文章编号：1001-0920(2007)08-0864-05

π演算的Petri网语义研究

于振华[1]，蔡远利[1]，徐海平[2]
(1. 西安交通大学 电子与信息工程学院，西安 710049；2. 马萨诸塞州立大学达特茅斯分校 计算机与信息科学系，北达特茅斯 02747)

摘 要：为弥补π演算的固有缺陷，提出一种将π演算映射为Petri网语义的方法. 该方法将π演算分为基本单元、顺序、并发、选择和递归等几种基本结构，分别映射为Petri网，然后复合构成复杂的系统. π演算的Petri网语义可形象地描述系统的物理结构和动态行为，可直接从模型网络结构上定性分析系统的性质. 最后，利用该方法将移动汽车网络的π演算模型映射为Petri网，验证了方法的有效性.

关键词：π演算；Petri网；并发；结构特性；分析
中图分类号：TP301 **文献标识码**：A

On Petri nets semantics for π-calculus

YU Zhen-hua[1], *CAI Yuan-li*[1], *XU Hai-ping*[2]
(1. School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China; 2. Department of Computer and Information Science, University of Massachusetts Dartmouth, North Dartmouth 02747, USA. Correspondent: YU Zhen-hua, E-mail: zhenhua_yu@163.com)

Abstract: In order to remedy the deficiencies of π-calculus, π-calculus is mapped into Petri nets. π-calculus is divided into the basic elements, sequence, concurrency, choice and recursive modules. These modules are mapped into Petri nets respectively, which construct a complicated system. Petri nets semantics for π-calculus visually describe system structure as well as system behaviors, and the qualitative analysis of properties is proved directly on the structure of the nets. Finally, the mobile car network is used to show how the π-calculus model is mapped into Petri nets.

Key words: π-calculus; Petri nets; Concurrency; Structual characteristics; Analysis

1 引 言

Petri网和π演算以严格的数学理论为支撑，是研究并发系统的主要理论模型，在并发系统的建模、分析、设计和验证等方面得到了广泛应用[1]. π演算[2]可用来描述拓扑结构动态变化的并发系统，通过相应的数学分析方法，对系统的属性和行为进行分析. 但π演算的进程表达式比较繁杂，不能形象地反映系统物理结构信息，不能刻画系统真并发行为，而且支持工具较少，仅有MWB和HAL等几种自动验证工具. Petri网[3]是一种具有严格数学语义的形式化图形建模工具，适合描述并发、异步和分布式系统. 它侧重于系统的结构描述和性质分析，能有效地刻画系统的真并发语义. 目前有很多工具可以模拟、分析和验证Petri网模型[4].

由于π演算存在固有缺陷，本文考虑将π演算转换为Petri网语义，从而可有效利用Petri网的结构化分析方法和支持工具分析具有动态拓扑结构的并发系统. 近年来，已提出一些将π演算转换为Petri网语义的方法[5-8]. 尽管现有方法为π演算转换为Petri网语义提供了一定的支持，但有的方法引入了一些特殊的Petri网，不能利用现有Petri网的支持工具，而且大部分方法比较繁琐，不利于工程应用.

本文根据π演算进程的特点，将其分为基本单元、递归、顺序、并发和选择等几种基本结构，分别映射为Petri网. 复杂π演算进程的Petri网语义都可通过基本结构复合而成，因而具有较大的可操作性和可计算性.

2 π演算的Petri网语义

将π演算的进程 P 对应的Petri网模型记为

收稿日期：2006-04-27；**修回日期**：2006-11-29.
基金项目：国家863计划项目(2003AA721070).
作者简介：于振华(1977—)，男，山东乳山人，博士生，从事复杂系统建模与分析、多Agent系统理论与应用等研究；蔡远利(1963—)，男，贵州瓮安人，教授，博士生导师，从事智能制导、多Agent系统及应用等研究.

N_P，在 N_P 中采用着色 Token，同时弧带有弧表达式，变迁带有卫函数. π 演算的通道分为受限通道和非受限通道，受限通道只能在进程内部通信. 按照文献[8]的方法，与受限通道相关联的变迁和弧加标签(label)注明，不能与其他进程的 Petri 网模型交互. 库所分为输入、内部和输出库所，分别用 e(entry)，i(internal) 和 x(exit) 表示[8]. 同时表示进程的初始状态(即 P 尚未执行任何动作)、内部状态和终止状态(即进程 P 顺利结束)，且 e 的前置变迁集合为空，x 的后置变迁集合为空. π 演算中的动作相应于 Petri 网中的变迁. 变迁具有两种不同的类型：普通变迁和通信变迁 τ. 变迁 τ 由 π 演算的对偶名字映射而成.

在 π 演算中，为更好地描述系统的动态行为特征，参考 CSP，引入轨迹(trace)的概念[9].

定义 1　一个进程 P 所能依次执行的一系列动作称为 P 的一条执行轨迹，通常用 ⟨action1, action2,…⟩ 表示. 进程 P 所有可能的执行轨迹集合记为 traces(P)，并且空轨迹⟨⟩属于任何进程的轨迹集合，即对于任何进程 P，⟨⟩ ∈ traces(P).

定义 2　两条执行轨迹 s 和 t 的连接即为它们动作序列的连接，用 $s^\wedge t$ 表示.

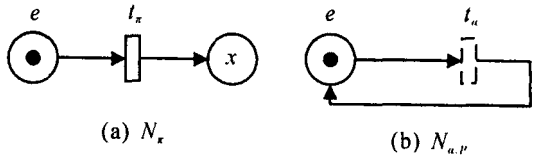
π 演算的执行轨迹类似于 Petri 网的发射序列 σ，也能刻画系统的动态行为特征. 将 π 演算的前缀表达式 $\pi \in \{\tau, y(x), \bar{y}x, \bar{y}(x)\}$ 作为进程的基本单元，并将进程归纳为递归结构、顺序结构、并发结构和选择结构，下面分别研究将其映射为 Petri 网语义的方法.

2.1　π 演算基本单元的 Petri 网语义

基本单元 $\pi \in \{\tau, y(x), \bar{y}x, \bar{y}(x)\}$ 表示进程的动作，进程由基本单元经过顺序、并发等操作组合而成.

映射规则 1　从 $\pi.0$ 到 N_π.

将动作 π 用一个变迁 t_π 表示，为变迁 t_π 增加包含一个 Token 的输入库所和输出库所，如图 1(a) 所示.



(a) N_π　　(b) $N_{a.P}$

图 1　基本单元及递归结构的 Petri 网语义

由图 1 知，traces($\pi.0$) = {⟨⟩, ⟨π⟩}，$\sigma_S(N_\pi)$ = {⟨⟩, ⟨t_π⟩}，π 和 t_π 是同一个动作的不同符号表示，因此可得如下结论：

结论 1　如果使用映射规则 1，由 π 演算模型 $\pi.0$ 得到 Petri 网模型 N_π，则 traces($\pi.0$) = $\sigma_S(N_\pi)$.

2.2　π 演算递归结构的 Petri 网语义

π 演算递归的定义类似于 CCS[2]，$P =_{def} \alpha.P$ 表示 P 中动作 α 执行无限次.

映射规则 2　从 $P =_{def} \alpha.P$ 到 $N_{\alpha.P}$.

假设进程 P 已经映射为 N_P，进程 $P =_{def} \alpha.P$ 映射到 $N_{\alpha.P}$ 的规则是：将动作 α 用一个变迁 t_α 表示，删除 N_P 的输出库所及其输入弧，在变迁 t_α 和输入库所之间加入一条弧，如图 1(b) 所示.

在本文中，如无特殊说明，虚线库所和变迁都表示抽象，可进一步精化到具体的内部实现；并假设进程 P 和 Q 已分别转化为 Petri 网模型 N_P 和 N_Q，且 traces(P) = $\sigma_S(N_P)$，traces(Q) = $\sigma_S(N_Q)$.

很显然，进程 $P =_{def} \alpha.P$ 的执行轨迹为 traces($[a=b]P$) = {⟨⟩, ⟨α⟩^{n}}，表示执行 n 次动作 α. 根据映射规则 2 和图 1(b) 知，$N_{\alpha.P}$ 的发射序列为 traces($N_{\alpha.P}$) = {⟨⟩, ⟨t_α⟩^{n}}. 由于 α 和 t_α 是同一个动作的不同符号表示，可得出如下结论：

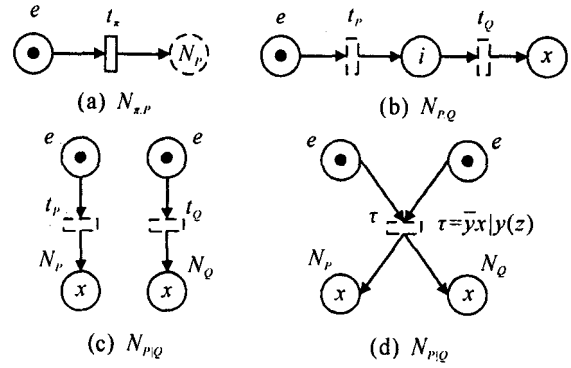
结论 2　如果使用映射规则 2，由 π 演算模型 $P =_{def} \alpha.P$ 得到 Petri 网模型 $N_{\alpha.P}$，则 traces($P =_{def} \alpha.P$) = $\sigma_S(N_{\alpha.P})$.

π 演算的复制算子 $!P = P \mid !P$ 表示 P 的 n 次复制，其 Petri 网语义也可根据映射规则 2 得到.

2.3　π 演算顺序结构的 Petri 网语义

映射规则 3　从 $\pi.P$ 到 $N_{\pi.P}$.

进程 $\pi.P$ 表示执行完动作 π 后，执行 P. $\pi.P$ 转化为 Petri 网模型的规则是：将动作 π 用变迁 t_π 表示，为变迁 t_π 增加包含一个 Token 的前置库所 e 作为模型的输入库所，把 N_P 抽象为一个抽象库所，则 t_π 作为该抽象库所的前置变迁，如图 2(a) 所示.



(a) $N_{\pi.P}$　(b) $N_{P;Q}$　(c) $N_{P|Q}$　(d) $N_{P|Q}$

图 2　顺序及并发结构的 Petri 网语义

进程 $\pi.P$ 的执行轨迹集合 traces($\pi.P$) = {⟨⟩, ⟨π⟩$^\wedge s \wedge s \in$ traces(P)}. 由映射规则 3 和图 2(a) 可知，$N_{\pi.P}$ 在变迁 t_π 发射之后的发射序列集合与进程 P 所对应的 Petri 网模型 N_P 的发射序列集合完全相同，即 $\sigma_S(N_{\pi.P})$ = {⟨⟩, ⟨t_π⟩$^\wedge \sigma_S(N_P)$}. 由于 π 和 t_π 是同一个动作的不同符号表示，可得出如下结论：

结论 3　如果使用映射规则 3，由 π 演算模型

$\pi.P$ 得到 Petri 网模型 $N_{\pi.P}$，则 $\text{traces}(\pi.P) = \sigma s(N_{\pi.P})$.

映射规则 4 从 $P.Q$ 到 $N_{P.Q}$.

进程 $P.Q$ 可表示顺序结构，进程 Q 在进程 P 之后执行. 进程 $P.Q$ 转化为 Petri 网模型 $N_{P.Q}$ 的规则是：删除 N_Q 的输入库所中的 Token，并将该库所与 N_P 的输出库所合并为一个库所作为 N_Q 的输入库所；N_P 的输入库所和 N_Q 的输出库所分别作为 $N_{P.Q}$ 的输入和输出库所，如图 2(b) 所示.

如果进程 P 执行 s_1 后能够成功结束，则进程 Q 能执行，那么 $\text{traces}(P.Q) = \{s_1 \wedge s_2 \wedge s_1 \in \text{traces}(P) \wedge s_2 \in \text{traces}(Q)\}$. 由映射规则 4 和图 2(b) 知，当进程 P 成功结束时，N_P 中的 Token 到达 N_Q 的输入库所，因此 $\sigma s(N_{P.Q}) = \{\sigma_1 \wedge \sigma_2 \wedge \sigma_1 \in \sigma s(N_P) \wedge \sigma_2 \in \sigma s(N_Q)\}$.

如果进程 P 死锁，则 $\text{traces}(P.Q) = \{\text{traces}(P)\}$，此时 N_P 死锁，其输出库所即 N_Q 的输入库所不包含 Token，N_Q 的变迁永远不满足使能条件，那么 $\sigma s(N_{P.Q}) = \{\sigma s(N_P)\}$. 于是可得如下结论：

结论 4 如果使用映射规则 4，由 π 演算模型 $P.Q$ 得到 Petri 网模型 $N_{P.Q}$，则 $\text{traces}(P.Q) = \sigma s(N_{P.Q})$.

2.4 π 演算并发结构的 Petri 网语义

π 演算使用并发操作符 $|$ 表示并发结构.

映射规则 5 从 $P \mid Q$ 到 $N_{P|Q}$.

进程 $P \mid Q$ 包含两种类型：1) 如果进程 P 和 Q 是孤立的，则 $P \mid Q$ 映射到 $N_{P|Q}$，如图 2(c) 所示，N_P 和 N_Q 独立并发执行；2) 如果进程 P 和 Q 通过对偶名字进行通信，如 $P = \bar{y}x$ 或 $P = \bar{y}(x)$，$Q = y(z)$，则进程 $P \mid Q$ 产生一个通信变迁 $\tau = \bar{y}x \mid y(z)$. 进程 P 和 Q 存在同步关系，利用握手协议进行通信[10]，将进程中的对偶名字合并成变迁 τ. 变迁 τ 是实际上的握手协议，可解释为内部动作或哑动作，它只被进程 P 和 Q 利用，不能包含在第 3 方的进程中，如图 2(d) 所示.

进程 $P \mid Q$ 的执行轨迹可表示为 $\text{traces}(P \mid Q) = \{s_1 \in \text{traces}(P) \wedge s_2 \in \text{traces}(Q)\}$. 由映射规则 5 和图 2(c) 知，对于类型 1)，$N_{P|Q}$ 的发射序列为 $\sigma s(N_{P|Q}) = \{\sigma_1 \in \sigma s(N_P) \wedge \sigma_2 \in \sigma s(N_Q)\}$；对于类型 2)，可将两个相应的通信动作（发送和接收，互为对偶）合并为一个变迁 τ，实际上输入输出动作也都得到了执行，发射序列与类型 1) 相同. 因此可得如下结论：

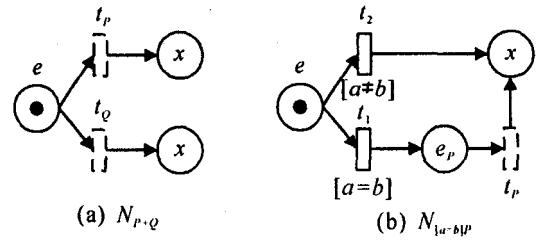
结论 5 如果使用映射规则 5，由 π 演算模型 $P \mid Q$ 得到 Petri 网模型 $N_{P|Q}$，则 $\text{traces}(P \mid Q) = \sigma s(N_{P|Q})$.

2.5 π 演算选择结构的 Petri 网语义

π 演算使用求和操作符 $+$ 表示选择结构；匹配算子 $[a = b]P$ 也可以看作一种特殊的选择结构，若 $a = b$，则执行 P，否则终止[11].

映射规则 6 从 $P+Q$ 到 N_{P+Q}.

进程 $P+Q$ 映射到 N_{P+Q} 的规则是：将 N_P 和 N_Q 的输入库所合并，作为公共的输入库所，如图 3(a) 所示.



(a) N_{P+Q} (b) $N_{[a=b]P}$

图 3 选择结构的 Petri 网语义

进程 $P+Q$ 的执行轨迹可表示为 $\text{traces}(P+Q) = \{s \mid s \in \text{traces}(P) \vee s \in \text{traces}(Q)\}$. 根据映射规则 6 和图 3(a) 知，$N_{P+Q}$ 的发射序列为 $\sigma s(N_{P+Q}) = \{\sigma \mid \sigma \in \sigma s(N_P) \vee \sigma \in \sigma s(N_Q)\}$. 因此可得如下结论：

结论 6 如果使用映射规则 6，由 π 演算模型 $P+Q$ 得到 Petri 网模型 N_{P+Q}，则 $\text{traces}(P+Q) = \sigma s(N_{P+Q})$.

映射规则 7 从 $[a = b]P$ 到 $N_{[a=b]P}$.

进程 $[a = b]P$ 映射到 $N_{[a=b]P}$ 的规则是：添加辅助变迁 t_1 和 t_2 作为 N_P 输入和输出库所的前置变迁，将匹配算子作为前置变迁的卫函数 (guard function)，判断名字是否匹配，决定下一步的动作. 然后为 t_1 和 t_2 添加带有一个 Token 的输入库所，如图 3(b) 所示.

进程 $[a = b]P$ 的执行轨迹可表示为 $\text{traces}([a = b]P) = \{\langle\rangle, \langle[a \neq b]\rangle, \langle[a = b]\rangle \wedge s \wedge s \in \text{traces}(P)\}$. 根据映射规则 7 和图 3(b) 知，$N_{[a=b]P}$ 的发射序列为 $\sigma s(N_{[a=b]}P) = \{\langle\rangle, \langle t_2\rangle, \langle t_1\rangle \wedge \sigma \wedge \sigma \in \sigma s(N_P)\}$. 由于 $[a = b]$ 和 t_1，$[a \neq b]$ 和 t_2 是同一个动作的不同符号表示，可得出如下结论：

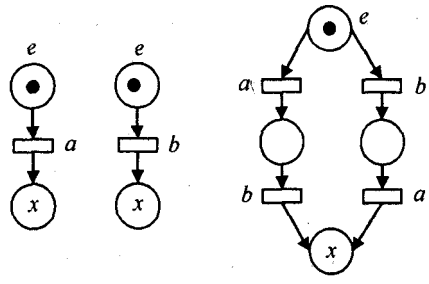
结论 7 如果使用映射规则 7，由 π 演算模型 $[a = b]P$ 得到 Petri 网模型 $N_{[a=b]P}$，则 $\text{traces}([a = b]P) = \sigma s(N_{[a=b]P})$.

根据以上 7 条映射规则，可得到 π 演算进程的 Petri 网语义，对其进行复合，可得到整个系统的模型.

2.6 π 演算的 Petri 网语义有效性评价

π 演算的 Petri 网语义不能改变系统的功能特性，可根据文献[12] 从以下 2 个标准来评价这种方法的有效性：并发性和功能一致性.

并发性　π 演算的 Petri 网语义应能表示其期望的并发性. π 演算以交织语义为基础，其并发性可能被约简为非确定性选择，无法体现出不确定性选择活动和并发活动的差别. 如进程 $P = a \mid b$ 和 $Q = a.b + b.a$，π 演算不能区分进程 P 和 Q 之间的差别，而将其映射为 Petri 网语义后(如图 4 所示)，才能真正区分并发(图 4(a)) 和非确定性选择(图 4(b)).



(a) 并发　　(b) 非确定性选择

图 4　并发性和非确定性选择

功能一致性　在 π 演算进程执行过程中，状态的集合可以看作是有向图的节点，而有向图的箭头相应于进程执行的动作，有向图可称为变迁系统 (TS)[8]，类似于 Petri 网中的可达图(RG). 由此可得出以下定义：

定义 3　设二元关系 $S \subseteq \mathrm{TS} \times \mathrm{RG}$，$M$ 为 Petri 网标识，如果 $(P,M) \in S$，则对于进程 P 的动作 α 和变迁 $t \in T$，下列条件成立：1) 如果 $P \xrightarrow{\alpha} P'$，则 $\exists M', M[t\rangle M'$，并且 $(P',M') \in S$；2) 如果 $M[t\rangle M'$，则 $P', P \xrightarrow{\alpha} P'$，并且 $(P', M') \in S$. 则称 S 是强互模拟关系.

π 演算的变迁系统和可达图的强互模拟关系表明，变迁系统中的动作必须对应于可达图中的变迁. 根据结论 1 ～ 7，进程 P 的每一条执行轨迹 trace，其 Petri 网模型 N_P 都有一条发射序列 σ 相对应，使得 trace 中的动作及其执行顺序与 σ 中的动作及其执行顺序相同，因此，P 的变迁系统与 N_P 的可达图具有强互模拟关系. 于是可得如下定理：

定理 1　π 演算进程 P 与其 Petri 网语义模型 N_P 在功能上是等价的.

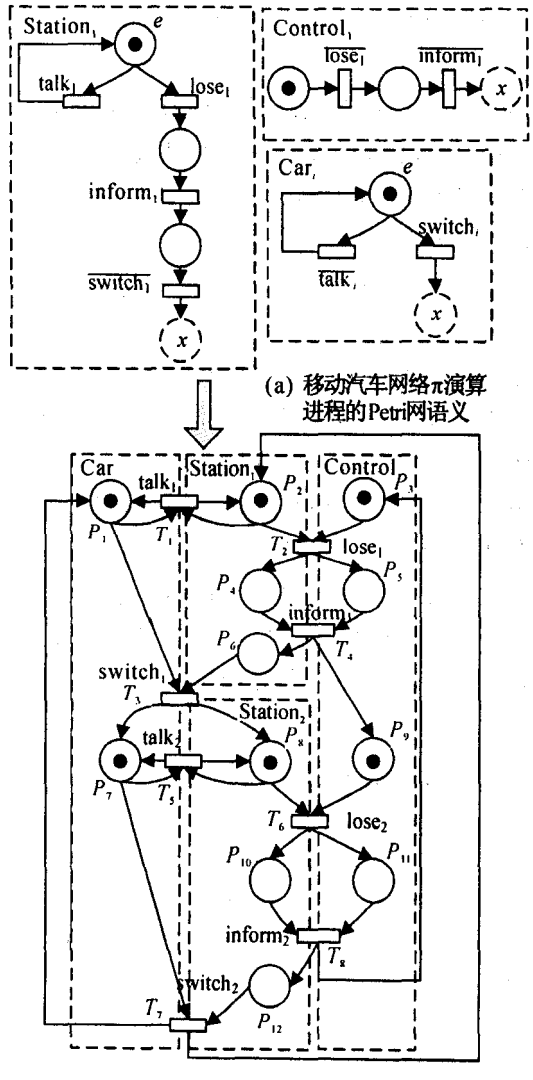
3　实例研究

为验证 π 演算 Petri 网语义的有效性，考虑一个利用 π 演算建模的典型例子：移动汽车网络[13]，包括两辆汽车、两个车站和一个中央控制器，汽车和车站进行通信. 假设汽车 1 从车站 1 行进到车站 2，然后返回；汽车 2 则反之. 在行进过程中，汽车必须把通信链接切换到离它最近的车站. 中央控制器通知某车站汽车已切换到其他的车站，同时为汽车提供新的通信链接. 把每一个车站、汽车、中央控制器都作为一个进程，设 $\alpha = \{\mathrm{talk}, \mathrm{switch}, \mathrm{gain}, \mathrm{lose}\}$，用 π 演算描述如下：

$$
\begin{aligned}
&\mathrm{Station}_i(\alpha_i) = \\
&{}_{\mathrm{def}}\mathrm{talk}_i.\,\mathrm{Station}_i\langle\alpha_i\rangle + \mathrm{lose}_i(t_i,s_i).\\
&\mathrm{inform}_i(t_j,s_j).\,\overline{\mathrm{switch}_i}\langle t_j,s_j\rangle.\,\mathrm{Station}_j(\alpha_j),\\
&\mathrm{Control}_i(\mathrm{lose}_i,\mathrm{inform}_i) = \\
&{}_{\mathrm{def}}\overline{\mathrm{lose}_i}\langle\mathrm{talk}_i,\mathrm{switch}_i\rangle.\\
&\overline{\mathrm{inform}_i}\langle\mathrm{talk}_j,\mathrm{switch}_j\rangle.\\
&\mathrm{Control}_j(\mathrm{lose}_j,\mathrm{inform}_j),\\
&\mathrm{Car}_i(\mathrm{talk}_i,\mathrm{switch}_i) = \\
&{}_{\mathrm{def}}\overline{\mathrm{talk}_i}.\,\mathrm{Car}_i\langle\mathrm{talk}_i,\mathrm{switch}_i\rangle + \\
&\mathrm{switch}_i(t_i,s_i).\,\mathrm{Car}_i\langle t_i,s_i\rangle.
\end{aligned}
\tag{1}
$$

根据上述的进程，系统可描述为

$$
\begin{aligned}
&\mathrm{System} = \\
&(\nu\alpha_i : i = 1,2)(\mathrm{Car}_1(\mathrm{talk}_1,\mathrm{switch}_1) \mid \\
&\mathrm{Station}_1(\alpha_1) \mid \mathrm{Control}_1(\mathrm{lose}_1,\mathrm{inform}_1) \mid \\
&\mathrm{Car}_2(\mathrm{talk}_2,\mathrm{switch}_2) \mid \\
&\mathrm{Station}_2(\alpha_2) \mid \mathrm{Control}_2(\mathrm{lose}_2,\mathrm{inform}_2)).
\end{aligned}
\tag{2}
$$



(b) 移动汽车网络系统 Petri网模型

图 5　移动汽车网络 π 演算模型的 Petri 网语义

根据π演算到Petri网语义的映射规则，首先将车站、汽车、中央控制器等进程映射到Petri网，如图5(a)所示. 其中：虚线库所是抽象库所，表示与其他进程的接口；$Station_1$ 和 $Station_2$ 与 $Control_1$ 和 $Control_2$ 的Petri网模型是类似的，这里只列出一个. 然后根据进程的对偶名字合并各Petri网模型中的相关变迁，成为一个通信变迁. 最后得到系统System的Petri网模型 N_{System}，如图5(b)所示. 其中：每个虚线框内的部分是各进程的Petri网语义；变迁 $talk_i$，$lose_i$，$switch_i$，$inform_i$ 是通信变迁(τ)，执行进程之间的通信. 从 N_{System} 可以看出，初始状态时，库所 P_1 和 P_7 各有一个Token，分别表示汽车1和汽车2. 汽车1和汽车2在通信网络中是并发执行的，Petri网模型能较好地体现系统的并发活动. 当汽车1在行进过程中超出车站1时，中央控制器通过车站通知汽车1转换到车站2的信道.

当π演算进程映射到Petri网后，可利用Petri网支持工具(如INA[14]等)对模型的特性进行分析和验证. 经分析知，N_{System} 是有界的，可达状态有21个，并且模型是活的，可以保证系统达到最终的稳定状态.

通过把移动汽车网络的π演算模型映射到Petri网，可较好地刻画系统的真并发语义，形象地描述系统的物理结构，并能根据Petri网的结构性质和工具，从模型网络结构上定性分析系统的性质.

4 结 论

本文将π演算分为基本单元、顺序、并发、选择和递归等几种基本结构，分别映射为Petri网，并复合构成复杂的系统，可自然地实现π演算进程的Petri网表示. 将π演算映射为Petri网语义的动机是因为Petri网比较直观、易懂和易用，可以描述系统的真并发语义，并可利用Petri网众多的支持工具对模型进行分析和验证. 最后，根据π演算的Petri网语义，将移动汽车网络的π演算模型映射为Petri网，并用INA对模型进行分析，验证了所提出方法的有效性.

参考文献(References)

[1] 蒋昌俊. Petri网的行为理论及其应用[M]. 北京：高等教育出版社，2003.
(Jiang Chang-jun. Behavior theory and applications of Petri nets [M]. Beijing: Higher Education Press, 2003.)

[2] Milner R, Parrow J, Walker D. A calculus of mobile processes[J]. J of Information and Computation, 1992, 100(1): 1-77.

[3] Murata T. Petri nets: Properties, analysis, and application [J]. Proc of the IEEE, 1989, 77(4): 541-580.

[4] Petri nets tools and software [EB/OL]. http://www.informatik.uni-hamburg.de/ TGI/PetriNets/tools/.

[5] Busi N, Gorrieri R. A Petri net semantics for π-calculus [J]. Lecture Notes in Computer Science, 1995, 962: 145-159.

[6] Engelfriet J. A multiset semantics for the π-calculus with replication [J]. Theoretical Computer Science, 1996, 153(1/2): 65-94.

[7] Devillers R, Klaudel H, Koutny M. Petri net semantics of the finite π-calculus [J]. Lecture Notes in Computer Science, 2004, 3235(2): 309-325.

[8] 曹木亮，吴智铭，杨根科. 一类新型的模块化高级Petri网—π网 [J]. 上海交通大学学报，2004，38(1)：52-58.
(Cao Mu-liang, Wu Zhi-ming, Yang Gen-ke. π-net: A new modular high level Petri nets[J]. J of Shanghai Jiaotong University, 2004, 38(1): 52-58.)

[9] Mazzeo A, Mazzocca N, Russo S, et al. Formal specification of concurrent systems: A structured approach [J]. The Computer Journal, 1998, 41(3): 145-162.

[10] Best E, Koutny M. Process algebra a Petri-net-oriented tutorial [J]. Lecture Notes in Computer Science, 2004, 3098: 180-209.

[11] 韩婷婷，陈韬略，颜锋，等. 同步和异步π演算的表达能力研究 [J]. 高技术通讯，2005，15(8)：18-22.
(Han Ting-ting, Chen Tao-lue, Yan Feng, et al. Study on expressive power of synchronous asynchronous π-calculus [J]. High Technology Letters, 2005, 15(8): 18-22.)

[12] Ribaudo M. Stachostic Petri net semantics for process algebras[C]. Proc of the 6th Int Workshop on Petri Nets and Performance Models. Los Alamitos: IEEE Press, 1995, 15(8): 148-157.

[13] Milner R. Communicating and mobile systems: The π-calculus [M]. Cambridge: Cambridge University Press, 1999.

[14] Roch S, Starke P H. INA: Integrated net analyzer [Z]. Version 2.2.