ISSN 1000-9825, CODEN RUXUEW
*Journal of Software*, Vol.18, No.7, July 2007, pp.1553−1562
DOI: 10.1360/jos181553

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

# 基于逻辑电路的 Petri 网化简方法*

叶剑虹[1,2+], 宋 文[2], 孙世新[1]

[1](电子科技大学 计算机科学与工程学院,四川 成都 610054)

[2](西华大学 数学与计算机学院,四川 成都 610039)

## A Reduction Technique of Petri Nets Based on Logic Circuit

YE Jian-Hong[1,2+], SONG Wen[2], SUN Shi-Xin[1]

[1](School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China)

[2](School of Mathematics and Computer Engineering, Xihua University, Chengdu 610039, China)

+ Corresponding author: Phn: +86-013194983160, E-mail: leafever@163.com, http://www.uestc.edu.cn

**Ye JH, Song W, Sun SX. A reduction technique of Petri nets based on logic circuit. *Journal of Software*, 2007, 18(7):1553−1562.** http://www.jos.org.cn/1000-9825/18/1553.htm

**Abstract**: In traditional methods, the local structure of Petri net is required to compare with all reduction rules. The process is complicate and does not fit for nets with inhibitor arcs. This paper presents a new reduction method. Firstly, Petri net is divided into several maximal acyclic subnets and each one is expressed with logic form. Then, logic algebra is used to reduce the logic form. Finally, the result is reconstructed and embedded in the original net. This paper establishes a method to find and reduce the maximal acyclic subnets and presents the correlative proofs. This method can be applied to Petri nets or subnets with inhibitor arcs and acyclic.

**Key words**: Petri net; reduction; logic algebra; maximal acyclic subnet

摘 要: 已有的 Petri 网化简方法需将网的局部结构与化简规则作逐一的比对,步骤较为繁琐,并且所提供的方法不适合于带抑止弧的网.采用一种与传统方法不同的化简思路,首先将网划分为若干个最大无圈子网,将每个最大无圈子网表达为若干个逻辑式.用逻辑代数来完成逻辑式的化简,最后将其结果还原为 Petri 网回嵌到原网中,完成整个网的化简.给出了寻找最大无圈子网、最大无圈子网的化简算法以及相关的证明.该方法将化简范围扩展到了带抑止弧的无回路的网或网的局部.

关键词: Petri 网;化简;逻辑代数;最大无圈子网

中图法分类号: TP301 文献标识码: A

Petri 网[1,2]是异步并发系统建模与分析的一种重要工具,在原型 Petri 网中加入抑止弧,给网系统的性质分析增加了更大的难度,但这个代价的付出,可以换来网系统模拟能力的增强.对于一般的网系统,往往存在着出现状态爆炸的危险,从而给分析带来困难.对此,有人提出化简的思想.化简是将一个较复杂的 Petri 网简化成一

* Supported by the National Natural Science Foundation of China under Grant No.60473030 (国家自然科学基金); the Fundamental Research Foundation of Science and Technology Bureau of Sichuan Province of China under Grant No.03226125 (四川省科学技术厅应用基础课题基金)

Received 2006-10-08; Accepted 2006-11-30

个比较简单的网的等价变换过程,这个过程减少了可达状态空间的规模,对简单网的分析可为理解原网提供充分的信息[3].

关于 Petri 网的化简方法已有较多的研究工作,Murata[4,5]针对 *T*-图提出若干化简规则,并讨论了这些规则对活性、安全性的保持关系;蒋昌俊[6]推广了 Murata 的工作,进一步提出了对加权 *T*-图的化简规则,并给出了化简运算对结构性质的保持条件;Aalst[7]在工作流过程验证中使用了这些技术;Mugarza[8]和 Ferscha[9]给出了并行程序的化简规则,并给出了相应的 Petri 网化简规则;林闯[10,11]运用自底向上逐步综合替代的分层分析方法给出了基本随机 Petri 网性能等价公式,并利用变迁可实施谓词和随机开关对模型进行了精化设计及化简.这些化简方法都是在权值为 1 的 Petri 网上考虑或者是在加权的特殊 Petri 网子类上考虑的;文献[12]将化简方法推广到加权的 P/T 网,并给出了其对结构性质及公平性、$S(T)$不变量的保持条件.文献[13,14]采用类似文献[12]的方法给出了工作流模型中的验证.这些方法都略显繁琐,化简过程中的主观性和随意性较大,难以算法实现,并且不能对带抑止弧的网进行化简.

对于简单的组合逻辑电路,通常可以通过基于逻辑代数的真值表分析,应用实验验证来达到逻辑验证或分析的目的[15,16].近几年来,基于 Petri 网的逻辑控制器的逻辑电路实现方法逐渐受到了重视.文献[17]给出了用 Petri 网来模拟逻辑电路、利用 Petri 网的变迁引发规则来分析原数字电路的方法.文献[18,19]给出了用 Petri 网变迁的引发规则确认逻辑电路设计中所需的驱动条件,进而选择适当的元件设计所需的组合逻辑电路或时序逻辑电路的方法.这些方法设计过程统一、简单.但传统的研究工作往往局限于用 Petri 网来模拟逻辑电路的设计和分析,而对逻辑电路中一些成熟的技术在 Petri 网的应用却鲜有文献涉及.本文提出了一种新的基于逻辑代数的化简方法,简称为 RM-LC 方法(reduction methods based on logic circuit),先将需化简的 Petri 网划分为若干个最大无圈子网,并对每个最大无圈子网用逻辑代数式来描述,并利用逻辑电路中的逻辑代数化简规则将其化简,最后再将逻辑代数式用 Petri 网还原并回嵌到原网中.

本文第 1 节给出化简相关的概念.第 2 节给出化简的方法.第 3 节给出化简的算法.第 4 节是实例.第 5 节是结论.

## 1 化简相关的基本概念

有关 Petri 网的基本概念来自于文献[1,2,20],本文在此不再赘述.

### 1.1 基本门电路的Petri网模拟及网性质的扩充

首先可用带抑止弧的网构造组成计算机硬件系统的基本元器件:逻辑电路和时序电路[21].数字逻辑中有与、或、非、与非、或非、与或非、异或、异或非等 8 个运算符,它们构成数字逻辑运算的完全集,其最小完全集是非、或.为了叙述方便,本文给出基本与、或、非门电路的 Petri 网模拟,如图 1 所示.

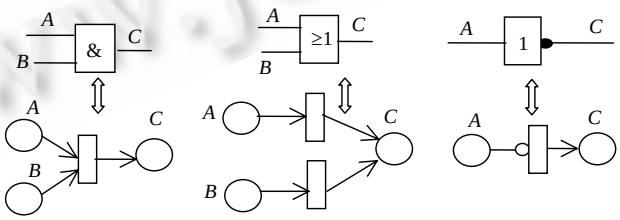


Fig.1 Simulation of gate circuit based on Petri net

图 1 基本门电路的 Petri 网模型

为便于在 RM-LC 化简过程中保持逻辑电路和 Petri 网性质的一致性,需给出以下定义和命题.

定义 **1.1**. $N=(S,T;F,I)$是一个带抑止弧的 E/N 网,其中,$I\subset S\times T$ 称为抑止弧集,$I\cap F=\varnothing$.$t\in T$,$t$ 的动力源定义为 $Power(t)=\{s|(s,t)\in F\}$;$t$ 的控制源定义为 $Control(t)=\{s|(s,t)\in I\}$.

定义 **1.2**. $\Sigma=(S,T;F,I,C)$是一个带抑止弧的 E/N 系统,其中,$(S,T;F,I)$是一个网,$C$ 是网的标识集.$\exists c\in C:c[t\rangle$,$t$ 的

托肯位置定义为 $Position(s,t)=\begin{cases}s, s\in Power(t)\wedge c(s)=1\\ \bar{s}, s\in Control(t)\wedge c(s)=0\end{cases}$.

图 2 中,$c_0$=(1,1,0,0,0),$Power(t)$={$s_1$,$s_2$};$Control(t)$={$s_3$};$Position(s_1,t)=s_1$;$Position(s_2,t)=s_2$; $Position(s_3,t)=\bar{s}_3$. 变迁 $t$ 发生后, $Position(s_4,t_1)=s_1\cdot s_2\cdot\bar{s}_3$; $Position(s_5,t_2)=s_1\cdot s_2\cdot\bar{s}_3$,这里的点乘代表逻辑运算中的合取.

$Position(s_i,t)$,$i$=1,2,3 称作 $s_4$,$s_5$ 的父本信息; $s_i$ 将 $Position(s_i,t)$,$i$=1,2,3 构建出两个副本分别传给 $s_4$,$s_5$.

命题 **1**. 给定一个带抑止弧的 E/N 系统$\Sigma$,$t\in T$,$\exists c$:$c[t\rangle c'$, $Position(s',t')=\prod_{s\in{}^\bullet t}Position(s,t)$,这里,$s'\in t^\bullet$, $t'\in(s')^\bullet$.即变迁 $t$ 的发生,将使得 $t^\bullet$中的所有库所同时继承相同的父本信息.这实际上是逻辑电路中与门的性质表现.

命题 **2**. 给定一个带抑止弧的 E/N 系统$\Sigma$, $t_1,...,t_\alpha\in T$,${t_1}^\bullet\cap...\cap{t_\alpha}^\bullet\neq\varnothing$,$c_1,c_2,\ldots,c_\alpha\in C$:$c_i[t_i\rangle$,$1\le i\le\alpha$,则存在着一个 $c_i$,某个 $t_i$ 的发生使得 ${t_1}^\bullet\cap...\cap{t_\alpha}^\bullet$ 的库所中得到的信息是 $Position(s,t)=\sum_{i=1}^{\alpha}Position(s_i,t_i)$,这里, $s\in{t_1}^\bullet\cap...\cap{t_\alpha}^\bullet$, $t\in s^\bullet$.

经典的 E/N 系统称 ${t_1}^\bullet\cap...\cap{t_\alpha}^\bullet$ 处存在冲撞(contact)[2].命题 2 中,某个使能变迁的发生将导致其非空的 ${t_1}^\bullet\cap...\cap{t_\alpha}^\bullet$ 后置库所中含有一个具有多父本析取信息的托肯资源.这实际上是逻辑电路中或门的性质表现.

如图 3 所示,$c$=(1,1,0),$Position(s_1,t_1)=s_1$;$Position(s_2,t_2)=s_2$,$t_1$ 或 $t_2$ 发生后,$Position(s_1,t)=s_1+s_2$,这里的“+”代表逻辑运算中的析取.

命题 **3**. 给定一个带抑止弧的 E/N 系统$\Sigma$,$\exists t\in T$,若${}^\bullet t=\{s\}$,$t^\bullet=\{s'\}$,$Control(t)=\{s\}$,$c[t\rangle c'$. $Position(s',t')=\bar{s}$,这里, $t'\in(s')^\bullet$.

如图 4 所示,$c$=(0,0),$c[t\rangle c'$,变迁 $t$ 发生后, $Position(s',t')=\bar{s}$.这实际上是逻辑电路中非门的性质表现.

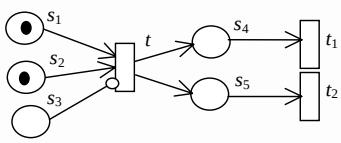


Fig.2 Illustrating the definition 1.2 and proposition 1

图 2 定义 1.2 和命题 1 的图示

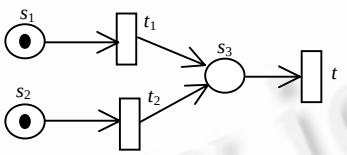


Fig.3 Illustrating the proposition 2

图 3 命题 2 的图示

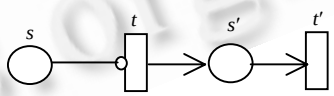


Fig.4 Illustrating the proposition 3

图 4 命题 3 的图示

由于逻辑运算规则与 Petri 网的发生规则具有差异性,需要对逻辑点乘运算作修正.首先定义几个基本结构:

定义 **1.3**. $N$=($S$,$T$;$F$,$I$)是一个带抑止弧的 E/N 网:

(1) $\underline{N}_0$ 是 $N$ 的一个 0 型结构,如果 $s_1,s_2\in S$.满足:① ${s_1}^\bullet\supseteq{s_2}^\bullet$;② $s_1\in Power(t_1)$;③ $s_1\in Control(t_2)$,$t_2\in{s_2}^\bullet$;

(2) $\underline{N}_1$ 是 $N$ 的一个 0 型结构,若进一步满足:${}^\bullet s_1={}^\bullet s_1$,则称 $\underline{N}_1$ 是 I 型结构;

(3) $\underline{N}_2$ 是 $N$ 的一个 0 型结构,若进一步满足:① $s_1$ 是一个或运算的输出库所;② 或运算的某个输入库所 $s'$ 满足${}^\bullet s'={}^\bullet s_2$,则称 $\underline{N}_2$ 是 II 型结构;

(4) $\underline{N}_3$ 是 $N$ 的一个 0 型结构,若进一步满足:① $s_1$ 是一个与运算的输出库所;② 与运算的某个输入库所 $s'$ 满足${}^\bullet s'={}^\bullet s_2$,则称 $\underline{N}_3$ 是 III 型结构.

给定一个带抑止弧的 E/N 系统$\Sigma$,$c[\sigma t_2\rangle c'$, $s^\omega\in{t_2}^\bullet$,$t\in(s^\omega)^\bullet$,有一个父本信息 $s$,使得 $Position(s_2,t_2)=s$.

(1) 对于 I 型的结构,又有 $Position(s_1,t_2)=\bar{s}$.显然,具有这种结构的网在满足条件的标识下允许变迁 $t_2$ 的发生,并随之产生有效的输出 $Position(s^{\omega},t)=\bar{s}\cdot s$.为了合理地表示 $s^{\omega}$可以得到托肯资源并继承父本信息 $s\cdot\bar{s}$.修正此时的逻辑点乘的规则为 $\bar{s}\cdot s=s$,如图 5(a)所示;

(2) 对于 II 型的结构,有 $Position(s_1,t_2)=\overline{s+s_{others}}$.同样,$t_2$ 的发生产生 $Position(s^{\omega},t)=\overline{s+s_{others}}\cdot s$.根据德·摩根(A.DeMorgen)律,$\overline{s+s_{others}}\cdot s=(\bar{s}\cdot\overline{s_{others}})\cdot s$,这与 I 型定义类似,故规定 $s\cdot\bar{s}=s$.如图 5(b)所示;

(3) 对于 III 型的结构,有 $Position(s_1,t_2)=\overline{s\cdot s_{others}}$,$t_2$ 的发生产生 $Position(s^{\omega},t)=\overline{s\cdot s_{others}}\cdot s$,$\overline{s\cdot s_{others}}\cdot s=(\bar{s}\cdot s+\overline{s_{others}}\cdot s)$,这实际上将 $s^{\omega}$的信息来源转换成等价的或门输入,因 $t_2$ 的动力源已含有父本信息 $s$,所以 $t_2$ 的发生所需的条件中控制源信息只要 $\bar{s}$ 或 $\overline{s_{others}}$ 二者取其一的条件成立即可,为了阐明 $s^{\omega}$还要受 $\overline{s_{others}}$ 信息的影响,舍弃 $\bar{s}\cdot s$ 项,选择含有更丰富输入信息的 $\overline{s_{others}}\cdot s$ 项,故修正此时的逻辑点乘的规则为 $\bar{s}\cdot s=0$,于是,$\bar{s}\cdot s+\overline{s_{others}}\cdot s=\overline{s_{others}}\cdot s$.

定义 **1.4**. $\Sigma$=(*S*,*T*;*F*,*I*,*C*)是一个带抑止弧的 E/N 系统,网 *N*=(*S*,*T*;*F*,*I*)中所含的 I 型、II 型结构在满足条件的标识下,修正局部逻辑点乘运算规则为 $\bar{s}\cdot s=s$;在 III 型结构下,修正局部逻辑点乘运算的规则为 $\bar{s}\cdot s=0$.

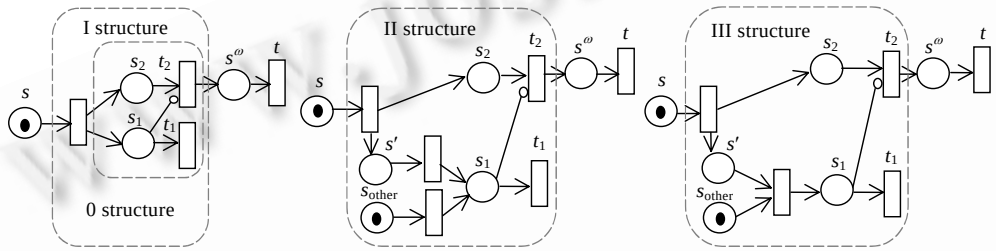


Fig.5 Illustrating the definition 1.3, definition 1.4

图 5 定义 1.3 和定义 1.4 的图示

## 2 RM-LC 化简方法

在 Petri 网中,冲突关系描述了系统的非确定性:在某种情况下有两个(或多个)事件都有权发生,但在实际运行过程中只有一个能真正发生.系统存在冲突之处,正是外界环境可以对其施加控制(加以选择)[20].因此,为了避免外部控制信息的丢失,以下章节的化简仅考虑无冲突的、带抑止弧的 E/N 网或有冲突网的无冲突部分的化简.首先给出一些相关的定义:

定义 **2.1**. *N* 是一个带抑止弧的 Petri 网,*w*=($s_0$,…,$s_n$)叫作长度为 *n* 的一条路径,当且仅当 $s_{i-1}^{\bullet}={}^{\bullet}s_i, i=1,...,n$,且 $1\le i\ne j\le n, s_i^{\bullet}\ne s_j^{\bullet}\wedge {}^{\bullet}s_i\ne {}^{\bullet}s_j$.*w* 始于 ${}^{\bullet}s_0$ 终于 $s_n^{\bullet}$;路径 *w* 是一个环,当且仅当 ${}^{\bullet}s_0=s_n^{\bullet}$.

定义 **2.2**. *N* 是一个带抑止弧的 E/N 网,*T*′⊆*T*,*N* 的 *T*-子网 *N*′=(*S*′,*T*′;*F*′,*I*′).其中,$S'={}^{\bullet}(T')\cup(T')^{\bullet}$,$F'=F\cap((S'\times T')\cup(T'\times S'))$,$I'=I\cap(S'\times T')$.

定义 **2.3**. 给定一个带抑止弧的 E/N 网 *N*=(*S*,*T*;*F*,*I*),*T*′⊆*T*,*N*′是 *T*′的 *T*-子网,*N*′是 *N* 的最大无圈子网,当且仅当 $\forall t\in T-T'$,由 $t\cup T'$所组成的 *T*-子网必产生环路.特别地,若 *T*′=*T*,则称网 *N* 为无圈网.

预备步. 找 *N* 中所有的最大无圈子网,具体实现由算法 3.1.1 给出.

以一个最大无圈 *N*′的化简为例,对最大无圈子网的化简步骤及规则定义阐述如下:

步骤 **1**. 最大无圈子网的划分.

规则 **1**. 选取网 *N* 的一个最大无圈子网 *N*′,满足:$S_I\subseteq S'$,${}^{\bullet}(S_I)\cap T'=\varnothing$, $\forall s\in S_I,\exists x\in S'$,s.t. $x<s$;$S_O\subseteq S'$,$(S_O)^{\bullet}\cap T'=\varnothing$, $\forall s\in S_O,\exists x\in S'$,s.t. $s<x$.

将 *N*′划分为 3 个部分:输入部分由库所集 $S_I$、变迁集 $T_I=\{t|{}^{\bullet}t=(Power(t)\cup Control(t))\subseteq S_I\}$及弧$((S_I\times T_I)\cup$

$(T_I\times(T_I)^\bullet))\cap F',(S_I\times T_I)\cap I'$组成;输出部分由库所集$S_O$组成;中间部分由库所集$S'_{medial}=S'-(S_I\cup S_O)$、变迁集$T'-T_I$及弧$((S'_{medial}\times T')\cup(T'\times S'_{medial}))\cap F',(S'_{medial}\times T')\cap I'$组成.

如图 6 虚线框所示,$S_I=\{s_0,s_1,s_2\}$,$S_O=\{s_5,s_6,s_7,s_8\}$.

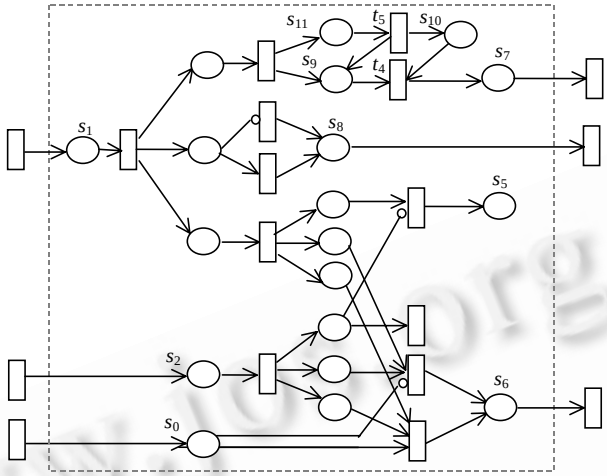


Fig.6 The maximal acyclic subnet $N'$

图 6 最大无圈子网 $N'$

步骤 **2**. 最大无圈子网的化简.

规则 **2**. 对$\forall t\in T_I$,$t$ 的动力源添加托肯资源,$t$ 的控制源不添加托肯资源(以下简称该操作为托肯的加载),构建变迁 $t$ 的使能标识,使得每一个 $T_I$ 中元素都发生至少一次,对应的发生序列为$\sigma$,$\sigma$中每一个变迁的发生都遵循命题 1~命题 3 的规定,将标记有"父本"信息的托肯向后置库所集传递.

规则 **3**. 设 $c_0[\sigma\rangle c'[\sigma'\rangle c'_O$,这里,$\sigma'\in(T'-T_I)^*$. $c'_O$ 是最大无圈子网上运行的终态标识.与 $c'_O(s)$ 对应的有 $Position(s,t)$,$s\in S_O$,$t\in(s)^\bullet$,$Position(s,t)$是论域为 $S_I$ 的逻辑函数.

若 $t\in T'$,$\forall c\in R(c_0)$:$\neg c[t\rangle$,则称变迁 $t$ 在该最大无圈子网中是死的.

令 $T_{dead}=\{t|t\in T',t$ 是死变迁$\}$;$S_{t\text{-}dead}=\{s|s\in S',t\in T_{dead},s\in{}^\bullet t\cup t^\bullet\}$; $S_{dead}=\{s\mid s\in S_{t-dead}\wedge c'_O(s)\neq 0\}$; $S_{T_{dead}-Logic}=\{Position(s,t)\mid\forall s\in S_{dead},t\in s^\bullet\}$;$S_{O\text{-}Logic}=\{Position(s,t)|\forall s\in S_O,t\in s^\bullet\}$.

规则 **4**. 经规则 3 所得的 $S_{O\text{-}Logic}$ 和 $S_{T_{dead}-Logic}$ 中的每一个表达式,若合取范式中含有形如 $\bar{s}\cdot s$, $\overline{s\cdot s_{\text{others}}}\cdot s$ 或 $\overline{s+s_{\text{others}}}\cdot s$ 的项,依据定义 1.4 对其进行修正,使得父本中的每个变量 $s$ 在每个合取范式中至多以 $s$ 或 $\bar{s}$ 出现一次.经修正后表达式与 Petri 网的发生规则相吻合.

规则 **5**. 将经规则 4 得到的表达式用公式化简法(或采用卡诺图(Karnaugh map),Q-M(quine McCluskey)化简法)将其化简到最简形式.整个规则 5 中化简规则的应用都遵循逻辑代数的方法,与 Petri 网无关.令 $S_{O\text{-}Logic}$, $S_{T_{dead}-Logic}$ 化简后的结果分别为 $S_{O\text{-}new}$, $S_{T_{dead}-new}$.

步骤 **3**. 化简后的表达式还原为新的子网 $N'_{new}$.

将经步骤 2 后所得的最简逻辑表达式集还原为新的子网.还原过程需考虑由于逻辑电路的逻辑值没有消耗性,而 Petri 网的发生序列会导致资源发生变异和迁移,这两者之间存在的差异性.

规则 **6**(解决集合 $\boldsymbol{S_{O\text{-}new}}$ 的还原问题).

(1) 如果 $S_{O\text{-}new}$ 中的元素由 $S_I$ 中的元素表示,为解决托肯资源具有消耗性的问题,对于 $S_I$ 中的元素 $s$,统计它在每一个表达式中以动力源信息形式出现的次数 $sum$,在还原时产生 $sum$ 个副本.依据命题 1~命题 3 的逆过程重构这些表达式所对应的网的局部.重构后的网结构显然是由 $S_I$ 中的元素与 $S_O$ 中的元素的某种连接关系构成.

(2) 对于 $Position(s,t)\in S_{O\text{-}new}\wedge Position(s,t)=0$,该表达式意味着$\sigma$或$\sigma'$的发生对 $s$ 不产生任何影响,还原时仍然画出 $s$,且在该子网中${}^\bullet s=\varnothing$.

(3) 对于 $Position(s,t)\in S_{O\text{-}new}\wedge Position(s,t)=1$,该表达式意味着,不论它的父本在初始标识下是否标识,$s$ 都能获得托肯.这是逻辑中的一个永真式.还原时画出该库所,并为其添加一个无前置库所的变迁.

规则 **7**(解决集合 $S_{T_{dead}-new}$ 的还原问题).

(1) 如果 $S_{T_{dead}-new}$ 中的元素由 $S_I$ 中的元素表示,对于 $S_I$ 中的元素 $s$,类似规则 6(1)的处理.

(2) 恢复所有 $T_{dead}$ 中的元素 $t$,对于$\forall t\in T_{dead}$,将其与 $S_{t\text{-}dead}$ 中的元素依据原 $N'$的连接关系重新作连接,对其中的 $s\in S_{dead}$,用本规则(1)所重构的网局部结构替换 $s$.

(3) 若 $t\in T_{dead}, s\in t^{\bullet}\wedge s\in S_O$,则$(t,s)$的弧可以删除.因为 $t$ 是“死”变迁,它不会发生,不会影响 $Position(s,t')$中的 $t'\in s^{\bullet}$.

规则 **8**. (1) 若 $s\in S_I,s$ 在原子网 $N'$中以动力源信息出现,但在 $S_{O\text{-}new}$ 或 $S_{T_{dead}-new}$ 中只以控制源信息的形式出现.如图 7(a)左图中的 $s'$,$Position(s',t)=s$,经过逻辑化简后,若 $Position(s',t)=\bar{s}$,则还原后的结果如图 7(a)右图所示.类似的控制源信息的情况如图 7(b)所示,说明略去.(2) 若 $s\in S_I,s$ 出现在 $S_{O\text{-}Logic}$ 和 $S_{T_{dead}-Logic}$ 的表达式中,但已不再出现在 $S_{O\text{-}new}$ 与 $S_{T_{dead}-new}$ 的任何表达式中,说明化简后的表达式不再含有任何关于 $s$ 的信息,此时仍然画出 $s$,且 $s^{\bullet}=\varnothing$.如图 8 中 $s_0$ 所示.

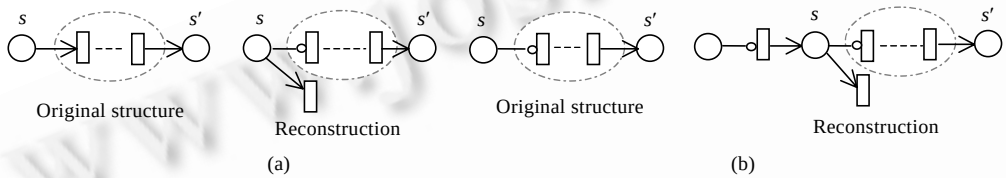


Fig.7 Illustrating the rule 8(1)

图 7 规则 8(1)的连接图示

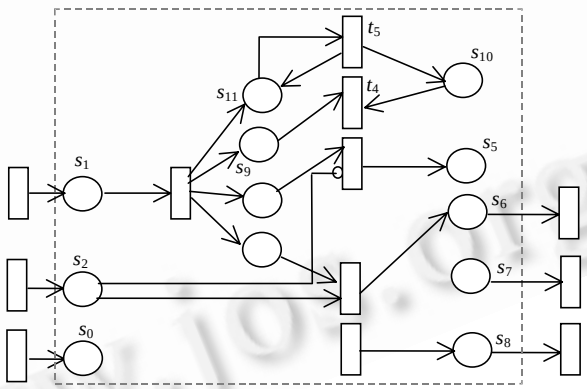


Fig.8 The reduction result of $N'$

图 8 $N'$经化简后的结果

规则 **9**. 将经过规则 6~规则 8 还原后的网结构做库所的连接并整理,合并重复出现的库所,得到最简形式的、还原后的新子网 $N'_{new}$.

步骤 **4**. 新子网代替原最大无圈子网回嵌到网 $N$ 中.

规则 **10**. 将经步骤 3 后所得到的新子网 $N'_{new}$ 代替最大无圈子网 $N'$回嵌到原网 $N$ 中,新子网保留了原最大无圈子网 $N'$的 $S_I$ 及 $S_O$,对于$\forall s\in S_I$,若$\exists t\in{}^{\bullet}s$,则恢复弧的连接$(t,s)$;对于$\forall s'\in S_O$,若$\exists t'\in s^{\bullet}$,则恢复弧的连接$(s',t')$.如此进行,可以得到 $N$ 化简后的结果.

定理 **2.1**. $N'_{new}$ 代替 $N'$回嵌到原网中是边缘保持的.

证明:根据步骤 3,新子网 $N'_{new}$ 的库所集由 $N'$中 $S_I$ 和 $S_O$ 的所有元素以及根据规则 6~规则 8 对 $S_I$ 中的元素

做的副本库所组成.新生成的库所与 $S_I$ 中的原库所是 $\underline{li}$ 关系,并不影响原库所的前置集.根据规则 1 的最大无圈子网的定义,$S_I$ 和 $S_O$ 中的元素是连接 $N'$与原网 $N$ 的边缘.既然新子网不改变 $S_I$ 和 $S_O$ 中的元素,则用 $N'_{new}$ 代替 $N'$回嵌到原网中对边缘的保持就是无损的. □

定义 **2.4**. $N_1,N_2$ 是网 $N$ 的子网,$N_3$ 是 $N_1,N_2$ 的连接子网,当且仅当 $N_1\subseteq N_3,N_2\subseteq N_3$.

定理 **2.2**. 若 $N',N''$都是网 $N$ 的最大无圈子网,则它们化简后的子网 $N'_{new}$, $N''_{new}$ 回嵌到原网后,得到的连接子网 $N^+$无须再作化简.

证明:(1) 若 $N^+-N'_{new}-N''_{new}\subseteq S$, $\exists s\in S_O\wedge s\in S''_I$,其中,$S_O$ 是 $N'$的输出部分, $S''_I$ 是 $N''$的输入部分.不妨假设连接子网 $N^+$可再做化简.这意味着存在 $t\in s^\bullet$,也即 $t\notin T'$但 $t\in T''$,其中,$T'$,$T''$是 $N'$,$N''$的 $T$-元.于是,$t\cup T'$可以构成一个比 $N'$更大的最大无圈子网.这与 $N'$已是最大无圈子网的定义 2.3 相矛盾.

(2) 若 $(N^+-N'_{new}-N''_{new})\cap T\neq\varnothing$,$\exists t\in T$但 $t\notin T'\cup T''$,不妨假设$^\bullet t\in S_O$,显然,根据最大无圈子网的定义,$t^\bullet$一定有属于 $S'$的元素,其中,$S'$是 $N'$的 $S$-元,即 $t$ 会导致 $N'$形成环路.无论 $t^\bullet\in S''_O$,还是 $t^\bullet\in S''_I$,其中的 $S''_O$ 是 $N''$的输出部分.经化简后的 $N'_{new}$, $N''_{new}$ 的连接 $N^+$中必含有环路.故它也无须再做化简. □

## 3 算法实现

定义 **3.1**. $N$ 的基于化简的修正关联矩阵是以 $S\times T$ 为序标集的 $m\times n$ 阶矩阵 $C^*$,其定义如下:(1) 若$(s_i,t_j)\in F$, $C^*(s_i,t_j)=-s_i$;(2) 若$(s_i,t_j)\in I$, $C^*(s_i,t_j)=-\bar{s}_i$;(3) 若 $(t_j,s_i)\in F$, $C^*(s_i,t_j)=\prod{}^\bullet t_j$ ; (4) 其他 $C^*(s_i,t_j)=0$.

### 3.1 算 法

算法 **3.1.1**. 求网 $N$ 的最大无圈子网的算法.

输入:网 $N=(S,T;F)$;

输出:网 $N$ 的所有最大无圈子网.

(1) 任意选取一个 $t_1\in T$ 作为初始变迁,令 $T'=\{t_1\}$,由 $t_1$ 及其前后集构造一个 $T$-子网,记为 $N'_1$.转向(2).

(2) 扫描$^\bullet t_1$ 与 $t_1^\bullet$,如果 $t_i\in{}^\bullet(^\bullet t_1)\cup(t_1^\bullet)^\bullet$ 但 $t_i$ 不属于 $T'$,考察$^\bullet t_i$ 与 $t_i^\bullet$ 是否会与 $N'_1$ 构成环路:若是,为 $t_i$ 打上标记;否则,将 $t_i$ 加入集合 $T'$,并根据 $T'$重构 $T$-子网 $N'_1$.转向(3).

(3) 重复步骤(2),直到不再有未被标记的变迁 $t\in T-T'$可以加入 $T'$的集合为止. $N'_1=(S'_1,T'_1,F'_1)$ 为网 $N$ 的一个最大无圈子网.转向(4).

(4) 考察那些未被打上标记且不属于 $T'$的变迁,选取其中的某个变迁替换步骤(1)中的 $t_1$ 作为新的初始变迁,类似步骤(2)、步骤(3),如果可能,直至构建出网 $N$ 的另一个最大无圈子网 $N'_2=(S'_2,T'_2,F'_2)$ .转向(5).

(5) 重复步骤(4),得到 $N'_1,N'_2,...,N'_j$ .若$\forall t\in T$,或者它已被打上标记,或者它已属于 $N'_1,N'_2,...,N'_j$ 中某个最大无圈子网的元素.转向(6).

(6) 输出所得到的所有的最大无圈子网 $N'_1,N'_2,...,N'_j$ .算法结束.

算法 **3.1.2**. 最大无圈子网的化简算法.

输入:网 $N$ 的一个最大无圈子网 $N'$;

输出:经化简后的 $N'_{new}$ 及根据其对 $C^*$所做的修正.

① 根据输入的最大无圈子网 $N'$分离出输入部分、输出部分以及中间层.转向②.

② 对输入部分 $S_I$ 中动力源和控制源进行托肯加载处理.转向③.

③ 让 $N'$中可发生的变迁都发生,分别求得 $S_O$ 及 $S_{T_{dead}}$ 中各库所的信息表达式 $S_{O\text{-}Logic}$ 和 $S_{T_{dead}-Logic}$ .转向④.

④ 化简 $S_{O\text{-}Logic}$, $S_{T_{dead}-Logic}$ 中各表达式,得到 $S_{O\text{-}new}$ 和 $S_{T_{dead}-new}$ .转向⑤.

⑤ 还原化简后表达式的网结构 $N'_{new}$ .转向⑥.

⑥ 用 $N'_{new}$ 置换原 $N$ 中的最大无圈子网 $N'$,得到 $N_1^*$ .转向⑦.

⑦ 生成 $N_1^*$ 的修正关联矩阵,对 $C^*$中与 $N_1^*$ 相关的部分做相应的添加和删除元素操作.算法结束.

下面分析算法 3.1.1、算法 3.1.2.显然,定理 2.2 保证了可停机,算法 3.1.1 是 P 类算法,因为给定一个网 $N$,它

的 $T$-元的个数是有限的,总可以在多项式的时间内判定它的元素是否被标记或属于某个最大无圈子网.算法 3.1.2 是 NP 类算法,其原因在于步骤④.逻辑电路的知识告诉我们,寻找高维最优覆盖解问题到目前为止仍然是一个 NP 难问题.尽管如此,RM-LC 化简方法仍可用于诸如无冲突的工作流逻辑网,或者是一些满足限制条件的网的局部化简.其效率显然比传统上采用将网结构与化简规则进行逐级比对要高.

## 4 应用实例

下面给出一个在电子政务中应用的简化的完整实例来说明 RM-LC 化简方法.

例 1:修改计划流程:项目组成员提出自己的计划修改申请,然后在组长的监督下进行修改;修改后,若外部条件提出新的更改要求,需根据新的要求做进一步修改;最后,在记录员的监督下填写修改记录,由记录员将修改后计划存档(如图 9 所示).

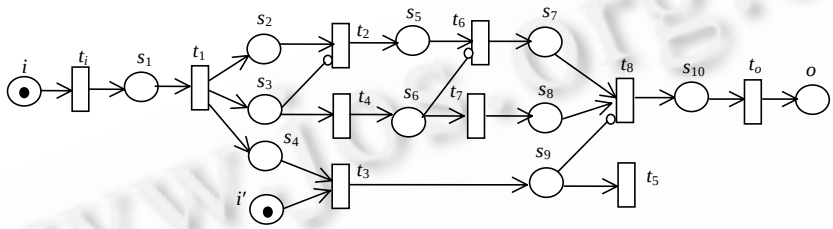


Fig.9 The example 1

图 9 例 1

模型中各变迁含义为:$t_i$ 表示开始;$t_1$ 表示项目组成员提出修改项目计划申请;$t_2$ 表示提出申请者本人修改计划;$t_3$ 表示新的外部条件对该计划提出变更要求;$t_4$ 表示项目组长监督;$t_5$ 表示根据新的外部条件对计划做修改;$t_6$ 表示填写修改记录;$t_7$ 表示记录员监督;$t_8$ 表示记录员将修改的计划存档;$t_o$ 表示结束.网 $N$ 是一个无圈网,$S_I$={$i$,$i'$},$S_O$={$o$}.下面给出化简程序的部分源码:

```
Matrix NetReduction(Matrix C*)                    //输入模型的修正关联矩阵 C*
{Set_acyclicSubnets=DivideNet(C*);                //划分出 C*所有最大无圈子网,并存储在集合 Set_acyclicSubnets
                                                  中,例 1 是一个无圈网,只有一个最大无圈子网
  DO {
     C_i* =Get(Set_acyclicSubnets);               //从集合 Set_acyclicSubnets 中提取出最大无圈子网 C_i*
     Set_acyclicSubnets=Set_acyclicSubnets− C_i*;
     DivideSubNet( C_i* ,S_I,S_O);                //对最大无圈子网 C_i* 进行输入、输出及中间部分的划分
     TransMatrix( C_i* ,S_O-Logic, S_Tdead−Logic );  //利用矩阵的初等变换完成相应变迁序列的发生,最终所得
                                                  信息表达式存放于 S_O-Logic, S_Tdead−Logic 中
     S_O-New=Q-M(S_O-Logic);                      //调用逻辑电路 Q-M 函数化简 S_O-Logic,结果存于 S_O-New 中
     S_Tdead−new =Q-M( S_Tdead−Logic );           //调用 Q-M 函数化简 S_Tdead−Logic ,结果存于 S_Tdead−new 中
     ReconstructExpression( C_i* ,S_O-New, S_Tdead−new ); //S_O-New, S_Tdead−new 中表达式还原回 C_i* 并对其进行修正
     RecoveryMatrix(C*,  C_i* );                  // C_i* 还原回 C*并对其进行修正
   }
  While (Set_acyclicSubnets!=Null);
  Output (C*);                                    //输出化简后的修正关联矩阵 C*
}
```

化简后的 $C^*$还原成网如图 10 所示.事实上,所有的化简规则均可借助矩阵及矩阵的初等变换完成.受篇幅

所限,本文不再讨论这些技术问题.

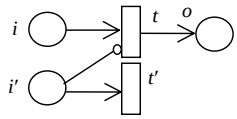


Fig.10 The net after reduction

图 10 化简后的网

## 5 结束语

(1) 本文利用已成熟并被广泛应用的逻辑电路及逻辑代数得到了一个新的、与传统思路完全不同的化简方法,并给出了可实现的算法.该方法首先将传统化简方法的应用范围扩展到带抑止弧的网;其次,对某些满足特殊限制条件的网的局部化简效率比传统的方法要高.从例 1 可以看出,对于一个无圈网,只需执行算法 3.1.1 及算法 3.1.2 各一次即可完成对整个网的化简工作.

(2) 将本文提出的化简方法与传统的 Murata 等人的代表性工作作比较,见表 1.

(3) 以后的工作是解决一般网的化简问题.

**Table 1** Comparison between RM-LC and classical reduction methods

表 **1** RM-LC 化简方法与传统方法的比较

| | Type of net | Corresponding reduction methods | Difference | Efficiency | Net with inhibitor arcs | Reduce for cycle |
|---|---|---|---|---|---|---|
| Murata | Ordinary net | Present six rules basic operations: Fusion or elimination of places and transitions | Stepwise reduction | Low, difficult to generate algorithm | No | Yes |
| RM-LC | Ordinary net without conflict | Present four steps for maximal acyclic subnet. Basic operations: Logic algebra | Area reduction | Finish the reduction maybe only ones. Easy to generate algorithm | Yes | No |

致谢 在此,向中国科学院数学与系统科学研究院数学研究所的陆维明研究员、中国科学院软件研究所的焦莉研究员表示感谢,他们给本文提出了重要的修改意见.同时,感谢审稿人详细的审稿及有益的建议.

**References**:

[1] Reisig W. Petri Nets: An Introduction. Berlin, Heidelberg: Springer-Verlag, 1985. 17−135.

[2] Yuan CY. The Theory and Application of Petri Nets. Beijing: Publishing House of Electronics Industry, 2005. 32−178 (in Chinese).

[3] Jiang CJ. The PN Theory of Discrete Event Dynamic System. Beijing: Science Press, 2000. 129−170 (in Chinese).

[4] Murata T, Koh JY. Reduction and expansion of live and safe marked graphs. IEEE Trans. on Circuit Systems, 1980,CAS-27(1): 68−70.

[5] Murata T. Petri nets: Properties, analysis, and applications. Proc. of the IEEE, 1989,77(4):541−580.

[6] Jiang CJ. Some reduction operations for a weighted T-graph. Journal of China Institute of Communications, 1994,15(2):97−102 (in Chinese with English abstract).

[7] van der Aalst WMP, Basten T. Inheritance of workflows: An approach to tackling problems related to change. Theoretical Computer Science, 2002,270(11):125−203.

[8] Mugarza JC, Camus H, Gentina JC, Teruel E, Silva M. Reducing the computational complexity of scheduling problems in Petri nets by means of transformation rules. In: IEEE Int'l Conf. on Systems, Man, and Cybernetics. 1998. 19−25.

[9] Ferscha A. Concurrent execution of timed Petri nets. In: Tew JD, Manivannan S, Sadowski DA, Seila AF, eds. Proc. of the Winter Simulation Conf. Orlando: Society for Computer Simulation International Press, 1994. 229−236.

[10] Lin C. On refinement of model structure for stochastic Petri nets. Journal of Software, 2000,11(1):104−109 (in Chinese with English abstract).

[11] Lin C, Qu Y, Zheng B, Tian LQ. An approach to performance equivalent simplification and analysis of stochastic Petri nets. ACTA ELECTRONICA SINICA, 2002,30(11):1620−1623 (in Chinese with English abstract).

[12] Xu AG, Jiang CJ. The reduction operations and their properties for P/T nets. Journal of Software,1997,8(7):493−504 (in Chinese with English abstract).

[13] Li JQ, Fan YS. Research of Petri nets based workflow model reduction methods. Information and Control, 2002,30(6):492−497 (in Chinese with English abstract).

[14] Zhou JT, Shi ML, Ye XM. A method for semantic verification of workflow processes based on Petri nets reduction technique. Journal of Software, 2005,16(7):1241−1251 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/16/1241.htm

[15] Palmer J, Perlman D, Wrote; Chen WK, Xu PP, Trans. Schaum's Outlines Introduction to Digital Systems. Beijing: Science Press, 2002. 21−68 (in Chinese).

[16] Liu BQ. Digital Circuit and System. Beijing: Tsinghua University Press, 1993. 15−198 (in Chinese).

[17] Schaefer DH. Petri net representations of computational and communication operators. In: Yakovlev A, ed. Hardware Design and Petri Nets. Boston: Kluwer Academic Publishers, 2000. 51−74.

[18] Yakovlev AV, Koelmans AM, Semenov A, Kinniment DJ. Modelling, analysis and synthesis of asynchronous control circuits using Petri nets. Integration, the VLSI Journal, 1996,21(3):143−170.

[19] Zhao BH, Jing L, Yan YG. Hardware implementation of Petri nets. Journal of Software, 2002,13(8):1652−1657 (in Chinese with English abstract). http://www.jos.org. cn/1000-9825/13/1652.pdf

[20] Wu ZH. An Introduction of Petri Nets. Beijing: China Machine Press, 2006. 197−212 (in Chinese).

[21] van der Aalsta WMP, ter Hofstede AHM. Verification of workflow task structures: A Petri-net-based approach. Information Systems, 2000,25(1):43−69.

## 附中文参考文献:

[2] 袁崇义.Petri 网原理与应用.北京:电子工业出版社,2005.32−178.

[3] 蒋昌俊.离散事件的动态系统的 PN 机理论.北京:科学出版社,2000.129−170.

[6] 蒋昌俊.加权 T-图的几种化简运算.通讯学报,1994,15(2):97−102.

[10] 林闯.随机 Petri 网模型的精化设计.软件学报,2000,11(1):104−109.

[11] 林闯,曲杨,郑波,田立勤.一种随机 Petri 网性能等价化简与分析方法.电子学报,2002,30(11):1620−1623.

[12] 许安国,蒋昌俊.P/T 网的化简运算及其性质研究.软件学报.1997,8(7):493−504.

[13] 李建强,范玉顺.基于 Petri 网化简方法的工作流模型验证.信息与控制,2002,30(6):492−497.

[14] 周建涛,史美林,叶新铭.一种基于 Petri 网化简的工作流过程语义验证方法.软件学报,2005,16(7):1241−1251. http://www.jos.org.cn/1000-9825/16/1241.htm

[15] Palmer J,Perlman D,著;陈文楷,徐萍萍,译.数字系统导论.北京:科学出版社,2002.21−68.

[16] 刘宝琴.数字电路与系统.北京:清华大学出版社,1993.15−198.

[19] 赵不贿,景亮,严仰光.Petri 网的硬件实现.软件学报,2002,13(8):1652−1657. http://www.jos.org.cn/1000-9825/13/1652.pdf

[20] 吴哲辉.Petri 网导论.北京:机械工业出版社,2006.197−212.

叶剑虹(1976－),男,福建厦门人,博士生,主要研究领域为分布式并行计算,Petri 网理论及应用.

孙世新(1940－),男,教授,博士生导师,主要研究领域为分布式并行计算,网格技术.

宋文(1956－),男,教授,CCF 高级会员,主要研究领域为 Petri 网理论及应用.