

CPN 原理及其在人工智能中的应用

Formal Principle of CPN and Its Application in Artificial Intelligence

姚绍文¹ 周明天² 龙华¹ 曾家智¹

(昆明理工大学通信与电子工程系 昆明650051)¹

(电子科技大学计算机学院 成都610054)²

Abstract After giving the formal description of Petri net, this paper presented the formal definition of CPN rather comprehensive and introduced CPN modeling tool Design/CPN for Linux. Taking the salesman problem in artificial intelligence as an example, we defined CPN model of the example in CPN formal method. Our paper ended with the implementation of visual model and analysis of the made model to show a variety of applications of CPN in systems modeling and analysis.

Keywords CPN, Occurrence-graph, Salesman problem, Markings

1 引言

Petri 网理论是由联邦德国的 Carl Adam Petri 于 1962 年在其博士论文中提出的一种建模方法。最初的网理论称为条件/事件网(CE-nets)。在随后的几年中,人们对新的网模型、网的基本概念及分析方法等进行了大量的研究。其中最有影响的是位置/变迁网(PT-nets), PT-nets 后来命名为 Petri 网(Petri Net)。这些早期的网模型在理论上都归结为基本网(EN-nets)。研究者通过对 PT-nets 进行了扩展,在网中增加了变迁的优先级、时间延迟、全程变量等一些概念以满足解决实际应用建模的具体要求。并在此基础上形成了比 CE-nets 和 PT-nets 更加通用化的高级网(High-level Nets)理论^[1-4]。最早的高级网是谓词/变迁网(PrT-nets),此后又产生了广义随机 Petri 网(GSPN)、着色 Petri 网(CPN)等较有影响的建模方法。CPN 是由丹麦的 Jensen Kurt 于 1981 年在 Petri 网基础上定义的一种高级网系统^[5]。CPN 理论不仅在描述系统静态模型方面进行了完备的形式化定义,例如支持在网模型中标记(token)的类型定义、模型的全程时钟、建立分层的子模型等,而且对系统模拟的动态行为的仿真分析也具有形式化定义和描述。此外在 Linux、Solaris 和 MacOS 等多种平台上都有 CPN 的建模分析工具。利用 CPN 的建模工具,可以建立描述系统的 CPN 静态模型,还可以对系统模型的动态行为进行仿真,分析系统的分布、并发、同步异步等特性,以及建立系统模型的状态空间并分析系统模型中的活性问题、可达性问题等。

由于 CPN 具有严格的网理论形式化的数学描述、以上的特性以及建模工具提供的仿真分析功能,因此在系统建模、计算机软件技术^[3]、自动控制系统、网络通信协议工程^[5]等得到了广泛的应用。本文在介绍 Petri 网理论的形式化定义的基础上,较全面地介绍了 CPN 理论关于静态模型、动态行为以及分析方法的形式化定义描述。然后介绍 Linux 平台上的 CPN 建模仿真工具 Design/CPN。最后以人工智能(AI)中推销员问题作为实例进行了 CPN 的形式化描述和模型建立,应用建模工具 Design/CPN 对该问题进行了仿真分析和求解。

2 CPN 的形式化定义

2.1 Petri 网定义

Petri 网的形式化定义^[6]为:

定义 1 四元组 $PN = (P, T, F, M_0)$, 其中, $P = \{P_1, P_2, \dots, P_m\}$ 为位置的集合; $T = \{T_1, T_2, \dots, T_n\}$ 为变迁的集合; $F \subseteq (P \times T) \cup (T \times P)$ 是弧的集合; $M_0: P \rightarrow N$, 初始标识, 用令牌(Token)表示, N 是自然数的集合。

Petri 网是一个图形表示的有向图,图中包括两种结点:位置和变迁,弧表示了位置结点与变迁结点之间的有向连结点关系。位置中令牌的初始分布构成了网系统的静止状态 M_0 。在 Petri 网系统的执行就是网中变迁事件的发生,使令牌分布发生变化而导致系统的状态发生变化。

2.2 CPN 概念及定义¹

·多集(Multi-set):多集 m 是非空集合 S 上的函

数, $m \in (S \rightarrow N)$, 设非空集合 S , 则 $m = \sum_{s \in S} m(s) \cdot s$ 是 S

上的多集, $m(s)$ 为集合 S 中元素 s 的系数。

·多集运算: 设 S_M 是 S 上所有多集的集合, $m_1, m_2 \in S_M, n \in N$, 则多集加、数乘、比较等运算操作定义如下:

i. 加法: $m_1 + m_2 = \sum (m_1(s) + m_2(s)) \cdot s$;

ii. 数乘: $n \cdot m = \sum_{s \in S, n \in N} (n \cdot m(s)) \cdot s$;

iii. 比较运算(关系)有 \neq, \leq, \geq 等。以 \neq 和 \leq 为例: $m_1 \neq m_2 = \exists s \in S: m_1(s) \neq m_2(s); m_1 \leq m_2 = \forall s \in S: m_1(s) \leq m_2(s)$;

iv. 减法: 当 $m_1 \leq m_2, m_2 - m_1 = \sum_{s \in S} (m_2(s) - m_1(s)) \cdot s$ 。

·CPN 定义: CPN 是一个元组 $CPN = (\Sigma, P, T, A, N, C, G, E, I)$, 其中:

i. Σ 是一个非空的类型有限集, 又称为类型集² (Color Set);

ii. P 是位置的有限集;

iii. T 是变迁的有限集;

iv. A 是弧的有限集, 满足: $P \cap T = P \cap A = T \cap A = \Phi$;

v. N 是节点函数, 定义为: $N: A \rightarrow (P \times T) \cup (T \times P)$;

vi. C 是一个类型 (Color) 函数, 定义为: $C: P \rightarrow \Sigma$;

vii. G 是一个变迁 t 的“保证”(Guard) 函数, 定义为从 T 到表达式的函数 $G(t)$ 。设 B 为具有 {真(true), 假(false)} 元素的布尔类型, 则表达式应满足:

$$\forall t \in T: [Type(G(t)) = B \cap Type(Var(G(t))) \subseteq \Sigma]$$

其中 $Type(v)$ 函数返回变量 v 的类型, $Var(expr)$ 返回表达式 $expr$ 的变量集;

viii. E 是一个弧表达式函数, 定义为从 A 到表达式的函数 $E(a)$, 满足:

$$\forall a \in A: [Type(E(a)) = C(p)_{MS} \cap Type(Var(E(a))) \subseteq \Sigma]$$

其中 p 是 $N(a)$ 中的位置, $C(p)_{MS}$ 返回位置 p 上多集的类型;

ix. I 是一个初始化函数 $I(p)$, 定义为从 P 到一个封闭表达式 (Closed Expression), 封闭表达式是指不

含任何变量的表达式, I 必须满足:

$$\forall p \in P: [Type(I(p)) = C(p)_{MS}]$$

即 $I(p)$ 函数对位置 p 进行初始化时, 其封闭表达式结果的类型必须同位置 p 上的多集的类型相一致。

2.3 CPN 模型仿真执行的形式化描述

·变迁绑定的定义: 变迁 t 的一个绑定定义为在 $Var(t)$ 之上的一个函数 b , 满足:

i. $\forall v \in Var(t): b(v) \in Type(v)$;

ii. $G(t)(b)$ 为“真”, $G(t)(b)$ 表示用绑定 b 计算“保证”表达式 $G(t)$ 的计算值。

变迁 t 的所有绑定的集合用 $B(t)$ 表示。

·标识 (Marking)、步³ (Step) 和初始标识的定义: 标识用于表示一个 CPN 模型的状态, 而步是用于描述引起 CPN 模型状态变化的事件发生步骤。

i. 标识 M 是在所有令牌元素的集合 TE 之上的一个多集;

ii. 步 Y 是在所有绑定元素的集合 BE 之上的非空且有有限多集;

iii. 初始标识 M_0 : 定义为: $\forall (p, c) \in TE: M_0(p, c) = (I(p))(c)$, 即 M_0 是通过计算初始化表达式函数得到的标识。所有标识和步的集合分别用 M 和 Y 表示。

·步的有效 (Enabled) 和发生 (Occur)⁴

i. 当 CPN 模型处于一个标识 (即状态) M 时, 我们称一个“步” Y 有效当且仅当满足:

$$\forall p \in P: \sum_{t \in T, EY} E(p, t)(b) \leq M(p) \text{。此时称绑定 } (t, b) \text{ 有效, 也称变迁 } t \text{ 有效;}$$

ii. 当 CPN 模型处于标识 M_1 且步 Y 有效时, 步 Y 可能发生, 步 Y 的发生定义为:

$$\forall p \in P: M_2(p) = (M_1(p) - \sum_{t \in T, EY} E(p, t)(b)) + \sum_{t \in T, EY} E(t, p)(b) \text{。}$$

步 Y 的发生引起 CPN 模型的状态从标识 M_1 改变到 M_2 , 并且称 M_2 是从标识 M_1 直接可达的标识。步 Y 的发生引起 CPN 模型标识变化的过程记为: $M_1 [Y > M_2$ 。

2.4 CPN 模型分析与模型动态特性的形式化描述

利用计算机上 CPN 建模仿真工具, 可以分析 CPN 模型的动态特性、CPN 模型状态标识构成的状态空间, 从而分析模型系统的并发性、同步性及其

¹CPN 支持分层结构和具有全程时钟时间特性的模型描述, 此处定义非分层无时间特性的单页模型。在分层结构中, 只是增加了不同层之间关系的定义描述。对 CPN 元组中部件附加上时间值实现时间特性。²传统的 Petri 网位置中有令牌用“黑点”表示, CPN 中加入了数据类型, 把令牌值称为颜色值以区分令牌, 因此在 CPN 中由于历史原因, 数据类型集被称之为颜色集 (color Set), 本文统一称为类型集。³此处“步”的概念实际上表示 CPN 模型中“事件的发生”行为, 从意义上说称为“事件”更恰当, 本文考虑到与外文文献中术语的一致, 把“事件发生的行为”称为“步”。

他系统特性,CPN模型的动态特性包括有界性(Boundedness)、回归性(Home)、活性(Liveness)以及公平性(Fairness)。

本文给出活性及与OG图(Occurrence Graph)相关概念的定义如下。

i. 事件发生序列(Occurrence Sequence):CPN模型的标识及步序列称为CPN模型的事件发生序列,具有形式: $M_1[Y_1>M_2[Y_2>M_3\cdots M_i[Y_i>M_{i+1}\cdots$,从标识 M' 可达的所有标识的集合表示为: $[M'>$ 。

ii. 活性:假定 $M\in M, X\subseteq BE, M$ 是“死”(dead)标识当且仅当: $\forall x\in BE: \neg M[x>$; X 是“死”的绑定元素集,当且仅当: $\forall M'\in [M>, \forall x\in X: \neg M'[x>$;反之, X 是“活”(live)的,当且仅当: $\forall M'\in [M0>, \exists M''\in [M'>, \exists x\in X: M''[x>$ 。

iii. OG图:OG图是一个有向图 $OG=(V, A, N)$,其中:

- $V=[M_0>$;
- $A=\{(M_1, b, M_2)\in V\times BE\times V | M_1[b>M_2\}$;
- $\forall a=(M_1, b, M_2)\in A: N(a)=(M_1, M_2)$ 。

OG图与有限状态自动机(FSA)类似,表示了CPN模型状态(即标识)空间。 $M_1[Y>M_2$ 在OG图中表示:模型在标识 M_1 状态下,步 Y 发生后,模型的状态标识变为 M_2 。CPN建模中存在组合爆炸问题,通过SCC(Strongly Connected Components)图、OG图中等价类的合并、OG图的对称性、无关状态集分解以及位置不变特性等处理,可以使OG图在较大程度上得到压缩^[5],本文不介绍这些技术和方法。

3 CPN建模分析工具——Design/CPN介绍

3.1 Design/CPN工具简介

Design/CPN^[6]是由美国麻省Meta软件公司与丹麦Aarhus大学合作开发的CPN建模工具,该工具通过X-Window图形界面,为用户提供了可视化的开发环境,利用该工具可以进行CPN模型的创建、仿真和验证。目前该工具最新版本为4.0,本文利用Linux平台下的Design/CPN3.1.1版本进行说明。

进入Design/CPN建模环境以后,在编辑状态下可以编辑可视化的CPN模型,模型可以编辑在不同的页上,同模型相关的类型集和函数定义与变量和常量声明需要创建一个全程声名节点(GDN)。CPN模型创建好以后,可以进入仿真状态仿真运行建立的CPN模型,CPN模型在仿真状态下可以单步执行,也可以连续执行。在仿真状态基础上,还可以进入OG图状态产生、分析OG图,从而分析所模拟的系统的各种状态特

⁴在传统的Petri网中,代表事件的变迁的发生称之为点火(Fired)。

性。在图1中给出了进入仿真状态之前、进入OG图状态之后建模工具的菜单和界面。

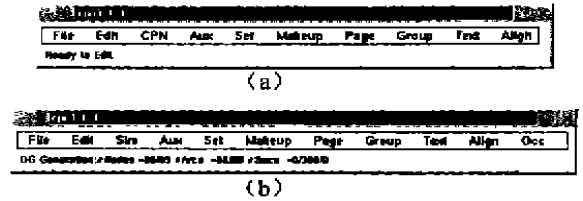


图1 Design/CPN建模环境和工具菜单

3.2 CPN ML

Design/CPN建模工具中所用的编程语言是CPN ML(CML),该编程语言是在一种称为SML的功能编程语言基础上进行扩展。在CPN模型中除了在GDN中进行定义和声明外,配合CPN模型的功能要求,可以在模型部件中很方便地增加CML代码。此外通常借助于CML编程分析OG图及CPN模型的动态特性。有关Design/CPN和CPN ML更深入的了解请参阅相关的资料^[5,6]。

4 实例建模

4.1 问题描述

推销员问题是人工智能领域中最经典的问题之一^[7-9],如图2所示。推销员从A地出发,走过B、C、D、E所有地方以后回到A地,任何两地之间的距离如图所示,问题的求解就是为推销员找出最短的行走路线。

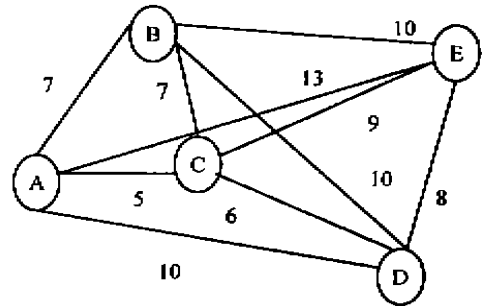


图2 推销员问题

4.2 CPN模型

用Design/CPN建模工具创建的推销员问题的CPN模型如图3所示。按照CPN形式化的定义,该图示化的CPN模型的部分的形式化描述如下:

- $\Sigma = \{LP, Position, DistanceBetweenTown, Price\}$;
- $P = (ArrivalPosition, CurrentPosition, Price, UsedPrice)$;

- T= {A,B,C,D,E};
- A= { (p,t) 或 (t,p) | p 在 P 的元素 p 和 T 的元素 t 之间的有向弧 };

模型中变迁 B,C,D,E 具有有效的绑定,因此当模型处于图中的状态标识下,变迁 B,C,D,E 均有可能发生,而变迁 A 此时没有有效的绑定。

在进入 OG 图状态分析之前,我们将以模型中变迁 D 及与之相关的位置、弧等网元素为例,对模型的形式化定义的其他部分作进一步的解释如下。

- N= { 返回与弧相连的节点的函数 | 节点是“位置”和“变迁”统称 }, 如图3中, $N((Price, D)) = \{(p, t) | p = Price, t = D\}$;

- 该函数是指位置 Price 和变迁 D 之间的弧所连的节点是位置 Price 和变迁 D;
- C= { 返回位置的令牌类型的函数 }, 图3中, $C(ArrivalPosition) = LP | LP \in \sum$;
- G= { 变迁 t 的保证函数 $t \in T$ }, 图3中, $G(D) = OkR(D, s, r, t, p)$ | 函数 OkR 在 GDN 中定义, 函数 OkR 的表达式满足 CPN 形式化定义中 G(t) 函数应满足的条件;
- E= { 在位置 p 和变迁 t 之间的弧的表达式函数 f }, 图3中, 弧 (Price, D) 上的表达式函数为: $E((Price, D)) = (s, r, p)$ (s, r, p) 是 GDN 中定义的元组类型的一个变量;
- I= { $I(p) | p \in P \wedge I(p)$ 是封闭表达式 }, 在图3中, $I(CurrentPosition) = I'A$;

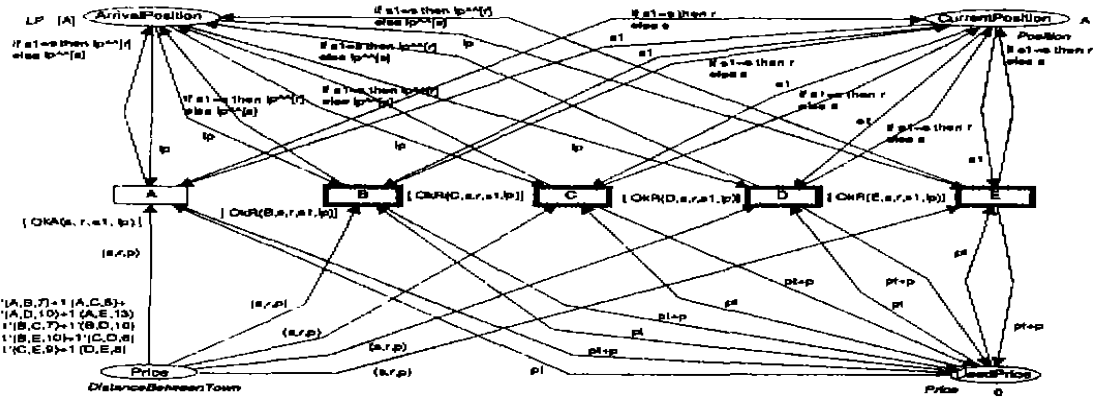


图3 推销员问题的 CPN 模型

图中的表达式、GDN 中的定义和声明都是用 CML 编写的。图中 CPN 模型的状态标识由图中四个位置的令牌类型集上多集构成的元组,表示了四种情况的组合:推销员所经历的地方、推销员当前所在的地方、推销员没有走过的路线以及推销员走过的总的距离。

5 动态分析

5.1 状态空间报告

模型建立以后,我们应用 OG 图分析模型。首先利用 Design/CPN 的 OG 图功能产生 CPN 模型的部分状态空间报告,如表1所示。从表1可知:该模型的 OG 图是一个完全 OG 图,它有 89 个节点、88 条弧,其中有 24 个“死”节点。利用我们的建模环境⁵计算 OG 图用了 1 秒钟。

很显然,在 OG 图中的 24 个“死”节点就是模型中推销员回到其最初的出发点的状态,问题的求解就是在这 24 个死节点中找出推销员走过的总的距离最小的

标识,即所有“死”节点标识中位置 UsedPrice 的最小值。

表1 Part of the property report

| Statistics: | Liveness Properties | Home Properties |
|------------------|---------------------------|---------------------|
| Occurrence Graph | Dead Markings: 24 | Home Markings: None |
| Nodes: 89 | [89, 88, 87, 86, 85, ...] | |
| Arcs: 88 | Dead transitions | |
| Secs: 1 | Instances: None | |
| Status: Full | | |

5.2 OG 图分析与问题求解

通过对 OG 图用 CML 进行编程^[1],可以求出问题的解并可以分析 OG 图的状态特性。为了表达上的直观,我们用 CML 编程将推销员可走的所有路线的总距离展示出来,如表2所示。此时通过观察就可以发现问题的解就是标识 M_{13} 和 M_{26} , 并且从初始标识 M_1 到 M_{13} 或 M_{26} 的可达标识节点序列分别为: $[1, 5, 16, 38, 62, 86]$ 和 $[1, 4, 12, 30, 54, 78]$ 。

⁵建模的硬件平台是 MMX/166MHz、64MRAM、4.3GHD。软件平台为 Slackware Linux 4.0 with kernel 2.2.6, 建模环境为 Design/CPN3.1.1。⁶可达标识序列及步的发生序列在传统的 Petri 网中称之为可达树。

表2 分析 OG 图使用的主要 CML 源程序及分析结果

| | |
|---|---|
| <pre>ListDeadMarkings(), map (Mark SinglePage'UsedPrice 1) (ListDeadMarkings()), map (Mark SinglePage'UsedPrice 1) (nthral(ListDeadMarkings(), 12)); nth (ListDeadMarkings(), 3), nth (ListDeadMarkings(), 11), NodesInPath(1, 86), NodesInPath(1, 78),</pre> | <pre>val it = [89,88 87,86,85,84,83,82,81,80,79,78,] Node list val it = [(1,43) " empty,(1,40) " empty,(1,44) " empty,(1,36) " empty,(1,44) " empty,(1,39) " empty,(1,41) " empty,(1,41) " empty,(1,45) " empty,(1,39) " empty,(1,42) " empty,(1,36) " empty,] Price ms list val it = [(1,46) " empty,(1,42) " empty,(1,49) " empty,(1,44) " empty,(1,41) " empty,(1,40) " empty,(1,49) " empty,(1,45) " empty,(1,46) " empty,(1,44) " empty,(1,41) " empty,(1,43) " empty] : Price ms list val it = 86 Node val it = 78 Node val it = [1,5,16,38,62,86] Node list val it = [1,4,12,30,54,78] Node list</pre> |
|---|---|

OG 图中关于问题的解的可达标识序列及步的
发生序列⁶自动产生出来如图4所示。从图中不难看出最短的
路线为{A,B,E,D,C,A}或{A,C,D,E,B,A},最短
的距离为36个单位,并且这两条最短的路线方向正

好完全相反。同时在初始标识状态下,该模型的并发性
发展有四种可能,因此利用 CPN 建模工具,借助于
OG 图功能对一些系统模型进行建模分析,可以深入
研究所仿真系统的并发特性。

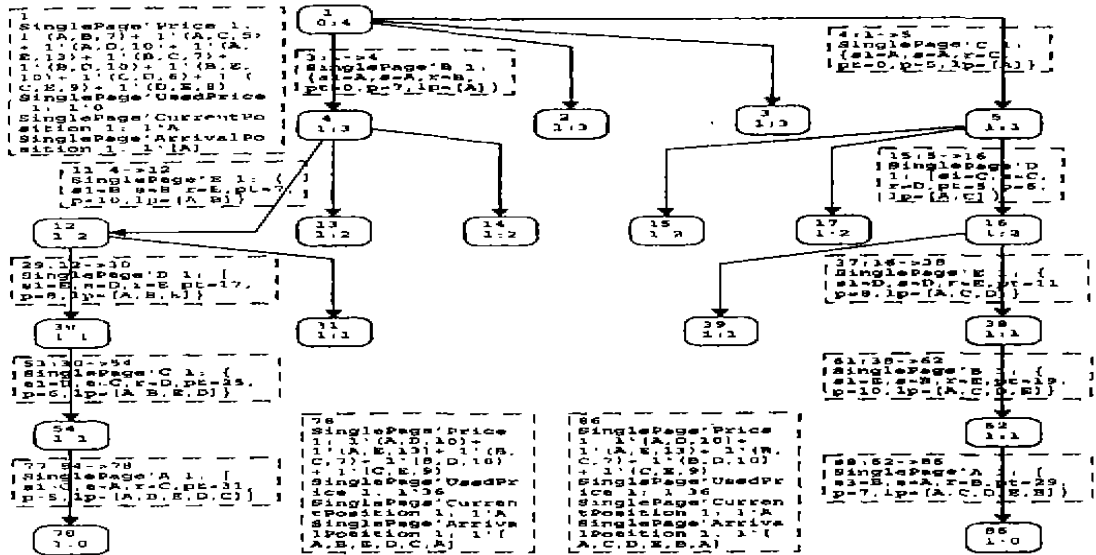


图4 问题的解的标识及事件发生序列

结论 本文在简述 Petri 网理论的基础上,比较全
面地介绍了 CPN 建模原理、仿真执行以及模型动态特
性的形式化定义。利用所介绍的 CPN 建模工具 De-
sign/CPN 对人工智能中经典的推销员问题进行建模
分析,深入分析了该模型的状态空间——OG 图。本文
展示了 CPN 建模工具在人工智能及其他系统建模仿
真研究中的建模方法和应用前景。

参考文献

- 1 蔡自兴,徐光佑. 人工智能及其应用(第二版). 北京:清华
大学出版社,1996
- 2 Nilsson N J. Artificial Intelligence, A New Synthesis
Morgan Kaufmann Publishers and China Machine Press,

- 1999
- 3 Tokmakoff A. A. Modeling, Analysis and Prototyping of
the ODP Trader using Colored Petri Nets and Java. [PhD.
Dissertation]. University of South Australia, Mar. 1998
- 4 张维明,邓苏,罗雪山,等. 信息系统建模技术与应用. 北
京:电子工业出版社,1997
- 5 Kurt J. Colored Petri Nets: Basic Concepts, Analysis
Methods and Practical Use. Springer-Verlag, 1992, 1995,
1997, Vol. 1, Vol. 2, Vol. 3
- 6 University of Aarhus CPN Group. Design/CPN Online.
http://www.dami.aau.dk/designCPN
- 7 Kurt J, et al Design/CPN Occurrence Graph Manual. U-
niversity of Aarhus, 1996
- 8 袁崇义. Petri 网原理. 北京:电子工业出版社,1998