1000-9825/2002/13(07)1257-07 ©2002 Journal of Software 软 件 学 报 Vol.13, No.7

关于 ESAC 网活性和有界性的一个多项式算法*

焦 莉, 陆维明

(中国科学院 数学与系统科学研究院 数学研究所,北京 100080)

E-mail: wmlu@math03.math.ac.cn

http://www.math.ac.cn

摘要: 寻找实际可行的多项式算法一直是 Petri 网应用的重要方面.给出了关于扩展强化非对称选择网(extended strong asymmetric choice nets,简称 ESAC 网)结构活和结构有界的一个判定算法.该算法可简单、有效地测试结构活结构有界的 ESAC 网的初始标识是否是活标识.ESAC 网覆盖了自由选择网,因此,该算法应用范围较为广泛.

关 键 词: 扩展强化非对称选择网(ESAC 网);结构活;结构有界;多项式算法

中图法分类号: TP311 文献标识码: A

Petri 网以其简单、潜在模拟能力强等特点被广泛用于离散事件系统的模拟和分析中.例如,多处理器计算机系统、计算机网络、交通控制系统等.在这些系统的设计和分析中,常常需要确定该系统从一个子系统转移到其他子系统而不会引起死锁,保持这一特点的系统一般被称为活的系统.

Petri 网理论提供了许多分析算法,但只要一涉及到 Petri 网的一些特点和性质,其算法的复杂度往往呈指数量级.因此,寻找多项式判定的工作一直在进行[1,2].对有一定限制的某些网类,比如状态机、标识图和自由选择网,活性和有界性的多项式判定算法切实存在[2~9].自由选择网活性和有界性的多项式判定算法的存在主要由著名的秩定理保证.而对比自由选择网大的网类,要判断它们的活性和有界性是相当困难的.

文献[3]扩展了文献[7]所研究的网类,证明了关于扩展强化非对称选择网(ESAC 网)结构活结构有界的充分必要条件.本文在文献[8,9]中所提算法的基础上,提出一个关于 ESAC 网活性和有界性判定的算法,并对结构活结构有界的这类网的初始标识,给出了其是否是活标识的有效判定.文献[8,9]都是关于自由选择网活性和有界性判定的多项式算法,本文中的算法应用范围更广.

本文第 1 节给出了一些相关的基本概念.第 2 节给出 ESAC 网活性和有界性判定算法的理论根据.第 3 节给出 ESAC 网结构活和结构有界性判定的算法.第 4 节提出了结构活和结构有界的 ESAC 网初始标识是否是活标识的多项式判定算法.第 5 节总结全文.

1 基本概念

假定读者知道 Petri 网,这里只引入本文最必要的几个定义.

定义 **1.1**. 设 $N=(P,T,F)$是 Petri 网,其中 P,T,F 分别是库所、变迁和流关系.

(1) $\forall x\in P\cup T, {}^{\bullet}x=\{y|(y,x)\in F\}$称为 x 的前集,$x^{\bullet}=\{y|(x,y)\in F\}$称为 x 的后集.

(2) N 的一条路径是结点序列$(x_1,x_2,...,x_n)$,其中 $x_i\in P\cup T, i\in\{1,2,...,n\}, (x_i,x_{i+1})\in F$ 对所有的 $i\in\{1,...,n-1\}$.

(3) N 是一个 S-图当且仅当$\forall t\in T:|{}^{\bullet}t|=|t^{\bullet}|=1$.

(4) Petri 网 $N'=(P',T',F')$是 N 的子网(记作 $N'\subseteq N$),当且仅当 $P'\subseteq P, T'\subseteq T, F'=F\cap((P'\times T')\cup(T'\times P'))$.

* 收稿日期: 2000-09-18; 修改日期: 2002-04-25

基金项目: 国家自然科学基金资助项目(60073013);国家重点基础研究发展规划 973 资助项目(G1998030416)

作者简介: 焦莉(1964-),女,河南嵩县人,博士,副教授,主要研究领域为 Petri 网,算法设计与分析;陆维明(1941-),男,浙江宁波人,研究员,博士生导师,主要研究领域为 Petri 网,软件工程,算法设计与分析.

(5) N 的子网 $N'=(P',T',F')$是由 $P'\subseteq P$ 生成的当且仅当 $T'={}^\bullet P'\cup P'^\bullet, F'=F\cap((P'\times T')\cup(T'\times P'))$.

(6) $N'\subseteq N$ 是 N 的一个 S-分支(S-component)当且仅当 N'是强连通的 S-图,并且 $T'={}^\bullet(P')\cup(P')^\bullet$.

(7) $P'\subseteq P$ 是 N 的一个非空死锁(陷阱)当且仅当 $P'\neq\varnothing, {}^\bullet(P')\subseteq(P')^\bullet((P')^\bullet\subseteq{}^\bullet(P'))$;$P'$是 N 的极小死锁(陷阱)当且仅当 P'的真子集都不是 N 的死锁(陷阱);一个死锁 N'是强连通的当且仅当 $N'=(P',{}^\bullet P',F')$是强连通的,其中 $F'=F\cap((P'\times{}^\bullet(P'))\cup({}^\bullet(P')\times P'))$.

定义 **1.2**. 设 $N=(P,T,F)$是 Petri 网,

(1) 函数 $M:P\rightarrow\{0,1,2,3,\ldots\}$叫作 N 的标识;

(2) $P'\subseteq P$ 在 M 被标识当且仅当$\exists p\in P':M(p)>0$;

(3) $t\in T$ 在 M 能点火,记作 $M[t\rangle$当且仅当$\forall p\in{}^\bullet t:M(p)>0$;

(4) W 表示 F 的特征函数,N 的关联矩阵 $C=(c_{ij})_{|P|\times|T|}$,其中 $c_{ij}=W(t_j,p_i)-W(p_i,t_j)$;

(5) 若 $M[t\rangle$,则产生一个新的标识 M',记为 $M[t\rangle M'$,其中 $M'(p)=M(p)-W(p,t)+W(t,p)$对$\forall p\in P$.称 M'为从 M 可达的标识;

(6) 称(N,M_0)为 Petri 网系统;

(7) 对(N,M_0),从 M_0 可达的标识,所有标识叫做 M_0 的可达集,记作 $R(M_0)$;

(8) (N,M_0)是有界的当且仅当$\exists k\in\{1,2,\ldots\}$,对$\forall p\in P,\forall M\in R(M_0):M(p)<k$;

(9) N 是结构有界的当且仅当$\forall M_0\in[P\rightarrow\{0,1,2,\ldots\}],(N,M_0)$是有界的;

(10) (N,M_0)是活的当且仅当$\forall t\in T,\forall M\in R(M_0):\exists M'\in R(M_0),M'[t\rangle$;

(11) N 是结构活的当且仅当$\exists M_0\in[P\rightarrow\{0,1,2,\ldots\}]:(N,M_0)$是活的.

定义 **1.3**. 令 N 是 Petri 网,

(1) N 是自由选择网(FC 网)当且仅当:$\forall p_1,p_2\in P,p_1^\bullet\cap p_2^\bullet\neq\varnothing\Rightarrow p_1^\bullet=p_2^\bullet$;

(2) N 是非对称选择网(AC 网)当且仅当:$\forall p_1,p_2\in P,p_1^\bullet\cap p_2^\bullet\neq\varnothing\Rightarrow p_1^\bullet\subseteq p_2^\bullet$或者 $p_2^\bullet\subseteq p_1^\bullet$;

(3) N 是扩展强化非对称选择网(ESAC 网)当且仅当:若$\exists p,q\in P,p^\bullet\cap q^\bullet\neq\varnothing$且 $p^\bullet\subset q^\bullet$则有${}^\bullet p\subseteq{}^\bullet q$.

容易看出,ESAC 网类包含 FC 网,是 Petri 网中一个较大的网类.

定义 **1.4**. 设 $N=(P,T,F)$是 Petri 网,$S,S'\subseteq P\cup T,S\cup S'=P\cup T,S\cap S'=\varnothing$,$N$ 中一条路径 $H=(x_0,x_1,\ldots,x_n)$是一个 handle 当且仅当 $x_0,x_n\in S,x_1,\ldots,x_{n-1}\in S',(x_i,x_{i+1})\in F,\forall i\in\{0,1,\ldots,n-1\},x_i\neq x_j,i,j\in\{1,2,\ldots,n-1\},i\neq j$.

2 扩展强化非对称选择网活性和有界性判定

本文只考虑连通的 Petri 网,因为对不连通的网,可逐个考虑它们的连通分枝.首先我们列出一些对本文有用的已知结果.

引理 **2.1**[10]. 令(N,M_0)是一个 Petri 网系统,如果(N,M_0)是活的和有界的,那么 N 是强连通的.

引理 **2.2**[10]. 令(N,M_0)是一个 AC 网系统,如果 N 的每个死锁都含有 M_0 标识的一个陷阱,那么(N,M_0)是活的.

引理 **2.3**[7,8]. 令 $N=(P,T,F)$是一个 AC 网,$H\subseteq P$ 是 N 的一个死锁.H 是极小死锁当且仅当 H 是 P 连通的(即 H 中元素相互可达),并且$\forall t\in H^\bullet:|{}^\bullet t\cap H|=1$.

定理 **2.1**[3]. 令 $N=(P,T,F)$是 ESAC 网,N 是结构活和结构有界的,当且仅当:

(1) N 的每个非空极小死锁 H 是陷阱;

(2) $\forall t\in H^\bullet:|{}^\bullet t\cap H|=|t^\bullet\cap H|=1$;

(3) $\forall p\in P$,存在一个极小死锁 H,使 $p\in H$.

现在考虑由极小死锁 H 生成的子网 $N_1=(H,H^\bullet,F_1)$,其中 $F_1=F\cap(({}^\bullet H\times H)\cup(H\times H^\bullet))$.

如果 N_1 满足下面的条件:

(1) N_1 是强连通的;

(2) $\forall t\in H^\bullet:|{}^\bullet t\cap H|=|t^\bullet\cap H|=1$;

则 N_1 是由 H 生成的一个 S-分支.

定理 **2.2**. 令 $N=(P,T,F)$是一个 AC 网,H 是其极小死锁,那么 H 生成 N 的一个 S-分支,当且仅当${}^\bullet H=H^\bullet$且

$\forall t\in H^{\bullet}:|^{\bullet}t\cap H|=|t^{\bullet}\cap H|=1$.

证明:充分性.因为$^{\bullet}H=H^{\bullet}$且$\forall t\in H^{\bullet}:|^{\bullet}t\cap H|=|t^{\bullet}\cap H|=1$,$H$ 是极小死锁,所以$\forall t\in H^{\bullet}:|^{\bullet}t\cap H|=1$,且 H 是强连通的.

下面证明 N_1 是强连通的.对$\forall x_1,x_2\in N_1$:

如果 $x_1,x_2\in H$,根据引理 2.3,H 是 P 连通的,即 x_1 与 x_2 相互可达.

如果 $x_1\in H,x_2\in H^{\bullet}$,由于$\exists p\in H$ 使 $x_2\in p^{\bullet}$.由 H 的 P 连通性可知 x_1 可达 p,所以 x_1 可达 x_2;又因为$|x_2^{\bullet}\cap H|=1$,所以$\exists p'\in H,x_2\in{}^{\bullet}p',p'$可达 x_1,所以 x_2 可达 x_1,故 x_1,x_2 相互可达.

如果 $x_1,x_2\in H^{\bullet}$,那么$\exists p_2\in H$ 使 $x_2\in p_2^{\bullet}$.又因为$|x_1^{\bullet}\cap H|=1$,所以存在 $p_1\in H$ 有 $x_1\in{}^{\bullet}p_1$.但 p_1 可达 p_2,故 x_1 可达 x_2.所以 N_1 是强连通的.故 H 生成 N 的一个 S-分支.

必要性.由于 H 生成 N 的一个 S-分支,所以对$\forall t\in H^{\bullet}:\forall t\in H^{\bullet}:|^{\bullet}t\cap H|=|t^{\bullet}\cap H|=1$.又由于 H 是死锁,所以$^{\bullet}H\subseteq H^{\bullet}$.为了证明$^{\bullet}H=H^{\bullet}$,我们只需证明 $H^{\bullet}\subseteq{}^{\bullet}H$ 即可.

对$\forall t\in H^{\bullet}$,由于$|t^{\bullet}\cap H|=1$,所以$\exists h\in H:t\in{}^{\bullet}h$,所以 $t\in{}^{\bullet}H$,故 $H^{\bullet}\subseteq{}^{\bullet}H$.所以$^{\bullet}H=H^{\bullet}$. □

由定理 2.1 和定理 2.2,我们可以直接得到定理 2.3.

定理 **2.3**. 设 $N=(P,T,F)$是一个 ESAC 网.N 是结构活结构有界的当且仅当 N 的每一极小死锁生成一个 S-分支,并且 N 被这些 S-分支覆盖.

一般来说,即使已知一个 Petri 网是结构活结构有界的,要判定它的一个任意初始标识是否活标识也是不容易的.但是如果一个网是结构活结构有界的 ESAC 网(LBESAC 网),我们可以通过检查下面定理 2.4 的条件来判定其初始标识是否活标识.

定理 **2.4**. 令 N 是一个 LBESAC 网,M 是 N 的一个任意初始标识.那么,M 是 N 的一个活标识当且仅当 N 的所有极小死锁在 M 下被标识.

证明:充分性.因为 N 是一个 LBESAC 网,由定理 2.1 可知,N 的每个非空极小死锁也是陷阱.由于 M 标识了 N 的所有极小死锁,即 N 的所有极小死锁都含有被 M 标识的陷阱,根据引理 2.1,M 是 N 的一个活标识.

必要性.因为 M 是 N 的一个活标识,由死锁的定义可知,N 的所有极小死锁在 M 下被标识. □

具体来说,要判定一个标识是否是LBESAC网的活标识,我们只需寻找是否存在由没有被 M_0 标识的库所构成的子集 D,判定 D 是否是 N 的一个死锁即可.

3 ESAC 网活性和有界性判定算法

依据上一节的主要结果,我们可以给出 ESAC 网活性和有界性判定算法的概要如下:

算法 **3.1**(概要).

输入:ESAC 网系统(N,M_0)

输出:Yes: 如果 N 是一个 LBESAC 网

No: 否则

第 1 步:检查网 N 是否强连通

如果 N 不是强连通的,输出 No 停止(根据引理 2.1)

第 2 步:对每个 $p\in P$ 寻找包含 p 的所有 S-分支

(1) 寻找包含 p 所有的极小死锁$\{H|p\in H\}$.如果不存在包含 p 的极小死锁,输出 No,停止(根据定理 2.3).

(2) 检查每个 H 是否生成一个 S-分支.如果有一个 H 所生成的子网不是一个 S-分支,输出 No,停止(根据定理 2.3).

判定一个 Petri 网是否强连通,实际上就是判定一个有向图是否强连通,具体算法可参见文献[11].

3.1 扩展强化AC网中极小死锁的计算

本节给出计算含有一个给定库所的强连通 ESAC 网的极小死锁的算法.

根据定理 2.3 我们知道,如果一个 ESAC 网 N 是结构活结构有界的,那么其极小死锁一定生成 N 的一个 S-分支.利用这一特点,考虑以下两种情况:(1) 如果找到 N 的一个不可能生成其 S-分支的极小死锁,那么可判定 N

不是结构活结构有界的;(2) 先寻找可能成为 N 的 S-分支的那些极小死锁,然后再进一步判定其是否真的能生成一个 S-分支.

算法 **3.1.1**. get-minimal-deadlock (P,T,F,p,T_D).

输入:一个强连通的 ESAC 网 $N=(P,T,F)$和 P 的一个库所 p

输出:含有 p 的一个极小死锁 D(其中 $T_D={}^{\bullet}D$)

Function: get-handle(S,S',F,p,t)

{该函数计算一个 handle($x_1,x_2,\ldots,x_{n-2},t,p$),其中 $x_0,p\in S,x_1,\ldots,x_{n-2},t\in S'$.handle 作为一个集合{$x_0,x_1,\ldots,x_{n-2},t,p$}返回}

```
Begin
  P′={p};T′=∅
  While (∃p′∈P′;∃t∈•p′;t∉T′)
    Begin
    H:=get-handle((P′∪T′),(P∪T)−(P′∪T′),F,p′,t);
    P′:=P′∪(H∩P);T′:=T′∪(H∩T);
    End (Begin)
  End (While)
  D:=P′;T_D=T′;
End (Begin)
```

下面的算法 3.1.2 把寻找可能生成 S-分支的极小死锁的 handle 以及可能生成的极小死锁不是 S-分支的判定结合起来考虑,后者无须找出极小死锁,而只要输出 No 即可.

算法 **3.1.2**. get-handle(S,S',F,p,t).

输入:一个强连通的网 $N=(P,T,F)$,其中 $S,S'\subseteq(P\cup T),S\cup S'=P\cup T,S\cap S'=\varnothing,p\in S,t\in S'$ and $t\in{}^{\bullet}p$

输出:Handle $H=(x_0,\ldots,x_{n-2},t,p)$ or No

```
Function dfs(v)
Begin
  num(v):=i;i:=i+1;push(stack,v);
  对所有的(w∈•v)
  Begin
    if (num(w)=−1)   // handle 的始结点是库所//
    then push(stack,w); return Yes;
    if (num(w)=−2)   // handle 的始结点是变迁//
    then pop(stack,w); return No;
  End (Begin)
  对所有的(w∈•v)
  Begin
    if (num(w)=0)    //一个新结点被达到//
    then if (dfs(w) = Yes)
        then return Yes;
  End (Begin)
  pop(stack,v); return No;
End (Begin)
Begin
  i:=1; Stack:=empty-stack;
```

```
        num(x):=0,∀x∈S′;
        num(x):=−1,∀x∈S∩P;
        num(x):=−2,∀x∈S∩T;
        push(stack,p);
        if (dfs(t) = No)
        then 输出 No,停止;
        else 输出 stack,停止;
    End (Begin)
```

说明:栈(stack)被用来储存可能属于一个 handle 的结点,p 先入栈直到终止后才出栈.因为如果 handle 存在,p 一定是它的一个成员.如果 dfs(v)被调用,结点 v 入栈.搜索如果成功,dfs(v)结束时 v 不被弹出.如果没有找到以 v 为起始结点的 handle,v 就不是一个 handle 的元素,此时弹出 v.

算法 3.1.2 的正确性参见文献[9].该算法的效率取决于 dfs 函数的调用次数,对$\forall v'\in S'$,dfs 只被调用一次.任何与 v 为终点的弧至多被检查两次,但 ESAC 网 $N=(P,T,F)$有$(|P|+|T|)$个结点和$|F|$条弧.因此,算法 3.1.2 的最坏情况复杂度是 $O(|P|+|T|+|F|)$.由于一个单独的 handle 计算最坏情况复杂度是 $O(|P|+|T|+|F|)$,因此,极小死锁的计算至多需寻找 handle$|T|$次,所以算法 3.1.1 的最坏情况复杂度是 $O(|T|(|P|+|T|+|F|))$.

3.2 检查一个极小死锁是否生成一个*S*-分支

利用定理 2.2,本节构造算法 3.2.1,以便检查是否每一极小死锁都生成一个 S-分支.要检查定理 2.2 的条件是否满足,需要记数从$\forall t\in T'$到 $s\in D$ 的弧$(t,s)\in F$.

算法 **3.2.1**. Check-S-component(T,F,D,T_D).

输入:ESAC 网 $N=(P,T,F)$的极小死锁 $D,T_D={}^\bullet D$

输出:Yes,D 生成一个 S-分支

　　No,D 不生成一个 S-分支

```
Begin
  num(x) = 0 ∀x∈T−T_D;
  num(x) = 1 ∀x∈T_D;
  对所有的(s∈D)
  Begin
    对所有的(t∈T)
    Begin
      if ((t,s)∈F)
      then num(t)=num(t)−1;
      if (num(t)<0)
      then return No
      if (s,t)∈F and t∈(T−T_D)
      then return No;
    End (Begin)
  End (Begin)
  return Yes;
End (Begin)
```

D 和 T 的有限保证算法 3.2.1 可终止.显然,算法 3.2.1 的最坏情况复杂度是 $O(|D||T|)$.

假定 $\varphi(|P|,|T|)$是 ESAC 网 $N=(P,T,F)$的极小死锁的总数,那么算法 3.1 的最坏情况复杂度是 $O(\varphi(|P|,|T|)|T|(|P|+|T|+|F|))+O(\varphi(|P|,|T|)|D||T|)$.

4 LBESAC 网活标识判定的多项式算法

假定已知 ESAC 网 $N=(P,T,F)$是结构活和结构有界的,对任一标识 M_0,要判定 M_0 是否 N 的活标识,根据定理 2.4,必须检查是否每一极小死锁在 M_0 下被标识,即检查最大没有被 M_0 标识的死锁.下面的算法利用上述思想,令 U 是在 M_0 下没有被标识的库所的集合,不断减少 U 中阻止${}^\bullet U\subseteq U^\bullet$的元素,如果最后 $U=\varnothing$,那么 M_0 是一个活标识.否则 M_0 不是 N 的活标识.

算法 **4.1**. Check-initial-marking(N,M_0).

输入:一个 LBESAC 网 $N=(P,T,F)$和 N 的一个任意初始标识 M_0

输出:Yes M_0 是 N 的活标识

No M_0 不是 N 的活标识

```
Begin
  U={p|p∈P,M0 =0};
  T′ = •U; Done = false;
  While (U ≠ ∅ and not done)
   Begin
    if (∃t∈T′: U∩t• ≠ ∅ and U∩•t = ∅)
    then U: = U–t•, T′: = T′−{t}
    else done: = true;
   End (Begin)
  End (While)
  if (U = ∅) then 输出 Yes, 停止
           else 输出 No, 停止
End (Begin)
```

由于 U 是有限的,并且每次 while 循环至少从 U 中减少一个元素,所以算法 4.1 是可终止的.算法 4.1 的正确性可由定理 2.4 保证.很明显,该算法的复杂性由 while 循环的次数确定.初始化 U 和 T 至多需要时间 $O(|P||T|)$,在一次 while 循环中至多有$|T|$次调用.因为每次循环至多从 U 中除去一个库所,因此至多有 $\min(|T|,|P|)$次可能的循环.要寻找满足 $U\cap t^\bullet\neq\varnothing$和 $U\cap{}^\bullet t=\varnothing$的 t $(t\in T')$,要化费 $O(|P|)$时间.因此,算法 4.1 的最坏情况复杂度是 $O(|P|^2|T|)$.

5 结 论

Petri 网系统是动态系统的静态描述,但只要一涉及到活性和有界性,其算法的复杂性往往是呈指数量级的,难以付诸实用.因此寻找判定 Petri 网活性和有界性的多项式算法是极有意义的工作.

本文在文献[3,8,9]的基础上,提出了关于判定扩展强化非对称选择网(ESAC 网)结构活结构有界的算法,该算法还可对结构活结构有界 ESAC 网初始标识的活性进行判定.这一算法的应用范围覆盖了自由选择网,因此,这是一个有效实用的算法.

References:

[1] Schmidt, K. Stubborn sets for standard properties. LNCS,1999,1639:46~65.

[2] Pastor, E., Cortadella, J., Pena, M.A. Structural methods to improve the symbolic analysis of Petri nets, LNCS, 1999,1639:26~45.

[3] Jiao, Li, LU, Wei-ming. Liveness and boundedness on extended strong asymmetric choice nets. Journal of Software, 2001,12(9):1311~1317 (in Chinese).

[4] He, K.X., Lemmon, M.D. Liveness verification of discrete event systems modeled by *n*-safe ordinary Petri nets. LNCS, 2000, 1825:227~243.

[5] Van der Aalst, W., Kindler, E., Desel, J. Beyond asymmetric choice: a note on some extensions, Petri Net Newsletter, 1998,55: 3~13.
[6] Lu, Wei-ming, Zhen, Qiang. Study of Petri net system liveness. Journal of Computer Science, 1999,26(4):1~4 (in Chinese).
[7] Zhen, Qiang, Lu, Wei-ming. On liveness of asymmetric choice nets. Journal of Software, 1998,9(5):354~359 (in Chinese).
[8] Esparza, J., Silva, M. A polynomial time algorithm to decide liveness of bounded free choice nets. Journal of Theoretical Computer Science, 1992,102:185~205.
[9] Kemper, P., Bause, F. An efficient polynomial-time algorithm to decide liveness and boundedness of free choice nets. LNCS, 1992, 616:263~278.
[10] Murata, T. Petri nets: properties, analysis and applications. Proceedings of the IEEE, 1989,77(4):541~580.
[11] Zhu, Hong. Design and Analysis of Algorithms. Shanghai: Fudan University Press, 1984 (in Chinese).

附中文参考文献:

[3] 焦莉,陆维明.扩展强化非对称选择网的活性和有界性.软件学报,2001,12(9):1311~1317.
[6] 陆维明,甄强.Petri 网系统活性的研究.计算机科学,1999,26(4):1~4.
[7] 甄强,陆维明.论非对称选择网的活性.软件学报,1998,9(5):354~359.
[11] 朱洪.算法设计与分析.上海:复旦大学出版社, 1984.

A Polynomial Algorithm to Decide Liveness and Boundedness of ESAC Nets*

JIAO Li, LU Wei-ming

(*Institute of Mathematics*, *Academy of Mathematics and Systems Sciences*, *The Chinese Academy of Sciences*, *Beijing* 100080, *China*)

E-mail: wmlu@math03.math.ac.cn

http://www.math.ac.cn

Abstract: It has been an important direction of Petri net applications to find practical and efficient polynomial algorithms. In this paper, an algorithm is given to decide the structural liveness and structural boundedness of extended strong asymmetric choice (ESAC) nets. This algorithm can also decide whether an initial marking of a structurally live and structurally bounded ESAC net is a live marking or not. Since the class of ESAC nets contains free choice (FC) nets, this algorithm shows large area of applications.

Key words: ESAC nets; structural liveness; structural boundedness; polynomial algorithm.

* Received September 18, 2000; accepted April 25, 2002

Supported by the National Natural Science Foundation of China under Grant No.60073013; the National Grand Foundation Research 973 Program of China under Grant No.G1998030416