基于库所指标的 Petri 网分解方法*)

A Decomposition Method of Petri Net Based on the Index of Places

曾庆田　吴哲辉

（山东科技大学信息学院　泰安271019）

Abstract　A new decomposition method of Petri net by assigning an index for the place set is given in this paper. By this decomposition method, the structure of the sub net is simple in which the element number of input place set or the output place set of arbitrary transition is less than or equal to one. And, there is a projection relation of the reachable marking set and the language between the old net system and the sub net system. The properties of the large scale systems could be analyzed by decomposing them to the sub net system because the state and behavior of the old system could be got by synchronous composing of the sub systems.

Keywords　Petri net, Index of places, Decompose, Reachable marking set, Language

一、引言

Petri 网是系统模拟和分析的有效工具。对于一个规模较大的网系统，由于变迁和库所数目的增多，其分析显得比较麻烦。针对这个问题，国内外的许多学者作了大量的工作，他们提出了网化简、网运算，定义了化简子网、逐次化简以及针对特殊子网、特殊结构的化简方法[1~5]。文[6]和文[7]分别给出了网系统的“和分解”和“并分解”方法，并讨论了通过这两种分解方法得到的子网同原网在结构性质方面的对应关系。本文通过指定库所集的指标函数，给出一种基于库所指标集的 Petri 网分解方法，通过分解得到结构简单的子网系统：$|{}^{\bullet}t|\leqslant 1$且$|t^{\bullet}|\leqslant 1$，原系统的状态和语言与子系统之间存在着一种投影关系，通过子网间的同步合成可得到原系统的状态和行为，对用 Petri 网分析大规模系统提供了一种有效的方法。

由于篇幅所限，本文没有介绍 Petri 网的基本概念和术语，读者可参见文[3,4]。

二、基于库所指标的 Petri 网分解

定义2.1　设 $\Sigma=(P,T;F,M_0)$为一 Petri 网，函数 $f:P\to\{1,2,\cdots,k\}$满足：$\forall p_1,p_2\in P$，$\exists t\in T$，使得$\{p_1,p_2\}\subseteq{}^{\bullet}t$或$\{p_1,p_2\}\subseteq t^{\bullet}$，则 $f(p_1)\neq f(p_2)$，称 f 为 Σ 的库所指标函数，$f(p)$为库所 p 的指标。

定义2.2　设 $\Sigma=(P,T;F,M_0)$为一 Petri 网，函数 $f:P\to\{1,2,\cdots,k\}$为 Σ 的库所指标函数，称 $\Sigma_i=(P_i,T_i;F_i,M_{0i})$ $(i\in\{1,2,\cdots,k\})$为 Σ 基于库所指标的分解网。Σ_i 满足：

(1)$P_i=\{p\in P\mid f(p)=i\}$，$(i\in\{1,2,\cdots,k\})$；

(2)$T_i=\{t\in T\mid \exists p\in P_i, t\in{}^{\bullet}p\cup p^{\bullet}\}$，$(i\in\{1,2,\cdots,k\})$；

(3)$F_i=\{(P_i\times T_i)\cup(T_i\times P_i)\}\cap F$，$(i\in\{1,2,\cdots,k\})$；

(4)$M_{0i}=\Gamma_{P\to P_i}M_0$，($\Gamma_{P\to P_i}M_0$表示 M_0在 P_i 上的投影，$i\in\{1,2,\cdots,k\}$)。

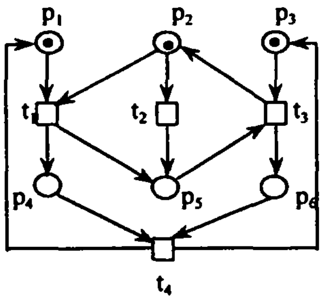
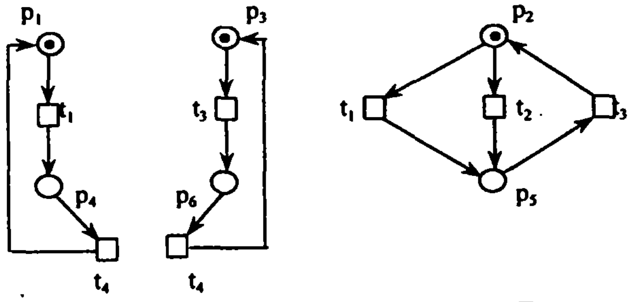


图1　Petri 网 Σ



(1) Σ_1　(2) Σ_2　(3) Σ_3

图2　Σ 的三个子网 Σ_1、Σ_2和 Σ_3

例2.1　图1给出了一个 Petri 网系统 $\Sigma=(P,T;F,M_0,P_f)$，取库所指标函数 $f:P\to\{1,2,\cdots,k\}$满足：

$f(p_1)=f(p_4)=1$；$f(p_3)=f(p_6)=2$；$f(p_2)=f(p_5)=3$；

显然 f 满足定义2.2的条件。按 f 分解 Σ 得到3个子网 Σ_1、Σ_2和 Σ_3分别如图2所示。

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)库所分别分到不同的子网中。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2.1　设 $\Sigma_i=(P_i,T_i;F_i,M_{0i})$ $(i\in\{1,2,\cdots,k\})$为 $\Sigma=(P,T;F,M_0)$的基于库所指标的分解网，则有：

1)$\forall i,j\in\{1,2,\cdots,k\}$，$i\neq j$，有 $P_i\cap P_j=\Phi$，$\bigcup_{i=1}^{k}P_i=P$。

*)国家自然科学基金(69873029)资助课题。曾庆田　硕士，主要研究方向为 Petri 网理论与应用。吴哲辉　教授，博士生导师，主要研究内容：Petri 网、算法设计与分析和形式语言等。

2)若 $k>1$,$\forall i\in\{1,2,\cdots,k\}$,$\exists j\in\{1,2,\cdots,k\}$,$i\neq j$,有 $T_i\cap T_j\neq\Phi$,$\bigcup_{i=1}^{k}T_i=T$。

证明:1)显然。

2)$\forall i\in 1,2,\cdots,k\}$,由于 $k>1$,故存在 $j\in\{1,2,\cdots,k\}$ 满足 $i\neq j$ 且至少存在 $p_i,p_j\in P$ 满足 $f(p_i)\neq f(p_j)$。

不妨设 $p_i\in P_i$,$p_j\in P_j$,由 f 的定义知:$\exists t\in T$ 使得:$(\{p_i,p_j\}\subseteq{}^{\bullet}t\vee\{p_i,p_j\}\subseteq t^{\bullet})$,所以 $t\in T_i\wedge t\in T_j$,即 $t\in T_i\cap T_j$,故 $T_i\cap T_j\neq\Phi$。

由定义2.2易知:$\bigcup_{i=1}^{k}T_i=T$。 □

定理2.2 设 $\Sigma_i=(P_i,T_i;F_i,M_{0i})(i\in\{1,2,\cdots,k\})$ 为 $\Sigma=(P,T;F,M_0)$ 的基于库所指标的分解网,则有:$\forall\Sigma_i(i\in\{1,2,\cdots,k\})$,$\forall t\in T_i$,$|{}^{\bullet}t|\leqslant1$ 且 $|t^{\bullet}|\leqslant1$。

证明:由定义2.2易证。 □

三、主要结果

定义3.1[7] 设 Petri 网 $\Sigma_i=(P_i,T_i;F_i,M_{0i})(i\in\{1,2\})$,令 $\Sigma=(P,T;F,M_0)$ 使得:

(1)$P=P_1\cup P_2$,$P_1\cap P_2=\Phi$;

(2)$T=T_1\cup T_2$,$T_1\cap T_2\neq\Phi$;

(3)$F=F_1\cup F_2$;

(4)$M_0(p)=\begin{cases}M_{01}(P) & p\in P_1\\ M_{02}(P) & p\in P_2\end{cases}$;

则称 Σ 为 Σ_1 与 Σ_2 的同步合成网,记作:$\Sigma=\Sigma_1 O_T\Sigma_2$ 或者 $\Sigma=\overset{2}{\underset{i=1}{O}}{}_T\Sigma_i$。

类似地可以定义多个网系统的同步合成。三个网系统的同步合成定义为 $\Sigma=(\Sigma_1O_T\Sigma_2)O_T\Sigma_3$,$n(n\geqslant3)$ 个网系统的同步合成记为:$\Sigma=\overset{n}{\underset{i=1}{O}}{}_T\Sigma_i=(\overset{n-1}{\underset{i=1}{O}}{}_T\Sigma_i)O_T\Sigma_n$。

定理3.1 设 $\Sigma_i=(P_i,T_i;F_i,M_{0i})(i\in\{1,2,\cdots,k\})$ 为 $\Sigma=(P,T;F,M_0)$ 的基于库所指标的分解网,则有:$\Sigma=\overset{k}{\underset{i=1}{O}}{}_T\Sigma_i$。

证明:根据定理2.1和定义2.2、定义3.1容易证明。 □

由定理2.3知,分解后的每个子网都具有很好的结构:$\forall t\in T_i$,$|{}^{\bullet}t|\leqslant1$ 且 $|t^{\bullet}|\leqslant1$。对每一个 $\Sigma_i=(P_i,T_i;F_i,M_{0i},S_{fi})$,我们可以比较容易和直接地分析其状态和行为特性,这也是我们将一个复杂的网系统进行分解的原因。由定理3.1知,所有的子网经同步合成又可得到原先的网系统。文[7]分析了系统同步合成过程中满足的动态不变性,由此我们就可以通过分析子网的特性来研究原系统的性质。下面我们主要分析分解过程中,原网系统与其子网系统在状态、语言方面满足的关系。

定理3.2 设 $\Sigma_i=(P_i,T_i;F_i,M_{0i})(i\in\{1,2,\cdots,k\})$ 为 $\Sigma=(P,T;F,M_0)$ 的基于库所指标的分解网,则有:$\forall M\in R(M_0)$,$\Gamma_{P\to P_i}(M)\in R(M_{0i})(i\in\{1,2,\cdots,k\})$。($\Gamma_{P\to P_i}(M)$ 表示 M 在 P_i 上的投影)。

证明:对 $\forall M\in R(M_0)$,记 $M_0[\sigma>M$,下面对 $|\sigma|$ 做归纳证明。

(1)当 $|\sigma|=0$ 时,由定义2.2(4)知:

$\Gamma_{P\to P_i}(M_0)=M_{0i}\in R(M_{0i})(i\in\{1,2,\cdots,k\})$

结论成立。

(2)设当 $|\sigma|=n$ 时结论成立,下面证明当 $|\sigma|=n+1$ 时结论也成立。

令 $\sigma=\sigma't$,则 $|\sigma'|=n$,记 $M_0[\sigma>M'[t>M$

则由归纳假设知:$\Gamma_{P\to P_i}(M')\in R(M_{0i})$

记 $\Gamma_{P\to P_i}(M)=M_i$ $\Gamma_{P\to P_i}(M')=M'_i$

那么 $M=[M_1^T,M_2^T,\cdots,M_k^T]$ $M'=[M'^T_1,M'^T_2,\cdots,M'^T_k]$

即有 $M'_i\in R(M_{0i})$ $(i=1,2,\cdots,k)$ (1)

不妨设 $t\in T_1\cap T_2\cap\cdots\cap T_h(h\leqslant k)$,$t\notin T_j$(当 $j\in\{h+1,\cdots,k\}$)

那么就有 a)$M'_i[t>M_i$ $(i=1,2,\cdots,k)$;

b)$M_j=M'_j$ $(j=h+1,\cdots,k)$;

综合 a)和 b)得到:$M_i\in R(M'_i)(i=1,2,\cdots,k)$ (2)

由(1)(2)知,即 $M_i\in R(M_{0i})$,即 $\Gamma_{P\to P_i}(M)\in R(M_{0i})$。 □

我们可以利用图1中所给的网系统验证该定理的正确性。$M_0=[1,1,1,0,0,0]$,$M_0[t_1t_3>M_1$,则 $M_1=[0,1,0,1,0,1]$。$\Gamma_{P\to P_1}(M_1)=[0,1]$,易知 $[0,1]\in R(M_{01})$;$\Gamma_{P\to P_2}(M_1)=[0,1]$,易知 $[0,1]\in R(M_{02})$;$\Gamma_{P\to P_3}(M_1)=[1,0]$,易知 $[1,0]\in R(M_{03})$。

Petri 网语言理论中,根据终止状态集的不同定义,把 Petri 网语言分为 L-型,G-型,T-型和 P-型语言[2]。根据变迁集到字母表映射的不同把每一型语言又分为无标注类、无空标注类和带空标注类。为讨论问题的方便,在本文中,我们以 L-型无标注类语言作为讨论的语言形式,取终止状态集为:

$Q_T\subseteq R(M_0)\wedge(\forall M_e\in Q_T,\forall p\in P-P_f:M_e(p)=0)$(其中 P_f 为指定的库所集)。

下面我们给出语言的形式定义:

定义3.2 设 $\Sigma=(P,T;F,M_0)$ 为一 Petri 网,取 $P_f\subseteq P$,定义 Petri 网 Σ 的语言为:$L(\Sigma)=\{\sigma|\sigma\in T^*\wedge M_0[\sigma>M_e\wedge(\forall p\in P-P_f,M_e(p)=0)\}$。$P_f$ 称为 Σ 的终止库所集。 □

为讨论语言的方便,任一 Petri 网直接记为 $\Sigma=(P,T;F,M_0,P_f)$,P_f 为 Σ 的终止库所集。设 $\Sigma_i=(P_i,T_i;F_i,M_{0i})$ 为 $\Sigma=(P,T;F,M_0)$ 的基于库所指标的分解子网,P_f 称为 Σ 的终止库所集,定义 Σ_i 的终止库所集为 P_{fi},$P_{fi}=P_f\cap P_i$,记子网为:

$\Sigma_i=(P_i,T_i;F_i,M_{0i},P_{fi})(i\in\{1,2,\cdots,k\})$。

定义3.3 设 $L(\Sigma_i)$ 分别为 Petri 网 $E_i=(P_i,T_i;F_i,M_{0i},P_{fi})(i\in\{1,2\})$ 的语言,称 L 为 $L(\Sigma_1)$ 与 $L(\Sigma_2)$ 的同步合成语言当且仅当:$T=T_1\cup T_2$,$\forall\sigma\in T^*\wedge\sigma\in L$,$\sigma_i=(\Gamma_{T\to T_i}\sigma)\in L(\Sigma_i)$($\Gamma_{T\to T_i}\sigma$ 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 $L=\overset{2}{\underset{i=1}{[\,]}}(\Sigma_i)$。 □

依次类推,定义 L 为语言 $L(\Sigma_1)$、$L(\Sigma_2)$ 和 $L(\Sigma_3)$ 的同步合成语言当且仅当:$L=(L(\Sigma_1)[\,]L(\Sigma_2))[\,]L(\Sigma_3)$ 或 $L=L(\Sigma_1)[\,](L(\Sigma_2)[\,]L(\Sigma_3))$,记作:$L=\overset{3}{\underset{i=1}{[\,]}}L(\Sigma_i)$。同样可以定义 k 个语言的同步合成语言,记 k 个语言 $L(\Sigma_i)(i\in\{1,2,\cdots,k\})$ 的同步合成语言为 $L(\Sigma)$ 为:$L(\Sigma)=L(\Sigma_k)[\,](\overset{k-1}{\underset{i=1}{[\,]}}L(\Sigma_i))$。

根据"[]"运算的定义,可得到下面的运算性质:

(1)$L[\,]L=L$;

(2)$L_1[\,]L_2=L_2[\,]L_1$;

(3)$L_1[\,](L_2[\,]L_3)=(L_1[\,]L_2)[\,]L_3$;

(4)$L_1[\,](L_2+L_3)=(L_1[\,]L_2)+(L_1[\,]L_3)$。

定理3.3 设 $\Sigma_i=(P_i,T_i;F_i,M_{0i},P_{fi})(i\in\{1,2,\cdots,k\})$ 为 $\Sigma=(P,T;F,M_0,P_f)$ 的基于库所指标的分解网,则

(1)$\Gamma_{T\to T_i}(L(\Sigma))\subseteq(\Sigma_i)$;

(2)$L(\Sigma)=\mathop{[\,]}\limits_{i=1}^{k}L(\Sigma_i)$。

证明:(1)的证明类似于定理3.2,下面证明(2)。

令 $M_f\subseteq R(M_0)$,$\forall\ M\in M_f$,满足:$\forall\ p\in P-P_f$,$M(p)=0$;

$M_{f_i}=\{\Gamma_{P\to P_i}M\mid M\in M_f\}$;

$L(\Sigma)=\{\sigma\mid(\sigma\in T^*)\wedge(M_0[\sigma>M)\wedge(M\in M_f)\}$。

对$|\sigma|$做归纳证明。

(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 $M_0[\sigma>M_1\wedge M_1\in M_f$、当且仅当 $\Gamma_{P\to P_1}(M_0)[\sigma_i>\Gamma_{P\to P_1}(M_1)\wedge\Gamma_{P\to P_1}(M_1)\in M_{f_i}$,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 $\sigma=\sigma'\circ t'$,其中$|\sigma'|=x$,则t'为σ的第 $x+1$个元素。由$\sigma\in L(\Sigma)$当且仅当 $M_0[\sigma'\circ t'>M_{x+1}\wedge M_{x+1}\in M_f$,记:

$M_0[\sigma'>M_x[t'>M_{x+1}$,则 $M_x\in R(M_0)$,令:

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

$M'_{f_i}=\{\Gamma_{P\to P_i}M_x\mid M_x\in M'_f\}(i\in\{1,2,\cdots,k\})$;

由归纳假设,$\exists\ \sigma'_i=\Gamma_{T\to T_i}(\sigma')$,$\sigma'_i\in L(\Sigma_i)$使得:

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

,$\sigma_i\in L(\Sigma_i)$当且仅当 $M_{0i}[\sigma'_i>M'_i\wedge M'_i\in M'_{f_i}$,即当且仅当:$M_{0i}[\sigma_i>M_{(x+1)i}\wedge M_{(x+1)i}\in M_{f_i}$,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。 □

由定理3.3知:通过基于网的库所指标对网进行分解,原网的语言等于子网语言同步合成。由此,求取任意 Petri 网的语言表达式的问题可转化成求取每个分解子网的语言表达式的问题。只要求出每个子网所对应的表达式 $L(\Sigma_i)(i\in\{1,2,\cdots,k\})$,$\mathop{[\,]}\limits_{i=1}^{k}L(\Sigma_i)$即为网的语言表达式,从而为求取任意 Petri 网的语言表达式的问题提供了一种有效的方法。

如:在上面给出的例子中,若取 $P_f=\{p_1,p_2,p_3\}$,则:

$L(\Sigma_1)=(t_1t_4)^*$,$L(\Sigma_2)=(t_3t_4)^*$,$L(\Sigma_3)=((t_1+t_2)t_3)^*$

从而 $L(\Sigma)=L(\Sigma_1)[\,]L(\Sigma_2)[\,]L(\Sigma_3)=(t_1t_4)^*[\,](t_3t_4)^*[\,]((t_1+t_2)t_3)^*$。

推论3.1 设 $\Sigma_i=(P_i,T_i;F_i,M_{0i})(i\in\{1,2,\cdots,k\})$为 $\Sigma=(P,T;F,M_0)$的基于库所指标的分解网,则有:$\forall\ \sigma\in L(\Sigma)$,$\Gamma_{T\to T_i}(\sigma)\in L(\Sigma_i)(i\in\{1,2,\cdots,k\})$。

例3.1 下面以生产者-消费者系统为例,通过基于库所指标的分解分析该系统的行为。

生产者生产产品并将其放到缓存区中,消费者从缓存区中取走产品并将其消费。如果缓存区中没有产品,消费者就需要等待。其 Petri 网模型 Σ_{p-c}如图3所示。

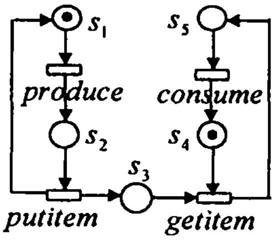
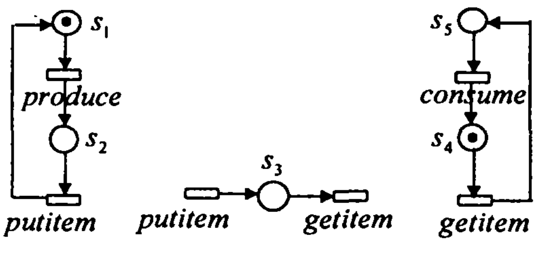


图3 生产者-消费者的 Petri 网模型 Σ_{p-c}



(1) Σ_{p-c1} (2) Σ_{p-c2} (3) Σ_{p-c3}

图4 按库所指标函数分解得到的三个子网

下面我们讨论这个系统的行为(语言),取 $S_f=\{s_1,s_4\}$。

(1)定义库所指标函数 f 满足:

$f(s_1)=f(s_2)=1$,$f(s_3)=2$,$f(s_4)=f(s_5)=3$。

(2)按库所指标函数 f 将 Σ_{p-c}分解成3个子网,分别如图4所示。分解的物理意义是明显的,Σ_{p-c1}对应了生产者,Σ_{p-c2}对应了缓存区,Σ_{p-c3}对应了消费者。

(3)分别写出各个子网所对应的语言。

$L(\Sigma_{p-c1})=(produce\cdot putitem)^*$,说明生产的产品与放入缓存区的产品数目相等,并且 *produce* 要在 *putitem* 的前面。

$L(\Sigma_{p-c2})=(putitem\cdot getitem)^{\otimes}$,在任何时刻放入缓存区的产品数目总要比从缓存区取走的产品数目多,但是 *putitem* 和 *getitem* 是并发关系。其中

$(putitem\cdot getitem)^{\otimes}=\{w\mid w\in\{putitem,getitem\}^*$ 且 $\forall\ s\in Pref(w)$,$\#(puritem,s)\geqslant\#(getitem,s)\}$

$Pref(w)$表示 w 的前缀,$\#(a,w)$表示 a 在 w 中出现的次数)。

$L(\Sigma_{p-c3})=(getitem\cdot consume)^*$,消费的产品数目等于从缓存区取走的产品数目,并且先从缓存区取走的产品消费者才能消费。

(4)生产者-消费者系统的行为描述为:

$$\begin{aligned}L(\Sigma_{p-c})&=L(\Sigma_{p-c1})[\,]L(\Sigma_{p-c2})[\,]L(\Sigma_{p-c3})\\&=(produce\cdot putitem)^*[\,](putitem\cdot getitem)^{\otimes}[\,]\\&\quad(getitem\cdot consume)^*\end{aligned}$$

四、讨论

由前面的分析可以看出,基于库所指标的 Petri 网分解得到的子网保持了原网系统的状态(定理3.2)和行为(定理3.3),即原网的特性在子网上都得到体现。对每个子网系统而言,由于结构自身简单($\forall\ t\in T_i$,$|{}^{\bullet}t|\leqslant1$且$|t^{\bullet}|\leqslant1$),其状态和语言性质分析起来都相对容易,这是我们对一个网系统进行库所指标分解的原由。虽然就单个子网系统而言,可能会改变在原系统中的状态和行为特性,但通过子网与子网之间的相互约束(通过公共变迁的同步合成),就会保持原系统中的状态和行为。如:图5所给的网系统 Σ_4,它的两个基于库所指标的分解网为 Σ_{41}和 Σ_{42}(取 $f(p_1)=f(p_2)=f(p_4)=1$,$f(p_3)=f(p_5)=2$)。显然分解后的子网 Σ_{42}增加了许多状态(p_3和p_5都变成了无界库所)和许多可发生的变迁序列(如$\{t_1^{n_1}t_3^{n_2}t_4^{n_3}\mid n_1\geqslant n_2\geqslant n_3\}$),但如果我们注意到 $Y=[2,1,1,1,1]^T$ 是原网Σ_4 的一个S-不变量,在把Σ_{41}和Σ_{42}的可达标识进行综合时,加

(下转第23页)

心代码开发的特点，来控制代码和文档的制定。对代码规范化的控制主要包括：(1)由于 Linux 源代码的复杂性，在定义变量的时候尽可能地用局部变量，同时在对全局变量和全局数据结构命名时，应加上适当的前缀，一旦命名与原有核心程序中的变量命名有重复，后果不堪设想；(2)使用子程序，即可以把循环和判断从子程序中拖出来，使其成为一个独立的子程序，以降低原有子程序的复杂性，尽量将新增程序或代码与原有的核心代码有明显的分离，若不得不嵌入其中，则必须有清晰统一的解释说明；(3)隐含顺序，例如在核心代码开发中经常涉及到对某一结构的操作集，为了降低阅读和审核的复杂性，应该把相关的代码以常规思维顺序集中在同一子程序中，相关的结构定义集中在同一头文件中等；(4)在首次开发安全操作系统时，可能有一些功能代码并未起作用，但是它可以作为一个功能程序族而存在，我们可以开发一些相关的接口，以便日后的移植或升级；(5)由于 Linux 已根据 POSIX 标准制定了完善的错误处理系统，我们可以利用它原有的错误处理机制来控制系统的运行。

我们参照 ISO9000-3中制定的相关软件产品开发文件编制指南，在安全操作系统开发的不同阶段制定相应的技术和帮助文档，文档要求格式统一、描述详细易懂、参考性强。对产生的文档要进行统一的管理和实时的更新，使之与当前开发进度和实际开发中采取的方案保持一致性，为以后的审核、测试提供必要的参考。

结合有效的标准，对代码以及文档进行规范化的管理，使得整个开发过程有精确的文字记录，可以有效地保证开发不偏离预期轨道，也有利于从中总结经验，为以后做其他类似的软件开发过程提供参考价值，提高企业软件开发过程的整体水平。

结束语 本文介绍了目前国际上软件产品开发中被广泛采用的过程和质量控制评估标准 CMM 和 ISO9000系列，并以此为基础，结合实际开发经验，介绍了基于 Linux 的安全操作系统开发中如何控制过程、提高开发质量的一些方法。采用国际开发标准、重视开发过程和产品质量是我国软件业走向国际市场必须重视的一个环节，而我们开发本安全操作系统的实际经验表明，结合要进行的开发过程的特点，采用合适的标准并进行相应的裁减，使之应用到开发过程中的方方面面，会使得软件产品的开发事半功倍。希望藉此能引发国内软件业关于过程和质量控制的研究。

参考文献

1 何新贵，王纬，王方德，等编著. 软件能力成熟度模型. 北京：清华大学出版社，2000. 11

2 杨一平，等. 软件能力成熟度模型 CMM 方法及其应用. 北京：人民邮电出版社，2001. 4

3 America Department of Defense. TRUSTED COMPUTER SYSTEM EVALUATION CRITERIA. CSC-STD-001-83，15 Aug 83

3 郑人杰，等. 实用软件工程. 北京：清华大学出版社，1997

4 MIL-STD-498. Software Development and Documentation

5 Paulk M C，Curtis B，Chrissis M B，Weber C. Capability Maturity Model，Version 1. 1. IEEE software，July 1993. 18～27

6 CMU-SEI-93-RT-24，Capability Model for Software，Version 1. 1. Feb. 1993

7 林汉川主编. ISO9000在质量管理中的应用. 广州：广东人民出版社，1999. 1

8 林汉川主编. ISO9000在服务、流程性材料、软件中的应用. 广州：广东人民出版社，1999. 1

(上接第17页)

上 S-不变量的条件，就可除去那些不属于原网的可达标识的标识(这一点将另文讨论)；至于语言(即变迁可发生序列)的综合，正如定理3. 3所指出，通过语言的同步合成运算就可以实现。

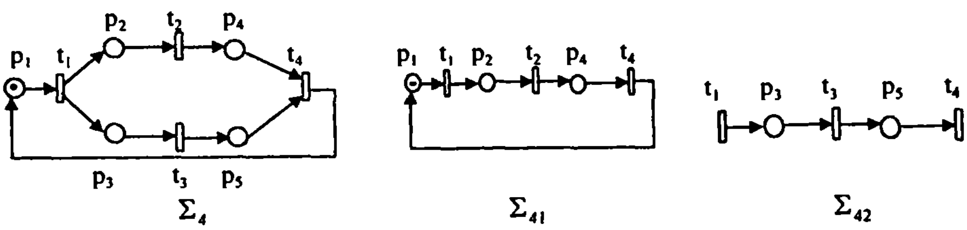


图5 一个 Petri 网 Σ_4和它的分解子网 Σ_{41}和 Σ_{42}

结束语 文[8]分析了系统合成(同步合成和共享合成)过程中合成网同子网之间满足动态不变性。文[6]和[7]提出的网的“和分解”和“并分解”，均是从保持系统的结构性质方面考虑系统的分解。本文提出的基于库所指标的 Petri 网分解技术，这种方法使分解得到的子系统保持原系统的行为和状态。同以往的工作相比，本文可以说是文[7]的逆过程；同文[5]和[6]相比，本文则是着重从保持系统的动态不变性(行为不变性和状态不变性)方面考虑分解网系统。

参考文献

1 Hyunglee K，et al. Generalized Petri Net Reduction Method. IEEE Transaction on Systems，Man and Cybernetics，1987，SMC–17(2)

2 Suzuki I. A Method for Stepwise Refinement and Abstraction of Petri Nets. J. of Computer and System Sciences，1983，27：51～76

3 Murata T. Petri Nets：Properties，Analysis and Applications. Proceedings of The IEEE，1989，77(4)

4 Jiang Changjun，Wu Zhehui. Net Operations. Computer Science and Technology，1992，7(4)：333～344

5 王培良，等. P/T 系统的和分解. 计算机科学，1999年(增刊)

6 王培良，等. Petri 网的并分解. 控制理论与应用，2001，18(1)：116～118

7 蒋昌俊. Petri 网的动态不变性. 中国科学(A 辑)，1997，27(6)：567～573

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