

多态 $\chi$ -演算

计算模型

计算机科学

②

多态 $\chi$ -演算<sup>\*)</sup>Polyadic  $\chi$ -Calculus

6-9

蒋翔宇 傅育熙

(上海交通大学计算机科学与工程系 上海 200030)

TP301.6

**Abstract** The paper proposes polyadic  $\chi$ -calculus. The language is stronger than  $\chi$ -calculus in communication. It can exchange several information once time. In the paper we define the syntax and semantics structure of polyadic  $\chi$ -calculus, and study the bisimulation relations of it.

**Keywords** Bisimulation, Polyadic  $\chi$ -calculus

## 1 引言

计算模型是计算机科学研究的重要问题之一。计算机科学家对顺序计算的研究已处于较成熟的阶段,并提出了相应的计算模型,其中尤以 $\lambda$ -演算最为经典。与顺序计算相比,我们对并行计算的认识处于较肤浅的水平。自 Milner 提出 CCS 以来,计算机科学家提出了一系列并行与并发计算模型。在这些模型中 Milner 等人提出的 $\pi$ -演算是较为成熟的。和 $\lambda$ -演算一样, $\pi$ -演算也是一封闭的数学模型,它在 CCS 的基础上通过增加传递通道名功能而使其表达能力大大强于 CCS。 $\pi$ -演算的最大特点是进程间的通讯的拓扑结构随进程演进而变化,而这一动态性却增加了研究 $\pi$ -演算代数语义的难度。本文第二作者傅育熙提出的 $\chi$ -演算旨在从概念上对 $\pi$ -演算进行简化。与 $\pi$ -演算相比, $\chi$ -演算更简单,代数性质也更好。本文所研究的多态 $\chi$ -演算是 $\chi$ -演算基础上的一个扩充,与 $\chi$ -演算所不同的是在一次通讯中进程间可交换多个名。多态性的加入使得多态 $\chi$ -演算比 $\chi$ -演算具有更强的控制能力。

2 多态 $\chi$ -演算的模型描述2.1 多态 $\chi$ -演算的语法结构

设 $\mathcal{N}$ 为一名集,其中的元素用小写字母表示, $\bar{x} \stackrel{\text{def}}{=} \{x | x \in \mathcal{N}\}$ 。 $\bar{x}$ 表示名 $x_1 \cdots x_n$ 的有限序列, $i \neq j$ 时 $x_i$ 和 $x_j$ 两者也可能相同。相应地 $(\bar{x})P$ 表示 $(x_1) \cdots (x_n)P$ 。 $|\bar{x}|$ 表示 $\bar{x}$ 的长度,当 $\bar{x}$ 的长度为0时, $(\bar{x})P$ 即 $P$ 。所有多维 $\chi$ -进程组成的集合 $C$ 由BNF定义的对象构成:

$$P ::= 0 \mid a[\bar{x}] \mid P \mid P + Q \mid P \mid Q \mid (x)P \mid !P$$

其意义解释如下:

1) 0 是没有任何行为的进程,不能与任何其它进程进行交互;

2)  $a[\bar{x}] \cdot P$ : 这里 $a$ 的形式为 $a$ 或 $\bar{a}$ , $a$ 和 $\bar{a}$ 分别代表一通道的两个端口。进程 $a[\bar{x}] \cdot P$ 先在通道 $a$ 的端口上发生一个通讯,交换 $n$ 个名,再启动 $P$ ;

3)  $P + Q$ : 这是进程的不确定形式。此进程将根据一定的诱因来激活 $P$ 或 $Q$ ,它的行为只是这个被激活进程的行为;

4)  $P \mid Q$ :  $P$ 和 $Q$ 既可并行地运行,也可相互进行通讯;

5)  $(x)P$ : 进程 $(x)P$ 中的 $x$ 是局部的,外界的其它进程无法通过 $x$ 与 $P$ 进行通讯;

6)  $!P$ : 表示 $P$ 的一组无限拷贝。

$ln(P)$ 和 $gn(P)$ 分别表示 $P$ 中所有的局部名和全局名的集合, $n(P)$ 为 $ln(P)$ 和 $gn(P)$ 的并。根据 $\alpha$ -协议,一个进程中的局部名可被另一新名替换而不改变这个进程的语法。在本文中,我们约定由 $\alpha$ -协议进行换名后得到的进程与原进程是同一进程。

2.2 多态 $\chi$ -演算的操作语义

在给出多态 $\chi$ -演算的操作语义之前,我们先定义一些符号。

我们称从 $\mathcal{N}$ 至 $\mathcal{N}$ 的函数 $\sigma: \mathcal{N} \rightarrow \mathcal{N}$ 为替换当其满足条件: $\{x | x \in \mathcal{N}, \sigma(x) \neq x\}$ 为有限。替换用符号 $\sigma$ 表示,并常记为 $[y_1/x_1, \dots, y_n/x_n]$ ,这里 $x_1 \neq y_1, \dots, x_n \neq y_n$ ,且 $[y_1/x_1, \dots, y_n/x_n]$ 表示如下的函数:

\*) 本文受国家自然科学基金(69873032)和国家863高科技项目(863-306-ZT06-02-2)资助。

$$\sigma(x) = \begin{cases} y_1, x = x_1 \\ \vdots \\ y_n, x = x_n \\ x, x \in \{x_1, \dots, x_n\} \end{cases}$$

$[y_1/x_1, \dots, y_n/x_n]$ 常略记为 $[\bar{y}/\bar{x}]$ ,恒等函数称为空替换,用 $[]$ 表示.我们之所以引入替换的定义,是因为为了给出多态 $\lambda$ -演算的标号迁移语义,为此必须引入一类新的操作—更新操作.例如下述迁移:

$$(x)\bar{m}[x, x, y]. P \mid (w)\bar{m}[u, v, w] Q \xrightarrow{[u/v, y/w]} P[u/v, y/w] \mid Q[u/v, y/w]$$

中的 $[u/v, y/w]$ 就是更新操作.更新操作最初由本文第二作者傅育熙在 $\lambda$ -演算中引入.在相关文献中已有较详细的阐述,本文就不加以展开了.

若 $\sigma$ 为 $[y_1/x_1, \dots, y_n/x_n]$ ,则用 $dom(\sigma)$ 及 $rng(\sigma)$ 分别表示 $\{x_1, \dots, x_n\}$ 和 $\{y_1, \dots, y_n\}$ .当 $dom(\sigma) \cap rng(\sigma) = \emptyset$ ,我们称 $\sigma$ 是正则的,研究证明多态 $\lambda$ -演算中涉及的替换均为正则替换,对于正则替换 $\sigma = [y_1/x_1, \dots, y_n/x_n]$ , $\sigma \uparrow z$ 表示如下定义的替换:

$$\sigma(x) = \begin{cases} y_i, & x = x_i \wedge x \neq z \wedge (1 \leq i \leq n) \\ x, & \text{其它} \end{cases}$$

显然 $\sigma \uparrow z$ 也为正则替换. $\sigma^* \{x_1, \dots, x_n\}$ 表示 $\{\dots(\sigma \uparrow z_1) \dots \uparrow z_n\}$ .例如 $[u/x, v/y, w/z] \uparrow y = [u/x, w/z]$ ,用 $\sigma^{-1}$ 表示函数的逆.替换操作作用后缀法表示,即 $P\sigma$ 是将 $P$ 按 $\sigma$ 换名后的进程.替换 $\sigma$ 和 $\sigma'$ 的复合记为 $\sigma\sigma'$ , $\sigma\sigma'$ 即函数 $\_ \xrightarrow{\sigma} \_ \xrightarrow{\sigma'} \_$ .显然一正则替换 $[y_1/x_1, \dots, y_n/x_n]$ 可表示成 $[y_1/x_1] \dots [y_n/x_n]$ .

**引理1** 若 $\sigma$ 和 $\sigma'$ 均为正则替换并且 $dom(\sigma) \cap rng(\sigma') = \emptyset$ ,则 $\sigma\sigma'$ 也为正则替换.

**证明:**显然有:

$$\begin{aligned} dom(\sigma\sigma') \cap rng(\sigma\sigma') &\subseteq (dom(\sigma) \cup dom(\sigma') \setminus rng(\sigma)) \\ &\cap (rng(\sigma) \cup rng(\sigma')) = (dom(\sigma) \cup dom(\sigma') \setminus rng(\sigma)) \\ &\cap rng(\sigma) \cup (dom(\sigma) \cup dom(\sigma') \setminus rng(\sigma) \cap \\ &rng(\sigma')) = dom(\sigma) \cap rng(\sigma') \cup dom(\sigma') \cap rng(\sigma') \\ &\setminus rng(\sigma) \cap rng(\sigma') = \emptyset. \end{aligned}$$

所以 $dom(\sigma\sigma') \cap rng(\sigma\sigma') = \emptyset$ .  $\square$

设 $\bar{x} = x_1, \dots, x_n$ 和 $\bar{y} = y_1, \dots, y_n$ 为两个长度为 $n$ 的名串,则 $x_1 = y_1, \dots, x_n = y_n$ 诱导出 $\mathcal{V}$ 上的一个等价关系,用 $\sigma_{\bar{x}=\bar{y}}$ 表示将该等价关系的每个等价类中的所有元素映射到该等价类中任意选定元素的函数.显然, $\sigma_{\bar{x}=\bar{y}}$ 是一个正则替换.

设 $x_1, \dots, x_n$ 和 $y_1, \dots, y_n$ 均是长度为 $n$ 的名串.称 $x_1, \dots, x_n$ 与 $y_1, \dots, y_n$ 为一致的当且仅当 $\forall i, j \in \{1, \dots, n\}, x_i = x_j \Leftrightarrow y_i = y_j$ .称 $y_1, \dots, y_n$ 可替换 $x_1, \dots, x_n$ 当且仅当 $\forall i, j \in \{1, \dots, n\}, x_i = x_j \Rightarrow y_i = y_j$ .显然,当 $y_1, \dots, y_n$ 可替换 $x_1, \dots, x_n$ 且 $\{x_1, \dots, x_n\} \cap \{y_1, \dots, y_n\} = \emptyset$ 时, $[y_1/x_1, \dots, y_n/x_n]$ 为正则替换.

以下的标号转移系统定义了多态 $\lambda$ -演算的操作语义,规则中的 $\mu$ 表示 $\{\tau\} \cup \{(\bar{y})a[\bar{x}] \mid \bar{y} \subseteq \bar{x} \in \mathcal{V}\}$ 中任一指定元素, $\delta$ 表示 $\{\tau\} \cup \{(\bar{y})a[\bar{x}] \mid \bar{y} \subseteq \bar{x} \in \mathcal{V}\} \cup \{\sigma \mid \sigma \text{ 为正则替换}\}$ 中任一指定元素.下文中,我们将 $\xrightarrow{\tau}$ 写成 $\xrightarrow{\tau}$ .所有对称的语义规则都被略去.

下面的规则定义了多态 $\lambda$ -演算的操作语义:

$$\text{顺序规则} \quad \frac{}{a[\bar{x}]. P \xrightarrow{\sigma[\bar{x}]} P} Sqn$$

$$\text{选择规则} \quad \frac{P \xrightarrow{\delta} P'}{P + Q \xrightarrow{\delta} P'} Sum$$

$$\text{并发规则} \quad \frac{P \xrightarrow{\mu} P' \quad Q \xrightarrow{\mu} Q'}{P \mid Q \xrightarrow{\mu} P' \mid Q'} Cmp_0$$

$$\frac{P \xrightarrow{\sigma_{\bar{x}=\bar{y}}} P' \quad Q \xrightarrow{\sigma_{\bar{x}=\bar{y}}} Q'}{P \mid Q \xrightarrow{\sigma_{\bar{x}=\bar{y}}} P' \mid Q'} Cmp_1$$

$$\text{通讯规则} \quad \left\{ \begin{array}{l} |\bar{x}| = |\bar{y}| \\ \{\bar{u}\} \cap \{\bar{v}\} = \emptyset \\ \{\bar{w}\} = rng(\sigma_{\bar{x}=\bar{y}}) \cap \{\bar{u}, \bar{v}\} \\ \frac{P \xrightarrow{(\bar{y})a[\bar{x}]} P' \quad Q \xrightarrow{(\bar{y})a[\bar{x}]} Q' \quad (\sigma_{\bar{x}=\bar{y}})^{-1}(\{\bar{w}\}) \subseteq \{\bar{u}, \bar{v}\}}{P \mid Q \xrightarrow{\sigma_{\bar{x}=\bar{y}} \uparrow \bar{u}, \bar{v}} (P'\sigma_{\bar{x}=\bar{y}} \mid Q'\sigma_{\bar{x}=\bar{y}})} Cmm \end{array} \right.$$

$$\text{局部规则} \quad \frac{P \xrightarrow{\delta} P' \quad x \in n(\delta)}{(x)P \xrightarrow{\delta} (x)P'} Loc_1$$

$$\frac{P \xrightarrow{\sigma} P' \quad x \in dom(\sigma)}{(x)P \xrightarrow{\sigma \uparrow x} P'} Loc_2$$

$$\frac{P \xrightarrow{(\bar{y})a[\bar{x}]} P' \quad z \in \{\bar{x}\} \setminus \{\bar{y}\} \quad a \in \{z, z\}}{(z)P \xrightarrow{(z)(\bar{y})a[\bar{x}]} P'} Loc_3$$

$$\text{复制规则} \quad \frac{P \xrightarrow{a} P'}{!P \xrightarrow{a} P' \mid !P} R$$

通讯规则比较复杂,所以举一个例子予以说明:

$$\frac{P \xrightarrow{(x)a[x, x, u]} P' \quad Q \xrightarrow{(v)\bar{u}[y, w, v]} Q'}{P \mid Q \xrightarrow{[y/w]} P[y/x, y/w, u/v] \mid Q[y/x, y/w, u/v]}$$

读者可自己加以验证,以便更好地理解通讯规则.

我们用 $\Rightarrow$ 表示 $\xrightarrow{\tau}$ 的自反传递闭包.以 $\xRightarrow{\mu}$ 代表 $\xrightarrow{\mu} \Rightarrow$ , $P \xRightarrow{\sigma} Q$ 表示存在正则替换 $\sigma_1, \dots, \sigma_n, n \geq 1$ ,满足以下条件:

$$1) \sigma = \sigma_1 \dots \sigma_n;$$

$$2) P \xrightarrow{\sigma_1} \dots \xrightarrow{\sigma_n} Q.$$

如果 $\mu \neq \tau (\delta \neq \tau)$ , $\xRightarrow{\mu} (\xRightarrow{\delta})$ 表示 $\xRightarrow{\mu} (\xRightarrow{\delta})$ ,否则表示 $\xRightarrow{\mu}$ .

在本节的剩余部分,我们将给出一些引理,并予以证明。

**引理2** 若  $P \xrightarrow{\delta} P'$  且  $\sigma$  为正则替换,则  $P\sigma \xrightarrow{\delta\sigma} P'\sigma$ 。

证明:对推导树的高度进行归纳。假设使用的最后一条规则是  $\text{Loc}_2$ 。根据归纳假设

$$P\sigma \xrightarrow{(\tilde{y}\sigma)\alpha\sigma[\tilde{x}\sigma]} P'\sigma$$
  
 $z\sigma \in \tilde{x}\sigma - \tilde{y}\sigma, \alpha\sigma \in \{z\sigma, z\sigma'\}$ 。用  $\text{Loc}_2$  规则,得:

$$(z\sigma)P\sigma \xrightarrow{(z\sigma \tilde{y}\sigma)\alpha\sigma[\tilde{x}\sigma]} P'\sigma$$

设所用的最后一条规则是  $\text{Cmm}$ 。由  $\alpha$ -协议及归纳前提:

$$\begin{aligned} P\sigma &\xrightarrow{(\tilde{y}\sigma)\alpha\sigma[\tilde{x}\sigma]} P'\sigma \\ Q\sigma &\xrightarrow{(\tilde{w}\sigma)\alpha\sigma[\tilde{y}\sigma]} Q'\sigma \end{aligned}$$

故由规则  $\text{Cmm}$  得:

$$\begin{aligned} (P|Q)\sigma &\equiv P\sigma|Q\sigma \xrightarrow{\sigma\tilde{x} \leftarrow \tilde{y}\sigma} (\tilde{w})(P'\sigma\sigma\tilde{x} \leftarrow \tilde{y}\sigma | Q'\sigma\sigma\tilde{x} \leftarrow \tilde{y}\sigma) \\ &\equiv (P'\sigma\sigma\tilde{x} \leftarrow \tilde{y} | Q'\sigma\sigma\tilde{x} \leftarrow \tilde{y})\sigma \end{aligned}$$

其它情况可类似证明。

**引理3** 设  $\alpha \in \text{gn}(P)$ 。如果  $(\tilde{x})(P|\alpha[\tilde{x}]) \xrightarrow{\tau} P'|\alpha[\tilde{y}]$ , 那么  $P \xrightarrow{[\tilde{y}/\tilde{x}]} P'$ 。

证明:显然  $(\tilde{x})(P|\alpha[\tilde{x}]) \rightarrow P'|\alpha[\tilde{y}]$  是通过应用规则  $\text{Loc}_1$  若干次后得到的, 因此  $P|\alpha[\tilde{x}] \xrightarrow{[\tilde{y}/\tilde{x}]} P'|\alpha[\tilde{y}]$ , 故  $P \xrightarrow{[\tilde{y}/\tilde{x}]} P'$ 。

**引理4** 设  $\alpha \in \text{gn}(P)$ 。如果  $(\tilde{x})(P|\alpha[\tilde{x}]) \Rightarrow P'|\alpha[\tilde{y}]$ , 那么  $P \Rightarrow P'$ 。

证明:证明思想与引理3的类似。

### 3 互模拟格

进程间的等价关系是进程代数研究的重点。对两个进程而言,当找不到第三个进程通过与它们进行通讯而加以区分的话,我们就认为这两个进程在观测上是等价的。互模拟等价是一种非常理想的观测等价。两个进程若是可以互相模拟对方的可观察行为,这两个进程就是互模拟的。本节我们将对多态  $\lambda$ -演算的互模拟等价进行系统的研究,并将引入互模拟格的概念。多态  $\lambda$ -演算是  $\lambda$ -演算的扩充,和在  $\lambda$ -演算一样,我们认为除了内部行为外的一些外部行为也是不可观测的。

在引入  $L$ -互模拟之前,需要先定义一些符号。我们用  $\alpha$  表示集合  $\{\alpha | [\tilde{x}] | \alpha \in \mathcal{N}, \tilde{x} \in \mathcal{N}\}$ ,  $\bar{\alpha}$  表示  $\{\bar{\alpha}[\tilde{x}] | \alpha \in \mathcal{N}, \tilde{x} \in \mathcal{N}\}$ ,  $\text{ro}$  表示  $\{(\tilde{y})\alpha[\tilde{x}] | \alpha \in \mathcal{N}, \emptyset \subset \{\tilde{y}\} \subseteq \{\tilde{x}\} \subseteq \mathcal{N}\}$ ,  $\bar{\text{ro}}$  表示  $\{(\tilde{y})\bar{\alpha}[\tilde{x}] | \alpha \in \mathcal{N}, \emptyset \subset \{\tilde{y}\} \subseteq \{\tilde{x}\} \subseteq \mathcal{N}\}$ ,  $\alpha$  表示  $\{\sigma | \sigma \text{ 为正则替换}\}$ ,  $\mathcal{L}$  表示  $\{\cup S | S \subseteq \{\alpha, \bar{\alpha}, \text{ro}, \bar{\text{ro}}, \alpha\} \wedge S \neq \emptyset\}$ 。

• 8 •

**定义1** 设  $\mathcal{R}$  是  $\mathcal{L}$  上的对称二元关系,  $L$  是  $\mathcal{L}$  中的一个元素。关系  $\mathcal{R}$  是一个  $L$ -互模拟, 当且仅当  $\mathcal{R}$  满足如下要求: 若  $P \mathcal{R} Q$  则对任意进程  $R$  和任意名序列  $\tilde{x}$  下述性质成立:

如果  $(\tilde{x})(P|R) \xrightarrow{\varphi} P'$ , 其中  $\varphi \in L \cup \{\tau\}$ , 那么存在一个  $Q'$  满足  $(\tilde{x})(Q|R) \xrightarrow{\hat{\varphi}} Q'$  且  $P' \mathcal{R} Q'$ 。

关系  $\approx_L$  是最大的  $L$ -互模拟。

上述定义实质上是31种  $L$ -互模拟的统一定义。由于其中一些互模拟相互等价, 这31种  $L$ -互模拟最终可归约为七种互模拟, 并且这七种互模拟在包含关系下形成一种良好的序结构。

**引理5** 若  $P \Rightarrow P_1 \approx_L Q$  且  $Q \Rightarrow Q_1 \approx_L P$  则  $P \approx_L Q$ 。

证明:由定义3.1可以很容易得证。□

设  $\varphi \in L$ ,  $(\varphi)$  表示这样一个进程:  $(i)(\varphi) \xrightarrow{\varphi} 0(ii)$  若  $(\varphi) \xrightarrow{\varphi} A$  则  $A \equiv 0$ 。

**引理6** 设  $\alpha \in \text{gn}(P|Q)$ , 则: (i)  $(\tilde{x})(P|\alpha[\tilde{x}]) \approx_L (\tilde{x})(Q|\alpha[\tilde{x}])$  蕴含  $P \approx_L Q$ ; (ii)  $P|\alpha[\tilde{x}] \approx_L Q|\alpha[\tilde{x}]$  蕴含  $P \approx_L Q$ 。

证明: (i) 设  $\varphi \in L$  且  $n(\varphi) \cap \text{gn}(P|Q) = \emptyset$ 。由  $(\tilde{x})(P|\alpha[\tilde{x}]) | (\alpha[\tilde{x}], (\varphi)) \Rightarrow (P|0)|0$ , 可知存在  $Q_1$  使得  $(\tilde{x})(Q|\alpha[\tilde{x}]) | (\alpha[\tilde{x}], (\varphi)) \Rightarrow (Q_1|0)|0 \approx_L (P|0)|0$  可推出  $(\tilde{x})(Q|\alpha[\tilde{x}]) \xrightarrow{\alpha[\tilde{x}]} Q_1|0 \approx_L P|0$ 。由此可得  $Q \Rightarrow Q_1$ 。同样我们可以得到  $P_1$  存在使得  $P \Rightarrow P_1 \approx_L Q$ 。由引理5得  $P \approx_L Q$ 。

(ii) 证明类似。□

**引理7** 若  $P \approx_L Q$ , 则对任意正则替换  $\sigma$  满足  $P\sigma \approx_L Q\sigma$ 。

证明: 设  $P \approx_L Q$ 。任一正则替换都可表示成形如  $[y/x]$  的替换的复合, 故只需证对  $x \in \text{gn}(P|Q)$  且  $y \neq x$ ,  $P[y/x] \approx_L Q[y/x]$  成立。令  $b$  为一新名。假设  $\varphi \in L$  并且  $n(\varphi) \cap \text{gn}(P|Q) = \emptyset$ 。由定义可知对  $(x)(P|(\bar{b}[y]|(\bar{b}[x], (\varphi)))) \Rightarrow P[y/x]|(0|0)$  必有:

$$(x)(Q|(\bar{b}[y]|(\bar{b}[x], (\varphi)))) \Rightarrow Q_1|(0|0) \quad (1)$$

如果  $(x)(Q|(\bar{b}[y]|(\bar{b}[x], (\varphi)))) \xrightarrow{\tau} (x')(Q_2|(\bar{b}[y]|(\bar{b}[x'], (\varphi))))$  那么由对称性和  $\alpha$ -变换可得:

$$(x)(Q|(\bar{b}[y]|(\bar{b}[x], (\varphi)))) \xrightarrow{\tau} (x)(Q_3|(\bar{b}[y]|(\bar{b}[x], (\varphi))))$$

从而  $Q \xrightarrow{\tau} Q_3$ , 也就是说(1)可以分解成以下步骤:

$$\begin{aligned} (x)(Q|(\bar{b}[y]|(\bar{b}[x], (\varphi)))) &\xrightarrow{\tau} (x)(Q' | (\bar{b}[y]|(\bar{b}[x], (\varphi)))) \\ &\xrightarrow{\tau} Q'[y/x]|(0|0) \\ &\Rightarrow Q_1|(0|0) \end{aligned}$$

其中  $Q \Rightarrow Q'$  并且  $Q'[y/x] \Rightarrow Q_1 \approx_1 P[y/x]$ . 由引理 2,  $Q[y/x] \Rightarrow Q'[y/x]$ . 同样  $P_1$  存在使得  $P[y/x] \Rightarrow P_1 \approx_1 Q[y/x]$ . 根据引理 5,  $P[y/x] \approx_1 Q[y/x]$ .  $\square$

**定理 1** 如果  $P \approx_L Q$  并且  $O \in \mathcal{C}$ , 那么: (i)  $a[\bar{x}]. P \approx_L a[\bar{x}]. Q$ ; (ii)  $P|O \approx_L Q|O$ ; (iii)  $(\bar{x})P \approx_L (\bar{x})Q$ .

**证明:** (ii) 和 (iii) 可由定义 1 直接证明, (i) 利用 (ii) 和 (iii), 可以很容易证明  $\{(\bar{y})(R|a[\bar{x}]. P), (\bar{y})(R|a[\bar{x}]. Q) | \bar{y} \text{ 是名集}, R \in \mathcal{C}, P \approx_L Q\} \cup \approx_L$  是一个  $L$ -互模拟.  $\square$

**定理 2**  $L$ -互模拟具有如下性质: (i)  $\approx_n \subseteq \approx_s$ ,  $\approx_n \subseteq \approx_x$ ; (ii) 如果  $L \cap (\bar{o} \cup \bar{r}o) = \emptyset$ , 那么  $\approx_L \subseteq \approx_{s \cup \bar{o}}$ ; (iii) 如果  $L \cap (\bar{o} \cup \bar{r}o) = \emptyset$ , 那么  $\approx_L \subseteq \approx_{s \cup \bar{o} \cup \bar{r}o}$ ; (iv) 如果  $L \cap (\bar{r}o) = \emptyset$ , 那么  $\approx_L \subseteq \approx_n$ ; (v) 如果  $L \cap (\bar{r}o) = \emptyset$ , 那么  $\approx_L \subseteq \approx_n$ ; (vi)  $\approx_L \subseteq \approx_s$ .

**证明:** (i) 要证明  $\approx_n \subseteq \approx_s$ , 只要证明如果  $P \approx_n Q$ , 并且  $P \xrightarrow{a[\bar{x}]} P'$  那么存在  $Q'$  使得  $Q \xrightarrow{a[\bar{x}]} Q'$  并且  $P' \approx_n Q'$ . 由  $P \approx_n Q$  我们可推得对  $(\bar{y})P \xrightarrow{(\bar{y})a[\bar{x}]} P'$ , 其中  $\bar{y} \subseteq \bar{x}$ ,  $a \in \bar{y}$ , 必存在一个  $Q'$  满足  $(\bar{y})Q \xrightarrow{(\bar{y})a[\bar{x}]} Q'$  并且  $P' \approx_n Q'$ . 显然  $(\bar{y})Q \xrightarrow{(\bar{y})a[\bar{x}]} Q'$  是通过应用规则  $Loc_2$  若干次后得到的. 因此  $Q \xrightarrow{a[\bar{x}]} Q'$ . 在上述过程中我们是假设  $y$  不能为空, 若允许  $y$  为空, 则结论显然成立. 我们通过一个例子来证明上述结论的反方向不成立. 很明显,  $(\bar{x})a[\bar{x}]. (\bar{b}[\bar{x}] | b[\bar{z}]) \not\approx_n a[\bar{x}]. (\bar{x})a[\bar{x}]. (\bar{b}[\bar{x}] | b[\bar{z}])$ , 而  $(\bar{x})a[\bar{x}]. (\bar{b}[\bar{x}] | b[\bar{z}]) \approx_n a[\bar{x}]. (\bar{x})a[\bar{x}]. (\bar{b}[\bar{x}] | b[\bar{z}])$ .

(ii) 可通过一个例子来证明.  $!(\bar{x})a[\bar{x}]. (\bar{b}[\bar{x}] | b[\bar{z}]) \not\approx_n a[\bar{x}]. !(\bar{x})a[\bar{x}]. (\bar{b}[\bar{x}] | b[\bar{z}])$ , 但是  $!(\bar{x})a[\bar{x}]. (\bar{b}[\bar{x}] | b[\bar{z}]) \approx_L a[\bar{x}]. !(\bar{x})a[\bar{x}]. (\bar{b}[\bar{x}] | b[\bar{z}])$ .

(iii) 结论可类似得证.

(iv)  $(z)\bar{a}[x, z] | !(\bar{y})(z)\bar{a}[y, z]. (\bar{b}[x] | b[y]) \not\approx_n !(\bar{y})(z)\bar{a}[y, z]. (\bar{b}[x] | b[y])$ , 而  $(z)\bar{a}[x, z] | !(\bar{y})(z)\bar{a}[y, z]. (\bar{b}[x] | b[y]) \approx_L !(\bar{y})(z)\bar{a}[y, z]. (\bar{b}[x] | b[y])$ .

(v) 结论可类似得证.

(vi) 要证明  $\approx_L \subseteq \approx_s$ , 只要证明对正则替换  $\sigma \equiv [y/x]$ , 如果  $P \approx_L Q$  并且  $P \xrightarrow{\sigma} P'$  那么存在  $Q'$  使得  $Q \xrightarrow{\sigma} Q'$  并且  $P' \approx_L Q'$ . 由  $P \xrightarrow{\sigma} P'$  可推出  $(\bar{x})(P | a[\bar{x}]) \xrightarrow{\sigma} P' | a[\bar{y}]$ ,  $a$  是一新名. 存在某个  $Q'$  使得由  $(\bar{x})(Q | a[\bar{x}]) \Rightarrow Q' | a[\bar{y}]$  可推出  $P' | a[\bar{y}] \approx_L Q' | a[\bar{y}]$ . 根据引理 6 可得  $P' \approx_L Q'$ . 同时由引理 4 得,  $(\bar{x})(Q | a[\bar{x}]) \Rightarrow Q' | a[\bar{y}]$ . 推出  $Q \xrightarrow{\sigma} Q'$ .  $\square$

由上述定理可得  $L$ -互模拟格结构图如图 1 所示.

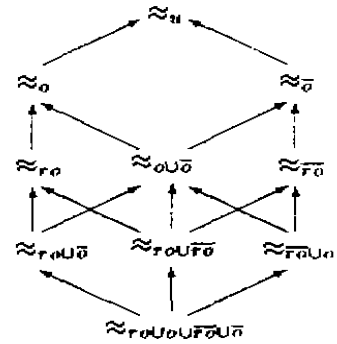


图 1 多态  $\chi$ -演算的  $L$ -互模拟格

## 参考文献

- 1 Fu Y. The  $\chi$ -Calculus. In: Proc. of the Interl. Conf. on Advances in Parallel and Distributed Computing. Shanghai, IEEE Computer Society Press, 1997. 74~81
- 2 Fu Y. A Proof Theoretical Approach to Communications ICALP'97, Bologna, Italy, Lecture Notes in Computer Science 1256, Springer, 1997. 325~335
- 3 Fu Y. Symmetric  $\pi$ -Calculus. Journal of Computer Science and Technology, 1998, 13: 202~208
- 4 Fu Y. Reaction Graphs. Journal of Computer Science and Technology, 1998, 13: 510~530
- 5 Fu Y. Bisimulation Lattice of Chi Processes. ASIAN'98, Manila, The Philippines, Lecture Notes in Computer Science 1538, Springer, 1998. 245~262
- 6 Fu Y. Variations on Mobile Processes. To appear in Theoretical Computer Science, 1999
- 7 Fu Y. Open Bisimulations of Chi Processes CONCUR'99, Eindhoven, The Netherlands, Lecture Notes in Computer Science, Springer, 1999. 24~27
- 8 Fu Y. An Open Problem of Mobile Processes. Submitted to Theoretical Computer Science
- 9 Fu Y. A Functional Presentation of Pi Calculus. Submitted to ASIAN'99
- 10 Fu Y. Bisimulation Congruence of Chi Calculus. Submitted
- 11 Fu Y. Algebraic Theory of Chi Calculus. Preprint
- 12 Milner R. et al. A Calculus of Mobile Processes. Information and Computation, 100: 1~40 (Part I), 41~77 (Part II), Academic Press
- 13 Milner R. et al. 19th ICALP, Lecture Notes in Computer Science 623, Springer, 1992. 685~695
- 14 Parrow J, Victor B. The Fusion Calculus: Expressive and Symmetry in Mobile Processes. LICS'98, 1998
- 15 Parrow J, Victor B. The Tau-Laws of Fusion. CONCUR'98, Lecture Notes in Computer Science, 1998
- 16 Sangiorgi D. A Theory of Bisimulation for  $\pi$ -Calculus. CONCUR'93, Lecture Notes in Computer Science 715, Springer, 1993
- 17 Victor B, Parrow J. Concurrent Constraints in the Fusion Calculus ICALP'98, Lecture Notes in Computer Science 1443, Springer, 1998. 455~469