



无标注 L 型 Petri 网语言属性判定的一种方法

栾尚敏^{1,2*}, 田立勤^{1,2}, 耿子林^{1,2}

1. 华北科技学院计算机系, 三河 065201

2. 河北省物联网数据采集与处理工程技术研究中心, 三河 065201

* 通信作者. E-mail: 623320726@qq.com

收稿日期: 2016-11-09; 接受日期: 2016-12-21; 网络出版日期: 2017-03-06

国家自然科学基金(批准号: 61472137)、中央高校基本科研业务项目(批准号: 3142014007, 3142015022)、河北省高等学校科学技术研究项目(批准号: Z2014038)和青海省重点研发项目(批准号: 2016-SF-130)资助

摘要 Petri 网和自动机是离散事件动态系统建模的两种重要方法, 研究这两种模型之间的关系, 对于更好地理解和控制离散事件动态系统的行为具有重要作用. 本文从形式语言的角度对该问题进行了研究, 提出了判定无标注 L 型 Petri 网语言属性的方法, 引入有效递增子和线性子等概念来刻画 Petri 网语言的性质. 对于一个 Petri 网 PN, 当 PN 没有有效递增子时, PN 对应的 Petri 网语言是正则语言. 当 PN 的某个有效递增子有两个及其以上线性子时, PN 对应的 Petri 网语言是上下文有关的. 当 PN 的所有有效递增子只有一个线性子时, 如果只有一个有效递增子, 则 PN 对应的 Petri 网语言是上下文无关的; 如果 PN 有两个及其以上有效递增子, 则顺序引发、嵌套引发和选择引发时 PN 对应的 Petri 网语言是上下文无关的, 并发引发和交叉引发时 PN 对应的 Petri 网语言是上下文相关的. 本文还对可达树进行了改进, 给出了利用改进后的可达树来判定有效递增子的方法, 从而使本文给出的判定无标注 L 型 Petri 网语言属性的方法具有可操作性.

关键词 Petri 网语言, 形式语言, 自动机理论, 可达树, 正则语言, 上下文无关语言, 上下文相关语言

1 引言

Petri 网^[1~3]和形式语言理论中的自动机^[4,5]是离散事件动态系统的两种重要的建模方法, 研究这两种模型之间的关系, 对于更好地理解和控制离散事件动态系统的行为具有重要作用^[6]. Petri 网的引发序列本身也构成一种形式语言, 称为 Petri 网语言^[1], 本文从形式语言的角度讨论了 Petri 网和自动机之间的关系. 根据终止状态的不同, Petri 网语言分为 L 型、 G 型、 T 型和 P 型^[7]. 给定一个 Petri 网, 判定其对应的 L 型 (G 型、 T 型或 P 型) Petri 网语言属于 Chomsky 谱系中的正规语言、上下文无关语言、上下文有关语言的问题也称为 Petri 网语言的属性判定问题^[8]. 关于该问题, 人们已

引用格式: 栾尚敏, 田立勤, 耿子林. 无标注 L 型 Petri 网语言属性判定的一种方法. 中国科学: 信息科学, 2017, 47: 696-714, doi: 10.1360/N112016-00056

Luan S M, Tian L Q, Geng Z L. An approach to distinguishing the levels of free-labeled L -type Petri net languages in the Chomsky hierarchy (in Chinese). Sci Sin Inform, 2017, 47: 696-714, doi: 10.1360/N112016-00056

经做了大量的研究工作^[1~3,6~12]. Peterson^[7] 证明了正规语言是 Petri 网语言; 他还举例说明了存在 Petri 网, 其对应的 Petri 网语言不是正则语言, 从而说明了正则语言只是 Petri 网语言的一个真子集. Peterson^[7] 还证明了不存在一个 Petri 网, 其对应的语言为回文. 回文是上下文无关语言, 由此说明了存在上下文无关语言不是 Petri 网语言; 同时还给出了一个 Petri 网, 它对应的语言是 $a^n b^n c^n$, 该语言不是上下文无关的, 而是上下文相关的, 从而说明了 Petri 网语言和上下文无关语言之间不存在包含关系; Peterson 最后证明了所有的 Petri 网语言都是上下文相关的. 吴^[8] 引入了减向量、有效减向量、传递向量、有效传递向量等概念, 给出了判定一个 Petri 网对应的 Petri 网语言属性的方法. 他证明了对于一个顺序引发的标准 Petri 网, 如果它不存在有效减向量, 则它对应的 Petri 网语言是正则语言; 如果存在有效减向量, 但不存在传递向量, 则它对应的 Petri 语言是上下文无关的; 如果存在有效减向量和有效传递向量, 则它对应的 Petri 网语言是上下文相关的. 蒋^[6] 给出了求极小受控可重复向量的算法, 使得吴^[7] 的判定 Petri 网语言属性的方法具有可操作性. Valk 和 Vidal-Naquet^[9] 指出了有界 Petri 网对应的 Petri 网语言是正则语言; 对于无界 Petri 网 PN, PN 对应的 Petri 网语言是正则语言的充要条件为: 存在一个整数 k , 对于 PN 的任意可达标识 M 、从 M 可达的标识 M' 和位置 p , 都有 $M'(p) \geq M(p) - k$. 对于位置集 P 的子集 P' , 他们还给出了 P' 是有一致下界的和极大无界的定义, 由此证明了对于一个 Petri 网 PN, 它对应的语言不是正则语言的充要条件为存在一个标识 M 以及位置集 P 的子集 P' , P' 关于 M 是极大无界的, 但关于 M 不是一致有界的. Mazurkiewicz^[10] 证明了只有一个位置的 Petri 网对应的 Petri 网语言是上下文无关语言, 并进一步证明了一个 Petri 网语言是有限个上下文无关语言的交集. Petri 网也被广泛地应用到了控制系统的描述中, 描述控制系统行为的一些 Petri 网所对应的 Petri 网语言是正则语言^[3~5,11]. 在形式语言理论中, “泵” 引理是一个重要的结论^[13], 在 Petri 网语言中也存在其对应的表现形式, 蒋和刘给出了这一结论^[12]. 尽管已经证明了带抑制弧的 Petri 网的计算能力等同于图灵机^[7], 目前对 Petri 网语言的研究主要是针对传统的 Petri 网的^[7,9], 对于带抑制弧的 Petri 网语言方面的研究还未见报道, 本文也是针对传统的无标注 L 型 Petri 网语言展开讨论. Petri 网 PN 对应的无标注 L 型 Petri 网语言是 PN 的从初始状态可达终止集 \dot{F} 中某个标识的引发序列的集合, 在讨论无标注 L 型 Petri 网语言时, PN 和 \dot{F} 是成对出现的, 用 (PN, \dot{F}) 表示 Petri 网 PN 和终止集 \dot{F} . 如无特别说明, 后面所说的 Petri 网语言均指无标注 L 型 Petri 网语言.

本文分析了无标注 L 型 Petri 网语言的特性、给出了判定无标注 L 型 Petri 网语言属性的方法. 影响无标注 L 型 Petri 网语言特性的因素是满足如下条件的变迁的集合: 该集合中的变迁能形成一个引发序列, 并且引发序列一旦引发就可以无限次的引发, 把这种变迁的集合称为循环子; 有的循环子的连续引发不会导致网中标记数量的连续增加, 把这样的循环子称为简单循环子; 有的循环子的引发会导致某些位置中的标记数量的增加, 把这样的循环子称为递增循环子, 简称递增子; 给定一个递增子, 如果对于任何一个正整数 N , 都存在一个引发序列使得初始标识可达终止标识, 且该递增子中的变迁在 τ 中出现的次数大于 N , 则称该递增子为有效递增子. 本文还进一步地给出了受限循环子和线性子等概念, 在此基础上, 详细讨论了无标注 L 型 Petri 网语言属性的判定方法, 该方法中最重要的一步就是寻找有效递增子, 为此, 对可达树进行了改进, 并给出了基于改进可达树的判定有效递增子的方法. 本文还用到了 Petri 网语言对某些运算的封闭性的性质, 这方面的结论请参见文献^[7,14~17]. 关于 Petri 网和形式语言的基础知识, 请参见文献^[1,13].

2 无标注 L 型 Petri 网语言的特征分析

本节主要分析无标注 L 型 Petri 网语言的特性、给出判定其属性的方法. 对于 Petri 网 PN 和任

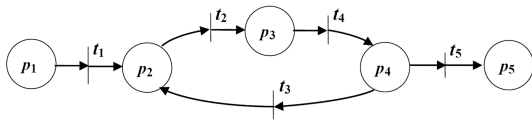


图 1 有界 Petri 网实例, $M_0 = (1, 0, 0, 0, 0)$
 Figure 1 A bounded Petri net with $M_0 = (1, 0, 0, 0, 0)$

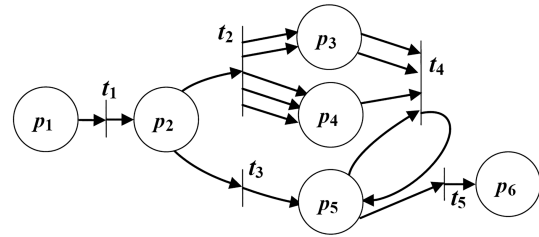


图 2 有无界位置, 但 Petri 语言是正则语言实例, $M_0 = (1, 0, 0, 0, 0, 0)$
 Figure 2 An unbounded Petri net generating a regular language, in which $M_0 = (1, 0, 0, 0, 0, 0)$

意终止集 \hat{F} , 如果 PN 是有界的, 则 PN 对应的 Petri 网语言是正则语言^[9]. 例如, 图 1 所示的 Petri 网, $\hat{F} = \{(0, 0, 0, 0, 1)\}$; 则它所对应的 Petri 网语言都是正则语言.

命题 1 对于 (PN, \hat{F}) , 如果 PN 是有界的, 则 PN 对应的 Petri 网语言是正则语言^[9].

定义 1 对于 $PN = (S, T, F, W, M_0)$, 设 τ 是 T 中某些变迁构成的序列. 对于任意的正整数 K , PN 总有一个从初始状态开始的引发序列, τ 在该序列中出现的次数不小于 K , 则称 τ 为循环引发序列.

循环引发序列是影响 Petri 网语言性质的因素之一, 为了便于详细地讨论这种影响, 下面进一步地对循环引发序列进行限定得到循环子、递增子和受限循环子等概念.

定义 2 对于 (PN, \hat{F}) , 其中 $PN = (S, T, F, W, M_0)$, 设 T_1 是 T 的非空子集, 若 T_1 中的变迁能构成一个循环引发序列, 且循环引发的次数不受限制, 但 T_1 的任意真子集都不具备该性质, 则称 T_1 为循环子, 用 \mathcal{R} 及其下标表示循环子, \mathcal{R} 的循环引发序列的引发简称为 \mathcal{R} 的引发.

对于图 1 中的 Petri 网, $\{t_2, t_3, t_4\}$ 是循环子.

定义 3 对于 (PN, \hat{F}) , 设 \mathcal{R} 是 PN 的循环子,

(1) 若 \mathcal{R} 的每次引发得到的标识仍然是引发前的标识, 则称 \mathcal{R} 为 PN 的简单循环子. 用 $\hat{\mathcal{R}}$ 及其下标表示简单循环子.

(2) 若 \mathcal{R} 的每次引发都会使得某些位置中的标记数量增加, 而其他位置中的标记数量不变, 则称 \mathcal{R} 为 PN 的递增循环子, 简称递增子. 用 $\hat{\mathcal{R}}$ 及其下标表示递增子.

图 1 中所示 Petri 网的循环子是简单循环子, 图 2 中所示 Petri 网的循环子是递增子.

命题 2 对于 (PN, \hat{F}) , \hat{C} 为 PN 的邻接矩阵. $\#(\tau, t)$ 表示 t 在 τ 中出现的次数.

(1) 设 $\hat{\mathcal{R}}$ 是简单循环子, τ 是 $\hat{\mathcal{R}}$ 的循环引发序列. 向量 e 满足, 对任意的 t_i , $e(i) = \#(\tau, t_i)$, 则 $\hat{C} \cdot e = 0$.

(2) 设 $\hat{\mathcal{R}}$ 是递增子, τ 是 $\hat{\mathcal{R}}$ 的循环引发序列. 向量 e 满足, 对任意的 t_i , $e(i) = \#(\tau, t_i)$, 则 $\hat{C} \cdot e > 0$.

证明 (1) 假设 τ 引发一次后使标识从 M_1 变为 M_2 , 即有 $M_2 = M_1 + \hat{C} \cdot e$. 根据简单循环子的定义可知 $M_1 = M_2$, 所以, $\hat{C} \cdot e = 0$.

(2) 假设 τ 引发一次后使标识从 M_1 变为 M_2 . 假设 $\hat{C} \cdot e \leq 0$, 则 $M_2 = M_1 + \hat{C} \cdot e \leq M_1$, 这与递增子的定义相矛盾. 所以 $\hat{C} \cdot e > 0$.

命题 3 对于 (PN, \hat{F}) , 如果 PN 中只有简单循环子, 则 PN 对应的 Petri 网语言是正则语言.

证明 简单循环子的引发不会改变标识中标记的数量, 所以 PN 是有界网. 根据命题 1, 有界

Petri 网对应的 Petri 网语言是正则语言, 可知 PN 对应的 Petri 网语言是正则语言.

对于包含递增子的 Petri 网, 其对应的 Petri 网语言可能是正则语言, 也可能是上下文无关语言, 还可能是上下文相关语言, 下面进行详细的分析.

命题4 对于 (PN, \hat{F}) , 若存在一个常数 K , 当递增子的引发次数大于 K 时, 就不存在可达终止标识的引发序列, 则 PN 对应的 Petri 网语言是正则语言.

证明 因为引发次数大于 K 时, 不存在可达终止标识的引发序列, 这样可以限定每个位置中的标记个数不超过 $m \cdot K$, m 是一个常数, 这样得到一个有界网, 且这两个网对应的 Petri 网是相同的. 根据命题 1 可知该命题成立.

如图 2 中的 Petri 网, 设 $\hat{F} = \{(0, 0, 0, 0, 0, 1)\}$, 则它对应的 Petri 网语言都是正则语言.

定义4 对于 (PN, \hat{F}) , \hat{R} 是 PN 的递增子. 对任意整数 k , 都存在一个从初始标识到达终止标识的引发序列 τ , 且 \hat{R} 中的每个变迁在 τ 中的出现次数均大于 k , 则称 \hat{R} 是 (PN, \hat{F}) 的有效递增子.

如图 3 中的 Petri 网, 设 $\hat{F} = \{(0, 0, 0, 1)\}$, 则 $\{t_2\}$ 是有效递增子. 之所以存在有效递增子, 是因为它有一个循环引发序列, 它每次引发都能减少递增子产生的标记, 所以能使得递增子产生的标记数降低到一个常数. 但, 图 2 中的 Petri 网, 却没有能循环引发的变迁来减少递增子产生的标记, 所以不存在有效递增子.

命题5 设 \hat{R} 是 (PN, \hat{F}) 的有效递增子, 则 PN 中至少存在一个循环引发序列, 它的每次引发都能使得有效递增子产生的标记的数量减少.

证明 设位置 p 是有效递增子 \hat{R} 所导致的无界位置. 假设 PN 中不存在循环引发序列, 它的每次引发都能使网中由有效递增子产生的标记的数量减少, 也就不存在循环引发序列, 它的引发能使 p 中标记数量减少. 因为 \hat{R} 是一个递增子, 它会导致 p 中的标记数量的无限增加, 也就是说, 随着 \hat{R} 的循环引发序列的引发次数的增加, 会使 p 中标记的数量大于终止标识中对应分量的值, 从而导致不能达到终止状态. 这与题设 \hat{R} 是有效递增子矛盾. 结论得证.

图 3 中 Petri 网的位置 p_2 是无界的, t_3 的引发次数只受位置 p_3 的限制, 而 p_3 中的标记数量只受循环子 $\{t_2\}$ 的限制, 因此 t_4 的引发次数只受循环子 t_2 的限制. 把这种引发次数受递增子引发次数制约的循环引发序列中的变迁构成的集合称为受限循环子, 具体的定义如下.

定义5 设 \hat{R} 是 (PN, \hat{F}) 的有效递增子, 其中 $PN = (S, T, F, W, M_0)$. 对于非空集合 $T_1 \subseteq T$, 如果 T_1 中的变迁能够形成循环引发序列, 且在到达终止标识的引发序列中, 其引发次数和有效递增子 \hat{R} 的引发次数呈线性关系, 但 T_1 的任意真子集都不具备这个性质, 则称 T_1 是 \hat{R} 的受限循环子. 用 \hat{R} 及其下标表示受限循环子.

当然, 受限循环子可以包含一个变迁, 例如图 3 中 Petri 网的变迁 t_3 构成一个受限循环子; 有的受限循环子包含多个变迁, 如图 4 中的 Petri 网, $\{t_4, t_5\}$ 是受限循环子.

图 3 中的 Petri 网, 设 $\hat{F} = \{(0, 0, 0, 1)\}$, 则它对应的 Petri 网语言是 $\{\text{str} \mid \text{str 是以 } t_1 \text{ 开头, } t_4 \text{ 结尾, } t_2 \text{ 和 } t_3 \text{ 出现的次数相等呈 } 2:3 \text{ 的关系, 且在任意前缀中 } t_2 \text{ 出现的次数和 } t_3 \text{ 出现的次数之比不小于 } 2:3\}$. 该语言是上下文无关语言, 而不是正则语言. 对于一般的情况, 该结论也成立.

命题6 假设 (PN, \hat{F}) 有有效递增子, 则 PN 对应的 Petri 网语言不是正则语言.

证明 为了简化证明, 这里假设 (PN, \hat{F}) 中只有一个有效递增子 \hat{R} . 若 (PN, \hat{F}) 有有效递增子, 根据命题 5 可知, 至少存在一个受限循环子, 它的每次引发会导致由有效递增子产生的标记数量的减少. 当然受限循环子可以有多个, 但为了简化证明, 这里假设只有一个受限循环子 \hat{R} . 根据 Petri 网的

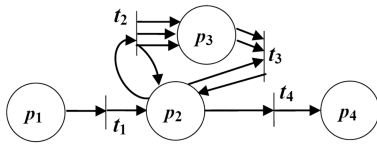


图 3 Petri 网语言是上下文无关语言的实例, $M_0 = (1, 0, 0, 0)$

Figure 3 A Petri net generating a context-free language, in which $M_0 = (1, 0, 0, 0)$

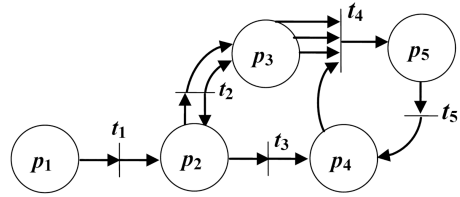


图 4 包含多个变迁的受限循环子的实例, $M_0 = (1, 0, 0, 0, 0)$

Figure 4 A Petri net with a constrained iteration containing more than one transition, in which $M_0 = (1, 0, 0, 0, 0)$

引发规则, 有效递增子 \hat{R} 的引发次数和受限循环子 \check{R} 的引发次数之间呈线性关系, 也就是在该 Petri 网对应的语言中, \hat{R} 和 \check{R} 的出现次数满足一定的线性关系, 也就是形如 $\dots \tau_1^{k_1 \cdot m} \dots \tau_2^{k_2 \cdot m + k_3} \dots$ 的序列, 这里 k_1, k_2 和 k_3 是常整数, τ_1 和 τ_2 分别是 \hat{R} 和 \check{R} 的可无限引发的序列, $\tau_1^{k_1 \cdot m}$ 和 $\tau_2^{k_2 \cdot m + k_3}$ 之间有有限个符号, 且不包含简单循环子的循环引发序列. 根据正则语言的“泵”引理可知 PN 对应的 Petri 网语言不是正则语言.

上述定理说明, 当一个 Petri 网中存在有效递增子时, 其对应的 Petri 语言就不是正则语言了, 那么该语言是上下文无关的呢? 还是上下文相关的呢? 下面进一步分析这个问题.

对于图 5 中的 Petri 网, 设 $\hat{F} = \{(0, 0, 0, 0, 1)\}$, $\{t_1\}$ 是递增子, 用 x_1 表示它的引发次数, $\{t_3\}$, $\{t_4\}$ 和 $\{t_5\}$ 都是受限循环子, 分别用 x_3, x_4, x_5 表示它们的引发次数, 要使无界位置 p_2 和 p_3 中的标记数为 0 需要满足 $3 \cdot x_1 = 2 \cdot x_3 + x_4 + x_5$ 和 $2 \cdot x_1 = 2 \cdot x_3$, 也就得到一个线性方程 $x_1 = x_4 + x_5$. 下面用线性子来描述这种关系.

定义 6 对于 (PN, \hat{F}) , 设 \hat{R} 是 PN 的有效递增子, L 是受限循环子的集合. 如果 \hat{R} 的引发次数和 L 中受限循环子的引发次数之间呈固定的线性关系, 但 \hat{R} 和 L 的任何真子集都不具备上述性质, 则称 L 为 \hat{R} 的线性子. 用 L 及其下标表示线性子.

给出了线性子这个概念后, 就可以用下面的命题来清晰地说明什么样的 Petri 网对应的 Petri 网语言是上下文无关的, 什么样的 Petri 网对应的 Petri 网语言是上下文有关的.

命题 7 假设 (PN, \hat{F}) 只有一个有效递增子 \hat{R} , 则

- (1) 若 \hat{R} 只有一个线性子, 则 PN 对应的 Petri 网语言是上下文无关的, 但不是正则的.
- (2) 若 \hat{R} 有两个或者两个以上的线性子, 则 PN 对应的 Petri 网语言是上下文相关的, 但不是上下文无关的.

证明 (1) 首先根据命题 6 可知 PN 对应的 Petri 网语言不是正则的. 下面构造一个下推自动机来识别 PN 对应的 Petri 网语言. 首先构造出 PN 的可达树, 该可达树有几个节点, 对应的下推自动机就有几个状态, 对于状态转换函数: 如果状态 q_i 对应的标识在 t 的引发下转化到了 q_j 对应的标识, 则 $(q_i, t) = q_j$; 若 t 的引发不会导致标识中无穷量的出现, 则堆栈的信息不变, 如果 t 的引发会导致无穷量的出现、或者导致无界位置中标记个数的增加, 则需要向堆栈中压入 t , 压入 t 的个数由弧的条数决定; 如果 t 的引发会导致无界位置中标记个数的减少, 则需要从堆栈中弹出信息, 弹出的个数也是由弧的条数决定. 如此构造的下推自动机所识别的语言和 PN 对应的 Petri 语言相同.

(2) 文献 [5] 已经证明了 Petri 网语言是上下文相关的. 再根据上下文无关语言的“泵”引理可知 PN 对应的 Petri 网语言不是上下文无关的.

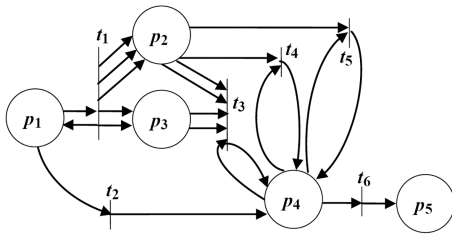


图 5 线性子中包含多个循环子的 Petri 网实例, $M_0 = (1, 0, 0, 0, 0)$

Figure 5 A Petri net with a linear containing more than one circles, in which $M_0 = (1, 0, 0, 0, 0)$

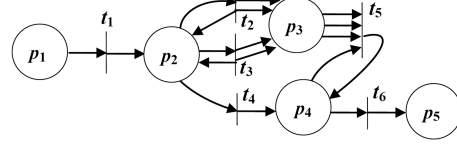


图 6 循环子不同, 但线性子相同的 Petri 网实例, $M_0 = (1, 0, 0, 0, 0)$

Figure 6 A Petri net with different iterations, but the same linearer, in which $M_0 = (1, 0, 0, 0, 0)$

上面的定理给出了只有一个递增子的情况下如何判断 PN 对应的 Petri 网语言的属性. 如果在一个 Petri 网中存在多个有效递增子, 首先考虑线性子个数最多的有效递增子, 如果线性子的个数在两个及其之上, 根据上述命题 7 以及上下文相关语言关于并、并发等运算的封闭性可知这样的 Petri 网对应的 Petri 网语言不是上下文无关的, 而是上下文相关的; 如果只有一个线性子, 需要分情况讨论. 假设 (PN, \dot{F}) 有两个有效递增子 $\hat{\mathcal{R}}_1$ 和 $\hat{\mathcal{R}}_2$, 它们的线性子分别为 L_1 和 L_2 , 则分为如下情况分别讨论:

- (1) 有效递增子的线性子都相同.
- (2) 有效递增子的线性子不相同的情况,
 - (i) 递增循环子 $\hat{\mathcal{R}}_1$ 和 $\hat{\mathcal{R}}_2$, 以及线性子 L_1 和 L_2 都能并发引发.
 - (ii) 递增循环子 $\hat{\mathcal{R}}_1$ 和 $\hat{\mathcal{R}}_2$, 以及线性子 L_1 和 L_2 中有 3 个能并发引发.
 - (iii) 递增循环子 $\hat{\mathcal{R}}_1$ 和 $\hat{\mathcal{R}}_2$, 以及线性子 L_1 和 L_2 中有 2 个能并发引发.
 - (iv) 递增循环子 $\hat{\mathcal{R}}_1$ 和 $\hat{\mathcal{R}}_2$, 以及线性子 L_1 和 L_2 都不能并发引发, 而只能顺序引发.

下面逐一讨论上述情况. 首先, 两个有效递增子都只有一个线性子, 且相同. 如图 6 中 Petri 网 PN, 设 $\dot{F} = \{(0, 0, 0, 0, 1)\}$, 则 $\{t_2\}$ 和 $\{t_3\}$ 都是 (PN, \dot{F}) 的有效递增子, $\{t_2\}$ 和 $\{t_3\}$ 的线性子都是 $\{t_5\}$, PN 对应的 Petri 网语言 $t_1 t_2^m t_3^n t_4 t_5^k t_6 (3 \cdot k = 2 \cdot m + 2 \cdot n)$ 是上下文无关的.

命题 8 假设 (PN, \dot{F}) 只有两个有效递增子, 若它们的线性子相同, 则 PN 对应的 Petri 网语言是上下文无关的.

证明 对于这种情况, 只可能是两个递增子都使得同样的位置变成了无界位置, 除此之外, 其他的就和命题 7 中的情况一样了, 类似地可以构造一个下推自动机来识别 PN 对应的 Petri 网语言.

定义 7 只要线性子中存在一个受限循环子和一个或者一组变迁能并发引发, 则称该线性子和该变迁或者该组变迁是可并发引发的.

第 1 种情况: 有效递增循环子 $\hat{\mathcal{R}}_1$ 和 $\hat{\mathcal{R}}_2$, 以及线性子 L_1 和 L_2 能并发引发. 如图 7 中的 Petri 网, 设 $\dot{F} = \{(0, 0, 0, 0, 1)\}$. t_2, t_3, t_4 和 t_6 都能并发引发, 所以存在引发序列 $t_1 t_2^m t_3^n t_4^k t_6^l t_5$. 根据上下文无关语言的“泵”引理可知, 该语言不是上下文无关的, 而是上下文有关的.

命题 9 设 (PN, \dot{F}) 只有两个有效递增子 $\hat{\mathcal{R}}_1$ 和 $\hat{\mathcal{R}}_2$, L_1 和 L_2 分别是它们的线性子, 且 $\hat{\mathcal{R}}_1, \hat{\mathcal{R}}_2, L_1$ 和 L_2 能并发引发, 则 PN 对应的 Petri 网语言不是上下文无关的, 而是上下文有关的.

证明 首先文献 [5] 中已经证明了 Petri 网语言是上下文有关的, 所以只要再证明这种 Petri 网对应的 Petri 网语言不是上下文无关的. 首先, 该语言的特征为 $\hat{\mathcal{R}}_1$ 和 L_1 的引发次数呈线性关系、 $\hat{\mathcal{R}}_2$ 和 L_2 的引发次数呈线性关系. 设 L_1 和 L_2 中都只有一个受限循环子, 分别为 $\hat{\mathcal{R}}_1$ 和 $\hat{\mathcal{R}}_2$. 既然这 4 个

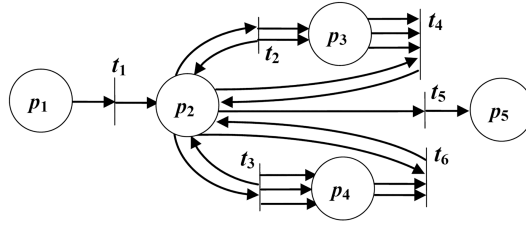


图 7 有效递增子和线性子都能并发执行的 Petri 网实例, $M_0 = (1, 0, 0, 0, 0)$

Figure 7 A Petri net whose valid incrementers and their linearers can be fired concurrently, in which $M_0 = (1, 0, 0, 0, 0)$

循环子是可以并发引发的, 可以限定它们的引发次序为 $\hat{\mathcal{R}}_1, \hat{\mathcal{R}}_2, \check{\mathcal{R}}_1$ 和 $\check{\mathcal{R}}_2$, 并且假设它们的循环引发序列分别为 τ_1, τ_2, τ_3 和 τ_4 , 则会出现如下的引发序列 $\tau_1^{k_1 \cdot m} \dots \tau_2^{k_2 \cdot n} \dots \tau_3^{k_3 \cdot m + k_5} \dots \tau_4^{k_4 \cdot n + k_6}$, 这里 k_1, k_2, k_3, k_4, k_5 和 k_6 都是常数, 且 $\tau_1^{k_1 \cdot m}$ 和 $\tau_2^{k_2 \cdot n}$ 之间、 $\tau_2^{k_2 \cdot n}$ 和 $\tau_3^{k_3 \cdot m + k_5}$ 之间、 $\tau_3^{k_3 \cdot m + k_5}$ 和 $\tau_4^{k_4 \cdot n + k_6}$ 之间都不存在不属于 L_1 和 L_2 的循环引发序列. 假设 N 就是上下文无关语言“泵”引理中的那个常数, 当 m 和 n 都大于 N 时, 上述引发序列不再满足上下文无关语言“泵”引理的条件, 据此可以知该命题成立. 对于线性子中包含多个受限循环子的情况, 也可以类似证明.

第 2 种情况: 有效递增子 $\hat{\mathcal{R}}_1$ 和 $\hat{\mathcal{R}}_2$, 以及线性子 $\check{\mathcal{R}}_1$ 和 $\check{\mathcal{R}}_2$ 中有 3 个能并发引发, 有一个和它们只能顺序引发, 这存在两种情况, 一种是某个递增子和其他 3 个顺序引发, 另外一种是一个线性子和其他 3 个顺序引发. 图 8 中的 Petri 网, 设终止标识集为 $\{(0, 0, 0, 0, 1)\}$, 则图 8 中的两个 Petri 网分别代表上述两种情况. 如图 8(a) 的 Petri 网中 t_3, t_4 和 t_5 都能并发执行, 所以存在引发序列 $t_2^m t_1 t_3^m t_4^m t_5^m t_6$; 如图 8(b) 的 Petri 网中 t_2, t_3 和 t_4 都能并发引发, 所以存在引发序列 $t_1 t_2^m t_3^m t_4^m t_5 t_6^n$. 根据上下文无关语言的“泵”引理可知, 它们都不是上下文无关的, 而是上下文有关的.

命题 10 设 (PN, \hat{F}) 只有两个有效递增子 $\hat{\mathcal{R}}_1$ 和 $\hat{\mathcal{R}}_2$, L_1 和 L_2 分别是它们的线性子, 且 $\hat{\mathcal{R}}_1, \hat{\mathcal{R}}_2, L_1$ 和 L_2 中有 3 个能并发引发, 则 PN 对应的 Petri 网语言是上下文有关的.

证明 假设 L_1 和 L_2 中都只有一个受限循环子. 因为 $\hat{\mathcal{R}}_1, \hat{\mathcal{R}}_2, L_1$ 和 L_2 中有 3 个能并发引发, 不妨假设 $\hat{\mathcal{R}}_1, \hat{\mathcal{R}}_2$ 和 L_1 是可以并发引发的, L_1 中可并发引发的受限循环子为 $\check{\mathcal{R}}_1$, 限定它们的引发次序为 $\hat{\mathcal{R}}_1, \hat{\mathcal{R}}_2$ 和 $\check{\mathcal{R}}_1$, 并且假设它们的循环引发序列分别为 τ_1, τ_2 和 τ_3 , 则会出现如下的引发序列 $\tau_1^{k_1 \cdot m} \dots \tau_2^{k_2 \cdot n} \dots \tau_3^{k_3 \cdot m + k_4}$, 这里 k_1, k_2, k_3 和 k_4 都是常数, 且 $\tau_1^{k_1 \cdot m}$ 和 $\tau_2^{k_2 \cdot n}$ 之间、 $\tau_2^{k_2 \cdot n}$ 和 $\tau_3^{k_3 \cdot m + k_4}$ 之间都不存在不属于 L_1 和 L_2 的循环引发序列. 假设 N 是上下文无关语言“泵”引理中的常数, 当 m 和 n 都大于 N 时, 上述引发序列不再满足上下文无关语言“泵”引理的条件, 由此可知该命题成立. 对于 L_1 和 L_2 中包含多个受限循环子的情况, 也可类似地证明.

第 3 种情况: 设有效递增子 $\hat{\mathcal{R}}_1$ 只有一个线性子 L_1 , 有效递增子 $\hat{\mathcal{R}}_2$ 只有一个线性子 L_2 , $\hat{\mathcal{R}}_1$ 和 L_1 其中之一能与 $\hat{\mathcal{R}}_2$ 和 L_2 其中之一并发引发. 这存在 3 种可能情况: 一种是两个递增子能并发引发; 第 2 种是一个递增子和一个线性子; 第 3 种情况就是两个线性子能够并发引发. 例如, 图 9 中的 Petri 网, 终止标识集为 $\{(0, 0, 0, 0, 0, 1)\}$, 这 3 个 Petri 网分别对应着上述 3 种情况. 对于图 9(a) 中的 Petri 网, t_2 和 t_4 能并发引发, 所以存在引发序列 $t_1 t_2^m t_4^m t_3 t_5^m t_6 t_7^n$; 对于图 9(b) 中的 Petri 网, t_4 和 t_6 能并发引发, 所以存在引发序列 $t_1 t_3^m t_2 t_4^m t_6^m t_5 t_7^n$; 对于图 9(c) 中的 Petri 网, t_6 和 t_7 能并发引发, 所以存在引发序列 $t_1 t_3^m t_2 t_4^m t_6^m t_5 t_7^n$. 根据上下文无关语言的“泵”引理可知, 它们都不是上下文无关的, 而是上下文有关的. 对于一般情况有如下结论.

命题 11 设 (PN, \hat{F}) 只有两个有效递增子 $\hat{\mathcal{R}}_1$ 和 $\hat{\mathcal{R}}_2$, L_1 和 L_2 分别是它们的线性子, 并且 $\hat{\mathcal{R}}_1$ 和 L_1 中有一个能与 $\hat{\mathcal{R}}_2$ 和 L_2 中的一个并发引发, 则 PN 对应的 Petri 网语言是上下文有关的.

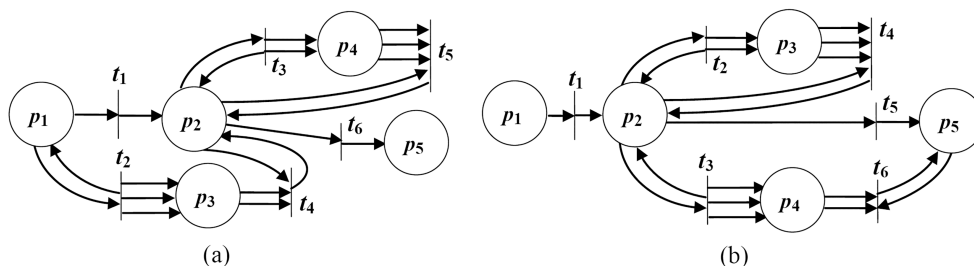


图 8 有效递增子和线性子中有 3 个能并发执行的 Petri 网实例

Figure 8 A Petri net, three of whose valid incrementers and linearers can be fired concurrently. (a) and (b) are different cases of the Petri net, in which $M_0 = (1, 0, 0, 0, 0)$

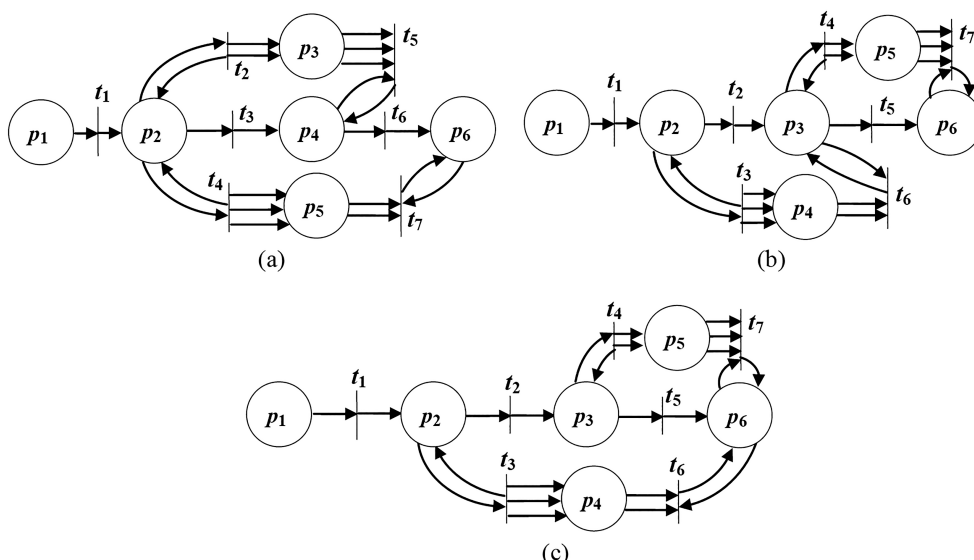


图 9 有效递增子和线性子中有两个能并发执行的 Petri 网实例

Figure 9 A Petri net, two of whose valid incrementers and linearers can be fired concurrently. (a)~(c) are different cases of the Petri net, in which $M_0 = (1, 0, 0, 0, 0, 0)$

证明 和前面几个命题的证明类似, 只要有一个循环子和一个受限循环子可以并发引发, 就存在一个引发序列不满足上下文无关语言的“泵”引理的情形, 从而说明该语言不是上下文无关的。

第 4 种情况: 有两个有效递增子 $\hat{\mathcal{R}}_1$ 和 $\hat{\mathcal{R}}_2$, L_1 和 L_2 分别是它们的线性子, 并且 $\hat{\mathcal{R}}_1$ 和 L_1 中没有任何变迁能与 $\hat{\mathcal{R}}_2$ 和 L_2 中的任意变迁并发引发, 存在如下 4 种可能的情况。

(1) 第 1 种情况是顺序引发: $\hat{\mathcal{R}}_1$ 和 L_1 先引发, $\hat{\mathcal{R}}_2$ 和 L_2 再引发。例如, 图 10(a) 和 (b) 中的 Petri 网 PN, 终止标识集为 $\{(0, 0, 0, 0, 0, 1)\}$, PN 对应的 Petri 语言仍然是上下文无关的。

(2) 第 2 种就是交叉引发: 按照 $\hat{\mathcal{R}}_1, \hat{\mathcal{R}}_2, L_1$ 和 L_2 的顺序进行引发。如图 10(c) 中的 Petri 网 PN, 终止标识集为 $\{(0, 0, 0, 0, 0, 1)\}$ 。 t_2, t_3, t_5 和 t_6 交叉引发, 引发序列位 $t_1^{2 \cdot m} t_2^{3 \cdot n} t_3^{3 \cdot m} t_5^{3 \cdot m} t_7 t_6^{2 \cdot n}$, 根据上下文无关语言的“泵”引理可知, PN 对应的 Petri 语言不是上下文无关的, 所以它是上下文有关的。

(3) 第 3 种情况是嵌套引发: 按照 $\hat{\mathcal{R}}_1, \hat{\mathcal{R}}_2, L_2$ 和 L_1 的顺序进行引发。如图 10(d) 中的 Petri 网, 终止标识集为 $\{(0, 0, 0, 0, 0, 1)\}$ 。 t_3 和 t_5 被嵌入到了 t_1 和 t_7 之间, 引发序列为 $t_1^{2 \cdot m} t_2 t_3^{3 \cdot n} t_5^{3 \cdot m} t_7 t_6^{2 \cdot n}$, PN 对应的 Petri 语言仍然是上下文无关的。

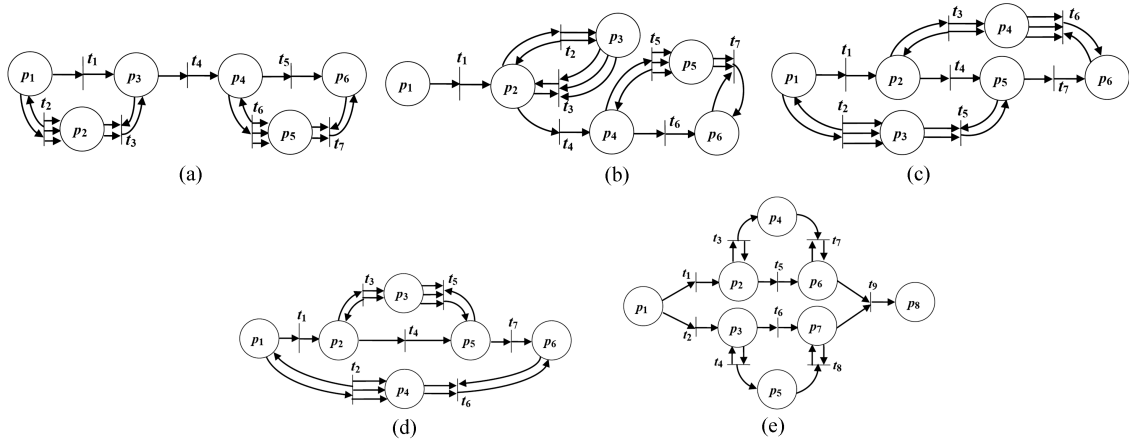


图 10 两个递增子和线性子的顺序引发、交叉引发、嵌套引发和选择引发实例

Figure 10 Illustrating that two valid incrementers and their linearizers can be fired sequentially, intersepectively, nestedly and alternatively. (a)~(e) are different cases of the Petri net, in which $M_0 = (1, 0, 0, 0, 0, 0)$

(4) 第 4 种情况是选择引发: 要么 $\hat{\mathcal{R}}_1$ 和 L_1 能引发, 要么 $\hat{\mathcal{R}}_2$ 和 L_2 能引发, 但它们只能有一组可以引发. 如图 10(e) 中的 Petri 网, 设 $\dot{F} = \{(0, 0, 0, 0, 0, 0, 0, 1)\}$, PN 对应的 Petri 网语言为 $t_1(t_3^m t_5 t_7^m | t_4^n t_6 t_8^n)t_9$, 即 $t_1 t_3^m t_5 t_7^m t_9 \cup t_1 t_4^n t_6 t_8^n t_9$.

命题12 设 (PN, \dot{F}) 只有两个有效递增子 $\hat{\mathcal{R}}_1$ 和 $\hat{\mathcal{R}}_2$, 它们分别只有一个线性子 L_1 和 L_2 , 且 $\hat{\mathcal{R}}_1$ 和 L_1 中没有任何变迁能与 $\hat{\mathcal{R}}_2$ 和 L_2 中任意变迁并发引发.

- (1) 如果 $\hat{\mathcal{R}}_1$ 和 L_1 与 $\hat{\mathcal{R}}_2$ 和 L_2 是顺序引发, PN 对应的 Petri 网语言是上下文无关的.
- (2) 如果 $\hat{\mathcal{R}}_1$ 和 L_1 与 $\hat{\mathcal{R}}_2$ 和 L_2 是交叉引发, PN 对应的 Petri 网语言不是上下文无关的, 而是上下文相关的.
- (3) 如果 $\hat{\mathcal{R}}_1$ 和 L_1 与 $\hat{\mathcal{R}}_2$ 和 L_2 是嵌入引发, PN 对应的 Petri 网语言是上下文无关的.
- (4) 如果 $\hat{\mathcal{R}}_1$ 和 L_1 与 $\hat{\mathcal{R}}_2$ 和 L_2 是选择引发, PN 对应的 Petri 网语言是上下文无关的.

证明 (1) 顺序引发时, 相当于语言间的连接运算, 两个上下文语言的连接仍是上下文无关语言, 所以 PN 对应的 Petri 网语言是上下文无关的.

(2) 因为 $\hat{\mathcal{R}}_1$ 和 L_1 与 $\hat{\mathcal{R}}_2$ 和 L_2 是交叉引发, 则存在循环引发序列 τ_1, τ_2, τ_3 和 τ_4 和引发序列 $\tau_1^{k_1 \cdot m} \dots \tau_3^{k_2 \cdot n} \dots \tau_2^{k_3 \cdot m + k_5} \dots \tau_4^{k_4 \cdot n + k_6}$, 这里 k_1, k_2, k_3, k_4, k_5 和 k_6 都是常数, 且 $\tau_1^{k_1 \cdot m}$ 和 $\tau_3^{k_2 \cdot n}, \tau_3^{k_2 \cdot n}$ 和 $\tau_2^{k_3 \cdot m + k_5}, \tau_2^{k_3 \cdot m + k_5}$ 和 $\tau_4^{k_4 \cdot n + k_6}$ 之间不包含不属于 L_1 和 L_2 的循环引发序列. 这个引发序列不满足上下文无关语言“泵”引理的条件, 所以 PN 对应的 Petri 网语言不是上下文无关的, 而是上下文有关的.

(3) 嵌入引发时, 可以构造一个下推自动机来识别该语言, 这个构造过程和命题 7 中 (1) 的证明类似, 不同的是这里需要两次压栈的过程, 两次退栈的过程. 据此可知, PN 对应的 Petri 网语言是上下文无关的.

(4) 选择引发时, 相当于两个语言的并运算. 两个上下文无关语言的并仍然是上下文无关的, 所以 PN 对应的 Petri 网语言是上下文无关的.

推论1 设 (PN, \dot{F}) 只有两个有效递增子 $\hat{\mathcal{R}}_1$ 和 $\hat{\mathcal{R}}_2$, 它们分别只有一个线性子 L_1 和 L_2 .

(1) 如果 $\hat{\mathcal{R}}_1$ 和 L_1 中的循环子与 $\hat{\mathcal{R}}_2$ 和 L_2 中的循环子是顺序引发、或是嵌入引发、或是选择引发, 则 PN 对应的 Petri 网语言是上下文无关的.

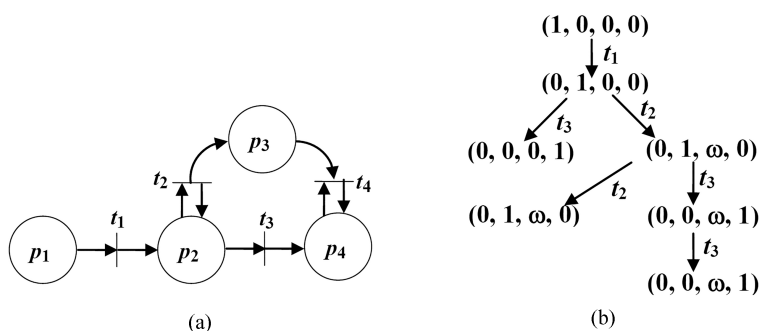


图 11 可达树实例

Figure 11 An instance of reachability tree. (a) $M_0 = (1, 0, 0, 0)$; (b) reachability tree

(2) 如果 $\hat{\mathcal{R}}_1$ 和 L_1 中的循环子与 $\hat{\mathcal{R}}_2$ 和 L_2 中的循环子有交叉引发, PN 对应的 Petri 网语言不是上下文无关的, 而是上下文相关的.

证明 根据命题 12 可知上述结论成立.

根据前面的命题和推论可以得到判断 Petri 网 PN 对应的 Petri 网语言属性的步骤如下:

第 1 步. 寻找有效的递增子.

第 2 步. 如果没有有效递增子, 则 PN 对应的 Petri 网语言为正则语言.

第 3 步. 如果有有效递增子, 则判断每一个有效递增子的线性子的个数.

(1) 若有效递增子有两个及其以上的线性子, 则 PN 对应的 Petri 网语言为上下文有关的.

(2) 若所有的有效递增子都只有一个线性子, 则

(a) 若只有一个有效线性子, 则 PN 对应的 Petri 网语言是上下文无关的.

(b) 若存在两个及其以上有效递增子,

- 如果存在两个有效递增子 $\hat{\mathcal{R}}_1$ 和 $\hat{\mathcal{R}}_2$, 它们的线性子分别是 L_1 和 L_2 . $\hat{\mathcal{R}}_1, \hat{\mathcal{R}}_2, L_1$ 和 L_2 中 4 个能并发引发、或者 3 个能并发引发、或者 $\hat{\mathcal{R}}_1$ 和 L_1 中有一个能与 $\hat{\mathcal{R}}_2$ 和 L_2 中的一个能并发引发、或者能交叉引发, 则 PN 对应的 Petri 网语言是上下文相关的.

- 如果任意两个有效递增子 $\hat{\mathcal{R}}_1$ 和 $\hat{\mathcal{R}}_2$ 以及它们的线性子 L_1 和 L_2 之间是顺序引发、或者嵌入引发、或者选择引发, 则 PN 对应的 Petri 网语言是上下文无关的.

3 判断有效递增子的方法

本节主要介绍有效递增子的判断方法. 有效递增子的判定分为两步, 第 1 步判断是否存在递增子, 第 2 步判断递增子的有效性.

3.1 判断是否存在递增子的方法

Petri 网的分析技术有多种^[1,2], 例如可达树、矩阵分析等, 本文采用改进的可达树来判断是否存在递增子. 首先用一个例子来说明如何根据可达树来判断递增子.

对于图 11(a) 的 Petri 网, 其可达树是图 11(b). 在初始标识状态下只有变迁 t_1 是使能的, 它引发后得到标识 $(0, 1, 0, 0)$, 这时变迁 t_2 和 t_3 是使能的. t_3 引发后, 得到标识 $(0, 0, 0, 1)$, 这时没有任何变迁是使能的; t_2 引发得到标识 $(0, 1, \omega, 0)$, 在该标识下 t_2 和 t_3 是使能的. 从可达树中可以看到, 在从根节

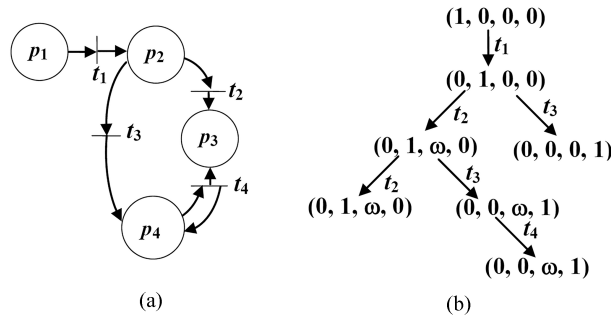


图 12 可达树导致信息丢失的 Petri 网实例

Figure 12 An instance showing information lost in the reachability tree. (a) $M_0 = (1; 0; 0; 0)$; (b) reachability tree

点到标识为 $(0, 1, \omega, 0)$ 的节点的路径上, 有一个节点的标识为 $(0, 1, 0, 0)$, 且有 $(0, 1, 0, 0) < (0, 1, \omega, 0)$. 标识 $(0, 1, \omega, 0)$ 出现了无穷量 ω , 从而可知从节点 $(0, 1, 0, 0)$ 到节点 $(0, 1, \omega, 0)$ 的路径上的变迁是可以重复引发的, 并且其引发会导致位置 p_3 中的标记数量的递增.

命题13 Petri 网 $PN = (S, T, F, W, M_0)$, 如果其可达树中某个节点 $Node_1$ 所对应的标识 M_1 中存在 ω , 则该 Petri 网中存在递增子. 假设 $Node_2$ 是从根节点到 $Node_1$ 的路径上一个节点, 其对应的标识小于 M_1 , 且 $Node_2$ 是满足上述条件的离 $Node_1$ 最近的节点, 则从 $Node_2$ 到 $Node_1$ 的路径上的变迁构成递增子.

证明 很显然, 不论从 $Node_2$ 到 $Node_1$ 的路径上的变迁序列是可以循环引发的, 所以这些变迁构成循环子. 再有, 它们的每次顺序引发都导致某位置中的标记数量的增加, 其他位置中的标记数量不变, 这样的循环子也是递增循环子, 结论成立.

命题14 如果 $PN = (S, T, F, W, M_0)$ 存在一个递增子, 则在可达树中肯定存在某个节点, 其对应的标识的某个分量中包含 ω .

证明 根据递增子的定义可知, 肯定存在某个引发序列, 它能循环引发, 并且每次引发都能使某些位置中的标记数量增加. 根据构造可达树的过程可知, 当出现某标识大于以前的某个标识的时候, 就把增大的分量变成了 ω , 所以结论成立.

因为 ω 的引入使得一些信息在可达树中丢失, 所以用可达树来分析标识的可达性具有很大的局限性. 通过可达树虽然能找到递增子, 但不能把全部的递增子都枚举出来. 例如, 图 12(a) 中的 Petri 网, 其可达树如图 12(b) 所示. 用上述方法来判断递增子, 只能得到 $\{t_2\}$ 是一个递增子, 但无法判断出 $\{t_4\}$ 也是递增子, 事实上 $\{t_4\}$ 也是一个递增子.

为了解决上述问题, 需要进一步地修改可达树判定的方法. 为了区分标识中的标记是由哪个变迁所产生的, 不仅要在标识中记下一个变迁引发导致的标记数量的变化, 还要记录下这个变化来自哪个变迁的引发. 为此, 用标记数和变迁的乘积表示该变迁的引发产生的标记的变化. 例如, 对于图 12(a) 中的 Petri 网, t_1 引发就产生标识 $(0, 1, 0, 0)$, 为了记住第 1 个分量从 1 变为了 0 和第 2 个分量中的 1 是变迁 t_1 的引发而产生的, 用形如 $(1 - 1 \cdot t_1, 1 \cdot t_1, 0, 0)$ 的向量表示. 在标识 $(0, 1, 0, 0)$ 下, t_2 和 t_3 都是使能的. 若 t_2 引发, 则得到标识 $(0, 1, 1, 0)$, 对应的向量为 $(1 - 1 \cdot t_1, 1 \cdot t_1 - 1 \cdot t_2 + 1 \cdot t_2, 1 \cdot t_2, 0)$, 化简后为 $(1 - 1 \cdot t_1, 1 \cdot t_1, 1 \cdot t_2, 0)$. 从标识的角度有 $(0, 1, 1, 0) > (0, 1, 0, 0)$; 从向量的角度来看, 也有 $(0, 1 \cdot t_1, 1 \cdot t_2, 0) > (0, 1 \cdot t_1, 0, 0)$. 根据 Peterson 的构造可达树的算法, 这时需要构造一个节点, 该节点对应的标识为 $(0, 1, \omega, 0)$. 在新型可达树中, 也构造一个标记为 $(1 - 1 \cdot t_1, 1 \cdot t_1, 1 \cdot \omega \cdot t_2, 0)$ 的节点, 其

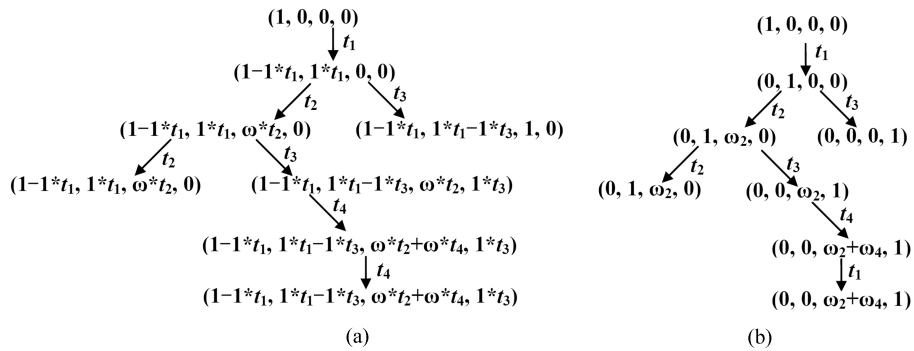


图 13 图 12(a) 中 Petri 网的 A 型可达树和 B 型可达树
 Figure 13 (a) A-type and (b) B-type reachability tree for the Petri net in Figure 12(a)

中分量 $1 \cdot \omega \cdot t_2$ 表示 t_2 的每次引发会导致该位置中增加一个标记. 为了书写方便, 把 $1 \cdot \omega \cdot t_2$ 简记为 $\omega \cdot t_2$. 图 12(a) 中 Petri 网的新型可达树如图 13(a). 把 $\omega \cdot t_i$ 变成 ω_i , 则该树就变成了图 13(b) 中的树. 把图 13(a) 中的树称为 A 型可达树, 把图 13(b) 中的树称为 B 型可达树. 从上面可以看到, 新型可达树有两种形式, 可达树中的节点也有两种形式.

为了准确定义修正后的可达树, 并讨论其性质, 给出如下规则:

- 规则 1 $\omega \cdot t_1 + \omega \cdot t_1 = \omega \cdot t_1$.
- 规则 2 $\omega \cdot t_1 + k \cdot t_1 = \omega \cdot t_1$.
- 规则 3 $\omega \cdot t_1 - k \cdot t_1 = \omega \cdot t_1$.
- 规则 4 $\omega \cdot t_1 + k \cdot t_2 > \omega \cdot t_1 (k > 0)$.
- 规则 5 $\omega_1 + \omega_2 > \omega_1$.
- 规则 6 $\omega_1 + \omega_1 = \omega_1$.
- 规则 7 $\omega_1 + k > \omega_1 (k > 0)$.

定义 8 变迁标记向量和 ω 标记向量:

- (1) 向量 V 中的分量是由“+”和“-”把常量、 $\omega \cdot t_i$ 和 $k \cdot t_i$ 连接成的表达式, 则称 V 为变迁标记向量.
- (2) 向量 V 中的分量是由“+”和“-”把常量、 ω_i 连接成的表达式, 则称 V 为 ω 标记向量.
- (3) 设 V 是变迁标记向量, 把其中的 t_i 用 1 替代, 并化简后得到向量 V' . t_i 是 V' 下使能的变迁, 对于每一个位置 p_k , 如果 t_i 的引发导致位置 p_k 中的标记增加 m 个, 则将 V 的第 k 个分量加上 $m \cdot t_i$; 如果导致位置 p_k 中的标记减少 m , 则将 V 的第 k 个分量减去 $m \cdot t_i$, 得到新的变迁标记向量 V'' , 把 V'' 称为 V 关于 t_i 的变迁标记向量.

定义 9 将变迁标记向量 V 中的变迁 t_i 用 1 代替, 化简得到标识 M , 称 M 为 V 的标识.

定义 10 两个变迁标记型向量 V_1 和 V_2 , 把 ω 和 t_i 看作正整数,

- (1) 给定两个由“+”和“-”把常量、 $\omega \cdot t_i$ 和 $k \cdot t_i$ 连接成的表达式 E_1 和 E_2 , 如果 E_1 中的常数项大于等于 E_2 中的常数项、 E_1 中 t_i 的系数大于等于 E_2 中 t_i 的系数, 则记为 $E_1 \geq E_2$; 如果上面的大于等于中有一个是大于但不等于, 则记为 $E_1 > E_2$.
- (2) 如果 V_1 的每个分量都大于等于 V_2 的对应分量, 则称 V_1 大于等于 V_2 , 记为 $V_1 \geq V_2$.
- (3) 设 $V_1 \geq V_2$, 且 V_1 中至少存在某分量大于 V_2 的对应分量, 则称 V_1 大于 V_2 , 记为 $V_1 > V_2$.

对于端点、重复节点和覆盖节点的定义, 和 Peterson^[5] 中的定义一样. 在做了如上准备后就可以定义 A 型可达树和 B 型可达树了.

定义11 A 型可达树是如下构造的一个树:

第 1 步. 构造初始标识对应的节点, 将该节点记为可扩展的, 并把它赋值给 root.

第 2 步. 选择一个可扩展的节点 Node, 设其对应的变迁标记向量为 V , V 的标识为 M . 对于在 M 下使能的变迁 t , 构造 V 关于 t 的变迁标记向量 V' , 并按照上述的规则 1~7 对向量中的表达式进行化简, 得到一个新的向量 V_1 .

(1) 如果存在某节点, 其对应的变迁标记向量和 V_1 相同, 则对 V_1 构造一个节点 $Node_1$, 把 $Node_1$ 作为 Node 的子节点, 并把该节点标记为重复节点.

(2) 如果在从根节点到 Node 节点的路径上, 存在某个节点, 它对应的向量小于 V_1 , 则对于 V_1 中大于 V 的对应分量如下处理: 该分量中的 $\omega \cdot t_i$ 的表达式不再处理; 对于形如 $k \cdot t_i$ 的表达式, 若 V 中对应的分量存在形如 $m \cdot t_i$ 的表达式, 则把 V_1 中的表达式 $k \cdot t_i$ 变为 $(k - m) \cdot \omega \cdot t_i$. 若 $k - m$ 为 1, 则把 $1 \cdot \omega \cdot t_i$ 简记为 $\omega \cdot t_i$. 这样处理后的向量记为 V_2 , 为 V_2 建立一个对应的节点作为 Node 后继节点, 并把该节点标记为可扩展.

(3) 如果上述两个条件都不满足, 则为 V_1 建立对应的节点, 并把它作为 Node 的后继节点.

第 3 步. 如果存在可扩展的节点, 则转第 2 步; 否则, 结束构造的过程.

定义12 把 A 型可达树中每个节点对应的向量中的 $\omega \cdot t_i$ 变成 ω_i , 并对表达式化简, 把这种新得到的向量作为树的节点, 树的边保持不变, 得到另外一棵树, 把这种树称为 B 型可达树.

命题15 在 B 型可达树中, 若节点 $Node_3$ 对应的标识中包含 ω_i 的表达式, 则在从根节点到 $Node_3$ 的路径上找到第一次出现 ω_i 的节点 Node, 设 Node 对应的标识为 M , 再在从根节点到 Node 的路径上找到节点 $Node_1$, $Node_1$ 对应的标识 M_1 小于 M , 则在从 $Node_1$ 到 Node 的路径上的变迁形成一个循环引发序列:

(1) 如果该引发序列的引发只导致某些位置中的标记数量的增加, 而不会使得任何位置的标记数量减少, 则该引发序列中的变迁构成一个递增子.

(2) 如果该引发序列的引发在导致某些位置中的标记数量的增加的同时, 还会使得某些位置的标记数量减少, 则该引发序列中的变迁构成一个受限循环子.

证明 (1) 根据 A 型可达树和 B 型可达树的定义, 当把 M 还原为原来的状态 M' 时, 也就是 ω_i 的值变成应有的常数时, M' 仍然大于 M_1 , 这也就决定了从 $Node_1$ 到 Node 之间路径上的变迁是可以循环引发的, 并且它的引发还导致了某些位置中的标记数量的增加, 再根据递增子的定义可知, 该引发序列中的变迁构成了一个递增子. 图 12(a) 中的 Petri 网的变迁 t_2 就属于这种情况.

(2) 根据给出的条件可知从 $Node_1$ 到 Node 的路径上的变迁形成的引发序列是能循环引发的, 因为它的引发能导致某些位置中标记的数量减少, 根据 A 型可达树和 B 型可达树的定义可知, 它把某些无界位置中的标记移走了. 如果它移走标记的位置不是无界, 而是有界, 则就不会出现 ω_i 这个符号, 这与命题的条件是矛盾的. 所以, 该引发序列中的变迁构成了一个受限循环子.

3.2 递增子的有效性判定

3.1 小节给出了判断递增子的方法, 本小节讨论如何判断递增子的有效性. 先来分析一下 A 型可达树和 B 型可达树. 对于图 5 中的 Petri 网, 它的 A 型和 B 型可达树如图 14 所示. 设 $\hat{P} = \{(0, 0, 0, 0, 0, 1)\}$, 引发序列 $t_1 t_3 t_5$ 得到的标识属于终止集. 在 B 型可达树中, 设标识 M 的某些分量包

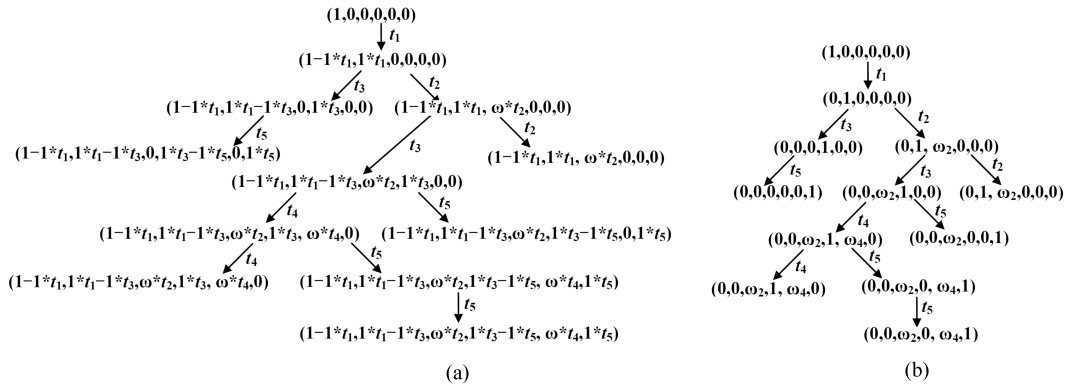


图 14 图 5 中 Petri 网的 A 型可达树和 B 型可达树

Figure 14 (a) A-type and (b) B-type reachability tree for the Petri net in Figure 5

含 ω , 如果在终止标识集中存在一个标识 M' , M 和 M' 中为常数的分量都相同, 则进一步检验是否存在一个引发序列使得 M 中包含 ω 的分量也和 M' 中的对应分量相同. 从其可达树中可以看出, 它有一个递增子 $\{t_2\}$. $\{t_4\}$ 是循环子, 随着 t_4 的引发, p_2 中的标记数量减少, 只要 t_4 的引发次数和 t_2 的引发次数相等, 则位置 p_2 中的标记数量也变为了 0, 和 M' 中的数值相同了, 也就是说 p_2 中的标记的数量由 t_4 和 t_2 的引发次数决定的. 如果用 x_i 表示变迁 t_i 引发的次数, 则 p_2 中标记的个数为 $x_2 - x_4$, 再由终止标识, 得到方程式 $x_2 - x_4 = 0$. 同样地, t_6 的引发, 也会导致位置 p_4 中标记数量的减少, 只要 t_6 的引发次数和 t_4 的引发次数相同, 则 p_4 中的标记数量也变为了 0, 也就是要求 $x_4 - x_6 = 0$. 据此得到如下方程组:

$$\begin{cases} x_2 - x_4 = 0, \\ x_4 - x_6 = 0. \end{cases} \quad (1)$$

因为 $\{t_2\}$ 是一个递增子, 它的引发次数不受限制, 而 t_4 和 t_6 的引发次数是受限制的, 所以, 可以把 x_2 看作为常量, x_4 和 x_6 作为变量, 求解该方程. 该方程有一个正整数解, 就存在一个引发序列到达终止标识集中的标识.

图 12(a) 中的 Petri 网的 B 型可达树如图 13(b) 所示. 设 $\dot{F} = \{(0, 0, 0, 1)\}$, 则标识 $(0, 0, 0, 1)$, $(0, 0, \omega_2, 1)$ 和 $(0, 0, \omega_2 + \omega_4, 1)$ 分别与 $(0, 0, 0, 1)$ 匹配, 对于标识 $(0, 0, \omega_2, 1)$ 来说, p_3 中的标记只由 t_2 的引发产生的, 并且由终止标识中 p_3 的标记数为 0 可知, $x_2 = 0$, 也就是 t_2 不引发. 同样地, 对于标识 $(0, 0, \omega_2 + \omega_4, 1)$ 来说, p_3 中的标记只由 t_2 和 t_4 的引发产生, 并且由终止标识中 p_3 的标记数为 0 可知, $x_2 + x_4 = 0$; 从而可知 $x_2 = 0$ 和 $x_4 = 0$, 也就是 t_2 和 t_4 都不能引发. 从该例中也得到一个结论: 如果该方程不存正整数解, 则就不存在一个引发序列到达终止标识集中的标识.

定义13 对于 (PN, \dot{F}) , 设 PN 的 B 型可达树为 \mathfrak{t} , $M \in \dot{F}$. 对于 \mathfrak{t} 中的一个节点 Node 对应的标识 M , 若 M 中为常量的分量和 M' 的对应分量都相等, 则令 M 中每一个包含 ω 的分量等于 M' 中对应分量建立一个方程, 且把 ω_i 变为 x_i . 这样的方程构成一个方程组, 称为 M 和 M' 的关联方程组.

定理1 对于 (PN, \dot{F}) , 设 \mathfrak{t} 为 PN 的 B 型可达树. 设 \mathfrak{t} 中节点 Node 对应的标识为 M , 如果 \dot{F} 中存在一个标识 M' , M 的常量分量和 M' 的对应分量都相等, 则 M 和 M' 的关联方程组有正整数解, 当且仅当存在包含递增子的引发序列 τ 满足 $M_0[\tau]M'$, 且每个变量的值就是其对应的循环子在 τ 中的出现次数.

证明 首先, 若 M 和 M' 的关联方程组有正整数解, 根据 B 型可达树的定义可知, 从根节点开始到 Node 的路径上, 各个变迁引发的次序按照路径上的次序即可, 对于出现在关联方程组中的变迁, 在其引发时可以连续的次数等于对应变量的值, 这就得到了引发序列使得 $M_0[\tau]M'$.

其次, 如果存在包含递增子的引发序列 τ 使得 $M_0[\tau]M'$. 因为有递增子在引发序列中出现, 在 B 型可达树中肯定有一个节点, 其对应的标识中存在 ω . 包含 ω 的分量对应的位置是无界的, 且满足 M 和 M' 的关联方程组, 也就是方程组存在正整数解.

定理2 对于 (PN, \dot{F}) , 设 t 为 PN 的 B 型可达树. 设 t 中某节点对应的标识为 M , \dot{F} 中存在一个标识 M' , M 和 M' 的常数分量对应相等, M 和 M' 的关联方程组有无穷多个正整数解, 当且仅当 PN 中存在有效递增子.

证明 根据定理 1 可知如果关联方程组的一个解对应着一个引发序列, 同样的一个引发序列对应着关联方程组的解. 所以, 如果关联方程组存在无穷多个正整数解, 当且仅当存在无穷多个引发序列能使初始状态到达终止状态. 如果存在无穷多个引发序列能使初始状态到达终止状态, 当且仅当对于任意的正整数 N , 总存在一个引发序列 τ , τ 的长度大于 N , 且 τ 的引发可以使初始状态到达终止状态, 这说明了 PN 中存在有效递增子.

根据上面的定理可以得到判断有效递增子的步骤如下:

第 1 步. 构造 PN 的 B 型可达树.

第 2 步. 对于树中的每一个节点 Node, 构造其对应的变迁标记向量 V 和 V 的标识 M .

第 3 步. 判断终止标识集中是否存在一个标识 M' , M 中为常量的分量和 M' 中的对应分量相等. 如果不存在这样的标识, 说明从该初始标识开始不存在引发序列可达标识集中的任何标识; 判断结束. 如果存在这样的节点, 进行第 4 步.

第 4 步. 构造 M 和 M' 的关联方程组, 如果方程组存在无数解, 则存在有效递增子.

4 相关工作的比较

本文主要做了两个方面的工作: 第一, 给出了判定无标注 Petri 网的语言属性的判定方法; 第二, 就是给出了判断有效递增子的方法. 吴^[8]研究的问题和本文研究的问题相同, 他给出了重复向量、传递向量的概念, 并通过是否存在重复向量、是否存在传递向量来判定一个 Petri 网对应的 Petri 网语言的属性. 吴给出的结论是针对变迁顺序引发的 Petri 网, 例如他在文中给出的如图 15 中的例子, 就是针对变迁顺序引发的情况. 对于该例, 采用本文给出的方法也能得到同样的判定结论. 吴^[8]讨论的是标准 Petri 网, 所以图 15 中 Petri 网的终止集分别为 $\{(0, 0, 0, 0, 0, 0, 1)\}$ 和 $\{(0, 0, 0, 0, 0, 0, 0, 0, 0, 1)\}$. 按照吴的方法, 图 15(a) 中的 Petri 网, 因为不存在有效传递向量, 所以该 Petri 网对应的 Petri 网语言是上下文无关的; 图 15(b) 中的 Petri 网, 因为存在有效传递向量, 所以该 Petri 网对应的 Petri 网语言是上下文有关的. 用本文的方法也能得到同样的结果, 不论图 15(a) 中的 Petri 网, 还是图 15(b) 中的 Petri 网, 它们都只有一个有效递增子, 图 15(a) 中的 Petri 网在终止集 $\{(0, 0, 0, 0, 0, 0, 1)\}$ 下只有一个有效递增子, 且该递增子只有一个线性子, 该线性子由受限循环子 $\{t_5, t_6\}$ 构成, 所以它对应的 Petri 网语言是上下文无关的; 图 15(b) 中的 Petri 网在终止集 $\{(0, 0, 0, 0, 0, 0, 0, 0, 0, 1)\}$ 下只有一个有效递增子, 该有效递增子有两个线性子, 这两个线性子分别由受限循环子 $\{t_5, t_6\}$ 和 $\{t_8, t_9\}$ 构成, 所以它对应的 Petri 网语言是上下文有关的.

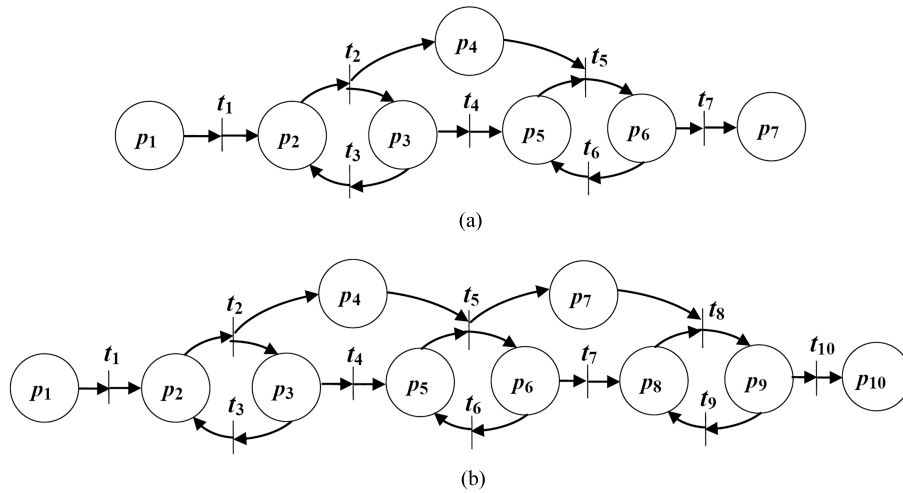


图 15 文献 [8] 中的实例

Figure 15 An Instance shown in reference [8]. (a) $M_0 = (1, 0, 0, 0, 0, 0, 0)$; (b) $M_0 = (1, 0, 0, 0, 0, 0, 0, 0, 0, 0)$

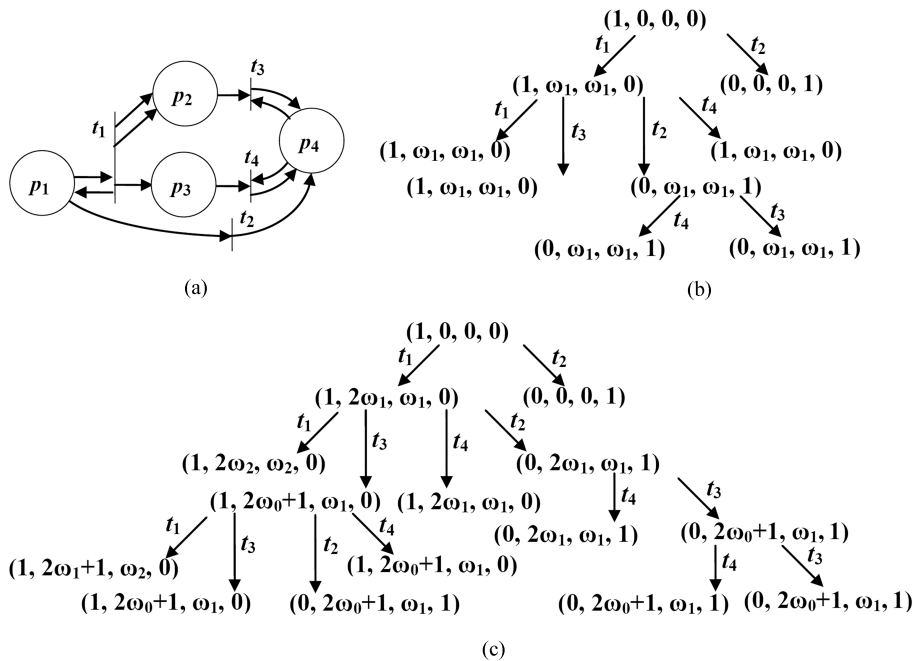


图 16 无传递向量的 Petri 网及其 NMRT 树

Figure 16 A Petri net without transitive vectors and its NMRT tree. (a) $M_0 = (1, 0, 0, 0)$; (b) B-type reachability tree; (c) NMRT tree

图 16(a) 中的 Petri 网, 设终止集为 $\{(0, 0, 0, 1)\}$, 虽然没有有效传递向量, 但其对应的 Petri 网语言也是上下文有关的. 这种情况, 用文献 [8] 的方法就不能进行判定了. 用本文的方法就可以判定它对应的 Petri 网语言是上下文有关的. 在该 Petri 网中, 存在一个有效的递增子, 该递增子有两个线性子, 所以该 Petri 网对应的 Petri 网语言是上下文有关的.

在判定递增子的有效性时, 本文对可达树进行了改进, 给出了 A 型可达树和 B 型可达树, 这样的可达树保留了更多的信息, 这些内容在 3.2 小节中已经进行了说明, 图 16(a) 中的 Petri 网的 B 型可达树如图 16(b) 所示. 还有一些其他文献也对可达树进行了改进, 这些改进主要是针对 ω 怎么表示展开的. 例如, Peterson^[2] 提出用 $a + b \cdot n_i$ 来替代符号 ω 的建议, 后来人们围绕怎么表示 ω 做了大量工作, 文献 [18~20] 都是围绕这一主题展开的. Wang 等^[18] 最早提出了用一个整数的子集来表示 ω : 称一个整数的子集 S 为 ω -数, 当且仅当存在一个正整数 k , 整数 n 和 q 使得 $S = \{i \cdot k + q | i \geq n, 0 \leq q < k\}$. 直观上说, 对于一个可循环引发的序列, 如果它的每次引发都能使某些位置中标记的数量增加, 那么每次增加的数量是一样的, 是个常数, 即 i 代表引发次数, q 代表该位置中的初始标记数, 则 $i \cdot k + q$ 表示该序列引发 i 次时位置中的标记数量, S 也可以表示为 $S = \omega(k, n, q) \equiv k \cdot \omega_n + q \equiv \{i \cdot k + q | k$ 是一个正整数, n 是一个整数, $i \geq n, 0 \leq q < k\}$; 也就是一个 ω 数代表的是一个无限个整数的集合. $\omega_n = \{n, n+1, n+2, \dots\}$. 据此可知 $\omega(1, 0, 0) = \omega_0 = \{0, 1, 2, 3, \dots\}$, $\omega(2, 1, 0) = 2 \cdot \omega_1 = \{2, 4, 6, \dots\}$. 他还详细定义了整数集以及 ω -数上的大小比较等, 在做了这些准备之后, 给出了它们的可达树的构造方法. Wang 等^[19, 20] 对 MRT 进行了改进, 改进后的树记为 NMRT. 图 16(a) 中的 Petri 网, 其 NMRT 如图 16(c) 所示. B 型可达树和 NMRT 的主要区别有 3 个方面.

(1) 图 16(b) 和 (c) 中都有 ω_1 , 但它们的意义不一样. 图 16(b) 中的 ω_1 表示 ω 是由于 t_1 的引发而导致的; 图 16(c) 中 ω_1 为集合 $\{1, 2, 3, 4, \dots\}$.

(2) 由 B 型可达树可以很容易地判断 ω 是由哪个变迁的引发而导致的, 这一点满足了判断递增子有效性的需要. 在 NMRT 树中判断一个 ω 是由哪个变迁引发而导致的就比较困难了.

(3) NMRT 只是针对 ω -独立 (ω -independent) Petri 网的; 本文给出的 B 型可达树是针对一般 Petri 网的.

文献 [18~20] 中使用集合来表示可达集, 可达集可能是个无穷集, 从计算的角度上来说, 不具有可操作性. 本文的方法根据改进的可达树导出一个方程组, 再通过方程组的解来判断可达性, 在可计算性上具有优势.

5 结论

本文给出了判断无标注 L 型 Petri 网语言属性的方法. 对于只有一个有效递增子的 Petri 网, 可以通过有效递增子和它的线性子的个数来判断一个 Petri 网 PN 对应的 Petri 网语言的属性: 如果没有有效递增子, 则 PN 对应的 Petri 网语言是正则语言; 如果只有一个有效递增子, 且该递增子只有一个线性子, 则 PN 对应的 Petri 网语言为上下文无关的; 如果只有一个递增子, 但该递增子有两个及其以上的线性子, 则 PN 对应的 Petri 网语言为上下文相关的. 对于包含多个有效递增子的情况, 如果存在某个有效递增子有两个及其以上线性子, 则 PN 对应的 Petri 网语言为上下文相关的; 如果每个有效递增子都只有一个线性子, 则顺序引发、嵌套引发和选择引发时 PN 对应的 Petri 网语言是上下文无关的, 并发引发和交叉引发时 PN 对应的 Petri 网语言是上下文相关的.

本文的结论是针对无标注 L 型 Petri 网语言展开的讨论, 将来的工作可以在两个方面开展, 一个是在无标注的 G 型、 T 型和 P 型 Petri 网语言的属性判定方面开展研究工作; 另一个是在本文的基础上进一步地探讨可达性判定的算法.

参考文献

- 1 Murata T. Petri nets: properties, analysis and applications. Proc IEEE, 1989, 77: 541-580

- 2 Peterson J L. Petri Net Theory and the Modeling of Systems. New Jersey: Prentice-Hall, 1981
- 3 Giua A, DiCesare F. Decidability and closure properties of weak Petri net languages in supervisory control. *IEEE Trans Autom Control*, 1995, 40: 906–910
- 4 Ramadge P J, Wonham W M. Supervision of discrete event processes. In: *Proceedings of the 21st IEEE Conference on Decision & Control*, Orlando, 1982. 7: 1228–1229
- 5 Ramadge P J, Wonham W M. Supervision of control discrete event processes. In: *Proceedings of the Joint Workshop on Feedback and Synthesis of Linear and Nonlinear Systems*. Berlin: Springer-Verlag, 1982. 39: 202–214
- 6 Jiang C J. An algorithm for finding effective (controlled) repetitive vectors. *J Comput*, 1994, 17: 580–587 [蒋昌俊. 求有效极小受控可重复向量的一个算法. *计算机学报*, 1994, 17: 580–587]
- 7 Peterson J L. Computation sequence sets. *J Comput Syst Sci*, 1976, 13: 1–24
- 8 Wu Z H. Petri net description of pumping lemma—a set of conditions for determining the type of a Petri net language. *J Comput*, 1994, 17: 852–858 [吴哲辉. Pumping 引理 —Petri 网语言属型的一组判定条件. *计算机学报*, 1994, 17: 852–858]
- 9 Valk R, Vidal-Naquet G. Petri nets and regular languages. *J Comput Syst Sci*, 1981, 23: 299–325
- 10 Mazurkiewicz A. Petri net languages revisited. *Theor Comput Sci*, 2012, 429: 227–235
- 11 Gaubert S, Giua A. Deterministic weak-and-marked Petri net languages are regular. *IEEE Trans Autom Control*, 1997, 41: 1802–1803
- 12 Jiang C J, Liu G J. The pumping lemma of Petri net language. *J Comput*, 2006, 29: 274–278 [蒋昌俊, 刘关俊. Petri 网语言的 Pumping 引理. *计算机学报*, 2006, 29: 274–278]
- 13 Hopcroft J E, Motwani R, Ullman J D. *Introduction to Automata Theory Languages and Computation*. London: Addison Wesley, 2006
- 14 Habermehl P, Meyer R, Wimmel H. The downward-closure of Petri net languages. In: *Proceeding of International Colloquium on Automata, Languages and Programming*. London: Springer, 2010. 466–477
- 15 Pelz E. Closure properties of deterministic Petri nets. In: *Proceedings of the 4th International Symposium on Theoretical Aspects of Computer Science*. London: Springer, 1987. 371–382
- 16 Starke P H. Free Petri net languages. In: *Proceedings of the 7th International Conference on Mathematical Foundations of Computer Science*. London: Springer, 1978. 506–515
- 17 Starke P H. On the languages of bounded Petri nets. In: *Proceedings of the 8th International Conference on Mathematical Foundations of Computer Science*. London: Springer, 1979. 425–433
- 18 Wang F Y, Gao Y Q, Zhou M C. A modified reachability tree approach to analysis of unbounded Petri nets. *IEEE Trans Syst Man Cybernet Part B: Cybernet*, 2004, 34: 303–308
- 19 Wang S G, Zhou M C, Li Z W, et al. A new modified reachability tree approach and its applications to unbounded Petri nets. *IEEE Trans Syst Man Cybernet Syst*, 2013, 43: 932–940
- 20 Wang S G, Zhou M C, Gan M D, et al. New reachability trees for unbounded Petri nets. In: *Proceedings of IEEE International Conference on Robotics and Automation*, Washington, 2015. 3862–3867

An approach to distinguishing the levels of free-labeled L -type Petri net languages in the Chomsky hierarchy

Shangmin LUAN^{1,2*}, Liqin TIAN^{1,2} & Zilin GENG^{1,2}

1. *Department of Computer, North China Institute of Science and Technology, Sanhe 065201, China;*

2. *Hebei Engineering Technology Research Center for IOT Data Acquisition & Processing, Sanhe 065201, China*

* Corresponding author. E-mail: 623320726@qq.com

Abstract Petri nets (PNs) and automatic machines are two approaches to modeling discrete event dynamic systems. The relationship between these approaches is important for controlling discrete event dynamic systems. This paper investigates the relationship between these approaches from the formal language point of view, i.e., we introduce an approach to distinguishing the levels of free-labeled L -type PN languages in the Chomsky hierarchy. To characterize PNs from the formal language point of view, the concepts of increaser, valid increaser, and linearer are introduced. For a PN with no valid increaser, its PN language is regular. For a PN with a valid increaser with at least two linearers, its PN language is context-sensitive. For a PN in which each valid increaser has only one linearer, assuming only one increaser in the PN, the PN language is context-free. If the PN has more than one increaser, the corresponding PN language is context-sensitive when valid increasers and their linearers can be fired currently or crossed. The language is context-free when the increasers and their linearers are fired alternatively, or an increaser and its linearer are fired sequentially relative to other increasers and their linearers, or when firing occurs after an increaser and before its linearer.

Keywords Petri net language, formal language, automata theory, reachability tree, regular language, context-free language, context-sensitive language



Shangmin LUAN graduated from Beihang University, China, with a Ph.D. degree in computer science and technology. Presently, he is engaged in teaching and computer science research. His research interests include theoretical computer science, artificial intelligence, Petri net theory and its applications, big data and its applications to production safety.



Liqin TIAN was born in 1970. He received his Ph.D. degree from the Information Engineering School, Beijing University of Science and Technology, China, in 2009, and his Master's degree from the Department of Computer Science and Technology, Tsinghua University, China, in 2003. He conducted post-doctoral research at Tsinghua University, from 2009 to 2011. Presently, he is a professor at the North China Institute of Science and Technology, the New

Century Excellent Talents at University of China. His research interests include computer networks, performance evaluation, network security, and Internet of things.



Zilin GENG was born in 1958 in Bayan County. He is a professor at the North China Institute of Science and Technology, where he teaches and conducts research in computer science. His research interests include computer science, and big data and its applications.