

有界Petri网的可达图 到网图的转换算法

吴哲辉 蒋昌俊

(山东矿业学院应用数学与软件工程系, 泰安, 271019)

AN ALGORITHM FOR CONSTRUCTING A BOUNDED PETRI NET ACCORDING TO ITS REACHABLE MARKING GRAPH

Wu Zhehui and Jiang Changjun

(Department of Applied Mathematics and Software Engineering,

Shandong Institute of Mining and Technology, Taian, 271019)

ABSTRACT

An algorithm for constructing a bounded Petri Net according to its reachable marking graph is presented in this paper. A proof of correctness and a valuation of complexity for the algorithm are also given. We show that the algorithm is of polynomial complexity, hence it is a valid algorithm.

摘 要

本文给出了有界Petri网的可达标识图到网图的转换算法, 对算法的正确性与复杂性分别进行了证明和估计, 结果表明该算法是一个多项式算法, 因而是有效的。

§ 1. 引 言

Petri网是系统模拟与分析的一种有效工具。Petri网研究的发展推动了人们去研究Petri网与自动机的关系, 文[1]已经阐明“一个有界Petri网的可达图与相应的有穷自动机状态转换图同构”。从此, 意义上来说一个有界Petri网就对应着一个有穷自动机, 这样有界Petri网的某些问题可转换为相应的有穷自动机的问题加以研究; 反之, 对于有穷自动机的某些问题的研究也可以化归为相应的有界Petri网的问题来研究。然而, 这种转化需要有效算法来完成。从有界Petri网的网图到它的可达图转换算法已经建立, 这可以从文献[2, 3]中见到。但是从有界Petri网的可达图到它的网图转换算法迄今为止尚未见到。本文正是给出这个转换的一个有效算法, 目的在于完善有界Petri网的网图与它的可达图之间的相互转换。

在本文中, 我们讨论的对象是有界P/T 网. P/T 网是一个六元组

$$N = (P, T; I, O, W, M_0)$$

其中 $P \cup T \neq \Phi, P \cap T = \Phi; I \subseteq P \times T, O \subseteq T \times P; W: I \cup O \rightarrow N^0$ (N^0 为非负整数集); $M_0: P \rightarrow N^0$. 并约定 $\forall p \in P, \forall t \in T, (p, t) \in I \rightarrow (t, p) \notin O$, 也就是说我们把讨论限制在纯网的范围内. 并假设 $(P, T; I, O)$ 构成一个连通图.

把从 M_0 可达的全部标识的集合记为 $R(M_0)$. 若存在正整数 k , 使 $\forall M \in R(M_0), \forall p \in P$ 都有: $M(p) \leq k$, 则称 P/T 网 N 为有界的. 显然对于有界的 P/T 网 $N, R(M_0)$ 是一个有限集.

假设 $|P| = m, |T| = n$, 一个 P/T 网可以用一个 n 行 m 列的整数矩阵 $A = [a_{ij}]_{n \times m}$ 来表示. 其中

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

有界 P/T 网 N 的一个可达标识图(简称可达图) 是一个以 $R(M_0)$ 为结点集的有向图 $RMG(N) = (R(M_0), E)$. 它的每一个结点 $M \in R(M_0)$ 对应着一个标识向量 $M = (M(p_1), M(p_2), \dots, M(p_m))^T$. 对 $\forall M_i, M_j \in R(M_0)$, 若 $\exists t \in T$, 使得 $M_i(t) > M_j(t)$, 则在 $RMG(N)$ 中, $(M_i, M_j) \in E$, 并对有向边 (M_i, M_j) 旁标以 t . 记为 $(M_i, M_j)/t$.

由于 N 是有界的, $RMG(N)$ 是一个有限图, 而且每个结点所对应的标识向量的各个分量均为非负整数, 也就是说, 在 $RMG(N)$ 的各个结点所对应的标识向量中, 不会出现无界量 ω .

§ 2. 数据结构的选择

由于我们的算法是用类Pascal 语言描述的, 因此有必要先说明一下选择的数据结构.

1. RMG 图的存贮

对于 $RMG(N) = (R(M_0), E)$ 图, 我们用一个 M -邻接表(见图1) 存贮. 其中①数组 $first$ 为表头数组, 用于存放各条链表的首结点地址. ②链表用于存放有邻接关系的结点; 例: 若 $(M_i, M_j) \in E$, 其中 $M_i, M_j \in R(M_0)$, 则在 M -邻接表中 M_i 结点所拉的链表中必有 M_j 对应的链结点. 每个链结点有三个域, 分别是:

- i) mark 域, 用于存放结点标号 M .
- ii) tran 域, 用于存放弧上的旁标 t .
- iii) link 域, 为链域.

2. 标识集的存贮

设 $|R(M_0)| = k$, 并对 $R(M_0)$ 中的标识编号为: $R(M_0) = \{M_0, M_1, \dots, M_{k-1}\}$, 我们可

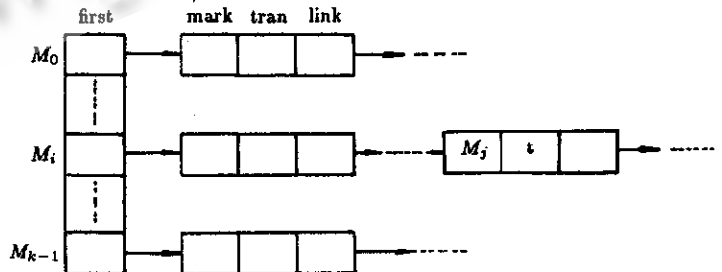


图1 M-邻接表

以用一个 m 行 k 列矩阵 $H = (M_0, M_1, \dots, M_{k-1})$ 来存贮 $RMG(N)$ 中全部标识的值, 其中 H 的每一列对应着一个标识向量.

3. 边标记的存贮

对于 $RMG(N) = (R(M_0), E)$ 图中的边集 E , 我们用一个 $r(r = |E|)$ 维的向量 e 来存贮, 并且定义

$$e((M_i, M_j)) = \begin{cases} 1 & \text{当边}(M_i, M_j) \text{ 访问过} \\ 0 & \text{否则} \end{cases}$$

其中 $(M_i, M_j) \in E$, 向量 e 主要作为遍历 $RMG(N)$ 中哪些边被访问过的标识.

4. 生成的 P/T 网图的存贮

对于生成的 P/T 网 N , 我们选用二个邻接表来存贮.

① PT 邻接表(见

图2)

用于存贮网 N 中的 $I \subseteq P \times T$ 的元素. 它有一个表头数组 FPT , 用于存放各条 $p-t$ 链的首结点的地址. 链表用于存放 (p, t) 弧及其值. 每个链结点有三个域, 分别是:

i) t 域, 用于存放 t - 结点标号 t_j .

ii) w 域, 用于存放 $(p, t) \in I$ 弧上的权值 $w(p, t)$.

iii) $link$ 域, 链结域.

例: 若 $(p_i, t_j) \in I$,

则在 PT 邻接表中的 p_i 所拉的 t - 链中必有存放 t_j 的链结点(见图2).

② TP 邻接表(见

图3)

用于存放网 N 中的 $O \subseteq T \times P$ 的元素. 结构与①类似, 这里不再赘述.

5. 另外, 在操作过程中还用到队 Q 及指针 S, R 等以及过渡集合 A 和 B .

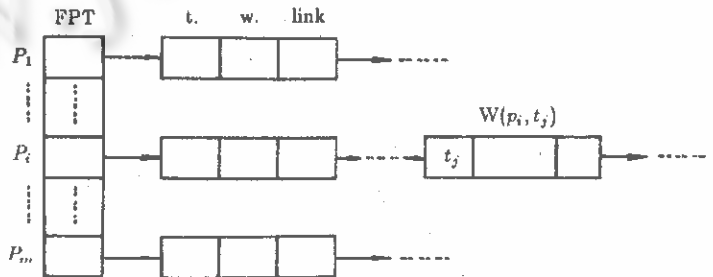


图2 PT 邻接表

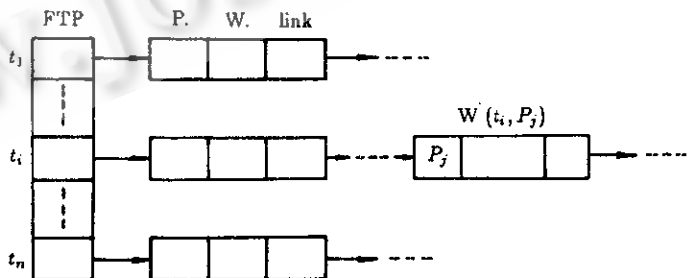


图3 TP 邻接表

§ 4. 算法思想与描述

为清楚起见, 我们先给出算法的基本思想, 然后用类Pascal语言描述算法.

4.1 基本思想

算法从初始标识 M_0 开始, 按照广度优先方式. 遍历 $RMG(N)$ 的每条边一次. 如遍历边 (M_1, M_2) 时(若 (M_1, M_2) 的旁标是 t_x 且 $M_1(t_x) > M_2$), 则算法中构造集合:

$$A = \{j | M_2(p_j) - M_1(p_j) > 0, p_j \in P\},$$

$$B = \{k | M_2(p_k) - M_1(p_k) < 0, p_k \in P\}.$$

则对 $\forall j \in A$, P/T 网 N 中一定有一条 (t_x, p_j) 的有向弧, 其权是 $W(t_x, p_j) = M_2(p_j) - M_1(p_j)$. 因以检查 P/T 网的 TP 邻接表中的 t_x 所拉的 $t-p$ 链中是否已有 p_j 链结点, 若有则转去检查 A 集中下一元素, 否则将 p_j 挂到 t_x 所拉的 $t-p$ 链上, 也转去检查 A 集中下一元素. 直至 A 集中元素检查完毕转去检查 B 集元素, 对 $\forall k \in B$, P/T 网图中一定有一条 (p_k, t_x) 的有向弧, 其权是 $w(p_k, t_x) = M_2(p_k) - M_1(p_k)$, 因此检查 P/T 网的 PT 邻接表中的 p_k 所拉的 $p-t$ 链中是否已有 t_x 链结点, 若有则转去检查 B 集中下一元素, 否则将 t_x 挂到 p_k 所拉的 $p-t$ 链上, 也转去检查 B 集中下一元素. 直至 B 集中元素检查完毕, 置 $e(M_i, M_j) = 1$, 转去遍历 M_2 所关联的边 (M_2, M_3) , (注意 $RMG(N)$ 中 M_2 可能关联若干条边, 按照广度优先次序 (M_2, M_3) 是其中的第一条, 这种顺序是靠队来实现的). 重复做上述工作, 直到 $RMG(N)$ 中所有边都被遍历一次(即 e 的分量全1)为止. 这样得到的 PT 邻接表和 TP 邻接表完全确定了 P/T 网 N .

4.2 算法描述

步骤1: 将 M_0 的地址进入队 Q 中,

步骤2: 当队 Q 为空时结束; 否则做 $1^0 - 4^0$,

1^0 , 将队 Q 中头元素出队且放入 S 中,

2^0 , $R := S$,

3^0 , $S := S \uparrow .first$,

4^0 , 当 S 为空时转步骤2, 否则做(1).

(1) 如果 $e(R \uparrow .mark, S \uparrow .mark) = 1$ 则转(1); 否则做(2)---(7).

(2) 将 $s \uparrow .mark$ 的地址进入队 Q 中,

(3) 置 $e(R \uparrow .mark, S \uparrow .mark) := 1$,

(4) 构造集合 $A = \{j | j \text{ 是 } (S \uparrow .mark) - (R \uparrow .mark) \text{ 中为正的 } \text{分量下标}\}$, $B = \{k | k \text{ 是 } (S \uparrow .mark) - (R \uparrow .mark) \text{ 中为负的 } \text{分量下标}\}$.

(5) 对 $\forall j \in A$, 若 $S \uparrow .tran$ 所拉的 $t-p$ 链中没有 p_j 链结点, 则将 p_j 挂到该 $t-p$ 链上, 其中将 $(S \uparrow .mark) - (R \uparrow .mark)$ 的第 j 个分量值送到 p_j 链结点的 W 域. 当 A 中元素检查完毕转(6).

(6) 对 $\forall k \in B$, 若 p_k 所拉的 $p-t$ 链中没有 $S \uparrow .tran$, 则将 $S \uparrow .tran$ 挂到该 $p-t$ 链中, 其中将 $(S \uparrow .mark) - (R \uparrow .mark)$ 的第 k 个分量值送到 $s \uparrow .tran$ 链结点的 W 域. 当 B 中元素检查完毕转(7).

(7) $S := S \uparrow .link$, 转 4^0 .

§ 5. 正确性证明与复杂性估计

首先我们对算法的正确性进行证明.

定理1: 设有界P/T网 $N = (P, T; I, O, W, M_0)$ 的可达图是 $RMG(N) = (R(M_0), E)$, H 是 $R(M_0)$ 对应的标识矩阵, 若满足条件

(i) $\forall p \in P, \exists M_i, M_j \in R(M_0)$, 使得 $M_i(p) \neq M_j(p)$

(ii) $\cup\{t|(M_i, M_j)//t\} = T. (M_i, M_j) \in E$

则算法从 $RMG(N)$ 生成的P/T网就是 N .

证明: 算法执行到(4)、(5)、(6)时, 将满足条件的 j 和 k 分别做:

① P_j 挂到 $S \uparrow .tran$ 所拉的 $t-p$ 链中(若该链中没有 P_j).

② $S \uparrow .tran$ 挂到 P_k 所拉的 $p-t$ 链中(若该链中没有 $S \uparrow .tran$).

若令 $M = R \uparrow .mark, M' = S \uparrow .mark$.

由于 $R \uparrow .mark, S \uparrow .mark$ 是算法遍历 $RMG(N)$ 得到的, 而且 $(R \uparrow .mark, S \uparrow .mark) \in E$, 因此有

$$(M, M') \in E, \quad M, M' \in R(M_0)$$

$$(M, M')//t = S \uparrow .tran \stackrel{\text{记为}}{=} t_x \in T$$

而且遍历的次序是从 M 到 M' , 即 $M(t_x > M')$. 根据状态方程有 $M' - M = A^T t_x$
即:

$$\begin{bmatrix} M'(p_1) - M(p_1) \\ \vdots \\ M'(p_i) - M(p_i) \\ \vdots \\ M'(p_m) - M(p_m) \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & a_{x1} & \cdots & a_{n1} \\ \vdots & & \vdots & & \vdots \\ a_{1i} & \cdots & a_{xi} & \cdots & a_{ni} \\ \vdots & & \vdots & & \vdots \\ a_{1m} & \cdots & a_{xm} & \cdots & a_{nm} \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \text{--- } X$$

即:

$$\begin{bmatrix} \Delta M(p_1) \\ \vdots \\ \Delta M(p_i) \\ \vdots \\ \Delta M(p_m) \end{bmatrix} = \begin{bmatrix} a_{x1} \\ \vdots \\ a_{xi} \\ \vdots \\ a_{xm} \end{bmatrix}$$

若 $\Delta M(p_j) > 0$, 则 $a_{xj} = W(t_x, p_j) > 0$ 对 $j \in A$ 进行①操作, 而由 $W(t_x, p_j) > 0$, 我们知道网 N 中有一条权为 $W(t_x, p_j) = \Delta M(p_j)$ 的从 t_x 到 p_j 的有向弧. 故我们说算法中(5)所挂的链正是建立网 N 的从 t 到 p 的有向弧.

同理可证算法中(6)所挂的链正是建立网 N 的从 p 到 t 的有向弧.

下面证明, 在网 N 中, $\forall (t, p) \in O$, 在 TP 邻接表中, p 必在 t 所拉的链中; $\forall (p, t) \in I$, 在 PT 邻接表中, t 必在 p 所拉的链中.

不妨设 $(t_x, p_j) \in O \subseteq T \times P$, 且权为 $W(t_x, p_j) > 0$. 根据条件(i)我们知道 $\forall p \in P, p$ 中都有标记的变化. 而由条件(ii)知道 $\forall t \in T, t$ 都参与了引发.

若设 $M, M' \in R(M_0)$, $M(t_x > M'$ 且 $W(t_x, p_j) = M'(p_j) - M(p_j) > 0$, 根据算法中的(4)、(5)操作知道将 p_j 挂到 t_x 所拉的 $t-p$ 链中(若该链中原先不含 p_j) 且 W 域中放入 $\Delta M(p_j) = W(t_x, p_j)$. 这说明网 N 的任何一条 (t, p) 有向弧在 TP 邻接表中都能找到.

同理可证网 N 的任何 (p, t) 弧在 PT 邻接表中也都能找到.

因此算法通过 $RMG(N)$ 生成网 N .

由于网 N 是有界的, 所以 $R(M_0)$ 有限, E 也有限, 故算法能够终止.

综上所述, 算法正确性得证.

注1: 若去掉定理1中的条件(i)和(ii), 则由此算法生成的网 N' 一般情况下是网 N 的子网. 因为条件(i)和(ii)实际上反映了 $\forall t \in T$ 和 $\forall p \in P$ 都分别参与了引发和标记流动, 这样 $RMG(N)$ 中都有所表现, 因此通过算法将整个网图生成出来. 否则只生成网图中参与活动的那一部分.

注2: 实际上只要知道 $RMG(N)$ 中的满足条件(i)和(ii)的一个连通子图就可以生成网图. 因为网图的每个 t, p 结点的活动在这样的子图中都已反映出来了.

现在我们对算法的复杂性做一估计.

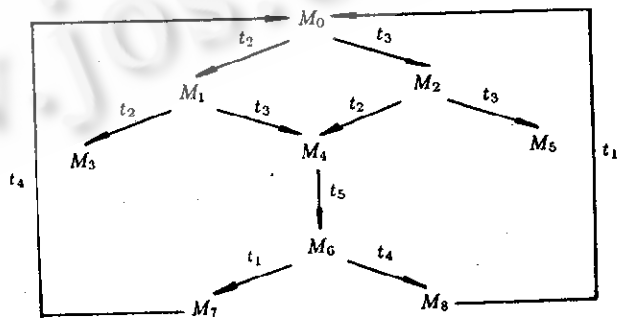
定理2: 设 $N = (P, T; I, O, W, M_0)$, $RMG = (R(M_0), E)$, $|P| = m, |T| = n, |E| = r$, 记 $a = \max(m, n), b = \min(n, m)$, 则算法的时间复杂性为 $O(rba^2)$.

证明: 不妨设 $m < n$, 即 $a = n, b = m$.

由于算法主要是通过遍历 $RMG(N)$ 对 M 进行操作, 在访问某条边 $(M_i, M_j) \in E$ 时, 对 $M_j - M_i$ 做了一次向量减运算, 即做了 m 次标量减运算. 在挂 $p-t$ 链时, 新增加的 t 结点要同该链前面的所有 t 链结点进行比较, 看是否已在其中, 最坏情况下挂好一条 $p-t$ 链要做的比较运算次数不超过 $1 + 2 + \dots + n = \frac{1}{2}n(n-1)$ 次, 即比较运算的阶为 $O(n^2)$. 由于 PT 邻接表共要挂 m 条 $p-t$ 链, 所以构成 PT 邻接表需要做的操作量不超过 $O(mn^2)$, 同理构成 TP 邻接表需要做的操作量不超过 $O(nm^2)$. 而算法对 RMG 的每条边遍历一次, 故总的复杂性为 $O(r(O(m) + O(nm^2) + O(n^2m)))$, 考虑到 $m < n$, 所以复杂性为 $O(rmn^2)$, 也即 $O(ra^2b)$.

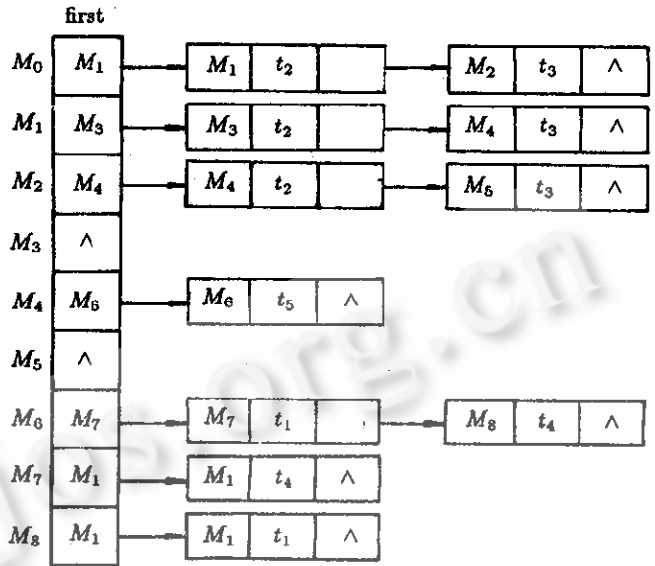
§ 6. 例子

已知一个 $RMG(N)$ 为

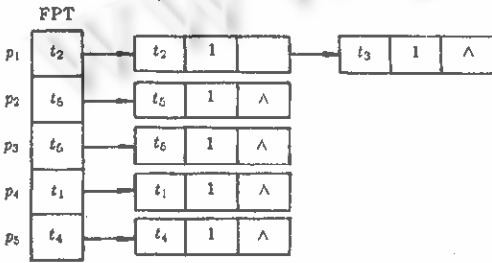


其中 $M_0 = (20000)^T, M_1 = (11000)^T, M_2 = (10100)^T, M_3 = (02000)^T, M_4 = (01100)^T, M_5 = (00200)^T, M_6 = (00011)^T, M_7 = (10001)^T, M_8 = (10010)^T$.

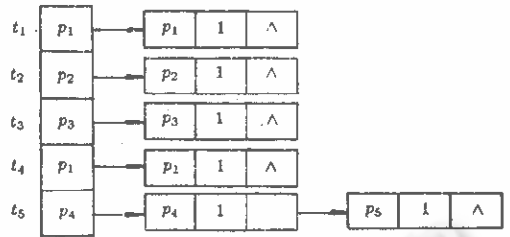
它的M—邻接表为:



根据前面算法运算的结果为: M_8

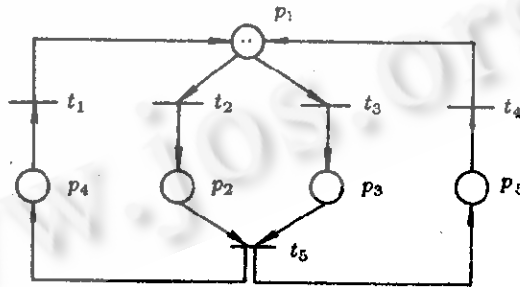


PT 邻接表



TP 邻接表

它所对应的网图为:



参考文献

- [1] 殷兆麟, "Petri 网与形式语言自动机", 中国矿院学报, No. 4 (1986).
- [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. 4 (1989).
- [3] 陆维明等, "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 Modeling of Systems", Prentice-Hall, Inc., Englewood Cliffs, N. J. U. S. A., 1981.
- [5] S. 巴斯, 《计算机算法: 设计与分析引论》, 朱洪等译, 复旦大学出版社, 1985 年版.