)\in M_{fi}$，即 $\sigma_i\in L(\Sigma_i)(i\in\{1,2,\cdots,k\})$.

(2) 设当$|\sigma|=x$ 时成立，下面证明$|\sigma|=x+1$时也成立.

令 $\sigma=\sigma'\cdot t'$，其中$|\sigma'|=x$，则 t'为 σ 的第 $x+1$ 个元素. 由 $\sigma\in L(\Sigma)$当且仅当 $M_0[\sigma'\cdot t'\rangle M_{x+1}\wedge M_{x+1}\in M_f$，记 $M_0[\sigma'\rangle M_x[t'\rangle M_{x+1}$，则 $M_x\in R(M_0)$.

令 $M'_f=\{M_x\mid M_0[\sigma'\rangle M_x[t'\rangle M_{x+1}\wedge M_{x+1}\in M_f\}$；

$M'_{fi}=\{\Gamma_{P\to P_i}M_x\mid M_x\in M'_f\}(i\in\{1,2,\cdots,k\})$；由归纳假设，$\exists\sigma'_i=\Gamma_{T\to T_i}(\sigma')$，$\sigma'_i\in L(\Sigma_i)$使得

$$\sigma_i=\Gamma_{T\to T_i}\sigma=\begin{cases}\sigma'_i\cdot t', & t'\in T_i\\ \sigma'_i, & t'\notin T_i\end{cases},$$

$\sigma_i\in L(\Sigma_i)$当且仅当 $M_{0i}[\sigma'_i\rangle M'_i\wedge M'_i\in M'_{fi}$，即当且仅当

$$M_{0i}[\sigma_i\rangle M_{(x+1)i}\wedge M_{(x+1)i}\in M_{fi},$$

即

$$\sigma_i\in L(\Sigma_i),\sigma_i=\Gamma_{T\to T_i}(\sigma)(i\in\{1,2,\cdots,k\}).$$

故归纳得证. 证毕.

由定理 5 知，只要求出每个子网的语言 $L(\Sigma_i)$ $(i\in\{1,2,\cdots,k\})$，$\overset{k}{\underset{i=1}{[]}}L(\Sigma_i)$即为原 Petri 网的语言. 文献[10]详细地分析了这些子网类型的语言特点及其语言求解方法. 这样，结构复杂的 Petri 网的语言求取方法就可由前面的讨论给出. 下面是具体算法.

算法 1. 求进程网系统的语言表达式.

输入：$\Sigma=(S,T;F,M_0)$

输出：Petri 网 Σ 的语言表达式 $L(\Sigma)$

1. 建立库所指标函数 $f:S\to\{1,2,\cdots,k\}$；

2. 按库所指标函数 f 分解 Σ 成 $\Sigma_i=(S_i,T_i;F_i,M_{0i})$ $(i\in\{1,2,\cdots,k\})$；

3. For $i=1$ to k Do

(1) 由文献[10]的方法写出 Σ_i的语言表达式 $L(\Sigma_i)$；

(2) $i\leftarrow i+1$；

4. $L(\Sigma)\leftarrow\overset{k}{\underset{i=1}{[]}}L(\Sigma_i)$，结束.

推论 2. 设 $\Sigma=(S,T;F,M_0)$为一个 Petri 网，$BP(\Sigma)$为基本进程段集，$\Sigma_P=(S_P,T_P;F_P,M_{0P})$为 Σ 的进程网系统，若

(1) 存在一一映射 $\xi:T_P\to BP(\Sigma)$；

① 文献[11]中定义了语言 L 的 α-闭包运算. 设 S 为语言 L 的 α-闭包，当且仅当 $S=\bigcup_{i=0,1,\cdots}L^{(i)}$，记为 $S=L^{\alpha}$，其中 $L^{(2)}=L||L$；$L^{(n)}=L||L^{(n-1)}(n\geq 2)$. 如 $L=\{a\cdot b\cdot c\}$，则

$$L^{\alpha}=\{w\mid\forall s\in Pre(w),\#(a,s)\geq\#(b,s)\geq\#(c,s)\wedge\#(a,w)=\#(b,w)=\#(c,w)\}$$

(其中，$Pre(w)$表示 w 的前缀，$\#(a,w)$表示 a 在 w 中出现的次数).

(2) 设 $\Sigma_{Pi}=(S_{Pi},T_{Pi};F_{Pi},M_{0Pi})(i\in\{1,2,\cdots,k\})$ 为 Σ_P 的基于库所指标的分解网，

则 Σ 的进程表达式 $Exp(P(\Sigma))=\xi(L(\Sigma_P))=\xi(\mathop{[\,]}\limits_{i=1}^{k} L(\Sigma_{Pi}))$.

证明. 由定理 2 和定理 5 可以证明该推论成立.

定理 2 将求取无界 Petri 网的进程表达式的问题转化为求取其进程网系统的语言表达式问题. 定理 5 给出了一种无界 Petri 网的进程网系统的语言求取方法. 推论 2 给出了求取无界 Petri 网的进程表达式的过程思想. 下面给出求取无界 Petri 网的进程表达式的具体算法.

算法 2.　求无界 Petri 网的进程表达式.

输入：无界 Petri 网 $\Sigma=(S,T;F,M_0)$

输出：Σ 的进程表达式 $Exp(\Sigma)$

1. 求取 Σ 的基本进程段集（具体方法见文献 [9]）；
2. 由定义 11 求出 Σ 的进程网系统 Σ_P；
3. 定义 Σ_P 的库所指标函数 f，将 Σ_P 分解成 $\Sigma_{Pi}=(S_i,T_i;F_i,M_{0i})(i\in\{1,2,\cdots,k\})$；
4. 由算法 1 求取 Σ_P 的语言表达式 $L(\Sigma_P)$；
5. $Exp(\Sigma)\leftarrow\xi(L(\Sigma_P))$（其中 $\xi:T_P\rightarrow BP(\Sigma)$ 的一一映射）；
6. 结束.

算法的正确性可由前面的分析保证.

例.　下面给出图 1 所给的无界 Petri 网 Σ 的进程表达式的求取过程. 首先由文献[9]的算法求得 Σ 的 5 个基本进程段如图 2 所示. 由定义 11，求得 Σ 的进程网系统 Σ_P 如图 3 所示. 为了求取 Σ 的进程表达式，我们先求取进程网系统 Σ_P 的语言. 根据第 4 节内容，按库所指标函数将 Σ_P 分解成两个子网 Σ_{P1} 和 Σ_{P2} 如图 4 所示. 由文献[10]的方法可写出 Σ_{P1} 和 Σ_{P2} 的语言：

$$L(\Sigma_{P1})=P_2{}^*P_1P_3{}^*P_4P_5{}^*;$$
$$L(\Sigma_{P2})=(P_1(P_3+P_4)P_5)^{\alpha}.$$

从而，$L(\Sigma_P)=L(\Sigma_{P1})[\,]L(\Sigma_{P2})=(P_2{}^*P_1P_3{}^*P_4P_5{}^*)[\,](P_2(P_3+P_4)P_5)^{\alpha}$. 故，$\Sigma$ 的进程表达式为 $Exp(\Sigma)=(P_2{}^*P_1P_3{}^*P_4P_5{}^*)[\,](P_2(P_3+P_4)P_5)^{\alpha}$.

可以验证，Σ 的任意满进程都是 $Exp(\Sigma)$ 所表示的语言的一个前缀.

6　结束语

本文解决了无界 Petri 网的进程表达式的求取问题. 主要创新之处有两点：

(1) 给出了与文献[6,7]求解进程表达式完全不同的方法. 借助进程网系统的概念，通过证明无界 Petri 网的进程同其进程网系统的语言之间的关系，将求取无界 Petri 网进程表达式的问题转化成进程网系统的语言求取问题.

(2) 通过定义库所的指标函数将复杂的网系统分解成结构简单的子系统，引入语言"同步交"运算的概念，给出了求取结构复杂 Petri 网语言表达式的一种方法，最终给出了求取无界 Petri 网的进程表达式的可行算法.

综合本文以及先前的工作[5~9]，可以得到表 1 所示的 Petri 网结构与进程表达式之间的关系.

表 1　Petri 网结构与进程表达式之间的关系

Petri 网分类	进程表达式的类型
有界 Petri 网	正规表达式
无界公平 Petri 网	正规表达式
任意无界 Petri 网	含有"[]"的表达式

下面的问题有待进一步研究：

(1)含有"同步交"运算的语言表达式同传统的语言表达式[12]之间表达能力强弱的比较；

(2)基于 Petri 网进程表达式的 Petri 网性质分析.

参　考　文　献

1　Gltz U, Reisig W. Processes of Place/Transition Net. LNCS 154, NewYork: Springer-Verlag, 1983. 264～277

2　Wolfgang Reisig. Petri Nets——An Introduction. Berlin: Springer Verlag, 1985

3　Lu Ru-Qian. P/R nets and P/R processes(Ⅰ). Science in China (Series E), 1992, 35(1): 21～31

4　Lu Ru-Qian. P/R nets and P/R processes(Ⅱ). Science in China (Series E), 1992, 35(1): 148～157

5　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. 计算机学报，2002，25(12): 1308～1315)

6　Wu Zhe-Hui. Process expression of bounded Petri net. Science in China (Series E), 1996, 39(1): 37～49

7　Wu Zhe-Hui, Wang Pei-Liang, Zhao Mao-Xian. Process expression of unbounded fair net. Chinese Journal of Computers, 2000, 23(4): 337～344 (in Chinese)
(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)

8　Zeng Qing-Tian, Wu Zhe-Hui, Ma Bing-Xian. The process and language expression of Petri net. Mini-Micro Systems, to appear (in Chinese)
(曾庆田，吴哲辉，马炳先. Petri 网的进程表达式与语言表达式. 小型微型计算机系统，待发表)

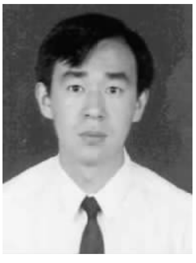
9 Zeng Qing-Tian, Wu Zhe-Hui. An algorithm for establishing the set of basic process sections for a Petri net. Computer Science, 2001, 28(Supplement): 5～11 (in Chinese)
(曾庆田，吴哲辉. Petri网基本进程段的一个求解算法. 计算机科学，2001,28(增刊): 5～11)

10 Zeng Qing-Tian, Wu Zhe-Hui. The language characters analysis of analogous S-graph. Computer Science, 2002, 29(5): 120～122 (in Chinese)
(曾庆田，吴哲辉. 类S-图的语言性质分析. 计算机科学，2002,29(5): 120～122)

11 Garg V K, Ragunath M T. Concurrent regular expressions and their relationship to Petri nets. Theoretical Computer Science, 1992, 96(2): 258～304

12 Hopcroft J, Ullman J. Introduction to Automata Theory Languages and Computation. Reading, MA: Addison-Wesley, 1979

13 Zeng Qing-Tian, Wu Zhe-Hui. Synchronous intersection operation of Petri net languages. Mini-Micro Systems, to appear (in Chinese)
(曾庆田，吴哲辉. Petri网语言的同步交运算. 小型微型计算机系统，待发表)



ZENG Qing-Tian, born in 1976, Ph. D. candidate. His research interests include large-scale knowledge processing and ontology analysis, V&V of knowledge-based system, Petri net theory and applications, and mathematical knowledge acquisition and analysis.

WU Zhe-Hui, born in 1941, professor, Ph. D. supervisor. His research interests include Petri net, algorithm design and analysis, and formal language.

第四届中国 Rough 集与软计算学术研讨会(CRSSC 2004)

征 文 通 知

由中国人工智能学会粗糙集专业委员会和中国计算机学会人工智能与模式识别专业委员会主办，浙江海洋学院承办，同济大学协办的第4届中国Rough集与软计算学术研讨会(CRSSC 2004)拟定于2004年10月中下旬在浙江舟山召开. 欢迎各界人士踊跃投稿.

Rough集理论经过国际众多研究人员的共同努力，其理论模型得到不断的完善和发展，并逐渐渗透到各个应用领域，而且形成了与其它软计算理论共同发展和优势互补的局面. 当前许多国内外学术会议和研讨班都把Rough集理论列入其研讨和交流的主要内容. 国内外学者也公认Rough集理论是研究数据挖掘、知识约简和粒计算的理论基础. Rough集理论与诸如Fuzzy集、粒计算、神经网络、遗传算法等其它一些软计算理论已经成为当前国内外计算机及相关专业的研究热点. 我国近年来在此领域的研究发展速度很快，研究理论逐年在深化，应用越来越广泛，研究队伍正在不断壮大. 现将有关征文事宜通知如下：

征文范围

Rough集理论及应用	计算智能	机器学习	文字计算	Fuzzy集理论及应用
粒度计算	软计算及其应用	演化计算	Petri网	软计算的逻辑基础
非经典逻辑	神经网络	软计算复杂性	空间推理	统计与概率推理
智能Agent	多准则决策分析	决策支持系统	知识发现与数据挖掘	多Agent技术
网络智能	集成智能系统	近似推理与不确定性推理	数据仓库	模式识别与图像处理
生物信息与生物计算	其他有关领域			

征文要求

(1)须未公开发表过，一般不超过6000字；
(2)论文包括题目、作者姓名、单位、地址、邮编、Email地址、联系电话，中英文摘要(一般不超过200字)、关键词、正文和参考文献；
(3)论文一律为A4打印稿，一式两份，用Word排版，欢迎用电子版投稿(Email递交)；
(4)会议论文集将由《计算机科学》专辑出版；
(5)征文请寄：浙江省舟山市浙江海洋学院信息学院 吴伟志 收(邮政编码：316004)
电子版投稿请送：crssc@zjou.net.cn
有关会议信息请登录浙江海洋学院主页：http://www.zjou.net.cn/

重要日期

截稿日期：2004年4月15日
录用日期：2004年5月30日
清样付印日期：2004年6月30日(收到日期)

联系方式

联系人：吴伟志；杨晓平
联系电话：0580-8180230；0580-8180228；13059890698(手机)
电子信箱：wuwz@zjou.net.cn