Petri 网的标注可达树

蒋昌俊　　吴哲辉*

(山东矿业学院应用数学与软件工程系,泰安 271019)

LABELED REACHABILITY TREE OF PETRI NETS

Jiang Changjun and Wu Zhehui

(*Shandong Institute of Mining and Technology*, *Taian* 271019)

Abstract The definition of labeled reachability tree of Petri nets is given in this paper. It is based on the concept of reachability tree of Petri nets. It can be proved that there exists an one correspondence between the set of Petri nets and the set of label reachability trees more over. An algorithm for transforming and in algorithm for transforming labeled reachability trees to their corresponding nets are given.

摘要 本文基于 Petri 网的可达树的概念,给出标注可达树定义,并且证明网 N 与其标注可达树是一一对应的.然后,我们给出了网 N 与相应的标注可达树的相互转换算法.

§0. 引　言

可达树分析方法是 Petri 网的一种重要分析手段.它对于 Petri 网的各种性态的分析起着十分重要的作用,如对有界性、可重复性、安全性、守恒性等等问题.每一个 Petri 网都有一个可达树,给定一个网,可以构造出相应的可达树,通过分析可达树来了解相应 Petri 网的性质.然而一个可达树可能对应若干个不同的 Petri 网.这样已知一个可达树,不能唯一确定一个相应的 Petri 网.因此我们对可达树附加一些注记,提出标注可达树,从而使得一个 Petri 网一一对应一个标注可达树.这样,从一个已知的标注可达树可以唯一地生成一个 Petri 网.基于这些,我们给出了网图与标注可达树的相互转换算法.在文[1]中,我们曾给出有界 Petri 网的状态图到 Petri 网图的一个有效转换算法.本文针对一般 Petri 网建立了标注可达树到相应网图的转换算法,从而讨论的范围较文[1]有了拓广.

---

* 本文1991年6月收到.本文受国家自然科学基金和中科院自动化所复杂系统控制开放实验室基金资助.作者**蒋昌俊**,副教授,1991年硕士毕业于山东矿业学院,现在中科院自动化所读博士,主要研究领域为算法研究,Petri 网理论及其应用,数字信号处理等.**吴哲辉**,教授,主要研究领域为 Petri 网理论及应用,计算机算法.

§1. 基本概念

本文中论及的是这样一个 Petri 网 $N=(P,T;I,O,M_0)$,其中:

(i)$P\cup T\neq\varnothing, P\cap T=\varnothing$;

(ii)$I\subseteq P\times T, O\subseteq T\times P$;

(iii)$M_0:P\to N^0$(N^0 为非负整数集);

记 $N_\omega=N^0\cup\{\omega\}$;$m=|P|, n=|T|$.

一个 Petri 网 $N=(P,T;I,O,M_0)$的结构可以用一个 $n\times m$ 的矩阵 $A=[a_{ij}]_{n\times m}$表示.其中:

$$a_{ij}=a_{ij}^{+}-a_{ij}^{-}\qquad a_{ij}^{+}=\begin{cases}1 & \text{当}(t_i,p_j)\in O\\ 0 & \text{否则}\end{cases}\qquad a_{ij}^{-}=\begin{cases}-1 & \text{当}(p_j,t_i)\in I\\ 0 & \text{否则}\end{cases}$$

为明显起见,有时也把 a_{ij}记作 $A=(t_i,p_j)$.

有关 Petri 网的引发规则,引发序列等其它一些概念本文不再赘述,请参见文[1,2].为讨论需要,给出可达树的定义.

定义 1:[3]设 $N=(P,T;I,O,M_0)$是一个 Petri 网,其可达树 $T(N)$定义为树(V,E),其中结点集 V 标以 $N_\omega^m=N_\omega\times N_\omega\times\cdots\times N_\omega$,E 为标以 T 的弧集,它由下列过程构造而成:

(i)根 root 标以 $M_0(M_0\in N_\omega)$

(ii)标有 $M\in N_\omega$ 的结点 v 没有后继结点当且仅当 root 到 v 的路上有一个结点 v'也标以 M 或者没有 $t\in T$,使得$\forall p\in P:p\in {}^{\cdot}t\to M(p)\geq1$;

(iii)若标有 M 的 v 不满足条件(ii),则对于使得$\forall p\in P:p\in {}^{\cdot}t\to M(p)\geq1$ 的所有 t 都使 v 产生一个后继结点 v',其中弧(v,v')标以 t,记作 $t/\!/(v,v')$,v'标以由下法确定的 M';若从 root 到 v'的路上有标以 M''的结点 v''满足$\forall p\in P:M''(p)\leq M(p)+A(t,p)$,则对所有满足 $M''(p)<M(p)+A(t,p)$的 $p\in P$,有 $M'(p)=\omega$,否则 $M'(p)=M(p)+A(t,p)$.其中 ω 满足这样的性质:对任意常数 a:$\omega\pm a=\omega, a<\omega, \omega\leq\omega$.

§2. 标注可达树及其性质

标注可达树是在可达树基础上对含 ω 的标识作进一步刻划和对自环做一些注记而得到的.值得注意的是:在标注可达树中,ω 参加的"±"是形式运算,得到的是形式表达式,因而对任何常数 $a\neq0$,满足 $\omega\pm a\neq\omega$.尽管这与前面规则相反,但它们是在两个不同的阶段,分别执行不同的含义,这样的处理不会导致错误的结论.

在给出标注可达树概念之前,需要定义一种逻辑运算"⊙".

定义 2:设 α,β 或者是非负整数,或者是含 ω 的表达式,定义:

$$\alpha\odot\beta=\begin{cases}1 & \text{当 }\alpha,\beta\text{ 均为含 }\omega\text{ 的表达式},\\ 0 & \text{否则}\end{cases}$$

定义 3:设 $T(N)=(V,E)$是 Petri 网 N 的可达树,令网 N 的标注可达树 $T'(N)=(V,E,flag)$.其中 $T'(N)$中的各结点标识由 $T(N)$中对应结点的标识经下面(1)、(2)过程得到,flag 如(3)给出.

(1)令 $M'_0=M_0$(根结点 v_0 的标识);

(2)对$\forall$ $v_1,v_2\in V,(v_1,v_2)\in E,t_j/\!/(v_1,v_2),M_1[t_j>M_2$,这里 M_1,M_2 是 T(N)中分别对应 v_1,v_2 的标识.对每个 $p_i\in P$,若 $M_1(p_i)\odot M_2(p_i)=0$,则令 $M'_2(p_i)=M_2(p_i)$,否则令:

$$M'_2(p_i)=\begin{cases}M'_1(p_i)+1 & \text{当}(t_j,p_i)\in O\wedge(p_i,t_j)\notin I;\\ M'_1(p_i)-1 & \text{当}(p_i,t_j)\in I\wedge(t_j,p_i)\notin O;\\ M'_1(p_i) & \text{否则}\end{cases}$$

其中 M'_1,M'_2 是 T′(N)中分别对应 v_1,v_2 的标识,且 $M'_1[t_j>M'_2$.

(3)flag:$M'(p)\to\{0,1\}$为 T′(N)的标注函数,定义为:①$flag(M'_0(p_i))=0$,对$\forall$ $p_i\in P$;②对 $v_1,v_2\in V$,及每一个 $p_i\in P$,若 $M'_1(p_i)=M'_2(p_i)\neq 0$ 且$(p_i,t_j)\in I\wedge(t_j,p_i)\in O$ 则令 $flag(M'_2(p_i))=1$;否则令 $flag(M'_2(p_i))=0$.

例如:图 1 是一个 Petri 网,它的可达树如图 2,它的标注可达树如图 3.

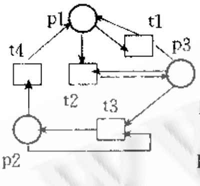

图1 Petri网N

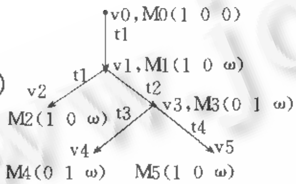

图2 N的可达树T(N)

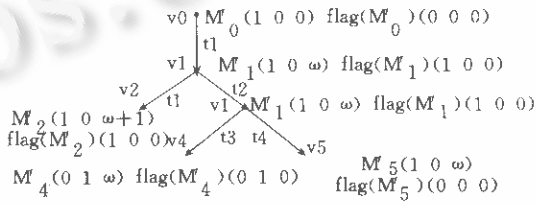

图3 N的标注可达树T′(N)

定理 1:对于一个 Petri 网 N,若(1)网 N 的每个变迁都至少有一次引发机会;(2)网 N 的每个位置的标志数都至少发生一次变化;则网 N 与它的标注可达树 T′(N)是一一对应的.

证明:(1)证网 N 唯一确定相应的 T′(N)

对于一个 Petri 网 N,由可达树生成算法的正确性确保生成唯一的可达树 T(N).根据定义 3 的规则(1)、(2)、(3)知 T′(N)由网 N 及 T(N)唯一确定,因此 T′(N)由网 N 唯一确定.

(2)证 T′(N)唯一地对应 N.

反证:假设某一标注可达树 T′(N)对应两个不同的网 N_1 及 N_2.

由于 N_1、N_2 都对应着 T′(N),而 T′(N)的结点标识 M 的维长$|P|$及边上旁标集的基数$|T|$是一定的(因为定理 1 的条件(1),(2)).因此易知$|P_1|=|P_2|$,$|T_1|=|T_2|$,且 T′(N)的根结点标识 $M'_{01}=M'_{02}$(M'_{0i}为 N_i 初始标识 i=1,2).因为 $N_1\neq N_2$,则存在 $t\in T_i,p\in P_i(i=1,2)$,使得$(t,p)\in O_1$ 但$(t,p)\notin O_2$ 或$(p,t)\in I_1$ 但$(p,t)\notin I_2$.不妨设$(t,p)\in O_1$ 但$(t,p)\notin O_2$.存在某个 $M'_{i1}\in R(M'_{01})$唯一对应着 $M'_{i2}\in R(M'_{02})$且 $M'_{i1}=M'_{i2}$而 $M'_{ij}[t>M'_{ij}(j=1,2)$(注:$R(M_0)$为 M_0 的可达集).若(t,p)在 N_1 的某自环中,而(t,p)不在 N_2 的相应自环中,根据定义 3 的(3)知的$\exists$ $M'_{ij}\in R(M'_{0j}),j=1,2$,使得 $flag(M'_{i1}(p))=1$ 但 $flag(M_{i2}(p))=0$,这里 $M'_{i1}=M'_{i2}$,从而 $flag(M'_{i1})\neq flag(M'_{i2})$,这样 T′(N)不唯一,与已知的一个 T′(N)矛盾!

若(t,p)均不在 $N_j(j=1,2)$的相应自环中,分三种情况讨论如下:

(i)如果 $M'_{ij}(p)$、$M''_{ij}(p)(j=1,2)$都不含 ω，由于$(t,p)\in O_1$ 而$(t,p)\notin O_2$，因此 $M''_{i1}(p)>M''_{i2}(p)$，这样 $M''_{i1}\neq M''_{i2}$，故 $T'(N)$不唯一，与已知的一个 $T'(N)$矛盾！

(ii)如果 $M'_{ij}(p)$不含 ω，而 $M''_{ij}(p)$含有 ω，$(j=1,2)$，根据定义 1(可达树生成算法)知：在$T(N)$中一定存在某个叶结点 $M_{ij}^{*\prime\prime}$及其直接前驱结点 $M_{ij}^{*\prime}$(也即 $M_{ij}^{*\prime}[t>M_{ij}^{*\prime\prime}$)，使得 $M_{ij}^{*\prime\prime}=M''_{ij}$且 $M_{ij}^{*\prime}=\omega(j=1,2)$，$M_{i1}^{*\prime}=M_{i2}^{*\prime}$. 由于 $M_{ij}^{*\prime}(p)\odot M_{ij}^{*\prime\prime}(p)=1$ 且$(t,p)\in O_1\wedge(p,t)\notin I_1$以及$(t,p)\notin O_2$，根据定义 3 的(2)可知在 $T'(N)$中 $M_{i1}^{*\prime\prime}(p)=M_{i1}^{*\prime}(p)+1$，而 $M_{i2}^{*\prime\prime}(p)=M_{i2}^{*\prime}(p)-1$(当$(p,t)\in I_2$)或 $M_{i2}^{*\prime\prime}(p)=M_{i2}^{*\prime}(p)$(当$(p,t)\notin I_2$)，又 $M_{i1}^{*\prime}=M_{i2}^{*\prime}$，因此 $M_{i1}^{*\prime}\neq M_{i2}^{*\prime\prime}$，从而 $T'(N)$不唯一，与假设的一个 $T'(N)$矛盾！

(iii)如果 $M'_{ij}(p)$，$M''_{ij}(p)$均含 ω，$(j=1,2)$，此种情况包含在(ii)中，从而导出矛盾！

故 $T'(N)$唯一地对应 N，定理得证.

§ 3. 网与标注可达树的相互转换算法

3.1 数据结构描述

3.1.1 标注可达树的数据结构

我们采用树的标准存贮结构，在这种存贮结构中，树的结点具有四个数组(如图 4)，其中：

1)mark ：ARRAY[1..m] OF integer

这里 m 为网 N 的位置个数，该数组用于存贮结点 v 的标识向量 M.

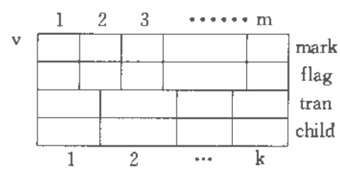

图4

2)flag ：ARRAY[1..m] OF bool

这里 m 同上，该数组用于存贮结点 v 的标注函数值.

3)child ：ARRAY[1..k] OF integer

这里 k 为标注可达树 $T'(N)$的度数，该数组用于存放该结点的儿子结点.

4)tran ：ARRAY[1..k] OF integer

这里 k 同上，该数组用于存放此结点对应状态下所有能引发的相应变迁.如对图 3 的标注可达树 $T'(N)$的存贮结构如图 5 所示.

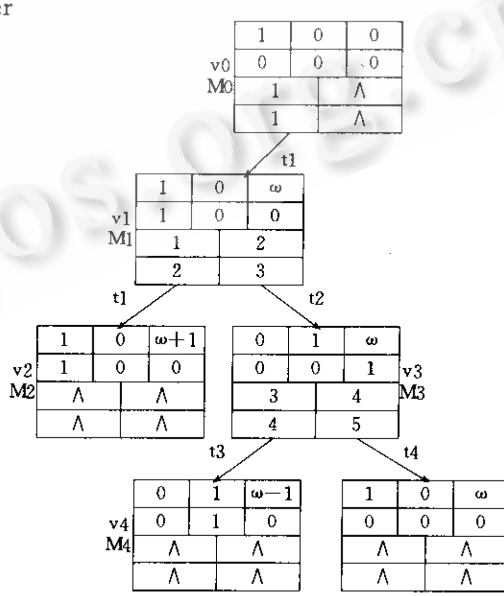

图5

3.1.2 网 N 的存贮结构

对于网 N 我们用二个邻接链表 PT 和 TP 来分别存贮网 N 的 I 集合和 O 集合(见文献[1]).

3.1.3 旁标集 T 的标志存贮

对于 $T'(N)$的旁标集 $T=\{t_1,t_2,\cdots,t_n\}$的处理过没有的标志，我们用

数组 T[1..n]来存贮,定义为:

$$T[j]=\begin{cases}1 & \text{当旁标为 } t_j \text{ 的边处理过,}\\ 0 & \text{否则}\end{cases}$$

此外,处理过程中还用到队 q、指针等一些数据结构.

3.2 网到标注可达树的转换算法

本文的标注可达树生成算法是基于可达树生成算法的,而从网生成可达树过程由定义1给出,因此下面我们描述一下可达树到标注可达树的生成过程.

算法 1:T(N)到 T′(N)的转换

输入:已存贮好的 T(N)及网 N.

输出:T′(N)

(1)T(N)的根结点入队 q,置 $flag(M_{根}(p_i)):=0, i=1,2,\cdots,m.$

(2)当队 q 为空时结束,否则转(3);

(3)将队头元素出队,并放到 v′;

(4)$v:=v'$;

(5)若 v 的所有儿子结点都访问过或 v 是叶结点则转(2),否则按层次顺序访问下一直接可达结点 v′;

(6)将 v′入队,并记(v,v′)的旁标为 t_j;

(7)记 v、v′对应的标识分别为 M、M′,则对 $p_i\in P(i=1,2,\cdots,m)$,若 $M(p_i)$含 ω,则

$$M'(p_i):=\begin{cases}M(p_i)+1 & \text{当}(t_j,p_i)\in O\wedge(p_i,t_j)\notin I\\ M(p_i)-1 & \text{当}(p_i,t_j)\in I\wedge(t_j,p_i)\notin O\\ M(p_i) & \text{否则}\end{cases}$$

若 $M(p_i)$不含 ω,则 $M'(p_i):=M'(p_i)$;

(8)对于 $p_i\in P(i=1,2,\cdots,m)$,若 $M(p_i)=M'(p_i)\neq 0$ 且 $(p_i,t_j)\in I\wedge(t_j,p_i)\in O$,则置 $flag(M'(p_i)):=1$,否则 $flag(M'(p_i)):=1$;

(9)转(5).

3.3 标注可达树到网的转换算法

算法 2:T′(N)到 N 的转换

输入:T′(N)

输出:N

(1)将根结点 v_0 入队 q;

(2)当 q 为空时结束,否则转(3);

(3)将队头元素出队放到 v 中;

(4)$j:=1$;

(5)当 $j>k$ 时转(2),否则转(6);

(6)如果 v 的第 j 个儿子结点 v↑.child(j)为空,则转(18),否则转(7);

(7)u:=v↑.child(j);

(8)t:=v↑.tran(j);

(9)如果 T[t]=1,则转(18),否则转(10);

(10)T[t]:=1;

(11)i:=1;

(12)当 i>m 时转(17),否则转(13);

(13)如果 v↑.mark(i)−u↑.mark(i)中含有 ω,则转(17),否则转(14);

(14)如果 v↑.mark(i)−u↑.mark(i)>0,则将 t 元素插到 PT 邻接表中 p_i 所拉的 p−t 链中去(若原先 t 不在其中的话),i:=i+1 转(12),否则转(15);

(15)如果 v↑.mark(i)−u↑.mark(i)=0,则将 t 元素插到 PT 邻接表中 p_i 所拉的 p−t 链中去(若原先 t 不在其中的话),将 p_i 元素插到 TP 邻接表中 t 所拉的 t−p 链中去(若原先 p_i 不在其中的话),i:=i+1 转(12);否则转(16);

(16)将 p_i 元素插到 TP 邻接表中 t 所拉的 t−p 链中去(若原先 p_i 不在其中的话),i:=i+1,转(12);

(17)将 u 进队 q;

(18)j:=j+1,转(5).

3.4 算法的正确性证明与复杂性分析

3.4.1 算法的正确性证明

定理 2:设 $T'(N)=(V,E,flag)$ 为网 N 的标注可达树,对于 $v,v'\in V$,记 $t_j /\!/ (v,v')$;M、M′分别是 v、v′对应的标识,若∃ $p_i\in P$,使得:$M(p_i)$ 为一常数,而 $M'(p_i)=\omega$,则必∃ $v'',v'''\in V$,使得 $t_j /\!/ (v'',v''')$,且 $M''(p_i)=\omega+\alpha_i$,$M'''(p_i)=\omega+\alpha_i+b_i$,其中:M″、M‴分别为 v″、v‴对应的标识,$\alpha_i$ 为一常数,$b_i\in\{-1,0,1\}$.

证明:由已知条件 $v、v'\in V$,$t_j /\!/ (v,v')$,∃ $p_i\in P$,使得 $M(p_i)$ 为一常数,而 $M'(p_i)=\omega$,则由定义 1 的(iii)可知,在 T(N)(在 T′(N)中也是如此)的根到 v 的路上必存在 u′,使得:$M_{u'}(p_i)>M'(p_i)$,这样 $u'\xrightarrow{t_{u'}}\cdots\rightarrow v\xrightarrow{t_j}v'$,且在 T(N)(或 T′(N))中存在 v″、v‴,使得 $u'\xrightarrow{t_{u'}}\cdots\rightarrow v\xrightarrow{t_j}v'\xrightarrow{t_{u'}}\cdots\rightarrow v''\xrightarrow{t_j}v'''$,在 T(N)中,$M''(p_i)=M'''(p_i)=\omega$,而根据定义 3 的(2)可知:在 T′(N)中 $M''(p_i)=\omega+\alpha_i$,$M'''(p_i)=\omega+\alpha_i+b_i$,其中 $|\alpha_i|\leqslant$ 序列 $t_{u'},\cdots,t_j$ 的长度. 而 $b_i\in\{-1,0,1\}$. 证毕.

定义 4:称 V_0 是 $T'(N)=(V,E,flag)$ 的基本点集当且仅当:

(1)V_0 是 V 的子集;

(2)∀ $v'',v'''\in V_0$,且 $v''\neq v'''$ 及∀ $p_i\in P$,都有 $M'''(p_i)-M''(p_i)$ 不含 ω;

(3)对∀ $t_j\in T$,存在且只存在唯一对应的 $v''、v'''\in V_0$,使得 $t_j /\!/ (v'',v''')$.

定义 5:在 T′(N)中,由 V_0 生成的森林称为 T′(N)的基本子林,记作 ST′(N).

定理 3:在 T′(N)中,ST′(N)所确定的 Petri 网就是 T′(N)所对应的 Petri 网.

证明:我们知道一个 Petri 网 N 可以由 $^{\cdot}T$ 及 $T^{\cdot}$ 唯一确定(这里 $^{\cdot}T=I$,$T^{\cdot}=O$),其中 $^{\cdot}T=\bigcup_{t\in T}{}^{\cdot}t$,$T^{\cdot}=\bigcup_{t\in T}t^{\cdot}$,而由定义 4 的(2)可知∀ $t_j\in T$,$^{\cdot}t_j$ 及 $t_j^{\cdot}$ 可唯一确定,由定义 4 的(3)可知:$\{t\,|\,t /\!/ (v'',v'''),v'',v'''\in V_0,v''\neq v'''\}=T$,这样 ST′(N)即可确定 T′(N)所对应的 Petri 网 N.

定理 4:对于一个 T′(N). 至少存在一个 ST′(N).

证明:由定理 2 可保证定义 4 中(2),对∀ $t_j\in T$,t_j 是 T′(N)中至少某条边 e 的旁标. 这样定义 4 中的(3)得到保证,根据定义 4、定义 5 可知 ST′(N)在 T′(N)中存在.

定理 5：设 Petri 网 $N=(P,T;I,O,M_0)$ 的标注可达树是 $T'(N)=(V,E,flag)$，则按算法 2（$T'(N)$ 到 N 的转换算法）构造的网就是 N.

证明：设 TP、PT 分别存放生成网 N 的 (t,p) 弧和 (p,t) 弧.

(1)我们只需证 TP 表中存放的所有生成元素等于 O 集，PT 表存放的正好是 I 集.

不失一般性，设在 TP 邻接表中 t_a 对应的 $t-p$ 链表上有一个 P_b 结点，根据算法 2 知：结点 $v,v'\in V$，$(v,v')\in E$，$t_a /\!/ (v,v')$，对应 v,v' 的标识为 M、M'，则 $M[t_a>M'$，且使得：① $M(p_b)-M'(p_b)<0$ 或② $M(p_b)-M'(p_b)=0$ 且 $flag(M'(p_b))=1$.

根据状态方程 $M'=M+A^Tx$ 有

$$M'(p_b)-M(p_b)=A(a,b)$$

对于①，由于 $M(p_b)-M'(p_b)<0$，从而 $A(a,b)>0$，由邻接矩阵定义便知：$(t_a,p_b)\in O$.

对于②，由于 $M(p_b)-M'(p_b)=0$ 且 $flag(M'(p_b))=1$，从而：$A(a,b)=0$ 且 $flag(M'(p_b))=1$，根据定义 3 的(3)便知 t_a 与 p_b 之间有一自环，从而有 $(t_a,p_b)\in O$. 从而(1)得证.

(2)要证网 N 中，$\forall\ (t_a,p_b)\in O$，在 TP 邻接表中 t_c 对应的 $t-p$ 链表上有 p_d 结点.

不失一般性，设 $(t_c,p_d)\in O$，则 $A^+(c,d)=1>0$.

(i)若 $(p_d,t_c)\notin I$，则 $A^-(c,d)=0$，即有 $M'(p_c)-M(p_c)=1$，也即 $M(p_c)-M'(p_c)=-1<0$，按算法 2 中的(16)有：在 TP 邻接表中 t_c 对应的 $t-p$ 链表中增加一个 p_d 结点（若原先不在其中的话）.

(ii)若 $(p_d,t_c)\in I$，则 $A^-(c,d)=1$，从而 $A(c,d)=0$，此时 $flag(M'(p_d))=1$，即 t_c,p_d 间有一自环. 按算法 2 中的(15)知：在 TP 邻接表中 t_c 对应的 $t-p$ 链表中增加一个 p_d 对应的结点（若原先不在其中的话）. 这就证明了(2).

再根据算法 2 中对 $T[t]=0$ 做处理，从而决定了由 $T'(N)$ 的一个子林 $ST'(N)$ 生成网 N，对 $ST'(N)$ 结点集是 $T'(N)$ 结点集 V 之子集，而 V 有限，故算法 2 有限步终止.

3.4.2 算法复杂性估计

定理 6：设 Petri 网 $N=(P,T;I,O,M_0)$，它的标注可达树 $T'(N)=(V,E,flag)$；记 $|P|=m$，$|T|=n$，则算法 2 的时间复杂性是 $O(mn.\max(n^2,m^2))$.

证明：算法 2 是根据 $T'(N)$ 的 $ST'(N)$ 生成网 N 的，算法执行过程中需要对 $ST'(N)$ 的 n 条边的每一条做一个 m 维的向量减法，因此共要做 nm 次标量减法. 而对每个标量减结果至多做两次 TP 或 PT 的链表上增加结点工作，而这样的工作的最坏情况下时间复杂性是 $O(m^2)$ 或 $O(n^2)$，故算法 2 的最坏时间复杂性是 $O(mnm^2)$ 或 $O(mnn^2)$，也即 $O(mn\max(m^2,n^2))$.

参考文献

1　吴哲辉、蒋昌俊，有界 Petri 网的可达图到网图的转换算法，《软件学报》，No. 1，1992.

2　吴哲辉，有界 Petri 网的活性与公平性的分析与实现，《计算机学报》，No. 1，1989.

3　陆维明、Merceron，A.，冻结标志——考核由 Petri 网模拟的分布式系统的一种途径，《中国科学》(A 辑)，No. 2，1987.

4　J. L. Peterson，Petri Net Theory and the Modelling of Systems，Prentice—Hall，Inc.，Englewoal Cliff，N. J. U. S. A.，1981.

5　B. Nash，Reachability Problems in Vector Addition Systems，The American Mathematical Monthly，Vol. 80，No. 3，1973.