

Petri 网中子网的分类与规范化研究^{*})

王志坚¹ 魏定国¹ 吴时霖²

(广东商学院信息学院 广州 510320)¹ (复旦大学计算机系 上海 200433)²

摘要 研究了 Petri 网中子网的外部环境、子网的接口、子网的边界等问题,给出了相关的定义;通过分析不同子网的特点,将子网分为 T 子网、P 子网、TP 子网和 PT 子网等不同类型;提出了接口标准化和子网规范化的思想。研究成果有助于基于子网的结构化设计方法的研究和推广。

关键词 子网,接口,标准化,规范子网

Taxonomy of Subnets in Petri Net and their Normalization

WANG Zhi-jian¹ WEI Ding-guo¹ WU Shi-lin²

(Information Science School, Guangdong University of Business Studies, Guangzhou 510320, China)¹

(Department of Computer Science, Fudan University, Shanghai 200433, China)²

Abstract The environment, interface and border of subnet were researched, corresponding definitions were given. On the analysis of the specialties of different subnets, they were classified into different types such as T-subnet, P-subnet, TP-subnet, PT-subnet and so on. The concept of interface standardization and subnet normalization were present. The results help to the research of subnet based structural system designing and its corresponding application.

Keywords Subnet, Interface, Standardization, Normalized subnet

1 引言

Petri 网提供了系统行为的图形化描述方法,同时引入了数学描述手段。Petri 网以一种简单直观的方式,反映出系统的并行、同步和资源共享等特性,提供强大的数学分析能力,拥有众多的应用软件。模块化与层次化是结构化设计的主要手段,基于 Petri 网的模块化与层次化设计中,子网是一个关键的概念。

子网研究的代表是文献[4],采用代数方法将一个复杂系统的 Petri 网模型分割成多个满足一定条件的子网。文献[5, 6]对于子网的分割方法及分割后的模型特性保持问题进行了研究,文献[7, 8]讨论了 Petri 网采用同步合成方式对系统建模后模型可回复性、活性与无死锁性的保持问题。较多的文献将子网研究与层次化方法结合起来^[9, 10]。文献[11]定义了一种 Petri 网的分层递归模型,文献[12]则是采用着色网形式统一定义了分层 Petri 网。文献[13]提出一类模块化 Petri 网结构——可控输出网(CO 网),用于制造系统建模,满足一定前提下,该网上下层变换之间可以保持模型的有界性、活性、可回复性等特性。文献[14]从变迁的细化与抽象角度讨论了类似的变换以及特性保持。

在已有的研究中,子网这个概念是不规范的,基本上各人根据自己的需求定义一个子网。本文首先对子网这一概念作出定义,在此基础上对接口、外部环境和子网结构进行分析,对子网进行分类。

定义 1 设 $N=(P, T; F)$ 是一个网,如果 $P_S \subseteq P, T_S \subseteq T, F_S = ((P_S \times T_S) \cup (T_S \times P_S)) \cap F$, 且 $SN=(P_S, T_S; F_S)$ 是连通的,则称 SN 为 N 的子网。

定义 1 源自文献[3],但增加了一个约束条件,即要求子

网是连通的,这是模块内聚性的体现,不把相互没有关联的功能部分放在同一个模块中。本文讨论的 Petri 网为 P/T 网,即容量函数 K 为任意函数,权函数 $W \equiv 1$,以下全部简称为网。本文所指的网都是连通的。

2 子网的环境

网中任意一个元素或子网均受到其环境的影响,相关定义如下。

定义 2 设 $N=(P, T; F)$ 是一个网,设 $x \in P \cup T, Y \subseteq P \cup T$:

- (1) 称 $x^* = x \cup x^*$ 为 x 的关联集;
- (2) 称 $Fx = \{(x, y) \in F\} \cup \{(y, x) \in F\}$ 为 x 的关联弧集;
- (3) 称 $loc(x) = \{x\} \cup x^* \cup x^*$ 为 x 的局部环境;
- (4) 记 $Y = \{y \mid \exists x \in Y, (y, x) \in F\}$;
- (5) 记 $Y^* = \{y \mid \exists x \in Y, (x, y) \in F\}$;
- (6) 称 $loc(Y) = \{Y\} \cup Y^* \cup Y^*$ 为 Y 的局部环境。

其中概念(3)、(4)和(5)引自文献[2],其他概念是本文所作的相应扩充。局部环境 $loc(Y)$ 实际上由两部分组成,即 Y 和 Y 以外的元素。如果 Y 的元素构成一个子网 SN ,在实际应用中,通常要对以上两部分分开来考虑,因此定义 3 中提出 Y 的外部环境的概念(用“ $>$ ”来代表)。对子网 SN 而言,也可称其为外部接口(“接口”的概念只对子网有效;对于一个集合而言,使用“环境”或“外延”更为合理)。

定义 3 设 $N=(P, T; F)$ 是一个网(或子网),设 $Y \subseteq P \cup T$:

- (1) 记 $>Y = Y^* - Y$;
- (2) 记 $Y^> = Y^* - Y$;
- (3) 称 $srd(Y) = >Y \cup Y^>$ 为 Y 的外部环境。

^{*})国家自然科学基金(60503007)、广东省自然科学基金(06300970)资助。王志坚 博士,副教授,研究方向为工业系统建模、Petri 网等;魏定国 教授,博士,研究方向为 Petri 网和电子商务;吴时霖 教授,博导,研究方向为计算机网络技术、Petri 网理论及应用。

对于子网 SN 而言, $\succ SN$ 是 SN 的外部输入变迁或库所(外部输入接口), $SN \succ$ 是 SN 的外部输出变迁或库所(外部输出接口), $srd(SN)$ 是 N 中和 SN 相互影响的部分。对于网的相关部分可以更详细地定义如下:

定义 4 设 $SN=(P_S, T_S; F_S)$ 是 $N=(P, T; F)$ 的一个子网,

- (1) $SN_{\bar{P}} = \bigcup_{p \in P_S} p^* - T_S$; (2) $\succ SN_P = \bigcup_{p \in P_S} p - T_S$;
- (3) $srd(SN_P) = \Rightarrow SN_P \cup SN_{\bar{P}}$;
- (4) $SN_{\bar{T}} = \bigcup_{t \in T_S} t^* - P_S$; (5) $\succ SN_T = \bigcup_{t \in T_S} t - P_S$;
- (6) $srd(SN_T) = \Rightarrow SN_T \cup SN_{\bar{T}}$;
- (7) $SN \succ = SN_{\bar{P}} \cup SN_{\bar{T}}$; (8) $\succ SN = \Rightarrow SN_P \cup \succ SN_T$;
- (9) $srd(SN) = srd(SN_P) \cup srd(SN_T)$ 。

定义 4 有助于在子网研究中,更清晰地区分 SN 各类不同元素与 N 中相应元素的交互作用关系。通常情况下,如果 $srd(SN) = \phi$, 那么 SN 本身就是一个独立的网(非子网)或者空网(没有任何元素,这种情况一般不讨论)。

3 子网接口与边界

相对于子网的外部环境或外部接口,另外一个重要的概念是子网的内部接口或简称接口。

定义 5 设 $SN=(P_S, T_S; F_S)$ 是 $N=(P, T; F)$ 的一个子网,

- (1) $SN_{I\bar{P}} = \{p | p \in P_S \wedge (* p - T_S \neq \phi)\}$ 称为 SN 的输入库所接口集;
- (2) $SN_{I\bar{P}}^+ = \{p | p \in P_S \wedge (p^* - T_S \neq \phi)\}$ 称为 SN 的输出库所接口集;
- (3) $SN_{I\#} = SN_{I\bar{P}} \cup SN_{I\bar{P}}^+$ 称为 SN 的库所接口集;
- (4) $SN_{IT} = \{t | t \in T_S \wedge (* t - P_S \neq \phi)\}$ 称为 SN 的输入变迁接口集;
- (5) $SN_{IT}^+ = \{t | t \in T_S \wedge (t^* - P_S \neq \phi)\}$ 称为 SN 的输出变迁接口集;
- (6) $SN_{I\#} = SN_{IT} \cup SN_{IT}^+$ 称为 SN 的变迁接口集;
- (7) $SN_{I^-} = SN_{I\bar{P}} \cup SN_{IT}$ 称为 SN 的输入接口集;
- (8) $SN_{I^+} = SN_{I\bar{P}}^+ \cup SN_{IT}^+$ 称为 SN 的输出接口集;
- (9) $SN_{I\#} = SN_{I\#} \cup SN_{I\#}$ 称为 SN 的接口集。

虽然文献[1]中提出了“边界”的概念,但实际被用来描述一组库所变迁集合(以及相应的弧连接),更接近于本文的“子网”(不过不一定是连通的)。“边界”一词更适合于描述一个网络的边沿部分,因此下文定义 6 中给出了“边界”新的定义与内涵。定义 5 中“接口”被用来描述网络两个部分之间(例如网与其某个子网,或将要被连接的两个网络之间)的连接部分,因此“边界”与“接口”是两个不同的概念。通常,每个网都存在边界(也存在例外,例如图 1 所示,这是一个最小的、没有边界的正常的网),而接口在讨论一个完整而独立的网时是没有意义的。边界对于网络的层次化与模块化,即子网的抽象、变迁与库所的细化等问题没有太多影响,这个问题涉及的主要是接口。

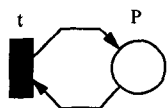


图 1 没有边界的正常的网

相对应于接口,给出边界的定义如下:

定义 6 设 $N=(P, T; F)$ 是一个网,

- (1) $NB_{\bar{P}} = \{p | p \in P \wedge (* p = \phi)\}$ 称为 N 的源库所边界集;
- (2) $NB_{\bar{P}}^+ = \{p | p \in P \wedge (p^* = \phi)\}$ 称为 N 的宿库所边界集;
- (3) $NB_{\#} = NB_{\bar{P}} \cup NB_{\bar{P}}^+$ 称为 N 的库所边界集;
- (4) $NB_{\bar{T}} = \{t | t \in T \wedge (* t = \phi)\}$ 称为 N 的源变迁边界集;
- (5) $NB_{\bar{T}}^+ = \{t | t \in T \wedge (t^* = \phi)\}$ 称为 N 的宿变迁边界集;
- (6) $NB_{\#} = NB_{\bar{T}} \cup NB_{\bar{T}}^+$ 称为 N 的变迁边界集;
- (7) $NB^- = NB_{\bar{P}} \cup NB_{\bar{T}}$ 称为 N 的源边界集;
- (8) $NB^+ = NB_{\bar{P}}^+ \cup NB_{\bar{T}}^+$ 称为 N 的宿边界集;
- (9) $NB_{\#} = NB_{\#} \cup NB_{\#}$ 称为 N 的边界集。

国外文献称定义 6 中式(5)描述的变迁“sink transition”,国内部分文献称之为“漏变迁”,似为不贴切,本文改称为“宿变迁”。

4 子网 SN 的接口标准化

定义 7 如果子网 SN 满足 $(SN_{I^+})^* = SN \succ \wedge^* (SN_{I^-}) = \succ SN$, 称子网 SN 为接口标准化的。

提出接口标准化的概念,是为了使子网与外界的接口更加清晰。标准化的接口中变迁的输入条件完全决定于其外部环境,不受子网内其他前提约束。从模块化的角度看,即由上层模块负责提供输入参数(即托肯),模块内部只决定如何运算,不干扰上层模块的运行。同样,标准化的接口中库所托肯的增加完全由外部环境中变迁决定,不受内部变迁影响。从模块化的角度看,它就是一个输入缓冲区,负责接受上层模块输入的参数;模块内部变迁决定如何使用这些参数,而不能决定或影响参数的输入情况。对于输出变迁接口或库所接口而言,情况是类似的。

对于非标准化接口的子网,可以先将其接口标准化。图 2 是接口标准化的几种典型方式,每组左边的是非标准化接口,右边是标准化后的结果。

5 子网的分类

根据接口的不同,我们将所有的子网分成不同的几种类型。

定义 8 设 $SN=(P_S, T_S; F_S)$ 是 $N=(P, T; F)$ 的一个子网:

- (1) 如果 $SN_{I\#} \neq \phi, SN_{I\#} = \phi$, 即 $SN_{I\#} = SN_{I\#}$, 称 SN 为 P 型子网;
- (2) 如果 $SN_{I\#} = \phi, SN_{I\#} \neq \phi$ 即 $SN_{I\#} = SN_{I\#}$, 称 SN 为 T 型子网;
- (3) 如果 $SN_{\bar{P}} \neq \phi, SN_{\bar{T}} \neq \phi, SN_{\bar{P}} = \phi, SN_{\bar{T}} = \phi$, 即 $SN_{I\#} = SN_{\bar{P}} \cup SN_{\bar{T}} \neq \phi$, 称 SN 为 PT 型子网;
- (4) 如果 $SN_{\bar{P}} = \phi, SN_{\bar{T}} = \phi, SN_{\bar{P}} \neq \phi, SN_{\bar{T}} \neq \phi$, 即 $SN_{I\#} = SN_{\bar{P}} \cup SN_{\bar{T}} \neq \phi$, 称 N 为 TP 型子网;
- (5) 如果 $SN_{\bar{P}} \neq \phi$ 且 $SN_{\bar{T}} \neq \phi$, 或者 $SN_{\bar{P}} \neq \phi$ 且 $SN_{\bar{T}} \neq \phi$, 称 N 为混合型子网。

由于混合型子网在抽象和细化过程中不便于处理,因此目前实际应用的子网一般可归入为前 4 种形式。定义 8 是从子网内部接口的角度来看待子网类型的。也可以从外部的角度来看,根据定义 8,可以简单得到以下推论。

推论 1 设 $SN=(P_S, T_S; F_S)$ 是 $N=(P, T; F)$ 的一个子

网:

- (1)如果 SN 为 P 型子网,那么 $srd(SN) \subseteq T-T_S$;
- (2)如果 SN 为 T 型子网,那么 $srd(SN) \subseteq P-P_S$;
- (3)如果 SN 为 PT 型子网,那么 $\supset SN \subseteq T-T_S \wedge SN \supset \subseteq P-P_S$;
- (4)如果 SN 为 TP 型子网,那么 $\supset SN \subseteq P-P_S \wedge SN \supset \subseteq T-T_S$;
- (5)如果 SN 为 TP 型子网,那么 $(srd(SN) \cap P-P_S) \neq \phi \wedge (srd(SN) \cap T-T_S) \neq \phi$.

显然,从子网的上层网来看, P 型子网和 T 型子网最简单,它们看起来就像一个库所或者一个变迁。从功能模块化的角度来看, T 型子网是最自然的一种子网形式。

P 型子网 SN 有两种特殊情况: $SN_P^+ = \phi$ 或 $SN_P^- = \phi$ 。

定义 9 对于 P 型子网 SN :

- (1)如果 $SN^{\#} = SN_P^- \neq \phi (SN_P^+ = \phi)$,称 SN 为 P^- 型子网(P 型宿子网);
- (2)如果 $SN^{\#} = SN_P^+ \neq \phi (SN_P^- = \phi)$,称 SN 为 P^+ 型子网(P 型源子网);

(3)如果 $SN_P^+ \neq \phi, SN_P^- \neq \phi$,称 SN 为 $P^{\#}$ 型子网(普通 P 型子网)。

T 型子网也有两种特殊情况: $SN_T^+ = \phi$ 或 $SN_T^- = \phi$ 。

定义 10 对于 T 型子网 SN :

- (1)如果 $SN^{\#} = SN_T^- \neq \phi (SN_T^+ = \phi)$,称 SN 为 T^- 型子网(T 型宿子网);
- (2)如果 $SN^{\#} = SN_T^+ \neq \phi (SN_T^- = \phi)$,称 SN 为 T^+ 型子网(T 型源子网);
- (3)如果 $SN_T^+ \neq \phi, SN_T^- \neq \phi$,称 SN 为 $T^{\#}$ 型子网(普通 T 型子网)。

源子网和宿子网的称谓是针对其上层网而言的,可以这样理解。对于其上层网 N 而言,如果子网 SN 是消耗托肯的,那么它是宿子网,记为“-”(P^-, T^-);如果子网 SN 是生产托肯的,那么它是源子网,记为“+”(P^+, T^+);如果子网 SN 既消耗托肯又生产托肯,那么它是普通子网,记为“#”($P^{\#}, T^{\#}$)。

几种常见的子网结构如图 3 所示,其中虚线框外为接口部分,框内为子网内部其他部分。

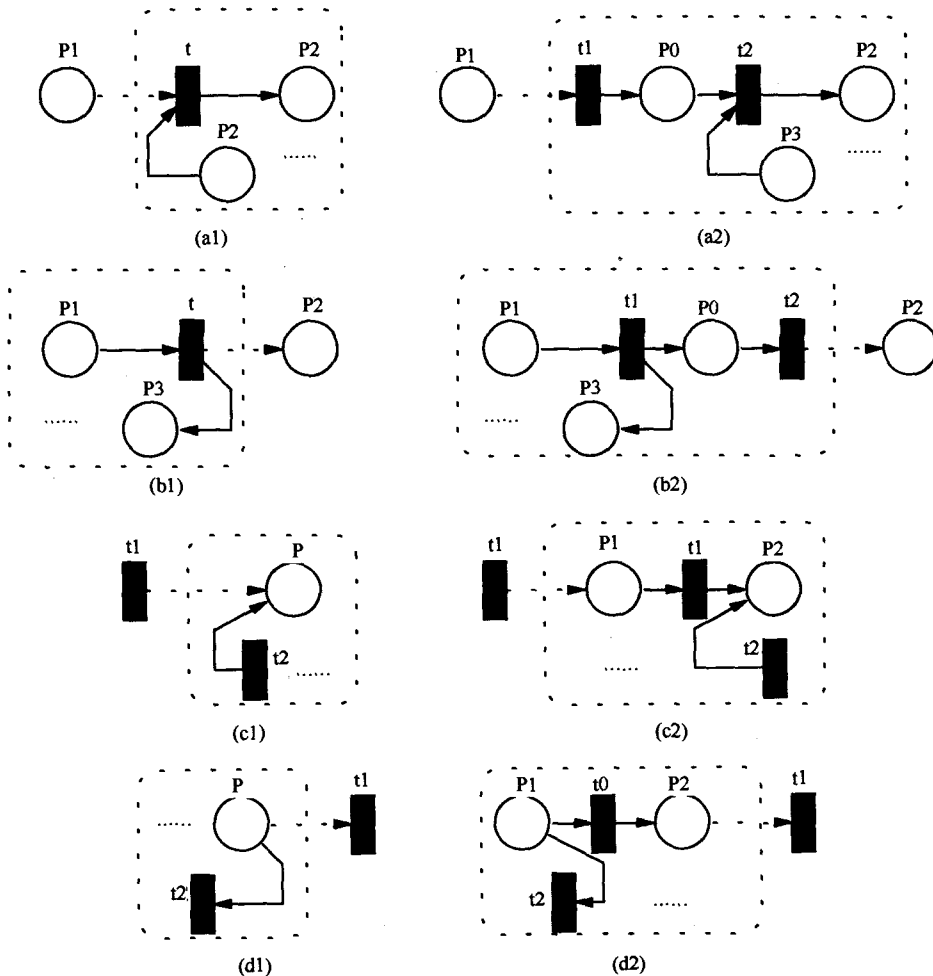


图 2 非标准化接口的标准化

定义 11 对于子网 SN :

- (1)如果 $|SNI^-| = 0$,称其为零输入接口的,记为 ZI (Zero Input);
- (2)如果 $|SNI^-| = 1$,称其为单输入接口的,记为 SI (Single Input);
- (3)如果 $|SNI^-| > 1$,称其为多输入接口的,记为 MI

(Multi Input);

- (4)如果 $|SNI^+| = 0$,称其为零输出接口的,记为 ZO (Zero Output);
- (5)如果 $|SNI^+| = 1$,称其为单输出接口的,记为 SO (Single Output);
- (6)如果 $|SNI^+| > 1$,称其为多输出接口的,记为 MO

(Multi Output)。

这里的输入、输出是针对子网 SN 而言的。相对于其上

层网 N; SN 的输入是 N 的输出, SN 的输出是 N 的输入。通过输入、输出的组合,总共可以形成 9 类子网。

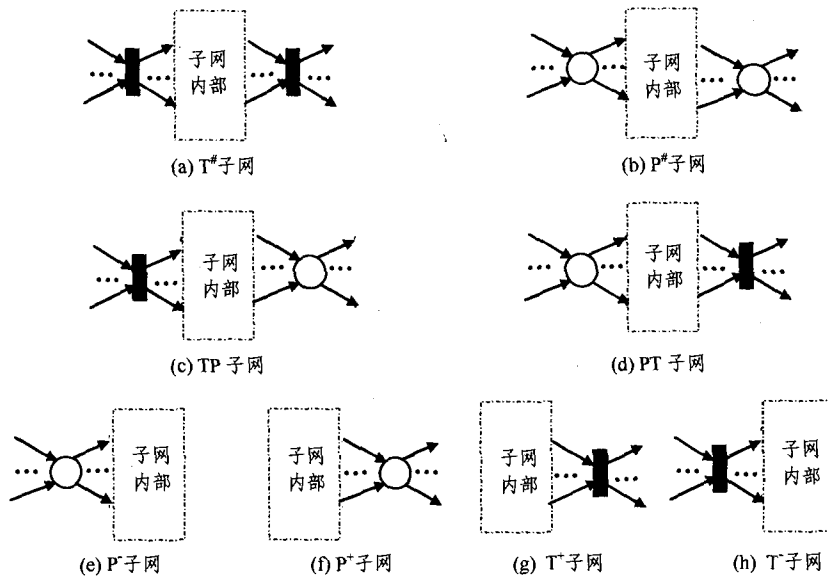


图 3 几种常见的子网结构

(1)ZIZO:无输入输出接口的独立网,严格说不属于子网范畴;

(2)ZISO:无输入接口,有一个输出接口,属于单输出的 P^+ 型或 T^+ 型源子网,如图 2 中(f)和(g);

(3)ZIMO:无输入接口,有多个输出接口,属于一般的 P^+ 型或 T^+ 型源子网;

(4)SIZO:单个输入接口,无输出接口,属于单输入的 P^- 型或 T^- 型宿子网,如图 2 中(e)和(h);

(5)SISO:单个输入接口,有一个输出接口,属于单输入单输出普通 $P^{\#}$ 型、 $T^{\#}$ 型子网、 PT 型子网或 TP 型子网,如图 2 中(a)、(b)、(c)和(d);

(6)SIMO:单个输入接口,有多个输出接口,属于单输入普通 $P^{\#}$ 型、 $T^{\#}$ 型子网、 PT 型子网或 TP 型子网;

(7)MIZO:多个输入接口,无输出接口,属于单输入的 P^- 型或 T^- 型源子网;

(8)MISO:多个输入接口,有一个输出接口,属于单输入的 $P^{\#}$ 型、 $T^{\#}$ 型子网、 PT 型子网或 TP 型子网;

(9)MIMO:多个输入接口,有多个输出接口,属于一般的 $P^{\#}$ 型、 $T^{\#}$ 型子网、 PT 型子网或 TP 型子网。

目前研究和应用得最多的是 SISO 型子网,文献[10,12,13]研究的均归属这一类。

6 规范子网

尽管对于网的抽象与细化是互逆的操作,不过它们未必一一对应。如果子网 SN 已确定,那么对于 N_1 的抽象是唯一的,不存在多种可选方案,即 $N_1 \xrightarrow{N_1} N_2$ 是一一映射的。对于细化操作而言,如果子网 SN 和子网变迁 t_N 已确定,原有的 t_N 关联弧到 SN 的对应方案有多种,即 $N_2 \xrightarrow{t_N, SN} N_1$ 是一到多的映射,这意味着细化操作中弧的定义实际上有很多种方案可以选择。单从模块化系统设计的角度考虑,只要输入(*t_N)、输出(t_N^*)和模块功能(${}^*t_N \rightarrow t_N^*$)确定,模块的具体实现可以灵活多样,在 ${}^*SN = {}^*t_N$ 且 $SN^{\#} = t_N^*$ 的前提下,

SN 的结构可以多种多样。不过考虑到变迁的触发特点,点火时要求所有输入同时成立 ($\forall p \in {}^*t_N: M(p) > 0$),点火后所有输出同时生成 ($\forall p \in t_N^*: M(p) = M(p) + 1$),因此子网应该满足一个条件,即子网的一次运行应该消耗所有 *SN 中库所中的一个托肯,而为所有的 $SN^{\#}$ 中库所生产一个托肯。

显然,对于同样的外部环境,通常有多种子网结构可以满足上述条件。对于一个正常的变迁 t_N (指 ${}^*t_N \neq \phi$ 且 $t_N^* \neq \phi$),最简单直观的做法,是将其分裂为如图 3(a)所示的输入、中间处理和输出三部分。基于这样的一种思路,提出规范子网的概念。

定义 12 称子网 SN 是规范的,如果满足以下条件:(1) SN 为接口标准化的子网;(2) $|SNI^-| \leq 1$;(3) $|SNI^+| \leq 1$ 。

显然,定义 12 等价于以下推论:

推论 2 任意接口标准化的 SISO, SIZO, ZISO 子网 SN 是规范子网。

分别简称接口标准化的 SISO, SIZO, ZISO 子网为 SISO, SIZO, ZISO 规范子网。对于规范 T 型子网,这三种情形分别为规范 $T^{\#}$ 子网、规范 T^- 子网、规范 T^+ 子网。

目前已有的对子网的研究,基本都属于规范的 T 子网,而且通常是其中的一种,即 SISO 型的子网,不过实际应用系统中经常涉及到 SIZO, ZISO 子网。

规范 T 型子网并非满足前述要求的唯一 T 型子网,不过它是接口结构最为简单的符合要求的类型,而且容易给出明确的定义。因此,本文将规范 T 型子网作为基于 T 型子网的细化与抽象的标准结构,其他类型满足一定条件的 T 型子网可以转化为规范 T 型子网。

结束语 根据子网的特点对子网分类有助于基于子网的结构化设计方法的研究和应用。子网标准化与规范化主要解决子网与变迁之间的“形似”问题,即使得子网与外部的衔接和原有的子网变迁是一致的,对于子网变换而言,不只需要形似,还要“神似”,即使模型在变换前后具有等价的行为特征。为了解决这些问题,首先要分析子网的行为,这通常是利用子

(下转第 167 页)

$$\begin{aligned}
&= r, Z = Z(r) - 2H(X | Q = r) \\
&= 2 \int p(X, Q = q) \log \frac{1}{p(X | Q = q)} dX - \int p(X, Z = (q), \\
&\quad Q = q) \log \frac{1}{p(X | Q = q, Z = Z(q))} dX \\
&+ \int p(X, Q = Z(r), Q = r) \log \\
&\quad \frac{1}{p(X | Q = r, Z = Z(r))} dX - 2 \int p(X, Q = r) \log \\
&\quad \frac{1}{p(X | Q = r)} dX \\
&\geq \int p(X, Q = q) \log \frac{p(X | Q = q, Z = Z(q))}{p(X | Q = q) p(X | Q = q)} dX \\
&+ \int p(X, Z = Z(r), Q = r) \log \frac{1}{p(X | Q = r, Z = Z(r))} dX \\
&- 2 \int p(X, Q = q) \log \frac{1}{p(X | Q = r)} dX \\
&= \int p(X, Q = q) \log \\
&\quad \frac{p(X | Q = q, Z = Z(q)) p(X | Q = r) p(X | Q = r)}{p(X | Q = q) p(X | Q = q)} dX \\
&+ \int p(X, Z = Z(r), Q = r) \log \frac{1}{p(X | Q = r, Z = Z(r))} dX \\
&\geq \int p(X, Z = Z(r), Q = r) \log \\
&\quad \frac{p(X | Q = q, Z = Z(q)) p(X | Q = r) p(X | Q = r)}{p(X | Q = q) p(X | Q = q) p(X | Q = r, Z = Z(r))} dX \\
&\geq \int p(X, Z = Z(r), Q = r) \log \\
&\quad \frac{p(X | Q = r, Z = Z(r)) p(X | Q = r) p(X | Q = r)}{p(X | Q = q) p(X | Q = q) p(X | Q = r, Z = Z(r))} dX \\
&= \int p(X, Z = Z(r), Q = r) \log \\
&\quad \frac{p(X | Q = r) p(X | Q = r)}{p(X | Q = q) p(X | Q = q)} dX \\
&\geq 0
\end{aligned}$$

(注:以上推导中利用了 $p(X, Q = q) \geq p(X, Z = Z(q), Q = q)$, $p(X, Q = q) \geq p(X, Z = Z(q), Q = q) \geq p(X, Z = Z(r), Q = r)$, $p(X | Q = q, Z = Z(q)) \geq p(X | Q = r, Z = Z(r))$, $p(X | Q = q) \geq p(X | Q = r)$, 这是由于 X 是 t 时刻的观测值, $Q = q$ 是 t 时刻的当前状态, 其对应的各相关概率必显然大于非当前时刻 t 对应状态下的各相关概率值)

(上接第 134 页)

网的测试网来进行的。如何使模型在变换前后继续保持原有属性, 是一个需要进一步深入研究的问题。

参考文献

- [1] Girault C, Valk R. Petri Nets for Systems Engineering: A Guide to Modeling, Verification, and Applications. Berlin: Springer-Verlag, 2002
- [2] Murata T. Petri Nets: Properties, Analysis and Applications. Proceedings of the IEEE, 1989, 77(4): 541-580
- [3] 袁崇义. Petri 网原理与应用. 北京: 电子工业出版社, 2005
- [4] Padberg J. Petri net modules. Journal of Integrated Design and Process Science, 2002, 6(4): 105-120
- [5] 王培良, 赵义军, 叶志宝. Petri 网的并分解. 控制理论与应用, 2001, 18(1): 116-119
- [6] Huang H, Cheung T Y, Mak W M. Structure and behavior preservation by Petri-net-based refinements in system design. Theoretical Computer Science, 2004, 328: 245-269
- [7] 蒲飞, 陆维明. 同步合成 Petri 网系统可回复性的保持性. 系统

以上证明显示了 FBMM 模型的区分度优于 HMM, 误识率低于 HMM, 而这两者正是评价模型优劣的主要标准。

性质 4 FBMM 较之于 HMM 具有更高的鲁棒性。

由于 FBMM 采用模糊随机变量刻画状态间、观测值间及其之间的相关性, 其值用具有弹性的区间来表示, 更好地反映了模型应用的实际情况, 并且 FBMM 能够对训练数据中的部分缺失进行优化填充, 因此其鲁棒性优于 HMM 是显见的。

结束语 本文提出一种对经典隐马尔可夫模型改进的模型——模糊深隐马尔可夫模型。该模型充分考虑到观测值间的相关性和转移概率值的不确定性, 通过相关理论证明可知模糊深隐马尔可夫模型与经典隐马尔可夫模型相比较具有在同一训练数据下优化程度高、模式区分度好、鲁棒性强, 且模型复杂度没有明显增加的优点。相信具有这些特性的模糊深隐马尔可夫模型将能够在高性能要求的复杂模式识别中具有广泛的应用前景。

参考文献

- [1] 龚光鲁, 钱敏平. 应用随机过程教程[M]. 北京: 清华大学出版社, 2004: 249-255
- [2] Bilmes J A. Buried Markov models for speech recognition//Proceeding of the IEEE international Conference on Acoustics, Speech, and Signal Processing. Phoenix, AZ, March 1999: 713-716
- [3] Bilmes J A. Buried Markov models- a graphical-modeling approach to automatic speech recognition. Computer Speech and Language, 2003, 17: 213-231
- [4] 冯前进, 陈武凡. 模糊马尔可夫场模型与图像分割新算法[J]. 南方医科大学学报, 2006(5): 579-583
- [5] Avrachenkov K E, Sanchez E. Fuzzy Markov Chains: Specificities and Properties//Proc IPMU. Madrid, Spain, July 2000: 1851-1856
- [6] Tran D, Wagner M. Generalised Fuzzy Hidden Markov Models for Speech Recognition//Proceedings of the International Conference on Fuzzy Systems, Calcutta, India, 2002: 84-91
- [7] Givan R, Leach S, Dean T. Bounded parameter markov decision processes. Artificial Intelligence, 2000, 122(1/2): 71-109
- [8] 茅晓泉, 胡光锐. 基于最大互信息的离散隐马尔可夫模型训练方法[J]. 上海交通大学学报, 2001(11): 1713-1717
- [9] Pearl J. Probabilistic Reasoning in Intelligent Systems: Net2 works of Plausible Inference. San Mateo, California: Morgan Kaufmann, 1988: 71-78
- [10] 张连文, 郭海鹏. 贝叶斯网引论[M]. 北京: 科学出版社, 2006: 36-50
- [11] Information theory as an extension of the maximum likelihood principle//Petrov V, Csaki F, eds. Second International Symposium on Information Theory. Budapest: Akailseoniai Kiado

工程理论与实践, 2005, 25(10): 83-94

- [8] 蒲飞, 陆维明. 同步合成 Petri 网系统活性与无死锁性的保持性. 软件学报, 2003, 14(12): 1977-1988
- [9] Bernadsky M, Sharykin R, Alur R. Formal Techniques, Modeling and Analysis of Timed Systems. Lecture Notes in Computer Science, 2004, 3253: 309-325
- [10] Tsinarakis G J, Valavanis K P. Studying Multi-operational Production Systems with Modular Hybrid Petri Nets//Intl. Conference on Intelligent Robots and Systems. Las Vegas, Nevada, 2003, 3028-3046
- [11] 张继军, 吴哲辉. Petri 网的分层递归模型. 系统仿真学报, 2003, 15(1): 89-92
- [12] Kristensen L M, Christensen S, Jensen K. The practitioner's guide to coloured Petri nets. International Journal on Software Tools for Technology Transfer, Springer, 1998, 2(2): 98-132
- [13] Proth J M. A Class of Petri Nets for Manufacturing System Integration. IEEE Transactions on Robotics and Automation, 1997, 13(3): 317-326
- [14] 夏传良, 焦莉, 陆维明. Petri 网精细化操作及其在系统设计中的应用. 软件学报, 2006, 17(1): 11-20