

18-20

一种基于线性逻辑的 Petri 网分析方法

An Analysis Method of Petri Nets Based on Linear Logic

高梅梅 吴智铭

TP271.8

(上海交通大学电子信息学院自动化系 上海 200030)

Abstract The paper presents a translation from Petri nets to linear logic with the objective of enhancing the analysis ability of Petri nets. The definition of Petri nets is described by defining the translations in Petri nets using the connectives in linear logic, and the behavior of Petri nets is constructed using the rules we defined. The behavior of Petri nets can be described precisely and clearly using this method.

Keywords Petri nets, Linear logic

1 引言

Petri 网是一种用网状图形表示系统模型的方法,能够从组织结构、控制和管理角度,精确描述系统中事件(变迁)之间的依赖(顺序)和不依赖(并发)关系。Petri 网理论提供了强大的分析方法,如不变量分析、系统性能分析(如活性)等以证明系统的正确性。近年来 Petri 网也被用来表示知识推理,例如用于诊断和监控,这就需要发展一种关于 Petri 网行为的推理主题。有些学者^[1,2]把经典逻辑和 Petri 网相结合,用 Petri 网表示产生式规则系统,网的框架代表基于产生式规则的知识结构,库所表示命题,如果命题为真,就在库所中标识上托肯,规则推理过程用 Petri 网中变迁的触发表示,命题与推理规则之间的因果联系用库所和变迁之间的有向弧表示。这种方法形式化地描述了规则的链接机制,其好处在于利用了 Petri 网的并行推理能力,利用 Petri 网的数学基础进行推理算法研究,实现自动推理。对托肯和变迁加入模糊参量则可以用于表示模糊产生式规则系统,这就导致了模糊 Petri 网^[3,4]的出现,模糊 Petri 网可形式化地选择多条规则的推理结果得到最后的决策,这种模糊推理 Petri 网最有代表的应用是模糊专家系统的建模和实现自动推理。这种方法虽然能有效地进行逻辑推理,但对传统 Petri 网的语义作了很大的改变,它没有利用 Petri 网中资源的概念,它表示的是经典逻辑,因为经典逻辑中处理的是永久真值,也就是说,当一个命题为真时,它将在整个推理过程中永远保持为真。但实际上,在用 Petri 网表示离散事件动态系统时,命题表示的是系统的状态,系统的运行是从一个状态转变到另一个状态,根据状态概念的定义,则系统的命题不会是“恒真”的。

线性逻辑(linear logic)^[5]表示的是状态的概念,命题在推理过程中表示资源,当命题作为假设而产生结论时,这些作为假设的命题则失去真值,状态的变化由假设和结论的因果关系表示。目前也有一些文章^[6]中提到用线性逻辑进行 Petri 网描述的思想,但未能全面地描述 Petri 网中的并发和冲突等问题,因此并不能很好地体现线性逻辑推理的优越性。本文提出用线性逻辑描述 Petri 网结构,定义了 Petri 网中变迁之间的各种触发关系和规则,并在此基础上全面而准确地证明 Petri 网的推理运行过程,以提高对 Petri 网的描述和分析能力。

传统 Petri 网理论分析方法有以下不足:

(1)基于可达状态图的描述。可达状态图是用于表示 Petri 网可达标识的集合,它描述的是在某一初始状态下,系统可能达到的状态。从可达状态图可以证明网的一些性质,如证明是否存在某变迁序列使系统从一个状态转变到另一个状态或证明是否某些状态是可达的,但可达状态图描述存在一些缺点:

- 它是针对某一个初始标识的,一个新的初始标识就意味着需要重新构造可达状态图;
- 当系统存在较多并发冲突的变迁时,会出现状态空间爆炸问题;
- 没有考虑并发问题,不能区分变迁之间的并发和冲突关系。

(2)线性代数描述法。Petri 网的线性代数分析法是基于以下算式:

$$\text{IF } M \xrightarrow{s} M' \quad \text{THEN } M' = M + C \cdot \bar{s}$$

其中 M 和 M' 是标识, s 是变迁序列, C 为关联矩阵, $\bar{s}$ 为特征向量。采用线性代数分析法可以从初始标识和

高梅梅 博士生,研究方向:离散事件系统理论及应用。吴智铭

教授、博士生导师,研究领域:离散事件动态系统和混杂系统。

变迁序列计算出可达标识,可以进行不变量计算,但这种方法既没有考虑触发顺序,也没有考虑关联矩阵中会有闭环消元的情况,即一个库所可能同时是一个变迁的输入和输出时,在关联矩阵中无法表达。

2 线性逻辑中的主要符号

线性逻辑是由 J. Y. Girard^[3]在经典逻辑的基础上提出的,旨在处理资源问题,所以对经典逻辑做了很大的变动。线性逻辑中与 Petri 网描述有关的主要连接符(connective)有:

- ⊗: 表示资源之间“与”的关系,与经典逻辑中的“and”具有基本相同的性质,但它不象“and”具有单义性,例如:对于命题 A, A and A=A, 而 $A \otimes A = 2A$ 。
- ⊕: 表示资源间“或”的关系,与经典逻辑中的“or”具有相同的性质,例如 $A \oplus A = A$ 。
- : 表示资源之间的因果关系,例如 $A \rightarrow B$ 表示由 A 命题产生 B 命题,而 A 命题不再成立。不象经典逻辑中, $A \Rightarrow B$ 表示 A 命题成立可导致 B 命题成立,且 A 命题仍然成立。
- ⊢: 表示资源之间的可证明关系。例如 $\Gamma \vdash \Delta$ 表示 Δ 可以由 Γ 证明得出,其中 Γ, Δ 为资源集合。

3 用线性逻辑描述 Petri 网

任何 Petri 网都可以用线性逻辑算式描述。首先,可以用线性逻辑表示 Petri 网中的托肯分布,即表示 Petri 网的标识。对于一 Petri 网,其当前标识可表示为:

$$M = \bigotimes_{p \in \text{set of marked places}} P_p^{m_p}$$

其中 P_p 为库所名称, m_p 表示库所 P_p 中有 m_p 个托肯。这样表示 Petri 网的标识只用列出有托肯的库所,不用列出所有的库所,如 $A \oplus B$ 则表示库所 A 和 B 中各有一个托肯。

另外, Petri 网中的变迁表示了输入库所和输出库所之间的因果关系,且变迁的触发意味着托肯从输入库所到输出库所的转移,所以可用线性逻辑中的符号“→”描述 Petri 网中的变迁。“→”的左边表示是变迁触发所需要的最小标识,右边表示由最小标识经变迁触发后可达的标识。即对于 Petri 网中的每个变迁 t_i :

$$t_i = \text{pre}(t_i) \rightarrow \text{post}(t_i)$$

例如图 1 中的变迁 t_1 的 $\text{pre}(t_1) = A \otimes A$, $\text{post}(t_1) = B \otimes D$, 则 t_1 可定义为:

$$t_1: 2A \rightarrow B \otimes D$$

同理 t_2 可定义为:

$$t_2: B \otimes D \rightarrow C$$

这样通过符号“⊗”和“→”定义 Petri 网中的变迁,实际上就对 Petri 网的结构进行了描述。例如图 1 中 Petri 网的结构可描述如下:

$$t_1: 2A \rightarrow B \otimes D$$

$$t_2: B \otimes D \rightarrow C$$

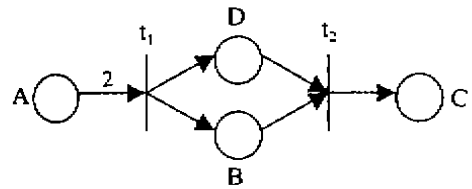


图 1

根据 Petri 网的描述,线性逻辑可以表示标识的运行状况。例如图 1:

$$2A, (t_1: 2A \rightarrow B \otimes D) \vdash B \otimes D$$

上式表示了标识的变化,从初始标识 $M(2A)$ 经过 t_1 触发达到标识 $M'(B \otimes D)$,即系统从状态 $M(2A)$ 到状态 $M'(B \otimes D)$ 是可证明的。在实际情况下,初始标识 M 有可能大于 t_1 触发所需要的最小标识,这时,多余的标识保持在可达标识中,例如:

$$2A \otimes C, (t_1: 2A \rightarrow B \otimes D) \vdash B \otimes D \otimes C$$

以上所讨论的是单个变迁触发的现象,下面我们讨论变迁序列的描述,如图 1:

$$(t_1: 2A \rightarrow B \otimes D)(t_2: B \otimes D \rightarrow C) \vdash$$

$$(t_1 t_2: 2A \rightarrow C)$$

上式给出了变迁序列 $s = t_1 t_2$ 的定义,它描述了变迁序列 $t_1 t_2$ 触发的最小标识和最小标识经变迁触发后的可达标识,与单个变迁的定义相同,用线性逻辑对 Petri 网经过变迁序列的标识运行的描述也和单个变迁的描述类似。例如:

$$2A \otimes D, (t_1 t_2: 2A \rightarrow C) \vdash C \otimes D$$

上式等价于:

$$2A \otimes D, (t_1: 2A \rightarrow B \otimes D)(t_2: B \otimes D \rightarrow C) \vdash C \otimes D$$

4 变迁之间不同的触发规则

4.1 规则的定义

上节讨论的例子中 t_1 和 t_2 是一种顺序触发关系,实际上, Petri 网中变迁的触发还有并发和冲突关系,为此,我们在线性逻辑的基础上,定义了三个规则 (SEQ 规则、PAR 规则和 ALT 规则) 分别表示变迁触发的顺序、并发和冲突关系。

定义 1 (SEQ 规则) 对于两个变迁 $t_1: M \rightarrow K$ 和 $t_2: K \rightarrow L$,若 t_2 的触发以 t_1 的触发为条件,则 t_1 和 t_2 是顺序触发关系。SEQ 规则表示为:

$$\frac{t_1: M \rightarrow K \quad t_2: K \rightarrow N}{t_1 \cdot t_2: M \rightarrow N} \text{SEQ}$$

其中连接符“·”表示变迁顺序触发。

定义 2 (PAR 规则) 对于两个变迁 $t_1: M \rightarrow K$ 和 $t_2: L \rightarrow N$, PAR 规则表示为:

$$\frac{t_1: M \rightarrow K \quad t_2: L \rightarrow N}{t_1 \parallel t_2: M \otimes L \rightarrow K \otimes N} \text{PAR}$$

其中连接符“||”表示变迁并发或互不相关独立触发。

定义3 (ALT 规则) 对于两个变迁 $t_1: M \rightarrow K$ 和 $t_2: M \rightarrow L$, 若 $pre(t_1)$ 和 $pre(t_2)$ 中含有相同的库所, 则 t_1 和 t_2 存在并发冲突, 对于有限的输入标识, t_1 和 t_2 只能选择其一触发。ALT 规则表示为:

$$\frac{t_1: M \rightarrow K \quad t_2: M \rightarrow L}{t_1/t_2: M \rightarrow K \oplus L} \text{ALT}$$

其中连接符“/”表示变迁并发冲突。

以上讨论了用线性逻辑对变迁或变迁序列的定义, 它是基于网结构的, 与初始标识无关。但实际上, 由于初始标识的不同, 变迁之间的触发关系不是绝对的, 而是可变的, 例如在上例中, 若初始标识中库所 A、B 和 D 中都有托肯 (A 中有两个托肯), 则 t_1 和 t_2 可以并发。在这种情况下, t_1 和 t_2 的触发序列的定义为:

$$\frac{t_1: 2A \rightarrow B \otimes D \quad t_2: B \otimes D \rightarrow C}{t_1 \parallel t_2: 2A \otimes B \otimes D \rightarrow B \otimes D \otimes C} \text{PAR}$$

上式表示 t_1 和 t_2 并发所需的最小标识为 $2A \otimes B \otimes D$ 。

4.2 用线性逻辑描述 Petri 网的运行行为

在第3节中, 我们实际上已经提到了用线性逻辑表示 Petri 网运行的方法, 在此, 再进行形式化定义。在 M 标识下, 经变迁 t 触发后, Petri 网的运行行为可表示为:

$$M, t \vdash M'$$

其中 M' 为 Petri 网运行后的标识, 同理经变迁序列 s 触发, Petri 网的运行行为可表示为:

$$M, s \vdash M'$$

这里值得再一次说明的是符号“ $\rightarrow$ ”和“ $\vdash$ ”的不同, 在描述 Petri 网的结构时, 我们用符号“ $\rightarrow$ ”定义 Petri 网的变迁和变迁序列, “ $\rightarrow$ ”的左边是变迁触发所需的最小标识, 即 $pre(t)$; 而“ $\vdash$ ”表示 Petri 网的运行, 它允许描述系统在某初始标识下 (该标识大于等于变迁触发的最小标识) 的运行状况。

基于上节所定义的变迁触发规则, 我们可以对系统的触发序列进行组合, 即:

$$\frac{M_1, s_1 \vdash M_2 \quad M_2, s_2 \vdash M_3}{M_1, s_1 \cdot s_2 \vdash M_3} \text{SEQ}$$

$$\frac{M_1, s_1 \vdash M_2 \quad M_2, s_2 \vdash M_4}{M_1 \otimes M_2, s_1 \parallel s_2 \vdash M_3 \otimes M_4} \text{PAR}$$

$$\frac{M_2, s_1 \vdash M_2 \quad M_1, s_2 \vdash M_2}{M_1, s_1/s_2 \vdash M_2 \oplus M_2} \text{ALT}$$

由此, 我们就可以采用多种规则的组合从而描述 Petri 网的行为。

5 举例

我们以图2为例, 说明本文的方法。此例看似简单, 实际上它包含了顺序、并发和冲突多种变迁触发关系。其 Petri 网结构可描述为:

$$\begin{aligned} t_1: P_1 &\rightarrow P_2 \\ t_2: P_3 &\rightarrow P_2 \\ t_3: P_2 &\rightarrow P_4 \\ t_4: P_2 &\rightarrow P_5 \end{aligned}$$

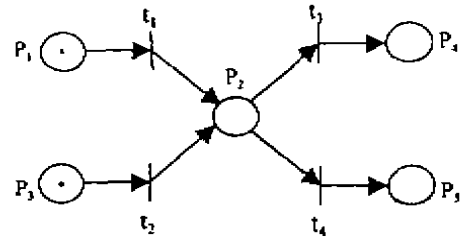


图2

我们可以通过本文所定义的规则, 证明在初始标识 $M_0(P_1 \otimes P_3)$ 下的 Petri 网运行行为, 首先, 可得以下变迁序列的定义:

$$\frac{t_1: P_2 \rightarrow P_4 \quad t_3: P_2 \rightarrow P_5}{t_1/t_3: P_2 \rightarrow P_4 \oplus P_5} \text{ALT}$$

$$\frac{t_1: P_1 \rightarrow P_2 \quad t_3/t_4: P_2 \rightarrow P_4 \oplus P_5}{t_1 \cdot (t_3/t_4): P_1 \rightarrow P_4 \oplus P_5} \text{SEQ}$$

$$\frac{t_2: P_3 \rightarrow P_2 \quad t_3/t_4: P_2 \rightarrow P_4 \oplus P_5}{t_2 \cdot (t_3/t_4): P_3 \rightarrow P_4 \oplus P_5} \text{SEQ}$$

则在初始标识 $M_0(P_1 \otimes P_3)$ 下可推出:

$$\frac{P_1, t_1 \cdot (t_3/t_4) \vdash P_4 \oplus P_5 \quad P_3, t_2 \cdot (t_3/t_4) \vdash P_4 \oplus P_5}{P_1 \oplus P_3, (t_1 \cdot (t_3/t_4)) \parallel (t_2 \cdot (t_3/t_4)) \vdash P_4 \oplus P_5} \text{PAR}$$

由此, 我们可得此 Petri 网在初始标识 $M_0(P_1 \otimes P_3)$ 的触发序列为 $(t_1 \cdot (t_3/t_4)) \parallel (t_2 \cdot (t_3/t_4))$, 它全面地描述了变迁之间的各种触发关系, 而可达状态图法和关联矩阵描述法则不能清楚地描述变迁之间的并发和冲突关系, 显然本文的方法优于可达状态图和关联矩阵描述法。

参考文献

- 1 Zisman M D. Use of production systems for modelling asynchronous concurrent processes. Pattern Directed Inference Systems. London: Academic Press, 1978. 53~68
- 2 Tzafestas S G, Capkovic F. Petri Net-Based Approach to Synthesis of Intelligent Control Systems for DEDS. Computer-Assisted Management and Control of Manufacturing Systems. Springer, 1997. 351~523
- 3 Looney C G. Fuzzy Petri nets for rule-based decisionmaking. IEEE Trans. Syst., Man, Cybern., 1988, SMC-18 (1): 178~183
- 4 Chen Shyi-Ming, et al. Knowledge Representation Using Fuzzy Petri Nets. IEEE Transaction on Knowledge and Data Engineering, 1990, 2(3): 311~319
- 5 Girard J Y. Linear Logic. Theoretical Computer Science. 1987
- 6 Valette R, Courvoisier M. Petri Nets and Artificial Intelligence. Modern Tools for Manufacturing Systems. Elsevier Science Publishers, 1993. 385~405