元 Chi 进程互模拟等价的一致性*

林 敏, 傅育熙

(上海交通大学 计算机科学与工程系,上海 200030)

E-mail: fu-yx@cs.sjtu.edu.cn

http://www.sjtu.edu.cn

摘要: Chi-演算是将 π-演算中两类受限名统一后得到的. 多态 Chi-演算扩充了 Chi-演算的通信能力,使得一次通信可传递多个信息. 元 Chi-演算是在省略多态 Chi-演算的前缀操作子之后得到的子语言. 研究了元 Chi-演算的互模拟等价关系,证明了在某种意义下,元 Chi-进程之间只有一个互模拟等价关系.

关键词: 进程代数;移动进程;互模拟

中图法分类号: TP301 文献标识码: A

计算机学科中有许多研究并发计算的分支,进程代数是其中的一个重要分支. 进程代数理论早期的工作有 Milner 的 CCS 和 Hoare 的 CSP,这两个进程演算只能描述系统的静态通信结构. 在 CCS 的基础上,20 世纪 90 年代初由 Milner,Parrow 和 Walker 提出的 π-演算[1]通过增加传递通道名功能而使其表达能力大大强于 CCS. π-演算的最大特点在于进程之间的通信的拓扑结构随进程演进而变化. 这一动态性增加了研究 π-演算代数语义的难度. 具体体现在两个方面:(1) 进程之间有各类互模拟等价,有关互模拟等价性质的讨论比较复杂;(2) 进程之间同余关系的公理化问题比较复杂. 正是由于这种复杂性,π-演算中还有不少尚未解决的问题.

作为并发计算的基本模型,π-演算有两点不足. (1) π-演算继承了 CCS 的传统,它有两类语法实体,即进程和通道名. 前者是必须的,后者的必要性则值得怀疑. 难道进程代数一定要使用通道吗? (2) π-演算有两类受限名. 进程 $a(x).P$ 和 $(x)P$ 中的两个 x 属于不同种类的受限名,两者的语义大相径庭. 对于一个基本模型而言,两类受限名是否多了一点? 与 λ-演算相比,这两点不足尤为突出. 文献[2]作者提出了 π-演算的一个对称形式——对称 π-演算,它只有一类受限名. 另外,文献[3,4]又提出一种新的语言——χ-演算,它也只有一类受限名. 与 π-演算相比,χ-演算更简单,代数性质更好[5~7].

χ-演算与 π-演算在语法上的区别是,前者将输入和输出前缀操作统一了起来. 在 π-演算中输入和输出进程语法形式分别为 $a(x).P$ 和 $\bar{a}x.P$. 在 χ-演算中,分别变成了 $a[x].P$ 和 $\bar{a}[x].P$. 显然,$a[x].P$ 和 $\bar{a}[x].P$ 具有统一的语法形式,x 在两个进程中均为自由出现. χ-演算与 π-演算在语义上的区别可从以下通信例子中看出:

$$\bar{m}[x].P|m[x].Q\xrightarrow{\tau}P|Q, \tag{1}$$

$$(x)(R|(\bar{m}[y].P|m[x].Q))\xrightarrow{\tau}R\{y/x\}|(P\{y/x\}|Q\{y/x\}), \tag{2}$$

$$(x)\bar{m}[x].P|(y)m[y].Q\xrightarrow{\tau}(z)(P\{z/x\}|Q\{z/y\}). \tag{3}$$

* 收稿日期: 1999-08-03; 修改日期: 2000-07-04

基金项目: 国家自然科学基金资助项目(69873032);国家 863 高科技发展计划资助项目(863-306-ZT06-02-2)

作者简介: 林敏(1974—),女,上海人,硕士,主要研究领域为并行理论;傅育熙(1962—),男,四川人,教授,博士生导师,主要研究领域为理论计算机科学.

在 χ-演算中,通信的目的是将局部名换成全局名.在式(2)中,局部名 x 被换成了全局名 y,在式(3)中,两局部名被等同.式(1)中的通信比较特殊,允许该类通信简化多态 χ-演算的操作语义.为了给出 χ-演算的标号迁移语义,必须引入一类新的操作——更新操作.例如,下述迁移中的$\lceil y/x\rceil$就是更新操作.

$$\overline{m}[x].P|m[y].Q\xrightarrow{\lceil y/x\rceil}P[y/x]|Q[y/x].$$

瑞典的 Parrow 和 Victor 研究了多态 χ-演算——Fusion 演算[8,9].多态 χ-演算的前缀操作进程的语法形式为 $a[x_1,\ldots,x_n].P$ 和 $\overline{a}[x_1,\ldots,x_n].P$.在一次通信中进程之间可交换多个名,例如,

$$(x)(y)\overline{m}[x,y].P|m[u,v].Q\xrightarrow{\tau}P[u/x,v/y]|Q[u/x,v/y].$$

与 χ-演算一样,多态 χ-演算的通信是对称的,所以有

$$(x)\overline{m}[x,y].P|(v)m[u,v].Q\xrightarrow{\tau}P[u/x,y/v]|Q[u/x,y/v].$$

但多态 χ-演算有一些 χ-演算所没有的现象,必须推广更新操作.下面的例子很能说明问题:

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

对称地,也有

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

读者若希望了解更多的有关多态 χ-演算的情况,请参阅文献[8,9].

多态 χ-演算有一种有趣的子语言,该子语言通过取消多态 χ-演算的前缀操作子而得到,它形如$a[x_1,\ldots,x_n].P$ 和 $\overline{a}[x_1,\ldots,x_n].P$ 的进程分别被 $a[x_1,\ldots,x_n]$和 $\overline{a}[x_1,\ldots,x_n]$所取代.我们称形如 $a[x_1,\ldots,x_n]$和 $\overline{a}[x_1,\ldots,x_n]$的进程为原子进程,称所得的子语言为元 χ-演算.本文的主要目的是研究元 χ-进程间的互模拟等价关系,主要贡献是证明了在某种意义上元 χ-进程间只有一种互模拟等价关系.

本文第 1 节定义元 χ-演算的操作语义并证明一些引理.第 2 节引入 31 个 L-互模拟并研究其间的关系.第 3 节总结全文.

1 操作语义

设 $\mathcal{N}$ 为名集,其元素用小写字母来表示.$\overline{\mathcal{N}}=\{\overline{x}|x\in\mathcal{N}\}$为对偶名集.用 α 表示 $\mathcal{N}\cup\overline{\mathcal{N}}$ 中任意一个指定元素.设 $\mathcal{P}$ 是所有元 χ-进程的集合.元 χ-进程由 BNF 定义如下:

$$P::=\alpha[\tilde{x}]|P|P\ \ (x)P|P+P.$$

这里,$\alpha[\tilde{x}]$是原子进程,它能在 α 表示的通道上引发一个通信,该通信将名串 $\tilde{x}$ 与其他进程进行交换.$P|Q$ 为进程 P 和进程 Q 的并发操作,P 与 Q 可单独运行,也可在共享通道上发生通信.在进程 $(x)P$ 中,x 为局部名,其他进程无法使用该名.非受限名称为全局名.P 中所有全局名的集合用 $gn(P)$来表示.本文采用 α-协定,即进程中的局部名可用新名替换而不改变这个进程的语法.进程 $P+Q$为选择进程,其行为或为 P,或为 Q.本文省去了复制运算符$!P$.如果加上该运算符,本文的结论仍然成立.下文中 0 代表$(x)x[x]$,此进程没有任何计算行为.

一个有限名串 $x_1,\ldots,x_n$ 常略记为 $\tilde{x}$.相应地,$(\tilde{x})P$ 表示$(x_1)\ldots(x_n)P$.用$|\tilde{x}|$表示 $\tilde{x}$ 的长度.当 $\tilde{x}$ 的长度为 0 时,$(\tilde{x})P$ 即 P.

若从 $\mathcal{N}$ 至 $\mathcal{N}$ 的函数 $\sigma:\mathcal{N}\to\mathcal{N}$ 满足条件:$\{x|\sigma(x)\neq x\}$为有限,则称其为替换.替换常用 σ 表示,并常记为$[y_1/x_1,\ldots,y_n/x_n]$,这里 $x_1\neq y_1,\ldots,x_n\neq y_n$ 且$[y_1/x_1,\ldots,y_n/x_n]$表示如下的

函数:

$$\sigma(x)=\begin{cases} y_1, & x=x_1 \\ \vdots & \\ y_n, & x=x_n \\ x, & x\notin\{x_1,\ldots,x_n\} \end{cases}.$$

$[y_1/x_1,\ldots,y_n/x_n]$常略记为$[\tilde{y}/\tilde{x}]$.恒等函数称为空替换,用$[]$表示.若σ为$[y_1/x_1,\ldots,y_n/x_n]$,则$dom(\sigma)$及$rng(\sigma)$分别定义为$\{x_1,\ldots,x_n\}$和$\{y_1,\ldots,y_n\}$.若$dom(\sigma)\cap rng(\sigma)=\varnothing$,则称$\sigma$为正则的.对于正则替换$\sigma\equiv[y_1/x_1,\ldots,y_n/x_n]$,$\sigma\uparrow z$定义如下:

$$\sigma\uparrow z(x)=\begin{cases} y_i, & x=x_i\wedge x\neq z\wedge(1\leqslant i\leqslant n) \\ x, & \text{otherwise} \end{cases}.$$

显然,$\sigma\uparrow z$也为正则替换.$\sigma\uparrow\{z_1,\ldots,z_n\}$表示$(\ldots(\sigma\uparrow z_1)\ldots)\uparrow z_n$.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$\sigma\sigma'$,$\sigma\sigma'$是类型为$\mathcal{N}\xrightarrow{\sigma}\mathcal{N}\xrightarrow{\sigma'}\mathcal{N}$的函数.显然,正则替换$[y_1/x_1,\ldots,y_n/x_n]$可表示成$[y_1/x_1]\ldots[y_n/x_n]$.

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

证明:显然有

$$\begin{aligned} dom(\sigma\sigma')\cap rng(\sigma\sigma') &\subseteq (dom(\sigma)\cup dom(\sigma')\backslash rng(\sigma))\cap(rng(\sigma)\cup rng(\sigma')) \\ &=(dom(\sigma)\cup dom(\sigma')\backslash rng(\sigma))\cap rng(\sigma)\cup(dom(\sigma)\cup \\ &\quad dom(\sigma')\backslash rng(\sigma)\cap rng(\sigma') \\ &=dom(\sigma)\cap rng(\sigma')\cup dom(\sigma')\cap rng(\sigma')\backslash rng(\sigma)\cap rng(\sigma') \\ &=\varnothing. \end{aligned}$$

所以,$dom(\sigma\sigma')\cap rng(\sigma\sigma')=\varnothing$. □

设$\tilde{x}=x_1,\ldots,x_n$和$\tilde{y}=y_1,\ldots,y_n$为两个长度为n的名串,则$x_1=y_1,\ldots,x_n=y_n$诱导出$\mathcal{N}$上的一个等价关系,用$\sigma_{x=y}$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,$\sigma_{x=y}$是一个正则替换.

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

以下的标号转移系统定义了χ-演算的操作语义.规则中的μ表示$\{\tau\}\cup\{(\tilde{y})\alpha[\tilde{x}]\mid\tilde{y}\subseteq\tilde{x}\in\mathcal{N}\}$中任一指定元素,$\delta$表示$\{\tau\}\cup\{(\tilde{y})\alpha[\tilde{x}]\mid\tilde{y}\subseteq\tilde{x}\in\mathcal{N}\}\cup\{\sigma\mid\sigma$为正则替换$\}$中任意一个指定元素.下文中,我们将$\xrightarrow{\{\}}$写成$\xrightarrow{\tau}$.所有对称的语义规则都被略去.

$$\frac{}{\alpha[\tilde{x}].P\xrightarrow{\alpha[\tilde{x}]}P}Sqn,\qquad \frac{P\xrightarrow{\delta}P'}{P+Q\xrightarrow{\delta}P'}Sum,$$

$$\frac{P\xrightarrow{\mu}P'}{P|Q\xrightarrow{\mu}P'|Q}Cmp_0,\qquad \frac{P\xrightarrow{\alpha}P'}{P|Q\xrightarrow{\alpha\sigma}P'|Q\sigma}Cmp_1,$$

$$\frac{\begin{cases}|\tilde{x}|=|\tilde{y}|\\ \{\tilde{u}\}\cap\{\tilde{v}\}=\varnothing\\ \{\tilde{w}\}=rng(\sigma_{\tilde{x}=\tilde{y}})\cap\{\tilde{u},\tilde{v}\}\\ P\xrightarrow{(\tilde{u})a[\tilde{x}]}P' \quad Q\xrightarrow{(\tilde{v})\bar{a}[\tilde{y}]}Q' \quad (\sigma_{\tilde{x}=\tilde{y}})^{-1}(\{\tilde{w}\})\subseteq\{\tilde{u},\tilde{v}\}\end{cases}}{P|Q\xrightarrow{\sigma_{\tilde{x}=\tilde{y}}\uparrow\{\tilde{u},\tilde{v}\}}(\tilde{w})(P'\sigma_{\tilde{x}=\tilde{y}}|Q'\sigma_{\tilde{x}=\tilde{y}})}Cmm,$$

$$\frac{P\xrightarrow{\delta}P' \quad z\notin n(\delta)}{(z)P\xrightarrow{\mu}(z)P'}Loc_0, \qquad \frac{P\xrightarrow{\sigma}P' \quad x\in dom(\sigma)}{(x)P\xrightarrow{\sigma\uparrow x}P'}Loc_1,$$

$$\frac{P\xrightarrow{(\tilde{y})a[\tilde{x}]}P' \quad z\in\{\tilde{x}\}\backslash\{\tilde{y}\} \quad a\notin\{z,\bar{z}\}}{(z)P\xrightarrow{(z)(\tilde{y})a[\tilde{x}]}P'}Loc_2.$$

下面举例说明如何使用 Cmm 规则：

$$\frac{(x)a[x,y,z].P\xrightarrow{(x)a[x,y,z]}P \qquad (x')(y')\bar{a}[x',y',z'].Q\xrightarrow{(x')(y')\bar{a}[x',y',z']}Q}{(x)a[x,y,z].P|(x')(y')\bar{a}[x',y',z'].Q\xrightarrow{[z'/z]}(x)(P[x/x',y/y',z'/z]|Q[x/x',y/y',z'/z])}.$$

这里，$\tilde{u}=x,y,z,\tilde{v}=x',y',z',\tilde{w}=x,\sigma_{\tilde{u}=\tilde{v}}=[x/x',y/y',z'/z]$. 读者可以验证，这些数据满足 Cmm 的边界条件.

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

(1) $\sigma=\sigma_1\ldots\sigma_n$;

(2) $P\Rightarrow\xrightarrow{\sigma_1}\Rightarrow\ldots\Rightarrow\xrightarrow{\sigma_n}\Rightarrow Q$.

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

下面，我们陈述一些引理，它们的证明是对推导规则或语法结构的归纳.

引理 2. 如果 $P\xrightarrow{\delta}P'$ 且 σ 为正则替换，那么 $P\sigma\xrightarrow{\delta\sigma}P'\sigma$.

证明：设所用的最后一条规则是 Loc_2. 根据 α-协议，可假定 $(dom(\sigma)\cup rng(\sigma))\cap\{\tilde{y},z\}=\varnothing$. 由归纳前提，$P\sigma\xrightarrow{(\tilde{y})a\sigma[\tilde{x}\sigma]}P'\sigma$. 由 Loc_2，$(z)P\sigma\xrightarrow{(z)(\tilde{y})a\sigma[\tilde{x}\sigma]}P'\sigma$. 设所用的最后一条规则是 Cmm. 由 α-协议及归纳前提，

$$P\sigma\xrightarrow{(\tilde{v})a\sigma[\tilde{x}\sigma]}P'\sigma,$$

$$Q\sigma\xrightarrow{(\tilde{w})\bar{a}\sigma[\tilde{y}\sigma]}Q'\sigma,$$

故由规则 Cmm 得

$$(P|Q)\sigma\equiv P\sigma|Q\sigma\xrightarrow{\sigma_{\tilde{x}\sigma=\tilde{y}\sigma}}(\tilde{u})(P'\sigma\sigma_{\tilde{x}\sigma=\tilde{y}\sigma}|Q'\sigma\sigma_{\tilde{x}\sigma=\tilde{y}\sigma}\equiv(P'\sigma_{\tilde{x}=\tilde{y}}|Q'\sigma_{\tilde{x}=\tilde{y}})\sigma.$$

其他情况可类似证明. □

引理 3. 设 $a\notin gn(P)$. 如果 $(\tilde{x})(P|a[\tilde{x}])\rightarrow P'|a[\tilde{y}]$，那么 $P\xrightarrow{[\tilde{y}/\tilde{x}]}P'$.

证明：显然，$(\tilde{x})(P|a[\tilde{x}])\rightarrow P'|a[\tilde{y}]$是通过应用规则 Loc_1 若干次后得到的. 因此 $P|a[\tilde{x}]\xrightarrow{[\tilde{y}/\tilde{x}]}P'|a[\tilde{y}]$，故 $P\xrightarrow{[\tilde{y}/\tilde{x}]}P'$. □

引理 4. 设 $a\notin gn(P)$. 如果 $(\tilde{x})(P|a[\tilde{x}])\Rightarrow P'|a[\tilde{y}]$，那么 $P\overset{[\tilde{y}/\tilde{x}]}{\Longrightarrow}P'$.

证明：证明思想与引理 3 的类似. □

引理 5. 以下性质成立：若 $P\xrightarrow{(\tilde{y})a[\tilde{x}]}\xrightarrow{\sigma}P'$ 且 $\sigma^{-1}(\{\tilde{y}\}\cap rng(\sigma))\subseteq\{\tilde{y}\}$，则 $P\xrightarrow{\sigma\uparrow(\tilde{y})}\xrightarrow{(\tilde{y}')a[\tilde{x}\sigma]}P'$，

其中$\{\tilde{y}'\}=\{\tilde{y}\}\backslash dom(\sigma)$.

证明:根据元 χ 进程的 BNF 定义,P 有以下几种情况:

· P 形如 $a[z]$,不可能.

· P 形如 P_1+P_2,则或者 $P_1\xrightarrow{(\tilde{y})a[\tilde{x}]}{}^{\sigma}\rightarrow P'$,或者 $P_2\xrightarrow{(\tilde{y})a[\tilde{x}]}{}^{\sigma}\rightarrow P'$. 无论哪种情况,用归纳前提即可得到.

· P 形如 $P_1|P_2$. 若两个动作由 P_1 或 P_2 单独完成,则由归纳前提可得证. 否则,$\{\tilde{y}\}\cap(dom(\sigma)\cup rng(\sigma))=\varnothing$,此时命题显然成立.

· P 形如$(v)P_1$. 有几种情况:

(1) $P\xrightarrow{(\tilde{y})a[\tilde{x}]}{}^{\sigma}\rightarrow P'$ 是由 $P_1\xrightarrow{(\tilde{y})a[\tilde{x}]}{}^{\sigma'}\rightarrow P_1'$ 引起的,其中 $v\in\{\tilde{x}\}$ 且 $\sigma=\sigma'\upharpoonright v$. 运用归纳前提.

(2) $P\xrightarrow{(\tilde{y})a[\tilde{x}]}{}^{\sigma}\rightarrow P'$ 是由 $P_1\xrightarrow{(\tilde{y}')a[\tilde{x}]}{}^{\sigma}\rightarrow P'$ 引起的,其中 $\{\tilde{y}\}=\{\tilde{y}'\}\cup\{v\}$. 由归纳前提得 $P_1\xrightarrow{\sigma\upharpoonright\{\tilde{y}'\}}\xrightarrow{(\tilde{y}'')a[\tilde{x}\sigma]}P'$,其中 $\{\tilde{y}''\}=\{\tilde{y}'\}\backslash dom(\sigma)=\{\tilde{y}\}\backslash dom(\sigma)$. 故 $(v)P_1\xrightarrow{\sigma\upharpoonright\{\tilde{y}',v\}}\xrightarrow{(\tilde{y}'')a[\tilde{x}\sigma]}P'$,即 $(v)P_1\xrightarrow{\sigma\upharpoonright\{\tilde{y}\}}\xrightarrow{(\tilde{y}'')a[\tilde{x}\sigma]}P'$.

命题得证. □

推论 1. 若 $P\xrightarrow{(\tilde{y})a[\tilde{x}]}{}^{\tau}\rightarrow P'$,则 $P\xrightarrow{\tau}\xrightarrow{(\tilde{y})a[\tilde{x}]}P'$.

证明:这是引理 5 的特例. □

2 互模拟等价

进程代数主要研究进程之间的等价关系及其公理化问题. 进程间的等价一般指观测等价,其中最精细的一类是互模拟等价. 相对于 CCS 进程间互模拟等价而言,π-演算中的互模拟等价要复杂得多,至少有早、迟、开[10]三大类互模拟. 所幸的是,在 χ-演算中没有早语义和迟语义之分,并且任何一个已知的互模拟都有相应的开模拟刻画. 文献[3]中引入了局部互模拟这一概念,给出了 χ-进程间的第一个弱互模拟等价关系. 文献[5]中推广了局部互模拟概念,引入了一大类互模拟——L-互模拟,并证明 χ-进程间只有 4 个不同的 L-互模拟. 这 4 个 L-互模拟在集合包含关系下构成一个格——互模拟格. L-互模拟覆盖面很广,例如,格的最小元对应开互模拟,最大元对应 barbed 互模拟[11].

L-互模拟的基本思想是,不同的观测者所能看到的进程行为是不同的,他们所能区别的进程对也是不一样的. L-互模拟试图将所有不同的观测行为进行分类,定义出相应的进程间的等价关系. 本节研究元 χ-进程之间的 L-互模拟,主要结论是所有 L-互模拟都相互等价.

在对 L-互模拟正式定义之前,有一些符号需要说明. o 表示集合$\{a[\tilde{x}]\mid a\in\mathcal{N},\tilde{x}\in\mathcal{N}^*\}$,$\bar{o}$ 表示$\{\bar{a}[\tilde{x}]\mid a\in\mathcal{N},\tilde{x}\in\mathcal{N}^*\}$,$ro$ 表示$\{(\tilde{y})a[\tilde{x}]\mid a\in\mathcal{N},\varnothing\subset\{\tilde{y}\}\subseteq\{\tilde{x}\}\subseteq\mathcal{N}\}$,$\overline{ro}$表示$\{(\tilde{y})\bar{a}[\tilde{x}]\mid a\in\mathcal{N},\varnothing\subset\{\tilde{y}\}\subseteq\{\tilde{x}\}\subseteq\mathcal{N}\}$,$u$ 表示$\{\sigma\mid\sigma$ 为正则替换$\}$. $\mathcal{L}$ 为集合$\{\bigcup S\mid S\subseteq\{o,\bar{o},ro,\overline{ro},u\}\wedge S\neq\varnothing\}$.

定义 1. 设 $\mathcal{R}$ 为 $\mathcal{P}$ 上的一个对称二元关系,L 为 $\mathcal{L}$ 中的一个元素. 对任意进程 P 和 Q,如果 $P\mathcal{R}Q$,那么对任意进程 R 和名序列 $\tilde{x}$,都满足以下性质:

如果$(\tilde{x})(P|R)\xrightarrow{\varphi}P'$,$\varphi\in L\cup\{\tau\}$,则存在某一 Q',$(\tilde{x})(Q|R)\xRightarrow{\hat{\varphi}}Q'$ 且 $P'\mathcal{R}Q'$,那么关系 $\mathcal{R}$ 为一个L-互模拟. L-互模拟等价$\approx_L$ 是最大的 L-互模拟关系. 这是对 31 个 L-互模拟的统一的定义. $\approx_L$ 是当一个观察者能且只能观察 L 中的动作时他/她所无法区别的进程对.

在本节余下部分,我们设 L 为 $\mathscr{L}$ 中一确定的元素.首先建立一些引理.下面一个引理可由定义直接得到.

引理 6. 如果 $P\Rightarrow P_1\approx_L Q$ 且 $Q\Rightarrow Q_1\approx_L P$,那么 $P\approx_L Q$.

设 $\varphi\in L$,我们用 $\langle\varphi\rangle$ 来表示满足如下条件的任意一个进程:(i) $\langle\varphi\rangle\xrightarrow{\varphi}0$;(ii) 如果 $\langle\varphi\rangle\xrightarrow{\varphi}A$,那么 $A\equiv 0$.

引理 7. 设 $a\notin gn(P|Q)$.下述性质成立:

(i) 若 $(\tilde{x})(P|a[\tilde{x}])\approx_L(\tilde{x})(Q|a[\tilde{x}])$,则 $P\approx_L Q$;

(ii) 若 $P|a[\tilde{x}]\approx_L Q|a[\tilde{x}]$,则 $P\approx_L Q$.

证明:(i) 设 $\varphi\in L$ 且 $n(\varphi)\cap gn(P|Q)=\varnothing$.因为 $(\tilde{x})(P|a[\tilde{x}])|(\overline{a}[\tilde{x}]+\langle\varphi\rangle)\Rightarrow(P|0)|0$,存在 Q_1 使

$$(\tilde{x})(Q|a[\tilde{x}])|(\overline{a}[\tilde{x}]-\langle\varphi\rangle)\Rightarrow(Q_1|0)|0\approx_L(P|0)|0.$$

这意味着 $(\tilde{x})(Q|a[\tilde{x}])\xrightarrow{(\tilde{x})a[\tilde{x}]}Q_1|0\approx_L P|0$,即 $Q\Rightarrow Q_1$.类似地,存在 P_1 使 $P\Rightarrow P_1\approx_L Q$.由引理 6 可知,$P\approx_L Q$.

(ii) 可被类似地证明. □

引理 8. 如果 $P\approx_L Q$,那么对任一正则替换 σ,有 $P\sigma\approx_L Q\sigma$.

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,故只需证明对 $x\in gn(P|Q)$ 且 $y\neq x$,$P[y/x]\approx_L Q[y/x]$ 成立.设 b 为一新名,$\varphi\in L$ 且 $n(\varphi)\cap gn(P|Q)=\varnothing$.根据定义 $(x)(P|(\overline{b}[y]|(b[x]+\langle\varphi\rangle)))\Rightarrow P[y/x]|(0|0)$,必由

$$(x)(Q|(\overline{b}[y]|(b[x]+\langle\varphi\rangle)))\Rightarrow Q_1|(0\ 0) \tag{4}$$

来匹配.如果 $(x)(Q|(\overline{b}[y]|(b[x]+\langle\varphi\rangle)))\xrightarrow{\tau}(x')(Q_2|(\overline{b}[y]|(b[x']+\langle\varphi\rangle)))$,那么由对称性和 α-转换可得此推导与下面的推导相同:

$$(x)(Q|(\overline{b}[y]|(b[x]+\langle\varphi\rangle)))\xrightarrow{\tau}(x)(Q_3|(\overline{b}[y]|(b[x]+\langle\varphi\rangle))),$$

且 $Q\xrightarrow{\iota}Q_3$.式(4)可被分解为以下动作:

$$(x)(Q|(\overline{b}[y]|(b[x]+\langle\varphi\rangle)))\Rightarrow(x)(Q'|(\overline{b}[y]\ (b[x]+\langle\varphi\rangle)))\xrightarrow{\tau}Q'[y/x]|(0|0)$$
$$\Rightarrow Q_1|(0|0).$$

其中 $Q\Rightarrow Q'$,$Q'[y/x]\Rightarrow Q_1\approx_L P[y/x]$.由引理 2 可知,$Q[y/x]\Rightarrow Q'[y/x]$.类似地,存在 P_1 使 $P[y/x]\Rightarrow P_1\approx_L Q[y/x]$.由引理 6 得到 $P[y/x]\approx_L Q[y/x]$. □

定理 1. 如果 $P\approx_L Q$ 且 $O\in\mathscr{P}$,那么(i) $P|O\approx_L Q|O$;(ii) $(x)P\approx_L(x)Q$.

证明:由定义 1 很容易得证. □

引理 9. $\approx_L\subseteq\approx_u$.

证明:要证明 $\approx_L\subseteq\approx_u$,只需证明:如果 $P\approx_L Q$ 且 $P\xrightarrow{\alpha}P'$,那么存在 Q' 使 $Q\Rightarrow Q'$ 且 $P'\approx_L Q'$.设 $\sigma=[y_1/x_1,\ldots,y_n/x_n]$.$P\xrightarrow{\alpha}P'$ 意味着 $(x_1)\ldots(x_n)(P|a[x_1,\ldots,x_n])\xrightarrow{\tau}P'\ a[y_1,\ldots,y_n]$,其中 a 为一个新名.因此存在 Q' 使得 $(x_1)\ldots(x_n)(Q|a[x_1,\ldots,x_n])\Rightarrow Q'|a[y_1,\ldots,y_n]$ 且 $P'|a[y_1,\ldots,y_n]\approx_L Q'|a[y_1,\ldots,y_n]$.由引理 7 可知,$P'\approx_L Q'$.由引理 4 可知,$(x_1)\ldots(x_n)(Q_1|a[x_1,\ldots,x_n])\xrightarrow{\tau}Q_2|a[y_1,\ldots,y_n]$ 意味着 $Q_1|a[x_1,\ldots,x_n]\xrightarrow{\alpha}Q_2|a[y_1,\ldots,y_n]$.因而 $Q\xrightarrow{\alpha}Q'$. □

引理 10. 如果 $P\approx_L Q$ 和 $P\xrightarrow{(\tilde{y})\alpha[\tilde{x}]}P'$,那么存在 Q' 使 $Q\xrightarrow{(\tilde{y})\alpha[\tilde{x}]}Q'\approx_L P'$.

证明:设 $P\approx_L Q$ 且 $P\xrightarrow{(\tilde{y})a[\tilde{x}]}P'$. 不失一般性,假定 $P\xrightarrow{(\tilde{y})a[\tilde{x}]}P'$ 为 $P\xrightarrow{(x_1,\ldots,x_m)a[x_1,\ldots,x_n]}P'$,其中 $0\leqslant m\leqslant n$ 且 $1\leqslant n$. 设 $\varphi\in L$ 且 $n(\varphi)\cap gn(P|Q)=\varnothing$,再设 $z_1,\ldots,z_n$ 为与 $x_1,\ldots,x_n$ 一致的新名. 显然有 $P|(\bar{a}[\tilde{z}]+\langle\varphi\rangle)\xrightarrow{[x_{m+1}/z_{m+1},\ldots,x_n/z_n]}P'[z_1/x_1,\ldots,z_n/x_n]|0$. 由引理 9 可知,必然存在某个 Q' 使得

$$Q|(\bar{a}[\tilde{z}]+\langle\varphi\rangle)\xrightarrow{[x_{m+1}/z_{m+1},\ldots,x_n/z_n]}Q'|0\approx_L P'[z_1/x_1,\ldots,z_n/x_n]|0.$$

由此推出 $Q\xrightarrow{(\tilde{z})a[\tilde{y}]}\overset{\sigma}{\Rightarrow}Q'[x_1/z_1,\ldots,x_n/z_n]\approx_L P'$ 使得 $\{\tilde{y}\}\subseteq\{\tilde{z}\}$. 因而由引理 5 可得 $Q\xrightarrow{(x_1,\ldots,x_m)a[x_1,\ldots,x_n]}Q'[x_1/z_1,\ldots,x_n/z_n]$. □

推论 2. 以下关于 L-互模拟的性质成立:(i) $\approx_L\subseteq\approx_{ro}=\approx_{\tilde{o}}$;(ii) $\approx_L\subseteq\approx_o=\approx_{\tilde{o}}$.

证明:由引理 10 可以直接得到. □

定理 2. 所有 31 个 L-互模拟都相等.

证明:由引理 9 和推论 2 可以直接得到. □

由此,我们得出元 χ-进程之间在本质上只有一个 L-互模拟.

3 结 语

Parrow 和 Victor 在文献[8]中讨论了多态 χ-进程间的超等价(hyperequivalence)和 barbed 等价关系,其所谓的超等价实际上就是开(open)等价. 同时他们还"证明了"超等价和 barbed 等价是相同的. 文献[6]中给出了一个反例,说明在 χ-演算中开等价与 barbed 等价不相同. 该反例同样适用于多态 χ-演算. 值得指出的是,该反例必须使用前缀操作子. 一个很自然的问题是,是否有不使用前缀操作子的反例? 本项研究的原始动机就是要回答这个问题.

本文证明了元 χ-进程间的所有 31 个 L-互模拟均相同. 易证,该 L-互模拟与元 χ 进程间的开互模拟和 barbed 互模拟也相同. 这些结论在一定程度上表明元 χ-演算只有一类互模拟等价关系. 我们的结果也回答了上述问题,即不存在不使用前缀操作子的反例.

关于多态 χ-演算的另外一个重要研究方向是同余关系的公理化. Parrow 和 Victor 在文献[8]中讨论了超同余关系的公理化问题,但其结论值得商榷. 问题出在 Hennessy 引理上,其叙述如下:

$P\approx Q$ 当且仅当 $\tau.P=Q$ 或 $P=Q$ 或 $P=\tau.Q$.

该引理在 CCS 中成立,但在 π-演算和 χ-演算中不成立,其问题在于假定了 Hennessy 引理在多态 χ-演算中成立. 由于 Hennessy 引理不成立,在讨论 π-演算和 χ-演算弱同余关系的公理化时,Milner 著名的 3 个 τ-等式不再是充分的. 文献[7]给出了第 4 个 τ-等式,并证明这 4 个 τ-等式足以将 π-进程间的强开同余的完全公理系统扩充为弱开同余的完全公理系统. 利用此方法,讨论了 χ-进程之间 L-互模拟同余的完全公理系统[7]. 该方法同样可用于多态 χ-演算的公理化问题的研究,这方面的内容我们将另文讨论.

References:

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

[2] Fu, Yu-xi. Symmetric π-calculus. Journal of Computer Science and Technology, 1998,13(3):202~208.

[3] Fu, Yu-xi. A proof theoretical approach to communications. In: Degano, P., Gorrieri, R., Marchetti-Spaccamela, A., eds. ICALP'97. Lecture Notes in Computer Science 1256, Berlin: Springer-Verlag, 1997. 325~335.

[4] Fu, Yu-xi. Reaction graphs. Journal of Computer Science and Technology, 1998,13(6):510~530.

[5] Fu, Yu-xi. Bisimulation lattice of Chi processes. In: Hsiang, J., Ohori, A., eds. Proceedings of the ASIAN'98. Lecture

Notes in Computer Science 1538, Berlin: Springer-Verlag, 1998. 245～262.

[6] Fu, Yu-xi. Variations on mobile processes. Theoretical Computer Science, 1999,221(2):327～368.

[7] Fu, Yu-xi. Open Bisimulations of Chi Processes. In: Baeten, J., Mauw, S., eds. Proceedings of the CONCUR'99. Lecture Notes in Computer Science 1664, Berlin: Springer-Verlag, 1999. 304～319.

[8] Victor, B., Parrow, J. Concurrent Constraints in the Fusion Calculus. In: Kim, Guldstrand Larsen, Sven, Skyum, Glynn, Winskel, eds. Proceedings of the ICALP '98. Lecture Notes in Computer Science 1443, Berlin: Springer-Verlag, 1998. 455～469.

[9] Parrow, J., Victor, B. The Tau-Laws of Fusion. In: Sangiorgi, D., de Simone, R., eds. Proceedings of the CONCUR'98. Lecture Notes in Computer Science 1466, Berline: Springer-Verlag, 1998. 99～114.

[10] Sangiorgi, D. A theory of bisimulation for π-calculus. Acta Informatica, 1996,33(1):69～97.

[11] Milner, R., Sangiorgi, D. Barbed Bisimulation. In: Werner, Kuich, ed. Proceedings of the ICALP'92. Lecture Notes in Computer Science 623, Berlin: Springer-Verlag, 1992. 685～695.

Uniformity of Bisimulation Equivalences of Chi Processes*

LIN Min, FU Yu-xi

(*Department of Computer Science and Engineering, Shanghai Jiaotong University, Shanghai* 200030, *China*)

E-mail: fu-yx@cs.sjtu.edu.cn

http://www.sjtu.edu.cn

Abstract: The Chi calculus is obtained from the π calculus by unifying two classes of restricted names. The polyadic Chi calculus extends Chi calculus in that more than one pieces of information can be passed around in a communication. And the atomic Chi calculus is the subcalculus of the polyadic Chi calculus by removing the prefix combinatory. In this paper, the bisimulation equivalences for the atomic Chi calculus are investigated. The main result of this paper is that, in a certain sense, there is only one bisimulation equivalence on the atomic Chi processes.

Key words: process algebra; mobile process; bisimulation

* Received August 3, 1999; accepted July 4, 2000

Supported by the National Natural Science Foundation of China under Grant No. 69873032; the National High Technology Development 863 Program of China under Grant No. 863-306-ZT06-02-2