软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2014,25(11):2433−2451 [doi: 10.13328/j.cnki.jos.004523]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

带 Mismatch 算子的高阶π演算*

徐 贤[1,2]

[1](华东理工大学 计算机科学与工程系,上海 200237)

[2](上海交通大学 计算机科学与技术系,上海 200240)

通讯作者: 徐贤, E-mail: xuxian@ecust.edu.cn

摘 要: 主要研究带 mismatch 的高阶进程演算的公理化问题.首先,建立存在 mismatch 时高阶进程的开弱高阶互模拟理论,证明了等价关系、同余性等重要性质;其次,沿用线性的方法,构建得到带 mismatch 的有限进程上的公理系统;最后,基于对开弱高阶互模拟的刻画,证明了该公理系统的完备性定理.该工作为带 mismatch 的高阶进程上互模拟判定的有效算法的设计与实现,进而为相关的应用建模工作提供了理论借鉴.

关键词: 公理化;互模拟;mismatch;线性;高阶;π演算;进程演算

中图法分类号: TP301

中文引用格式: 徐贤.带 mismatch 算子的高阶π演算.软件学报,2014,25(11):2433−2451. http://www.jos.org.cn/1000-9825/4523.htm

英文引用格式: Xu X. Higher-Order π-calculus with the mismatch operator. Ruan Jian Xue Bao/Journal of Software, 2014,25(11): 2433−2451 (in Chinese). http://www.jos.org.cn/1000-9825/4523.htm

Higher-Order π-Calculus with the Mismatch Operator

XU Xian[1,2]

[1](Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai 200237, China)

[2](Department of Computer Science and Technology, Shanghai Jiaotong University, Shanghai 200240, China)

Corresponding author: XU Xian, E-mail: xuxian@ecust.edu.cn

Abstract: This paper mainly studies the axiomatization of higher-order process calculi with the mismatch operator. Firstly, it formulates the theory of open weak higher-order bisimulation, and shows important properties such as equivalence and congruence. Secondly, following the method on linearity, it builds up an axiom system for finite processes. Finally, based on the characterization of open weak higher-order bisimulation, it proves the completeness of the axiom system. The work of this paper provides the basis for designing and implementing an effective algorithm for checking the bisimulation equivalence of higher-order processes with the mismatch operator, and a theoretical reference for relevant applications of modeling with higher-order processes.

Key words: axiomatization; bisimulation; mismatch; linear; higher-order; π-calculus; process calculus

高阶进程演算研究传进程(process-passing)机制与一阶进程的传名(name-passing)有本质的区别[1−3].高阶进程,如(plain)CHOCS[4−6]和高阶π演算[7],由 Thomsen 和 Sangiorgi 在文献[8,9]中给出初步研究,此后引起研究者的广泛兴趣,主要工作来自以下几个方面,这些相关领域背景为本文工作提供了动机与借鉴.

互模拟理论的逐步演化.Thomsen 首先提出并研究了(应用)高阶互模拟((applicative)higher-order bisimulation)[4,5],提供了高阶进程互模拟的基本框架,但该互模拟因过于精细而有悖于直观.此后,Sangiorgi 提出了广为人知的上下文互模拟(context bisimulation)[9].它将被传递的进程和剩余进程放在同一环境中进行分析比

* 基金项目: 国家自然科学基金-法国国家科研署联合项目 PACE(61261130589, 12IS02001); 国家自然科学基金(61202023, 61173048)

收稿时间: 2008-08-21; 修改时间: 2012-10-19; 定稿时间: 2013-11-11

较,更符合直观观测;同时,Sangiorgi 发现了正规互模拟(normal bisimulation)以简化上下文互模拟的验证工作[7].作为补充,Cao 采用加标技术证明了:在强条件下,上下文互模拟仍与正规互模拟重叠[10],其提出的加标归约互模拟,将不同形式的互模拟定义在一定程度上统一起来[11],并可扩展至移动灰箱演算(另一种高阶演算).基于文献[11]的工作,文献[12]研究了高阶进程上一种基于环境带标记的转换的互模拟,并证明了其同余性.此外,Li 与 Liu 在文献[13]中提出了一种测试观点的语义系统,其上的互模拟理论具有较简单的性质刻画(如同余)且不脱离直观.而 Deng 在文献[14]中研究了层次化高阶π演算(高阶π演算的变体),具有较简洁的语法和形式体系与较易证明的性质.总体而言,以上这些工作为理解高阶进程的等价提供了基础,也为本文的互模拟理论部分提供了借鉴.但这些工作中并未系统地涉及高阶进程的公理化问题,并未考虑 mismath 操作对相关理论体系可能的影响.

高阶进程与一阶进程的比较.Thomsen 研究了用一阶π演算[1]编码高阶 CCS 以及用高阶 CCS 编码一阶π演算[5],并给出了基本的编码策略,但均没有证明重要的全抽象(full abstraction)性质(即编码前的进程等价当且仅当编码后的进程等价).Sangiorgi 改进了该工作,提出了一阶π演算对高阶π演算的编码[9],并证明了全抽象定理.在文献[15]中,我们证明了传进程机制的表达能力严格弱于传名.以上这些工作反映了高阶进程与一阶进程的本质不同,并为本文的公理化工作提供了理论基础与技术上的参考.

高阶进程演算在软件工程、程序语言等领域内得到一定程度的应用.文献[16]的工作采用高阶π演算描述了支持演化的反射式需求规约,并论述了其有效性.在文献[17]中,提出了一种有时间特性的高阶π演算,以支持实时软件系统中使用的构件式设计方法.在文献[18,19]中,高阶π演算(其核心描述机制)被应用于软件设计中体系结构的建模与分析,并通过应用案例说明了方法的可行性.事实上,在程序语言领域,高阶函数(可传递函数自身)是非常重要的概念,其将函数自身视为值的特性,为许多应用提供了更为便利的(编程)模型(如 Haskell,Clojure,Scala 等),并相应地提供了新型的编程思想.高阶进程某种意义上可认为是高阶编程在并发领域的一个扩展,并发在提供更多语义的同时,也使得编程、测试、验证等工作更加复杂.在本文最后,我们结合实际并发环境及与网络安全有关的应用,给出了参考建模原型.尽管在一些工作中有所述及(如文献[20,21]),但对于高阶进程的完备公理系统,仍缺乏较完整的讨论.

以上的领域背景向高阶进程理论提出了更高的关于验证进程间等价性的要求,这需要一个完备的公理系统来支持.对一阶π演算,已有针对不同子语言或不同互模拟下的多种公理系统[22−25];然而,高阶进程演算不同于一阶进程的通信能力使得构造其上可靠而完备的公理系统变得困难,很长时间并没有相关的工作.Fu 对该问题做了进一步的研究[26],关注高阶π演算的线性片段,并得到了公理化方面的进展.线性片段是高阶π演算的一个子演算,它限制一个进程中并发位置上的进程变量的出现次数(不超过 1 次).下面的例子中,进程 P 是线性的,而进程 Q 是非线性的: $P \triangleq c(X).(\overline{d}y.0 \mid X + e(z).0), Q \triangleq c(X).(X \mid \overline{d}y.0 \mid X)$. Fu 成功地构造出该演算上的一个可靠而完备的公理系统.该系统的重要特点在于提供了用于处理高阶前缀以及相对于高阶操作子的(局部)同余的一系列规则(rule),其中的互模拟关系称为局部互模拟(local bisimulation)[26].它与上下文互模拟略有不同,是开型(open style)的[27,28],并且采用了更一般的非 delayed 形式的互模拟[5,7].利用线性性质,高阶进程间的等价性检测可被转换成更轻量形式的比较,这为公理化创造了条件.基于公理系统,Fu 还构造了相应的用于检测两个线性高阶进程是否等价的算法.

Mismatch 操作子与 match 操作子对偶,在一阶 CCS 和π演算中已有过研究.Mismatch 的引入,将引起互模拟理论的一些调整,因为它将影响等价(关系)的封闭性.然而,mismatch 的使用可以让表达一些进程行为变得方便,从而有利于进行系统规约的描述.同时,mismatch 提供了在构建公理系统中的一些便利[22,23].在已有工作的启发下,本文研究带 mismatch 操作子的高阶π演算的公理化,即构造公理系统.为实现这一目标,我们探讨带 mismatch 操作子的(线性)高阶π演算的互模拟理论,据此构建公理系统并证明其完备性.本文的要点如下:

- 定义了带 mismatch 操作子的高阶π演算及其线性片段.这里的演算包含(非确定)选择操作子.
- 给出了开弱高阶互模拟的定义,证明了它是等价关系和同余关系;更重要的是,探讨并建立了局部环境下开弱高阶互模拟的同余性质,给出了进程等价与构成它们的子进程间等价关系的刻画,为构建公理系统提供了条件.

- 建立了带 mismatch 操作子的线性高阶π(有限)进程的公理系统.由于存在 mismatch,我们需要使用完备的头范式(complete head normal form),并建立相应的技术引理.最终,我们证明了公理系统的完备性.
- 总结本文工作,并提出若干后续工作.同时,给出了一个利用带 mismatch 的(线性)高阶π进程进行建模的原型.

1 带 mismatch 的高阶π演算

在这一节中,我们给出带 mismatch 操作子的高阶π演算的语法和语义.我们将该演算称为 mHOPi.

1.1 语 法

mHOPi 的语法如图 1 所示.我们用大写字母 $A,B,E,F,P,Q,\ldots$表示 mHOPi 进程,用小写字母 $a,b,c,x,y,z,\ldots$表示名(name).一个 mHOPi 进程由以下成分结构化地构成:

- 零进程:0.
- 4 个前缀(prefix):一阶输入($a(x)$)、一阶输出 ($\overline{a}x$)、高阶输入($a(X)$)和高阶输出 ($\overline{a}P$).
- 并行复合(parallel composition,$P|P$).
- 限制(restriction,$(x)P$).
- match($[x=y]P$).
- mismatch($[x\neq y]P$).
- 非确定选择(non-deterministic choice,$P+P$).

X 是进程变量,或称变量.它们的优先级是:限制、前缀、match、mismatch、并行复合、选择.

$$P ::= 0, X, \pi.P, P \mid P, (x)P, [x=y]P, [x\neq y]P, P+P, \pi ::= a(x), \overline{a}x, a(X), \overline{a}P$$

Fig.1 Syntax of mHOPi

图 1 mHOPi 语法

我们假设某种结构同余[29],即在任何上下文中,并行复合、限制、选择、match 和 mismatch 各自具有交换性.≡表示语法相等.我们用 $Y,Z,M,N,\ldots$表示名的集合 N 的子集.名向量表示为 $\tilde{x}$,即 $x_1,\ldots,x_n$,向量中的名两两不同.$E[X]$表示最多带自由的(不被输入绑定)进程变量 X 的进程,类似地有 $E[\tilde{X}]$.

我们用 $E[\tilde{A}]$ 表示 $E[X_1,\ldots,X_n]\{A_1/X_1\}\ldots\{A_n/X_n\},\{A_n/X_n\}$表示进程替换.我们有以下一些导出前缀:

$$\tau.P \triangleq (m)(m(x)\mid \overline{m}m.P)(m \text{ fresh}), \overline{a}(x).P \triangleq (x)\overline{a}x.P.$$

fresh 表示不在当前进程中出现.我们将 $0|P$ 直接写为 P,并有简记:a 表示 $a(x).0$; $\overline{a}$ 表示 $\overline{a}(x).0$; τ表示$\tau.0$.名 x 在 $a(x).P$ 和$(x)P$ 中称为受限(或局部)(local,bound,restricted),否则是自由的(free).进程变量 X 在 $a(X).P$ 中是受限的,否则是自由的. $fn(\tilde{P}),bn(\tilde{P})$; $fv(\tilde{P}),bv(\tilde{P})$ 分别表示 $\tilde{P}$ 的自由名、局部名、自由变量和受限变量.不含自由变量的进程称为闭的(closed),否则是开的(open).我们通常考虑闭的进程,直接称其为进程.我们假设α-conversion 会自动作用以防止名的捕获(capture).名替换 $P\{y/x\}$和进程替换 $P\{A/X\}$可基于语法常规地定义.σ表示名替换,而Σ表示进程替换, $\{\tilde{y}/\tilde{x}\}(\{\tilde{A}/\tilde{X}\})$ 表示一系列对应名替换(进程替换).$range(\sigma)$表示σ的值域,通常用 $a\sigma$表示$\sigma(a)$.进程上的二元关系$\mathcal{R}$关于名替换封闭,指的是对于任意替换σ,$(P,Q)\in\mathcal{R}$则$(P\sigma,Q\sigma)\in\mathcal{R}$.类似地,可定义关于进程替换封闭.上下文(context)是一些带空位(hole)的进程,这些空位可以容纳进程,其定义是常规的[30].局部上下文(local contexts)是具有 $(\tilde{x})([\cdot]\mid O)$ 形式的环境,常见的是 $(x_1)\ldots(x_n)(\overline{c}_1x_1\mid\ldots\mid\overline{c}_nx_n\mid[\cdot])\,(x_1)$(或简记为 $(\tilde{x})(\widetilde{\overline{c}x}\mid[\cdot])$,它将在局部同余(local congruence)讨论中使用.

- match 和 mismatch

这里给出 match 和 mismatch 的定义和性质(相关证明参考文献[24]).μ表示 match 序列,δ表示 mismatch 序列,$\varphi,\psi,\phi,\ldots$表示它们的混合序列.$n(\varphi)$表示φ中出现的名集合.φ的长度是指其中 match 和 mismatch 的个数,注意,该长度可以是 0,此时,φP 即 P.φ的闭包 $clo(\varphi)$:对任意φ,如果$\varphi\Rightarrow x\Rightarrow y(\varphi\Rightarrow x\neq y)$,那么 $x=y(x\neq y)$在 $clo(\varphi)$中.$\Rightarrow$表示

逻辑蕴含:$\varphi\Rightarrow\psi$当φ能逻辑地推出ψ.$\varphi\Leftrightarrow\psi$定义为$\varphi\Rightarrow\psi$并且$\psi\Rightarrow\varphi$.$\bot$表示逻辑假.$\bigvee_{i\in I}\varphi_i$和$\bigwedge_{i\in I}\varphi_i$表示条件的析取和合取.$\varphi_x$表示将$\varphi$所有涉及 x 的条件去除.$\neg\varphi$表示φ的否,例如,$\neg[a=b][c\neq d]P$ 定义为$[a\neq b]P+[c=d]P$.下面是match/mismatch 序列完备的定义及一个简单性质:

定义 1. 假设 M 是名集合,φ在 M 上完备(complete on M),如果:

1. $n(\varphi)=M$.
2. 对于 M 中的每一对 x,y,$\varphi\Rightarrow x=y$ 或$\varphi\Rightarrow x\neq y$ 成立.

引理 2. 设φ在 M 上完备,并且 $n(\varphi)\subseteq M$,则$\varphi\psi\Leftrightarrow\varphi$或$\varphi\psi\Leftrightarrow\bot$成立.

下面给出了 match、mismatch 和替换之间的关系及几个不难证明的性质.

定义 3. 设φ是 match 和 mismatch 的序列,σ是一个替换,则有下面的定义:

- σ保持(respects)φ,如果$\varphi\Rightarrow x=y$ 蕴含$\sigma(x)=\sigma(y)$,并且$\varphi\Rightarrow x\neq y$ 蕴含$\sigma(x)\neq\sigma(y)$;
- φ保持(respects)σ,如果$\sigma(x)=\sigma(y)$蕴含$\varphi\Rightarrow x=y$,并且$\sigma(x)\neq\sigma(y)$蕴含$\varphi\Rightarrow x\neq y$;
- σ与φ一致(σ agrees with φ),或者φ与σ一致,如果它们互相保持(respect);
- σ可由φ导出(induced),如果σ与φ一致并且 $range(\sigma)\subseteq n(\varphi)$.

引理 4. 设φ,ψ在 M 上完备,我们有:

1. 若它们均与σ一致,则$\varphi\Leftrightarrow\psi$.
2. 若$\varphi\Rightarrow\psi$,则$\varphi\Leftrightarrow\psi$.

在定义互模拟时,我们需要一些辅助操作.$P^{[x\neq y]}$在进程 P 的每个前缀前加上$[x\neq y]$(结构化定义).

设 Y 是$\{y_1,y_2,\ldots,y_n\}$,则 $P^{[x\notin Y]}\triangleq(\ldots(P^{[x\neq y_1]})^{[x\neq y_2]}\ldots)^{[x\neq y_n]}$,$P^{[Y\not\subseteq Z]}\triangleq(\ldots(P^{[y_1\notin Z]})^{[y_2\notin Z]}\ldots)^{[y_n\notin Z]}$.

此外,$(P^{[x\neq y]})\sigma$定义为$(P\sigma)^{[x\neq(y\sigma)]}$,$(P^{[x\notin Y]})\sigma$和 $(P^{[Y\not\subseteq Z]})\sigma$ 分别定义为$(P\sigma)^{[x\notin(Y\sigma)]}$和 $(P\sigma)^{[Y\not\subseteq(Z\sigma)]}$.

1.2 语 义

图 2 给出了 mHOPi 的操作语义:

- 进程的动作(actions)有:一阶输入 $a(y)$、一阶输出 $\overline{a}y$ 、一阶受限输出 $\overline{a}(y)$; 高阶输入 $a(B)$(有时用 $a[B]$)、高阶输出 $(\tilde{x})\overline{a}A$ (有时用 $(\tilde{x})\overline{a}[A]$).
- 内部动作(internal,silent)τ.我们用$\lambda,\beta,\ldots$表示动作,用 $sub(\lambda),obj(\lambda),fn(\lambda),bn(\lambda)$分别表示动作$\lambda$的主体(subject)、对象(object)、自由名和受限名.语义规则大部分是常规的[9].

$$\frac{}{a(x).P\xrightarrow{a(y)}P\{y/x\}}\quad\frac{}{\overline{a}x.P\xrightarrow{\overline{a}x}P}\quad\frac{bn(P)\cap fn(A)=\varnothing}{a(X).P\xrightarrow{a(A)}P\{A/X\}}\quad\frac{}{\overline{a}A.P\xrightarrow{\overline{a}A}P}\quad\frac{P\xrightarrow{\lambda}P'}{P+Q\xrightarrow{\lambda}P'}$$

$$\frac{P\xrightarrow{\lambda}P'}{P|Q\xrightarrow{\lambda}P'|Q}bn(\lambda)\cap fn(Q)=\varnothing\quad\frac{P\xrightarrow{a(x)}P',Q\xrightarrow{\overline{a}x}Q'}{P|Q\xrightarrow{\tau}P'|Q'}\quad\frac{P\xrightarrow{a(x)}P',Q\xrightarrow{\overline{a}(x)}Q'}{P|Q\xrightarrow{\tau}(x)(P'|Q')}\quad\frac{P\xrightarrow{a(A)}P',Q\xrightarrow{(\tilde{x})\overline{a}[A]}Q'}{P|Q\xrightarrow{\tau}(\tilde{x})(P'|Q')}$$

$$\frac{P\xrightarrow{\lambda}P'}{(x)P\xrightarrow{\lambda}(x)P'}x\notin n(\lambda)\quad\frac{P\xrightarrow{\overline{a}x}P'}{(x)P\xrightarrow{\overline{a}(x)}P'}\quad\frac{P\xrightarrow{(\tilde{x})\overline{a}[A]}P'}{(y)P\xrightarrow{(y)(\tilde{x})\overline{a}[A]}P'}y\in fn(A)-\tilde{x}\quad\frac{P\xrightarrow{\lambda}P'}{[x=x]P\xrightarrow{\lambda}P'}\quad\frac{P\xrightarrow{\lambda}P'}{[x\neq y]P\xrightarrow{\lambda}P'}x\neq y$$

Fig.2 Operational semantics of mHOPi

图 2 mHOPi 的操作语义

在高阶输入中,接收的进程不应与接收环境产生名冲突.当输出中含有受限名时,它们将产生作用范围扩展.例如在高阶通信中,A 中的受限名将扩展以覆盖进程 P'和 Q'.弱迁移(weak transition)的定义如下:

$$\overset{\lambda}{\Rightarrow}\triangleq\Rightarrow\xrightarrow{\lambda}\Rightarrow\ (\Rightarrow\text{为}\xrightarrow{\tau}\text{的自反传递闭包}).$$

在没有 mismatch 时,我们通常有: $P\xrightarrow{\lambda}P'$ 蕴含 $P\sigma\xrightarrow{\lambda\sigma}P'\sigma$; 引入 mismatch,使得它一般不再成立,但我们有下面的引理 5.在高阶(进程)替换中不会有类似问题,我们有引理 6.

引理 5. 如果 $P\sigma\xrightarrow{\lambda'}P''$, 则 $P''\equiv P'\sigma$且$\lambda'\equiv\lambda\sigma$.

引理 6. 设 $fv(P)=\{X_i|i=1,\ldots,n\},\{b_i|i=1,2,\ldots,n\}$是 fresh 名:

1. 若 $P\xrightarrow{\lambda}P'$,则 $P\{P_1/X_1,\ldots,P_n/X_n\}\xrightarrow{\lambda\{P_1/X_1,\ldots,P_n/X_n\}}P'\{P_1/X_1,\ldots,P_n/X_n\}$.

2. 若 $P\{b_1/X_1,...,b_n/X_n\}\xrightarrow{\lambda\{b_1/X_1,...,b_n/X_n\}}P'\{b_1/X_1,...,b_n/X_n\}$,则 $P\xrightarrow{\lambda}P'$.

1.3 线性片段

高阶中线性的概念由 Fu 提出[26](在文献[7]中也有所述及),被证明可以蕴含完备公理系统.我们将使用类似的方法来进行 mHOPi 的公理化.线性强调同一个进程变量 X 不在并发的位置上出现超过 1 次,这在一定程度上有效地降低了演算的表达能力,从而使得公理系统的构建成为可能.下面给出线性片段的定义.

定义 7(linear fragment). mHOPi 的线性片段对进程变量的并发出现次数施加限制.在高阶输出 $\overline{a}Q.P$ 和并行复合 $P|Q$,要求 $fv(P)\cap fv(Q)=\varnothing$.我们调整 mHOPi 语义中的输入规则如下:$\dfrac{fv(A)\cap cp(P,X)=\varnothing}{a(X).P\xrightarrow{a(A)}P\{A/X\}}$.即,输入进程 A 中的进程变量不应与同 X 在并发位置上的进程变量有重叠,否则将破坏线性.其中,$cp(P,X)$操作计算与 X 处于并发位置上(并行复合和高阶输出)的进程变量,其定义是常规的结构化定义(参见文献[26,31]).

下面是关于线性 mHOPi 进程的简单性质,其证明是常规的.

引理 8. 设 $E[X],F[Y],P,A$ 是线性 mHOPi 进程:

1. 如果 $fv(F)\cap cp(E,X)=\varnothing$,则 $E[F[Y]]$也是线性 mHOPi 进程.
2. 如果 $P\xrightarrow{\lambda}P'$ 且λ是一阶或高阶动作或τ动作,则 P'也是线性 mHOPi 进程.

2 互模拟

在这一节中,我们给出 mHOPi 中的互模拟及相关性质.

2.1 互模拟定义

在互模拟定义中我们考虑闭进程.下面的方法可将某个互模拟≈扩展到开进程上:$E[X]\approx F[X]$,若 $E[A]\approx F[A]$ 对任意闭进程 A 成立.如不特别说明,则$\mathcal{R}$表示 mHOPi 进程上的二元关系,$P\mathcal{R}Q$ 表示$(P,Q)\in\mathcal{R}$.

2.1.1 强高阶互模拟

强高阶互模拟的概念(此处称为结构等价)最早由 Thomsen 提出[4,5].结构等价是一种非常强的互模拟形式,它可以在互模拟讨论中作为工具使用.

定义 9(structural equivalence). $\mathcal{R}$(对称)是一个结构等价,如果它关于名替换封闭,且 $P\mathcal{R}Q$ 蕴含:

1. 如果 $P\xrightarrow{\lambda}P'$, 其中,λ是内部动作、一阶输入、一阶输出、一阶受限输出或者高阶输入,则存在 Q'(为简化论述,以后在类似的场合,我们并不每次注明“存在 Q'”(或其他对象)),满足 $Q\xrightarrow{\lambda}Q'$, 并且 $P'\mathcal{R}Q'$.
2. 如果 $P\xrightarrow{(\tilde{x})\overline{a}A}P'$, 则存在 B,Q',满足 $Q\xrightarrow{(\tilde{x})\overline{a}B}Q'$, 并且 $P'\mathcal{R}Q'$以及 $A\mathcal{R}B$.

P,Q(记为 $P\sim_s Q$),如果存在一个结构等价$\mathcal{R}$满足 $P\mathcal{R}Q$,即$\sim_s$是最大的结构等价.

2.1.2 开强高阶互模拟

开强高阶互模拟是开型互模拟,考虑了局部名的独异性[23,26,27].

定义 10(open strong HO bisimulation). $\mathcal{R}$(对称)是一个开强高阶互模拟,如果每当 $P\mathcal{R}Q$ 则下面的性质对所有的σ成立:

1. 如果 $P\sigma\xrightarrow{\lambda}P'$, λ是 $\tau,a(x),\overline{a}x,a(A)$, 则 $Q\sigma\xrightarrow{\lambda}Q'$ 且 $P'\mathcal{R}Q'$.
2. 如果 $P\sigma\xrightarrow{\overline{a}(x)}P'$, 则 $Q\sigma\xrightarrow{\overline{a}(x)}Q'$, 且有 $P'^{[x\notin Y]}\mathcal{R}Q'^{[x\notin Y]}$,其中,$Y=fn(P'|Q')\backslash\{x\}$.
3. 如果 $P\sigma\xrightarrow{(\tilde{x})\overline{a}A}P'$, 则 $Q\sigma\xrightarrow{(\tilde{y})\overline{a}B}Q'$, 且对任意的 $E[X]$满足 $\tilde{x}\tilde{y}\cap fn(E)=\varnothing$, 有:
$$(\tilde{x})(E[A]\,|\,P')^{[\tilde{x}\notin Z]}\mathcal{R}(\tilde{y})(E[B]\,|\,Q')^{[\tilde{y}\notin Z]},$$
其中, $Z=(fn(P'\,|\,Q')\cup fn(E[X]))\backslash\{\tilde{x},\tilde{y}\}$.

P 与 Q 是开强高阶互模拟的(写作 $P\sim_{oh}Q$),如果存在一个开强高阶互模拟$\mathcal{R}$满足 $P\mathcal{R}Q$.

注意,若 $Y=fn(P'|Q')\backslash\{x\}$,则 $Y\sigma\triangleq fn(P'\sigma|Q'\sigma)\backslash\{x\}$.强互模拟太精细,因而从观察角度讲不具备很好的现实意义,虽然它们可以作为研究的某种起始.下面我们给出弱的版本.

2.1.3 开弱高阶互模拟

定义 11(open weak HO bisimulation). $\mathcal{R}$(对称)是开(弱)高阶互模拟,如果每当 $P\mathcal{R}Q$ 则有下面的性质对所有的替换σ成立:

1. 如果 $P\sigma \xrightarrow{\tau} P'$, 则 $Q\sigma \Rightarrow Q'$且 $P'\mathcal{R}Q'$.
2. 如果 $P\sigma \xrightarrow{\lambda} P'$, λ是 $a(x),\overline{a}x,a(A)$, 则 $Q\sigma \overset{\lambda}{\Rightarrow} Q'$ 且 $P'\mathcal{R}Q'$.
3. 如果 $P\sigma \xrightarrow{\overline{a}(x)} P'$, 则 $Q\sigma \overset{\overline{a}(x)}{\Rightarrow} Q'$, 且 $P'^{[x\notin Y]}\mathcal{R}Q'^{[x\notin Y]}$,其中,$Y=fn(P'|Q')\backslash\{x\}$.
4. 如果 $P\sigma \xrightarrow{(\tilde{x})\overline{a}A} P'$, 则 $Q\sigma \overset{(\tilde{y})\overline{a}B}{\Rightarrow} Q'$, 且对任意 $E[X]$满足 $\tilde{x}\tilde{y}\cap fn(E)=\varnothing$, 有:

$$(\tilde{x})(E[A]\mid P')^{[\tilde{x}\notin Z]}\mathcal{R}(\tilde{y})(E[B]\mid Q')^{[\tilde{y}\notin Z]},$$

其中, $Z=(fn(P'\mid Q')\cup fn(E[X]))\setminus\{\tilde{x},\tilde{y}\}$.

P 与 Q 是开弱高阶互模拟的(写作 $P\approx_{oh}Q$),如果存在一个开弱高阶互模拟$\mathcal{R}$满足 $P\mathcal{R}Q$.

我们可以给出一些定义 11 中各个模拟子句的其他版本,如迟(late)、早(early)、delayed 及其组合,并讨论它们的关系.因为与公理化工作关系不大,我们在这里不深入展开(可参考文献[26,32]).下面我们不加证明地给出几个简单的引理,它们将在后面用到.

引理 12. 设 x,y 不同,如果 $P\approx_{oh}Q$,则 $P^{[x\neq y]}\approx_{oh}Q^{[x\neq y]}$.即$\approx_{oh}$对操作$(\cdot)^{[x\neq y]}$封闭.

引理 13. 下面的性质成立:

1. 若 $P^{[x\neq y]}\approx_{oh}Q^{[x\neq y]}$且$\{x,y\}\not\subseteq fn(P|Q)$,则 $P\approx_{oh}Q$.
2. 若 $P^{[x\neq y]}\approx_{oh}Q^{[x\neq y]}$且 $x\neq y$,则 $P\approx_{oh}Q$.
3. 若 $P^{[\tilde{x}\not\subseteq Y]}\approx_{oh}Q^{[\tilde{y}\not\subseteq Y]}$, 则对 fresh 名 x',y',有 $(x')P^{[\tilde{x}x'\not\subseteq Y]}\approx_{oh}(y')Q^{[\tilde{y}y'\not\subseteq Y]}$.

2.1.4 互模拟 Up-to

Up-to 技术是互模拟的证明技术,其基本想法是:尽可能地利用已知的互模拟结果,以降低互模拟证明中所需构造的互模拟关系的大小.在文献[30]中,对 up-to 技术有较为系统的论述.在 mHOPi 中,我们可以定义开弱高阶互模拟 up-to $\sim_s$、开弱高阶互模拟 up-to $\approx_{oh}$等 up-to 技术,用于证明进程间的开弱高阶互模拟.它们的定义是常规的(可参考文献[26,30,32]),例如,开弱高阶互模拟 up-to $\sim_s$的定义是,将开弱高阶互模拟定义(定义 11)的子句中的$\mathcal{R}$替换为$\sim_s\mathcal{R}\sim_s$.基于此,我们有下面的 up-to 证明技术.

引理 14. 若$\mathcal{R}$是开弱高阶互模拟 up-to $\sim_s$关系或开弱高阶互模拟 up-to $\approx_{oh}$关系,则$\mathcal{R}\subseteq\approx_{oh}$.

2.2 等价关系

为了证明$\approx_{oh}$是等价关系,我们先给出一些引理.互模拟引理(bisimulation lemma,简称 BL)由 Fu 提出[26],是一个有用的证明工具.

引理 15(bisimulation lemma). 若对任意σ,$P\sigma\Rightarrow\cdot\approx_{oh}Q$ 且 $Q\sigma\Rightarrow\cdot\approx_{oh}P$,则 $P\approx_{oh}Q$.

证明:我们可以看到如下事实:

(1) 存在开弱高阶互模拟$\mathcal{R}_1$,对某个 P_1 满足 $P\sigma\Rightarrow P_1\sigma\mathcal{R}_1Q$;

(2) 存在开弱高阶互模拟$\mathcal{R}_2$,对某个 Q_1 满足 $Q\sigma\Rightarrow Q_1\sigma\mathcal{R}_2P$.

由此,一个 $P\sigma$的动作可以这样模拟:首先做一系列τ动作到达 $Q_1\sigma$,然后按照$\mathcal{R}_2$ 中的互模拟方式进行模拟;反过来,对于 $Q\sigma$的一个动作也类似.这样,我们构造一个开弱高阶互模拟$\mathcal{R}$,它由$\mathcal{R}_1$ 和$\mathcal{R}_2$ 的并加上(P,Q)得到.因此,我们得到 $P\approx_{oh}Q$. □

下面两个引理是证明$\approx_{oh}$为等价关系的基础,证明类似.在早期工作中[4],出于技术原因,弱迁移是 delayed 风格的:$\overset{\lambda}{\Rightarrow}\triangleq\Rightarrow\xrightarrow{\lambda}$,互模拟引理使我们可以采用一般的弱迁移形式而保持$\approx_{oh}$为等价关系.

引理 16. 如果$(x)(P|a.R)\approx_{oh}(x)(Q|a.R)$,$a$ 为 fresh,则$(x)(P|R)\approx_{oh}(x)(Q|R)$.

证明:因为 a 是 fresh, $(x)(P\mid a.R)\sigma\xrightarrow{a\sigma}(x)(P\mid R)\sigma$ 一定可以用下面的方式模拟:

$$(x)(Q\mid a.R)\sigma \Rightarrow (x)(Q_1\mid a.R)\sigma \xrightarrow{a\sigma} (x)(Q_1\mid R)\sigma \Rightarrow Q' \approx_{oh} (x)(P\mid R),$$

而这一过程可以被重写为 $(x)(Q\mid a.R)\sigma \xrightarrow{a\sigma} (x)(Q\mid R)\sigma \Rightarrow Q' \approx_{oh} (x)(P\mid R)$.

类似地,我们有$(x)(P|R)\sigma \Rightarrow P' \approx_{oh} (x)(Q|R)$.因此用互模拟引理,我们得到$(x)(P|R) \approx_{oh} (x)(Q|R)$. □

引理 17. 设 A,B,P,Q 是闭进程,并且 $G[X]$是最多具有 X 的进程,则 $(\tilde{x})(\overline{a}[A]\mid P) \approx_{oh} (\tilde{y})(\overline{a}[B]\mid Q)$ (a 为 fresh)当且仅当 $(\tilde{x})(G[A]\mid P)^{[\tilde{x}\notin Z]} \approx_{oh} (\tilde{y})(G[B]\mid Q)^{[\tilde{y}\notin Z]}$, 其中, $Z = (fn(P'\sigma \mid Q'\sigma) \cup fn(G[X])) \setminus \{\tilde{x},\tilde{y}\}$.

下面我们来证明$\approx_{oh}$是等价关系,其基础是引理 15~引理 17.

定理 18. 开弱高阶互模拟$\approx_{oh}$是一个等价关系.

证明:我们检查等价关系应满足的条件:自反性、对称性较为直接,我们关注传递性.

假设 $P \approx_{oh} Q \approx_{oh} R$ 而σ是任意一个替换,考虑典型的高阶输出,其他的相对容易,下面我们给出关键的模拟步骤.假设 $P\sigma \xrightarrow{(\tilde{x})\overline{a}A} P'$.

- 要模拟 P,有 $Q\sigma \Rightarrow Q_1\sigma \xrightarrow{(\tilde{y})\overline{a}B} Q_2\sigma \Rightarrow Q'$, 并且 $(\tilde{x})(\overline{a}[A]\mid P') \approx_{oh} (\tilde{y})(\overline{a}[B]\mid Q')$, 其中,$a$ 为 fresh;
- 为了模拟 Q,有 $R\sigma \Rightarrow R_1\sigma \overset{(\tilde{z})\overline{a}C}{\Rightarrow} R_2\sigma$, 并且 $(\tilde{y})(\overline{a}[B]\mid Q_2\sigma) \approx_{oh} (\tilde{z})(\overline{a}[C]\mid R_2\sigma)$.

于是, $(\tilde{z})(\overline{a}[C]\mid R_2\sigma) \Rightarrow (\tilde{z})(\overline{a}[C]\mid R') \approx_{oh} (\tilde{y})(\overline{a}[B]\mid Q')$. 这是因为 $(\tilde{y})(\overline{a}[B]\mid Q_2\sigma) \Rightarrow (\tilde{y})(\overline{a}[B]\mid Q')$, 这一定是由 $R_2\sigma \Rightarrow R'$带来的.

小结一下, $R\sigma \overset{(\tilde{z})\overline{a}C}{\Rightarrow} R'$ 满足 $(\tilde{x})(\overline{a}[A]\mid P') \approx_{oh} (\tilde{y})(\overline{a}[B]\mid Q') \approx_{oh} (\tilde{z})(\overline{a}[C]\mid R')$.

这样,由引理 17 我们有: $(\tilde{x})(E[A]\mid P') \approx_{oh} (\tilde{y})(E[B]\mid Q') \approx_{oh} (\tilde{z})(E[C]\mid R')$ 对任意 $E[X]$ $(fn(E) \cap \tilde{x}\tilde{y}\tilde{z} = \varnothing)$ 成立.

由此,我们完成了模拟过程. □

2.3 同余性

在这一节中,我们建立开弱高阶互模拟$\approx_{oh}$的同余性质.

引理 19. 设 $P \approx_{oh} Q$,则下面的性质成立:

1. $a(x).P \approx_{oh} a(x).Q$.
2. $\overline{a}x.P \approx_{oh} \overline{a}x.Q$.
3. $P|R \approx_{oh} Q|R, R|P \approx_{oh} R|Q$.
4. $(x)P \approx_{oh} (x)Q$.
5. $[x=y]P \approx_{oh} [x=y]Q$.
6. $a(X).P \approx_{oh} a(X).Q$.
7. $\overline{a}A.P \approx_{oh} \overline{a}A.Q$.

证明:我们沿用文献[26]中的方法,构造下面的一系列关系:

$$\mathcal{S}_0 \triangleq \approx_{oh},$$

$$\mathcal{S}_{i+1} \triangleq \{(a(x).P, a(x).Q), (\overline{a}x.P, \overline{a}x.Q), (a(X).P, a(X).Q), (\overline{a}A.P, \overline{a}A.Q),$$
$$(P|R, Q|R), ((x)P, (x)Q), ([x=y]P, [x=y]Q) \mid P\mathcal{S}_iQ\},$$

$$\mathcal{R} \triangleq \bigcup_{i\in\omega} \mathcal{S}_i.$$

设$\mathcal{S}_0 \cup \ldots \cup \mathcal{S}_i$ 中所有进程对满足定义 11 中的性质,我们可以证明$\mathcal{S}_{i+1}$ 所有的进程对也满足这些性质,进而证明$\mathcal{R}$是一个开弱高阶互模拟 up-to $\sim_s$.这一过程是常规的(具体细节参见文献[32]). □

要注意的是:证明高阶演算中的互模拟的同余性质通常并不容易[5,7],这里采用的方法借鉴了领域内的已有方法(如文献[33–35]).弱互模拟一般来说对选择操作子不封闭,对于$\approx_{oh}$ 也是如此.为了得到完整的同余性质,需要将弱的互模拟限制到较小的满足同余性的子关系上[36].

定义 20. 进程 P 和 Q 是开弱高阶同余,记为 $P \simeq_{oh} Q$,如果 $P \approx_{oh} Q$,并且下面的性质成立:

1. 如果 $P\sigma \xrightarrow{\tau} P'$, 则 $Q\sigma \overset{\tau}{\Rightarrow} Q' \approx_{oh} P'$.

2. 如果 $Q\sigma \xrightarrow{\tau} Q'$, 则 $P\sigma \overset{\tau}{\Rightarrow} P' \approx_{oh} Q'$.

3 开弱高阶互模拟的特征刻画

在这一节中,我们考察演算 mHOPi 中开弱高阶互模拟$\approx_{oh}$的一些重要特征.局部环境(上下文)代表了某种具有并可以发送一些局部名的环境,我们用它来建模实际中可以持有一些局部名以帮助算法进行的计算环境.通常,一个局部环境表示为 $(\tilde{x})(\widetilde{\overline{c}x}\,|\cdot)$, 即 $(\tilde{x})(\overline{c}_1x_1\,|\ldots|\,\overline{c}_nx_n\,|\cdot)$, 其中, $\tilde{c}$ 和 $\tilde{x}$ 中的名各自两两不同.当名向量长度为 0 时,局部环境就退化为全局环境(因而是特例).我们下面将考察$\approx_{oh}$ 在局部环境中的同余性质,称为局部同余(local congruence).作为准备,我们先给出带前缀进程和剩余进程间在互模拟上的关系,涉及一阶前缀(命题 21)和高阶前缀(定理 22、定理 25).它们的证明对于局部环境的处理具有某种一般性,可以使用相同的方法(限于篇幅,具体细节可参见文献[31,32]).这些性质将成为公理化的基础.

- 一阶前缀.我们有下面的关于一阶前缀的性质.

命题 21. 设 $a \notin \tilde{c}, x \notin \tilde{z}$, 则:

1. $(\tilde{z})(\widetilde{\overline{c}z}\,|\,a(x).P) \approx_{oh} (\tilde{z})(\widetilde{\overline{c}z}\,|\,a(x).Q)$ 当且仅当 $(\tilde{z})(\widetilde{\overline{c}z}\,|\,P) \approx_{oh} (\tilde{z})(\widetilde{\overline{c}z}\,|\,Q)$.
2. $(\tilde{z})(\widetilde{\overline{c}z}\,|\,\overline{a}y.P) \approx_{oh} (\tilde{z})(\widetilde{\overline{c}z}\,|\,\overline{a}y.Q)$ 当且仅当 $(\tilde{z})(\widetilde{\overline{c}z}\,|\,P) \approx_{oh} (\tilde{z})(\widetilde{\overline{c}z}\,|\,Q)$.

- 高阶输入.我们有下面的关于高阶输入的定理.开弱高阶互模拟中高阶输入的比较,某种意义上被简化至一阶的情形,在文献[31]中有类似的更加细致的相关讨论.

定理 22(abstraction theorem). 设 $a,b,I_b \triangleq b(Z).Z,E[X]$是任意进程,则下面的式子是相互等价的:

1. $(\tilde{z})(\widetilde{\overline{c}z}\,|\,a(X).P) \approx_{oh} (\tilde{z})(\widetilde{\overline{c}z}\,|\,a(X).Q)$.
2. $(\tilde{z})(\widetilde{\overline{c}z}\,|\,P\{I_b/X\}) \approx_{oh} (\tilde{z})(\widetilde{\overline{c}z}\,|\,Q\{I_b/X\})$.
3. $(\tilde{z})(\widetilde{\overline{c}z}\,|\,P\{b/X\}) \approx_{oh} (\tilde{z})(\widetilde{\overline{c}z}\,|\,Q\{b/X\})$.
4. $(\tilde{z})(\widetilde{\overline{c}z}\,|\,P\{E/X\}) \approx_{oh} (\tilde{z})(\widetilde{\overline{c}z}\,|\,Q\{E/X\})$.

- 高阶输出.我们首先给出两个线性进程上的引理,它们的证明是常规的,如建立互模拟的常规技术及归纳方法,我们在此不深入讨论细节(可参考文献[31,32]).

引理 23. 设 a,b 是 fresh 名,$E[X]$是任意的最多带变量 X 的线性 mHOPi 进程,A 是线性 mHOPi 进程,我们有下面的性质:

1. 如果 $(\tilde{x})E[A] \xrightarrow{\lambda} P$, 其中,$A$ 参与动作,则 $(\tilde{x})(E[a]\,|\,\overline{a}.(A+b)) \overset{\lambda}{\Rightarrow} P'$, 并且 $P \sim_{oh} P'$.
2. (反之)如果 $(\tilde{x})(E[a]\,|\,\overline{a}.(A+b)) \overset{\lambda}{\Rightarrow} P'$, 并且 a,b 不在 P'中出现,则 $(\tilde{x})E[A] \overset{\lambda}{\Rightarrow} P$, 并且 $P' \sim_{oh} P$.

引理 24. 设 $a,b,\tilde{c}$ 是 fresh,并且 E,F,A,B 是线性 mHOPi 进程.如果 $(\tilde{x})(\widetilde{\overline{c}x}\,|\,(\tilde{y})(\overline{a}.(A+b)\,|\,E[a])) \approx_{oh} (\tilde{x})(\widetilde{\overline{c}x}\,|\,(\tilde{z})(\overline{a}.(B+b)\,|\,F[a]))$, 则 $(\tilde{x})(\widetilde{\overline{c}x}\,|\,(\tilde{y})E[A]) \approx_{oh} (\tilde{x})(\widetilde{\overline{c}x}\,|\,(\tilde{z})F[B])$.

现在我们可以得到下面的关于高阶输出的定理.

定理 25(concretion theorem). 设 $x \notin \tilde{z}$, a,b 是 fresh, $Z = (fn(P\,|\,Q) \cup fn(E[X])) \setminus \{\tilde{x},\tilde{y}\}$, 并且 $E[X]$是最多带变量 X 任意进程,那么下面的式子是相互等价的:

1. $(\tilde{z})(\widetilde{\overline{c}z}\,|\,(\tilde{x})\overline{a}[A].P) \approx_{oh} (\tilde{z})(\widetilde{\overline{c}z}\,|\,(\tilde{y})\overline{a}[B].Q)$.
2. $(\tilde{z})(\widetilde{\overline{c}z}\,|\,(\tilde{x})(\overline{b}[A]\,|\,P)^{[\tilde{x}\notin Z]}) \approx_{oh} (\tilde{z})(\widetilde{\overline{c}z}\,|\,(\tilde{y})(\overline{b}[B]\,|\,Q)^{[\tilde{y}\notin Z]})$.
3. $(\tilde{z})(\widetilde{\overline{c}z}\,|\,(\tilde{x})(E[A]\,|\,P)^{[\tilde{x}\notin Z]}) \approx_{oh} (\tilde{z})(\widetilde{\overline{c}z}\,|\,(\tilde{y})(E[B]\,|\,Q)^{[\tilde{y}\notin Z]})$.
4. 如果 H,E 是线性 mHOPi 进程,则下面的式子也等价:

$$(\tilde{z})(\widetilde{\overline{c}z}\,|\,(\tilde{x})(\overline{b}.(A+a)\,|\,P)^{[\tilde{x}\notin Z]}) \approx_{oh} (\tilde{z})(\widetilde{\overline{c}z}\,|\,(\tilde{y})(\overline{b}.(B+a)\,|\,Q)^{[\tilde{y}\notin Z]}).$$

下面是定理 25 的一个推论,它的证明可以由引理 13 得出.

推论 26. 设 $x \notin \tilde{z}$, a,b 是 fresh,并且 $E[X]$是最多带变量 X 的任意进程,则下面的式子是相互等价的:

1. $(\tilde{z})(\widetilde{\overline{c}z}\,|\,(\tilde{x})\overline{a}[A].P) \approx_{oh} (\tilde{z})(\widetilde{\overline{c}z}\,|\,(\tilde{y})\overline{a}[B].Q)$.
2. $(\tilde{z})(\widetilde{\overline{c}z}\,|\,(\tilde{x})(\overline{b}[A]\,|\,P)) \approx_{oh} (\tilde{z})(\widetilde{\overline{c}z}\,|\,(\tilde{y})(\overline{b}[B]\,|\,Q))$.
3. $(\tilde{z})(\widetilde{\overline{c}z}\,|\,(\tilde{x})(E[A]\,|\,P)) \approx_{oh} (\tilde{z})(\widetilde{\overline{c}z}\,|\,(\tilde{y})(E[B]\,|\,Q))$.
4. 如果 H,E 是线性 mHOPi 进程,则下面的式子也是等价的:

$$(\tilde{z})(\widetilde{\overline{c}z}\,|\,(\tilde{x})(\overline{b}.(A+a)\,|\,P)) \approx_{oh} (\tilde{z})(\widetilde{\overline{c}z}\,|\,(\tilde{y})(\overline{b}.(B+a)\,|\,Q)).$$

3.1 局部环境中的同余

在本节中,我们考虑局部环境中的同余.首先给出$\simeq_{oh}$(定义 20)同余性质,其证明是常规的.

命题 27. 设 $P\simeq_{oh}Q$,则下面的性质成立:

1. $a(x).P\simeq_{oh}a(x).Q$.
2. $\overline{a}x.P \simeq_{oh} \overline{a}x.Q$.
3. $a(X).P\simeq_{oh}a(X).Q$.
4. $\overline{a}A.P \simeq_{oh} \overline{a}A.Q$.
5. $P|R\simeq_{oh}Q|R$.
6. $(x)P\simeq_{oh}(x)Q$.
7. $P+R\simeq_{oh}Q+R$.
8. $[x=y]P\simeq_{oh}[x=y]Q$.
9. $[x\neq y]P\simeq_{oh}[x\neq y]Q$.

下面我们开始考虑局部同余,即局部环境中进程对于各类操作的封闭性.先给出几个准备引理,它们的证明是对互模拟引理(引理 15)的应用.

引理 28. 设 b 是 fresh,M 是$(fn(P)\cup fn(Q))\backslash\{x\}$,如果 $(\tilde{z})(x)(\overline{b}x\,|\,(\widetilde{\overline{c}z}\,|\,P)) \simeq_{oh} (\tilde{z})(x)(\overline{b}x\,|\,(\widetilde{\overline{c}z}\,|\,Q))$, 则

$$(\tilde{z})(\widetilde{\overline{c}z}\,|\,(x)P^{[x\notin M]}) \simeq_{oh} (\tilde{z})(\widetilde{\overline{c}z}\,|\,(x)Q^{[x\notin M]}).$$

引理 29. 设 a,b 是 fresh 名,$a\neq b$,并且 M 是$(fn(P)\cup fn(Q))\backslash\{x\}$,如果 $(x)(\overline{a}x\,|\,\overline{b}x\,|\,P) \approx_{oh} (x)(\overline{a}x\,|\,\overline{b}x\,|\,Q)$, 则

$$(x)(\overline{a}x\,|\,P^{[x\notin M]}) \approx_{oh} (x)(\overline{a}x\,|\,Q^{[x\notin M]}).$$

下面的两个引理是上面两个的推论:

引理 30. 设 b 是 fresh,如果 $(\tilde{z})(x)(\widetilde{\overline{c}z}\,|\,\overline{b}x\,|\,P) \simeq_{oh} (\tilde{z})(x)(\widetilde{\overline{c}z}\,|\,\overline{b}x\,|\,Q)$, 则 $(\tilde{z})(\widetilde{\overline{c}z}\,|\,(x)P) \simeq_{oh} (\tilde{z})(\widetilde{\overline{c}z}\,|\,(x)Q)$.

引理 31. 设 a,b 是 fresh 名,$a\neq b$,如果 $(x)(\overline{a}x\,|\,\overline{b}x\,|\,P) \approx_{oh} (x)(\overline{a}x\,|\,\overline{b}x\,|\,Q)$, 则 $(x)(\overline{a}x\,|\,P) \approx_{oh} (x)(\overline{a}x\,|\,Q)$.

现在我们给出$\simeq_{oh}$的局部同余性质.

引理 32. 设 $(\tilde{z})(\widetilde{\overline{c}z}\,|\,P) \simeq_{oh} (\tilde{z})(\widetilde{\overline{c}z}\,|\,Q)$, 则下面的性质成立:

1. $(\tilde{z})(\widetilde{\overline{c}z}\,|\,P\,|\,R) \simeq_{oh} (\tilde{z})(\widetilde{\overline{c}z}\,|\,Q\,|\,R)$.
2. $(\tilde{z})(\widetilde{\overline{c}z}\,|\,\tau.P) \simeq_{oh} (\tilde{z})(\widetilde{\overline{c}z}\,|\,\tau.Q)$.
3. $(\tilde{z})(\widetilde{\overline{c}z}\,|\,a(x).P) \simeq_{oh} (\tilde{z})(\widetilde{\overline{c}z}\,|\,a(x).Q)$.
4. $(\tilde{z})(\widetilde{\overline{c}z}\,|\,\overline{a}x.P) \simeq_{oh} (\tilde{z})(\widetilde{\overline{c}z}\,|\,\overline{a}x.Q)$.
5. $(\tilde{z})(\widetilde{\overline{c}z}\,|\,a(X).P) \simeq_{oh} (\tilde{z})(\widetilde{\overline{c}z}\,|\,a(X).Q)$.
6. $(\tilde{z})(\widetilde{\overline{c}z}\,|\,\overline{a}A.P) \simeq_{oh} (\tilde{z})(\widetilde{\overline{c}z}\,|\,\overline{a}A.Q)$.
7. $(\tilde{z})(\widetilde{\overline{c}z}\,|\,(x)P) \simeq_{oh} (\tilde{z})(\widetilde{\overline{c}z}\,|\,(x)Q)$.
8. $(\tilde{z})(\widetilde{\overline{c}z}\,|\,(P+R)) \simeq_{oh} (\tilde{z})(\widetilde{\overline{c}z}\,|\,(Q+R))$.
9. $(\tilde{z})(\widetilde{\overline{c}z}\,|\,[x=y]P) \simeq_{oh} (\tilde{z})(\widetilde{\overline{c}z}\,|\,[x=y]Q)$.
10. $(\tilde{z})(\widetilde{\overline{c}z}\,|\,[x\neq y]P) \simeq_{oh} (\tilde{z})(\widetilde{\overline{c}z}\,|\,[x\neq y]Q)$.

证明:证明主要关注互模拟的部分,提升至同余性由条件出发不难验证.

(1) 不失一般性,设 R 中被 $\tilde{z}$ (即 $fn(R)\cap\tilde{z}$)捕获的名的集合是 $z_{i_1},z_{i_2},...,z_{i_m}$, 定义:

$$E\triangleq c_{i_1}(x_{i_1}).c_{i_2}(x_{i_2}).....c_{i_m}(x_{i_m}).(R\,|\,\overline{c}_{i_1}x_{i_1}\,|\,\overline{c}_{i_2}x_{i_2}\,|\,...\,|\,\overline{c}_{i_m}x_{i_m}),$$

则 $((\tilde{z})(\widetilde{\tilde{c}z}\,|\,P)\,|\,E)\sigma\overset{\tau}{\Rightarrow}\cdot\sim_s(\tilde{z})(\widetilde{\tilde{c}z}\,|\,P\,|\,R)\sigma$ 可以被下面模拟:

$$((\tilde{z})(\widetilde{\tilde{c}z}\,|\,Q)\,|\,E)\sigma\overset{\tau}{\Rightarrow}\cdot\sim_s(\tilde{z})(\widetilde{\tilde{c}z}\,|\,Q\,|\,R)\sigma\Rightarrow\cdot\approx_{oh}(\tilde{z})(\widetilde{\tilde{c}z}\,|\,P\,|\,R)\sigma.$$

对称地,有 $(\tilde{z})(\widetilde{\tilde{c}z}\,|\,P\,|\,R)\sigma\Rightarrow\cdot\approx_{oh}(\tilde{z})(\widetilde{\tilde{c}z}\,|\,Q\,|\,R)\sigma$. 这样,利用互模拟引理可得结论.

(2) 它们具有类似的论证方式(其中,性质 8~性质 10 较为直接).对于前缀相关的部分,我们定义:

$$\mathcal{R}\triangleq\{((\tilde{z})(\tilde{y})(O\,|\,\lambda.P),(\tilde{z})(\tilde{y})(O\,|\,\lambda.Q))\,|\,(\tilde{z})(\widetilde{\tilde{c}z}\,|\,P)\simeq_{oh}(\tilde{z})(\widetilde{\tilde{c}z}\,|\,Q),\tilde{c}\text{ fresh}\}.$$

由与命题 21 类似的讨论,我们可以证明$\mathcal{R}$是开弱高阶互模拟 up-to $\sim_s$.我们以性质 5 为例,其中最有趣的情形是:当 $a=z_i\in\tilde{z}$ 并且 $(\tilde{z})(\tilde{y})(O\,|\,z_i(X).P)\sigma\xrightarrow{\tau}(\tilde{z})(\tilde{y})(\tilde{x})(O'\,|\,P\{A/X\})$ (因为 $O\sigma\xrightarrow{(\tilde{x})\overline{z}_iA}O'$).

定义 $E\triangleq c_i(x).\overline{x}[A].\overline{c}_ix$, 则

$$\begin{aligned}(E\,|\,(\tilde{z})(\widetilde{\tilde{c}z}\,|\,z_i(X).P))\sigma&\xrightarrow{\tau}\sim_s(\tilde{z})((\overline{z}_i[A].\overline{c}_iz_i)\,|\,(\overline{c}_1z_1\,|\,\overline{c}_2z_2\,|\,...\,|\,\overline{c}_{i-1}z_{i-1}\,|\,\overline{c}_{i+1}z_{i+1}\,|\,...\,|\,\overline{c}_nz_n\,|\,z_i(X).P))\\&\xrightarrow{\tau}\sim_s(\tilde{z})(\widetilde{\tilde{c}z}\,|\,P\{A/X\})\end{aligned}$$

一定可以被下面模拟:

$$\begin{aligned}(E\,|\,(\tilde{z})(\widetilde{\tilde{c}z}\,|\,z_i(X).Q))\sigma&\xrightarrow{\tau}\sim_s(\tilde{z})((\overline{z}_i[A].\overline{c}_iz_i)\,|\,(\overline{c}_1z_1\,|\,\overline{c}_2z_2\,|\,...\,|\,\overline{c}_{i-1}z_{i-1}\,|\,\overline{c}_{i+1}z_{i+1}\,|\,...\,|\,\overline{c}_nz_n\,|\,z_i(X).Q))\\&\xrightarrow{\tau}\sim_s(\tilde{z})(\widetilde{\tilde{c}z}\,|\,Q\{A/X\})\Rightarrow\sim_s(\tilde{z})(\widetilde{\tilde{c}z}\,|\,Q')\approx_{oh}(\tilde{z})(\widetilde{\tilde{c}z}\,|\,P\{A/X\}).\end{aligned}$$

这样,我们有 $(\tilde{z})(\tilde{y})(O\,|\,z_i(X).Q)\sigma\overset{\tau}{\Rightarrow}(\tilde{z})(\tilde{y})(\tilde{x})(O'\,|\,Q')\approx_{oh}(\tilde{z})(\widetilde{\tilde{c}z}\,|\,P\{A/X\})$. □

由引理 32,下面的定理是我们所需要的关于局部环境中的同余性的结果.

定理 33(local congruence). 设 $(\tilde{z})(\widetilde{\tilde{c}z}\,|\,P)\simeq_{oh}(\tilde{z})(\widetilde{\tilde{c}z}\,|\,Q)$, 则对任意的上下文 $C[\cdot]$(其中,$C[\cdot]$中没有 $\tilde{c}$ 和 $\tilde{z}$ 的出现),有 $(\tilde{z})(\widetilde{\tilde{c}z}\,|\,C[P])\simeq_{oh}(\tilde{z})(\widetilde{\tilde{c}z}\,|\,C[Q])$.

4 公理化

在本节,我们构建有限线性 mHOPi 进程关于$\simeq_{oh}$的公理系统(axiom system),这是构建有效算法的基础.这里的公理系统由文献[26]的工作演变而来,增加了关于 mismatch 操作子的部分,其中重要的技术路线是类似的,因而我们将省略一些技术引理的证明细节(可参考文献[32]).传统的展开定理(expansion theorem/law)在 mHOPi 中具有以下形式(设 $P\triangleq\sum_{i\in I}\varphi_i\lambda_i.P_i,Q\triangleq\sum_{j\in J}\varphi_j\lambda_j.Q_j$, I,J 是下标集合,Σ表示有限次选择操作):

$$\begin{aligned}P\,|\,Q=&\sum_{i\in I}\varphi_i\lambda_i.(P_i\,|\,Q)+\sum_{j\in J}\varphi_j\lambda_j.(P\,|\,Q_j)+\sum_{i\in I,j\in J}^{\lambda_i\equiv\overline{a}_ix_i,\lambda_j\equiv b_j(y)}\varphi_i\varphi_j[a_i=b_j]\tau.(P_i\,|\,Q_j\,\{x_i/y\})+\\&\sum_{i\in I,j\in J}^{\lambda_i\equiv a_i(x),\lambda_j\equiv\overline{b}_jy_j}\varphi_i\varphi_j[a_i=b_j]\tau.(P_i\{y_j/x\}\,|\,Q_j)+\sum_{i\in I,j\in J}^{\lambda_i\equiv(\tilde{x})\overline{a}_iA_i,\lambda_j\equiv b_j(Y)}\varphi_i\varphi_j[a_i=b_j]\tau.(\tilde{x})(P_i\,|\,Q_j\,\{A_i/Y\})+\\&\sum_{i\in I,j\in J}^{\lambda_i\equiv a_i(X),\lambda_j\equiv(\tilde{y})\overline{b}_jB_j}\varphi_i\varphi_j[a_i=b_j]\tau.(\tilde{y})(P_i\{B_j/X\}\,|\,Q_j).\end{aligned}$$

展开定理在完备性的证明中起着非常重要的作用,它将一个有限进程转换为一个等价的不含并行复合的进程,由此可以定义范式(normal form),并在完备性证明中只考虑范式.但在高阶场合,由于存在进程变量无法用展开定理有效展开进程,如进程 $P|X$,因而无法完全消除并行复合.所以我们考虑闭进程(不含自由变量),并据此定义头范式(head normal form).头范式和范式的不同之处在于:前者并没有将并行复合完全消除,例如进程 $a(X).(P|X)$,当 X 被实例化后,才可利用展开定理作展开.

我们在下面用二元整数向量定义进程的尺度(深度,depth),其中的第 1 个元素表示高阶前缀的深度,后者表示一阶前缀的深度(定义$\langle u_0,u_1\rangle+\langle v_0,v_1\rangle\triangleq\langle u_0+v_0,u_1+v_1\rangle$):

$$d(0) \triangleq \langle 0,0\rangle,$$
$$d(X) \triangleq \langle 0,0\rangle,$$
$$d(a(x).P) \triangleq \langle 0,1\rangle + d(P),$$
$$d(\overline{a}x.P) \triangleq \langle 0,1\rangle + d(P),$$
$$d(a(X).P) \triangleq \langle 1,0\rangle + d(P),$$
$$d(\overline{a}A.P) \triangleq \langle 1,0\rangle + d(P) + d(A),$$
$$d([x = y]P) \triangleq d(P),$$
$$d([x \neq y]P) \triangleq d(P),$$
$$d((x)P) \triangleq d(P),$$
$$d(P \mid Q) \triangleq d(P) + d(Q),$$
$$d(P + Q) \triangleq \max\{d(P), d(Q)\}.$$

在构建公理系统中,我们将需要以下若干引理(限于篇幅,对于前两个引理,可参见文献[32]中的证明细节):

- Normalization 引理(引理 39):它们分别将 mHOPi 进程转换成等价的不增加深度的头范式(head normal form)和完备头范式(complete head normal form),后者是出于对 mismatch 的考虑.
- Saturation 引理(引理 40):它给出了 mHOPi 的操作语义相关的等式刻画,将用于下面的 Promotion 引理的证明.
- Promotion 引理(引理 41)和完备性定理(定理 43):前者提供了一种对公理系统完备性的某种弱化的结论,由它可以较直接地推导出完备性定理.

4.1 公理系统

表 1 中给出了公理系统,每一个等式组由对应的标签(label)标识(对应相应的操作子).例如,"T"开头的法则(laws)表示 TAU 法则,即与内部动作相关的等式.

Table 1 Basic axioms

表 1 基本公理

E1	$P=P$	**M3**	$[x=y](P+Q)=[x=y]P+[x=y]Q$
E2	$Q=P$,若 $P=Q$	**M4**	$[x\neq y](P+Q)=[x\neq y]P+[x\neq y]Q$
E3	$P=R$,若 $P=Q=R$	**M5**	$[x\neq x]P=\mathbf{0}$
L1	$(x)\mathbf{0}=\mathbf{0}$	**M6**	$[x\neq y]\lambda.P=[x\neq y]\lambda.[x\neq y]P$,若 $\{x,y\}\cap bn(\lambda)=\varnothing$
L2	$(x)X=X$	**S1**	$P+\mathbf{0}=P$
L3	$(x)\lambda.P=\lambda.(x)P$,若 $x\notin n(\lambda)$	**S2**	$P+Q=Q+P$
L4	$(x)\lambda.P=\mathbf{0}$,若 $x\in subj(\lambda)$	**S3**	$P+(Q+R)=(P+Q)+R$
L5	$(x)(P\mid Q)=P\mid(x)Q$,若 $x\notin fn(P)$	**S4**	$[x=y]P+P=P$
L6	$(x)(y)P=(y)(x)P$	**S5**	$[x=y]P+[x\neq y]P=P$
L7	$(x)[y=z]P=[y=z](x)P$,若 $x\notin n\{y,z\}$	**T1**	$\lambda.\tau.P=\lambda.P$
L8	$(x)[x=y]P=\mathbf{0}$,若 $x\neq y$	**T2**	$P+\tau.P=\tau.P$
L9	$(x)[x\neq y]P=\mathbf{0}$,若 $x=y$	**T3**	$\lambda.(P+\varphi\tau.Q)=\lambda.(P+\varphi\tau.Q)+\varphi\lambda.Q$
L10	$(x)(P+Q)=(x)P+(x)Q$	**T4**	$\tau.P=\tau.(P+[x=y]\tau.P)$
M1	$\phi P=\psi P$,若 $\phi\Leftrightarrow\psi$	**T5**	$\sum_{i\in I}a(x).(P_i+\varphi\tau.Q)=\sum_{i\in I}a(x).(P_i+\varphi\tau.Q)+\varphi a(x).Q \vee_{i\in I}\varphi_i \Leftrightarrow \varphi, x\notin n(\varphi)$
M2	$[x=y]P=[x=y]P\{y/x\}$		

表 2 中列出了一些导出等式,它们的证明是常规的[22,23,26].

Table 2 Derived laws

表 2 导出法则

D1	$\varphi P+P=P$	**D6**	$(x)[x\neq y]P=(x)P$,若 $x\neq y$
D2	$[x\neq y]P+P=P$	**D7**	$(x)[y\neq z]P=[y\neq z](x)P$,若 $x\notin\{y,z\}$
D3	$[x=x]P=P$	**D8**	$(x)P=P$,若 $x\notin fn(P)$
D4	$[x=y]\mathbf{0}=\mathbf{0}$	**D9**	$\tau.P=\tau.\left(P+\sum_{i\in I}\varphi_i\tau.P\right)$
D5	$[x\neq y]\mathbf{0}=\mathbf{0}$	**D10**	$[x=y]\lambda.P=[x=y]\lambda.[x=y]P$,若 $\{x,y\}\cap bn(\lambda)=\varnothing$

作为一个例子,我们在下面说明如何由 L7 和 S5 推导出 D7,其中,每行括号中的标签说明了在这一步推导中用到的主要的法则.

$$\begin{aligned}(x)[y\neq z]P &= [y=z](x)[y\neq z]P+[y\neq z](x)[y\neq z]P && \text{(S5,M3)}\\ &= (x)[y=z][y\neq z]P+[y\neq z](x)[y\neq z]P && \text{(L7)}\\ &= [y\neq z](x)[y\neq z]P && \text{(M1,M5,L1,S1)}\\ [y\neq z](x)P &= [y\neq z](x)([y=z]P+[y\neq z]P) && \text{(S5)}\\ &= [y\neq z][y=z](x)P+[y\neq z](x)[y\neq z]P && \text{(M4,L7)}\\ &= [y\neq z](x)[y\neq z]P && \text{(M1,M5,S1)}\end{aligned}$$

此外,当$\phi\Leftrightarrow$true 时,D1 退化为 $P+P=P$;D2 可以由 S5 和 D1 导出.我们说两个规则集合是等效的,如果它们可以(在保持剩余公理不变的情况下)互相推导出.

下面我们再给出一些导出法则(推导方法可参考文献[23]).

首先是由 M6 导出的等式(见表 3),注意,其中的标签反映了它们与某个基本公理的关系.例如,等式 L11a 和 L11b 的合并效果与 L11 相当,而 M6a 与 M6 具有相当的效果.

Table 3 M6 derived laws

表 3 M6 导出法则

L11	$(x)C[[x=y]P]=(x)C[\mathbf{0}]$,若 $x,y\notin bn(C[\cdot]),x\neq y$	**L11b**	$(x)P^{[x\neq y]}=(x)P$,若 $x\notin bn(P),x\neq y$
L11a	$(x)C[[x\neq y]P]=(x)C[P]$,若 $x,y\notin bn(C[\cdot]),x\neq y$	**M6a**	$P^{[x\neq y]}=[x\neq y]P$

表 4 是由 TAU 法则导出的等式.例如:T3a,T3b 由 T3 演化而来;T4a,T4b 由 T4 演化而来(并与之等效);T5a 由 T5 演化而来.注意,在 T5a 中, $f(a,P,Q,\delta)\equiv\sum_{y\in Y}a(x).(P_y+\delta[x=y]\tau.Q)+a(x).(P+\delta[x\notin Y]\tau.Q)$.

Table 4 Derived TAU laws

表 4 导出 TAU 法则

T3a	$\overline{a}x.(P+\delta\tau.Q)=\overline{a}x.(P+\delta\tau.Q)+\delta\overline{a}x.Q$	**T4b**	$\tau.P=\tau.(P+\delta\tau.P)$
T3b	$a(x).(P+\delta\tau.Q)=a(x).(P+\delta\tau.Q)+\delta a(x).Q$	**T5a**	$f(a,P,Q,\delta)=f(a,P,Q,\delta)+\delta a(x).Q$
T4a	$\tau.P=\tau.(P+[x=y]\tau.P)$		

表 5 给出了公理系统中的规则.共有 5 个规则对应局部同余(local congruence),它们是 Prefix Rule(PR), Restriction Rule(RR),Match Rule(MR1),Mismatch Rule(MR2),Parallel Rule(ParR)和 Choice Rule(CR).其中的前缀 α是任意动作, $a,\tilde{c},c',b,d$ 均是 fresh,并且 $M=fn(P|Q)\backslash\{y\}$, $N=(fn(P\mid Q))\backslash\{\tilde{x},\tilde{y},\tilde{z}\}$. 另外两个规则用于处理高阶前缀,其中,Abstraction Rule(AR)对应高阶输入,Concretion Rule(ConR)对应高阶输出.此外,Derived Concretion Rule (DConR)是基于 M6 的导出规则.

Table 5 Rules

表 5 规则

Prefix Rule (PR): $\dfrac{(\tilde{x})(\widetilde{\overline{c}x}\mid P)=(\tilde{x})(\widetilde{\overline{c}x}\mid Q)}{(\tilde{x})(\widetilde{\overline{c}x}\mid\alpha.P)=(\tilde{x})(\widetilde{\overline{c}x}\mid\alpha.Q)}$	Choice Rule (CR): $\dfrac{(\tilde{x})(\widetilde{\overline{c}x}\mid P_1)=(\tilde{x})(\widetilde{\overline{c}x}\mid Q_1),(\tilde{x})(\widetilde{\overline{c}x}\mid P_2)=(\tilde{x})(\widetilde{\overline{c}x}\mid Q_2)}{(\tilde{x})(\widetilde{\overline{c}x}\mid(P_1+Q_1))=(\tilde{x})(\widetilde{\overline{c}x}\mid(P_2+Q_2))}$
Match Rule (MR1): $\dfrac{(\tilde{x})(\widetilde{\overline{c}x}\mid P)=(\tilde{x})(\widetilde{\overline{c}x}\mid Q)}{(\tilde{x})(\widetilde{\overline{c}x}\mid[y=z]P)=(\tilde{x})(\widetilde{\overline{c}x}\mid[y=z]Q)}$	Abstraction Rule (AR): $\dfrac{P\{c/X\}=Q\{c/X\}}{P=Q}$
Mismatch Rule (MR2): $\dfrac{(\tilde{x})(\widetilde{\overline{c}x}\mid P)=(\tilde{x})(\widetilde{\overline{c}x}\mid Q)}{(\tilde{x})(\widetilde{\overline{c}x}\mid[y\neq z]P)=(\tilde{x})(\widetilde{\overline{c}x}\mid[y\neq z]Q)}$	Concretion Rule (ConR): $\dfrac{(\tilde{x})(\widetilde{\overline{c}x}\mid(\tilde{y})(\overline{b}.(A+d)\mid P)^{[\tilde{y}\notin N]})=(\tilde{x})(\widetilde{\overline{c}x}\mid(\tilde{z})(\overline{b}.(B+d)\mid Q)^{[\tilde{z}\notin N]})}{(\tilde{x})(\widetilde{\overline{c}x}\mid(\tilde{y})\overline{a}A.P)=(\tilde{x})(\widetilde{\overline{c}x}\mid(\tilde{z})\overline{a}B.Q)}$
Parallel Rule (ParR): $\dfrac{(\tilde{x})(\widetilde{\overline{c}x}\mid P_1)=(\tilde{x})(\widetilde{\overline{c}x}\mid Q_1),(\tilde{x})(\widetilde{\overline{c}x}\mid P_2)=(\tilde{x})(\widetilde{\overline{c}x}\mid Q_2)}{(\tilde{x})(\widetilde{\overline{c}x}\mid(P_1\mid Q_1))=(\tilde{x})(\widetilde{\overline{c}x}\mid(P_2\mid Q_2))}$	Derived Concretion Rule (DConR): $\dfrac{(\tilde{x})(\widetilde{\overline{c}x}\mid(\tilde{y})(\overline{b}.(A+d)\mid P))=(\tilde{x})(\widetilde{\overline{c}x}\mid(\tilde{z})(\overline{b}.(B+d)\mid Q))}{(\tilde{x})(\widetilde{\overline{c}x}\mid(\tilde{y})\overline{a}A.P)=(\tilde{x})(\widetilde{\overline{c}x}\mid(\tilde{z})\overline{a}B.Q)}$

$\mathcal{AS}_{LM}$表示由表 1 定义的法则和表 5 定义的规则构成的系统.当进程 P,Q 可在$\mathcal{AS}_{LM}$中推出相等时,我们记为$\mathcal{AS}_{LM}\vdash P=Q$.我们知道:规则 PR,MR1,MR2 和 CR 相对于$\simeq_{oh}$是可靠的(sound),依据是引理 32 和引理 30;定理 22 和定理 25 则分别保证了规则 AR 和 ConR 是可靠的.从表 5 中的规则不难得到下面的引理:

引理 34. 如果 $(\tilde{z})(\widetilde{\bar{c}z}\mid P)=(\tilde{z})(\widetilde{\bar{c}z}\mid Q)$, 则对任意的上下文 $C[\cdot]$,有 $(\tilde{z})(\widetilde{\bar{c}z}\mid C[P])=(\tilde{z})(\widetilde{\bar{c}z}\mid C[Q])$.

引理 35. 如果 $(\tilde{x})(z)(\widetilde{\bar{c}x}\mid \bar{c}'z\mid P)=(\tilde{x})(z)(\widetilde{\bar{c}x}\mid \bar{c}'z\mid Q)$ 且 $z\notin fn(P,Q)$,则 $(\tilde{x})(\widetilde{\bar{c}x}\mid P)=(\tilde{x})(\widetilde{\bar{c}x}\mid Q)$.

4.2 范　式

我们给出头范式(head normal form)及相关性质.

定义 36. 如下形式的进程: $\sum_{i\in I}\psi_i\lambda_i.P_i$, 称为头范式或说在头范式中.

注意:在头范式中,正如我们此前的解释,加项(summand)P_i 不必是一个头范式;(头范式)在做一个动作之后,所得到的遗留进程可以转换成另一个头范式.下面是完备头范式(complete head normal form)的定义.

定义 37. 设 M 是有限名的集合,进程 P 称为 M 上的完备头范式,若它具有如下形式: $\sum_{i\in I}\psi_i\lambda_i.P_i$, 其中,对 $i\in I$, $bn(\lambda_i)\cap M=\varnothing$且$\varphi$在 M 上完备.

完备头范式的想法与头范式类似,其特点在于对 match 和 mismatch 构成的条件序列可以进行更加一致的处理,这样可以在完备性证明中的等式推理中获得一些便利.事实上,因为 mismatch 的存在,范式的完备性变得必须[23].下面是一个关于(完备)头范式的不难证明的引理:

引理 38. 如果 P 是一个(完备)头范式,则对任意的替换σ,$P\sigma$也是一个(完备)头范式.

下面的 Normalization 引理(正规化引理)说明:任意一个有限的线性 mHOPi 进程可以转换成一个等价的深度不增加的(完备)头范式,其证明是对进程的结构作归纳(可参考文献[32]).

引理 39(normalization). 设 P 是有限的线性 mHOPi 进程,则存在一个(完备)头范式 P'满足下面的性质:

1. $\mathcal{AS}_{LM}\vdash P=P'$.
2. $d(P')\leqslant d(P)$.
3. 对任意的替换σ, $P\sigma\xrightarrow{\lambda}P''$ 当且仅当 $P'\sigma\xrightarrow{\lambda}P''$.

4.3 Saturation性质

我们给出 Saturation 引理,它指出了 mHOPi 操作语义相关的基本等式刻画(限于篇幅,证明细节可以参考文献[22,23,32]).注意,需要用到的引理有引理 4、引理 38、引理 39 等.

引理 40(saturation). 设 P 是有限线性 mHOPi 进程,M 是有限名的集合满足 $fn(P)\subseteq M$,φ在 M 上完备,并且σ是由φ导出的替换.我们有下面的性质:

1. 如果 $P\sigma\overset{\tau}{\Rightarrow}P'$, 则$\mathcal{AS}_{LM}\vdash P=P+\varphi\tau.P'$.
2. 如果 $P\sigma\overset{a(x)}{\Rightarrow}P'$, 其中,$x\notin fn(P\sigma)$,则$\mathcal{AS}_{LM}\vdash P=P+\varphi a(x).P'$.
3. 如果 $P\sigma\overset{\bar{a}x}{\Rightarrow}P'$, 则 $\mathcal{AS}_{LM}\vdash P=P+\varphi\bar{a}x.P'$.
4. 如果 $P\sigma\overset{\bar{a}(x)}{\Rightarrow}P'$, 则 $\mathcal{AS}_{LM}\vdash P=P+\varphi\bar{a}(x).P'$.
5. 如果 $P\sigma\overset{a(c)}{\Rightarrow}P'\{c/X\}$, 其中,$c$ 是 fresh,则$\mathcal{AS}_{LM}\vdash P=P+\varphi a(X).P'$.
6. 如果 $P\sigma\overset{(\tilde{x})\bar{a}A}{\Rightarrow}P'$, 则 $\mathcal{AS}_{LM}\vdash P=P+\varphi(\tilde{x})\bar{a}A.P'$.

4.4 完备性

我们给出重要的 Promotion 引理,它本质上建立了我们的公理系统的完备性.注意:在后面利用公理系统进行推导时,有时会用带注释的等号(例如 $\overset{M1}{=}$)来表明在一步推理中主要用到的公理.

引理 41(promotion). 设 P,Q 是有限线性 mHOPi 进程, $\tilde{c}$ 是 fresh,如果 $(\tilde{x})(\widetilde{\bar{c}x}\mid P)\approx_{oh}(\tilde{x})(\widetilde{\bar{c}x}\mid Q)$, 则

$$\mathcal{AS}_{LM} \vdash (\tilde{x})(\widetilde{\overline{c}x} \mid \tau.P) = (\tilde{x})(\widetilde{\overline{c}x} \mid \tau.Q).$$

证明:由引理 39,我们假设 P,Q 是 M 上的完备头范式,其中,$(fn(P)\cup fn(Q))\subseteq M$,即

$$P \equiv \sum_{i\in I}\varphi_i\lambda_i.P_i \text{且} Q \equiv \sum_{j\in J}\varphi_j\lambda_j.Q_j.$$

进一步地,我们设 $\tilde{x}$ 不在任何 $\varphi_k(k\in(I\cup J))$中出现.

该引理的证明是对 $d(P)+d(Q)$的归纳.初始情况是 $d(P)+d(Q)=0$,即 $d(P)=d(Q)=0$,这一点不难验证.现在假设σ是一个由φ_i 导出的替换.我们给出并在后面证明两个对称的关于进程 P 和 Q 的构成(加项)关系的断言(claim).由这两个断言出发,我们可以在 TAU 法则的帮助下证明本引理的结论.

Claim 1. 存在不相交的集合 I_1,I_2,且假设它们的并$(I_1\cup I_2)I$,则:

(1) 对任意 $i\in I_1$,此时$\lambda_i\equiv\tau$,有 $\mathcal{AS}_{LM} \vdash (\tilde{x})(\widetilde{\overline{c}x} \mid \varphi_i\tau.P_i) = (\tilde{x})(\widetilde{\overline{c}x} \mid \varphi_i\tau.Q)$.

(2) 对任意 $i\in I_2$,存在 Q_i' 满足$\mathcal{AS}_{LM}\vdash Q=Q+\varphi_i\lambda_i.Q_{i'}$并且 $\mathcal{AS}_{LM} \vdash (\tilde{x})(\widetilde{\overline{c}x} \mid \varphi_i\lambda_i.P_i) = (\tilde{x})(\widetilde{\overline{c}x} \mid \varphi_i\lambda_i.Q_{i'})$.

Claim 2. 存在不相交的集合 J_1,J_2,且假设它们的并$(J_1\cup J_2)$为 J,则:

(1) 对任意 $j\in J_1$,此时$\lambda_j\equiv\tau$,有 $\mathcal{AS}_{LM} \vdash (\tilde{x})(\widetilde{\overline{c}x} \mid \psi_j\tau.Q_j) = (\tilde{x})(\widetilde{\overline{c}x} \mid \psi_j\tau.P)$.

(2) 对任意 $j\in J_2$,存在 P_j' 满足$\mathcal{AS}_{LM}\vdash P=P+\psi_j\lambda_j.P_{j'}$并且 $\mathcal{AS}_{LM} \vdash (\tilde{x})(\widetilde{\overline{c}x} \mid \psi_j\lambda_j.Q_j) = (\tilde{x})(\widetilde{\overline{c}x} \mid \psi_j\lambda_j.P_{j'})$.

下面我们证明上面的两个断言,因为它们是对称的,我们只证明第 1 个.下面的分析分成两个组:第 1 组中,P,Q 的加项$\varphi_i\lambda_i.P_i$ 的动作λ_i 之名不在 $\tilde{x}$ 中出现;第 2 个组中,动作λ_i 中的名在 $\tilde{x}$ 中出现.通常,一对匹配(模拟)的(子)进程的下标是不同的,为方便起见,我们假设用于模拟的进程(如 $Q\sigma$中)与被模拟的(子)进程(如 $P\sigma$中)具有相同的下标,并且在其上加上一撇.同时注意:用于模拟的进程不一定是Q(或P)的直接(最外层的)加项,因为我们考虑的是弱互模拟.

第 1 组:有若干种情况要讨论.我们关注通信以及高阶(输出)动作,对其余动作可做类似的处理.

- $\lambda_i\equiv\tau$.由已知条件可知:动作 $(\tilde{x})(\widetilde{\overline{c}x} \mid P\sigma)\xrightarrow{\tau}(\tilde{x})(\widetilde{\overline{c}x} \mid P_i\sigma)$ 可以由 $(\tilde{x})(\widetilde{\overline{c}x} \mid Q\sigma)$ 用一系列内部动作模拟,且又可以被分成若干情况,即,内部动作序列的长度是否为 0.
  - 如果模拟是空模拟,则 $(\tilde{x})(\widetilde{\overline{c}x} \mid P_i\sigma)\approx_{oh}(\tilde{x})(\widetilde{\overline{c}x} \mid Q\sigma)$. 由于进程的深度因产生了一个内部动作而变小,由归纳假设, $\mathcal{AS}_{LM} \vdash (\tilde{x})(\widetilde{\overline{c}x} \mid \tau.P_i\sigma) = (\tilde{x})(\widetilde{\overline{c}x} \mid \tau.Q\sigma)$. 然后用 Match Rule(MR1)和 Mismatch Rule (MR2)得到 $\mathcal{AS}_{LM} \vdash (\tilde{x})(\widetilde{\overline{c}x} \mid \varphi_i\tau.P_i\sigma) = (\tilde{x})(\widetilde{\overline{c}x} \mid \varphi_i\tau.Q\sigma)$.
  - 如果模拟不是空模拟,则 $(\tilde{x})(\widetilde{\overline{c}x} \mid Q\sigma)\overset{\tau}{\Rightarrow}(\tilde{x})(\widetilde{\overline{c}x} \mid Q_i'\sigma)\approx_{oh}(\tilde{x})(\widetilde{\overline{c}x} \mid P_i\sigma)$, 其中, $Q\sigma\overset{\tau}{\Rightarrow}Q_i'\sigma$. 因此,由引理 40, $\mathcal{AS}_{LM} \vdash Q = Q+\varphi_i\tau.Q_i'\sigma = Q+\varphi_i\tau.Q_i'$. 由于进程的深度因产生了一个内部动作而变小,由归纳假设, $\mathcal{AS}_{LM} \vdash (\tilde{x})(\widetilde{\overline{c}x} \mid \tau.Q_i'\sigma) = (\tilde{x})(\widetilde{\overline{c}x} \mid \tau.P_i\sigma)$; 由 Match Rule(MR1)和 Mismatch Rule(MR2),$\mathcal{AS}_{LM}\vdash (\tilde{x})(\widetilde{\overline{c}x} \mid \varphi_i\tau.Q_i'\sigma) = (\tilde{x})(\widetilde{\overline{c}x} \mid \varphi_i\tau.P_i\sigma)$. 然后有 $\mathcal{AS}_{LM} \vdash (\tilde{x})(\widetilde{\overline{c}x} \mid \varphi_i\tau.Q_i') = (\tilde{x})(\widetilde{\overline{c}x} \mid \varphi_i\tau.P_i)$.
- $\lambda_i\equiv(\tilde{y})\overline{a}_iA_i$.我们有下面的动作: $(\tilde{x})(\widetilde{\overline{c}x} \mid P\sigma)\xrightarrow{(\tilde{x}_1\tilde{y})\overline{a_i\sigma}[A_i\sigma]}(\tilde{x}_2)(\widetilde{\overline{c}x} \mid P_i\sigma)$, 其中, $\tilde{x}_1\tilde{x}_2$ 是 $\tilde{x}$. 设 d,e 是 fresh.由已知条件可知:存在 $\tilde{z}$,B_i 和 Q_i' 满足 $(\tilde{x})(\widetilde{\overline{c}x} \mid Q\sigma)\overset{(\tilde{x}_1'\tilde{z})\overline{a_i\sigma}[B_i\sigma]}{\Rightarrow}(\tilde{x}_2')(\widetilde{\overline{c}x} \mid Q_i'\sigma)$, 其中, $\tilde{x}_1'\tilde{x}_2'$ 是 $\tilde{x}$, 并且

$$(\tilde{x}_1\tilde{y})(E[A_i\sigma] \mid (\tilde{x}_2)(\widetilde{\overline{c}x} \mid P_i\sigma))^{[\tilde{x}_1\tilde{y}\not\subseteq M]}\approx_{oh}(\tilde{x}_1'\tilde{z})(E[B_i\sigma] \mid (\tilde{x}_2')(\widetilde{\overline{c}x} \mid Q_i'\sigma))^{[\tilde{x}_1'\tilde{z}\not\subseteq M]}$$

对每个 $E[X]$满足 $\tilde{x}\tilde{y}\tilde{z}\cap fn(E)=\varnothing$ 成立,其中, $M=(fn(P'\sigma|Q'\sigma)\cup fn(E[X]))\setminus\{\tilde{x},\tilde{y},\tilde{z}\}$.

从上面的模拟可以知道两个事实:

- 首先, $Q\sigma\overset{(\tilde{x}_1'\tilde{z})\overline{a_i\sigma}[B_i\sigma]}{\Rightarrow}Q_i'\sigma$. 由引理 40,有 $\mathcal{AS}_{LM} \vdash Q = Q+\varphi_i(\tilde{x}_1'\tilde{z})\overline{a_i\sigma}[B_i\sigma].Q_i'\sigma = Q+\varphi_i(\tilde{x}'\tilde{z})\overline{a}_i[B_i].Q_i'$;
- 其次,如果选择 $E\equiv\overline{d}.(X+e)$, 则得到:

$$(\tilde{x}_1\tilde{y})(\overline{d}.(A_i\sigma+e) \mid (\tilde{x}_2)(\widetilde{\overline{c}x} \mid P_i\sigma))^{[\tilde{x}_1\tilde{y}\not\subseteq M]}\approx_{oh}(\tilde{x}_1'\tilde{z})(\overline{d}.(B_i\sigma+e) \mid (\tilde{x}_2')(\widetilde{\overline{c}x} \mid Q_i'\sigma))^{[\tilde{x}_1'\tilde{z}\not\subseteq M]}.$$

同样地,如果取 $E\equiv X+e$,则得到 $(\tilde{x}_1\tilde{y})((A_i\sigma+e) \mid (\tilde{x}_2)(\widetilde{\overline{c}x} \mid P_i\sigma))^{[\tilde{x}_1\tilde{y}\not\subseteq M]}\approx_{oh}(\tilde{x}_1'\tilde{z})((B_i\sigma+e) \mid (\tilde{x}_2')(\widetilde{\overline{c}x} \mid Q_i'\sigma))^{[\tilde{x}_1'\tilde{z}\not\subseteq M]}$.

用一个比较简单的推理(注意引理 13)可以得到下面的结果:

$$(\tilde{x})(\widetilde{\overline{c}x}\,|\,(\tilde{y})(\overline{d}.(A_i\sigma+e)\,|\,P_i\sigma)^{[\tilde{x}\tilde{y}\not\in M]}) \approx_{oh} (\tilde{x})(\widetilde{\overline{c}x}\,|\,(\tilde{z})(\overline{d}.(B_i\sigma+e)\,|\,Q_i'\sigma)^{[\tilde{x}\tilde{z}\not\in M]}),$$

$$(\tilde{x})(\widetilde{\overline{c}x}\,|\,(\tilde{y})((A_i\sigma+e)\,|\,P_i\sigma)^{[\tilde{x}\tilde{y}\not\in M]}) \approx_{oh} (\tilde{x})(\widetilde{\overline{c}x}\,|\,(\tilde{z})((B_i\sigma+e)\,|\,Q_i'\sigma)^{[\tilde{x}\tilde{z}\not\in M]}).$$

从上面第 2 个式子,用常规的分析方法不难得到:

$$(\tilde{x})(\widetilde{\overline{c}x}\,|\,(\tilde{y})((A_i\sigma+e)\,|\,\tau.P_i\sigma)^{[\tilde{x}\tilde{y}\not\in M]}) \approx_{oh} (\tilde{x})(\widetilde{\overline{c}x}\,|\,(\tilde{z})((B_i\sigma+e)\,|\,\tau.Q_i'\sigma)^{[\tilde{x}\tilde{z}\not\in M]}).$$

因为局部环境中的进程(即 $(\tilde{y})(\overline{d}.(A_i\sigma+e)\,|\,P_i\sigma)^{[\tilde{x}\tilde{y}\not\in M]}$,$(\tilde{z})(\overline{d}.(B_i\sigma+e)\,|\,Q_i'\sigma)^{[\tilde{x}\tilde{z}\not\in M]}$ 和 $(\tilde{y})((A_i\sigma+e)\,|\,\tau.P_i\sigma)^{[\tilde{x}\tilde{y}\not\in M]}$,$(\tilde{z})((B_i\sigma+e)\,|\,\tau.Q_i'\sigma)^{[\tilde{x}\tilde{z}\not\in M]}$)的深度因产生了从高阶输出到一阶动作的转变而变小,由归纳假设可知:

$$\mathcal{AS}_{LM} \vdash (\tilde{x})(\widetilde{\overline{c}x}\,|\,\tau.(\tilde{y})(\overline{d}.(A_i\sigma+e)\,|\,P_i\sigma)^{[\tilde{x}\tilde{y}\not\in M]}) = (\tilde{x})(\widetilde{\overline{c}x}\,|\,\tau.(\tilde{z})(\overline{d}.(B_i\sigma+e)\,|\,Q_i'\sigma)^{[\tilde{x}\tilde{z}\not\in M]}) \quad (*)$$

$$\mathcal{AS}_{LM} \vdash (\tilde{x})(\widetilde{\overline{c}x}\,|\,\tau.(\tilde{y})((A_i\sigma+e)\,|\,\tau.P_i\sigma)^{[\tilde{x}\tilde{y}\not\in M]}) = (\tilde{x})(\widetilde{\overline{c}x}\,|\,\tau.(\tilde{z})((B_i\sigma+e)\,|\,\tau.Q_i'\sigma)^{[\tilde{x}\tilde{z}\not\in M]}) \quad (**)$$

将引理 34 应用到式(**)并且使用 T1 可得:

$$\mathcal{AS}_{LM} \vdash (\tilde{x})(\widetilde{\overline{c}x}\,|\,\overline{d}.(\tilde{y})((A_i\sigma+e)\,|\,\tau.P_i\sigma)^{[\tilde{x}\tilde{y}\not\in M]}) = (\tilde{x})(\widetilde{\overline{c}x}\,|\,\overline{d}.(\tilde{z})((B_i\sigma+e)\,|\,\tau.Q_i\sigma)^{[\tilde{x}\tilde{z}\not\in M]}) \quad (**')$$

现在,在下面的两个进程上使用 expansion 定理:

$$(\tilde{x})(\widetilde{\overline{c}x}\,|\,(\tilde{y})(\overline{d}.(A_i\sigma+e)\,|\,\tau.P_i\sigma)^{[\tilde{x}\tilde{y}\not\in M]}),(\tilde{x})(\widetilde{\overline{c}x}\,|\,(\tilde{z})(\overline{d}.(B_i\sigma+e)\,|\,\tau.Q_i'\sigma)^{[\tilde{x}\tilde{z}\not\in M]}).$$

在 Choice Rule(CR)的帮助下可以得到:

$$\mathcal{AS}_{LM} \vdash (\tilde{x})(\widetilde{\overline{c}x}\,|\,(\tilde{y})(\overline{d}.(A_i\sigma+e)\,|\,\tau.P_i\sigma)^{[\tilde{x}\tilde{y}\not\in M]}) = (\tilde{x})(\widetilde{\overline{c}x}\,|\,(\tilde{z})(\overline{d}.(B_i\sigma+e)\,|\,\tau.Q_i'\sigma)^{[\tilde{x}\tilde{z}\not\in M]}).$$

由 Concretion Rule(ConR),有 $\mathcal{AS}_{LM} \vdash (\tilde{x})(\widetilde{\overline{c}x}\,|\,(\tilde{y})(\overline{a_i\sigma}[A_i\sigma].\tau.P_i\sigma)) = (\tilde{x})(\widetilde{\overline{c}x}\,|\,(\tilde{z})(\overline{a_i\sigma}[B_i\sigma].\tau.Q_i'\sigma))$.

由 T1,有 $\mathcal{AS}_{LM} \vdash (\tilde{x})(\widetilde{\overline{c}x}\,|\,(\tilde{y})(\overline{a_i\sigma}[A_i\sigma].P_i\sigma)) = (\tilde{x})(\widetilde{\overline{c}x}\,|\,(\tilde{z})(\overline{a_i\sigma}[B_i\sigma].Q_i'\sigma))$.

由 Match Rule(MR1)和 Mismatch Rule(MR2),有:

$$\mathcal{AS}_{LM} \vdash (\tilde{x})(\widetilde{\overline{c}x}\,|\,\varphi_i(\tilde{y})(\overline{a_i\sigma}[A_i\sigma].P_i\sigma)) = (\tilde{x})(\widetilde{\overline{c}x}\,|\,\varphi_i(\tilde{z})(\overline{a_i\sigma}[B_i\sigma].Q_i'\sigma)).$$

由关于 match 和 mismatch 的公理,有 $\mathcal{AS}_{LM} \vdash (\tilde{x})(\widetilde{\overline{c}x}\,|\,\varphi_i(\tilde{y})(\overline{a}_iA_i.P_i)) = (\tilde{x})(\widetilde{\overline{c}x}\,|\,\varphi_i(\tilde{z})(\overline{a}_iB_i.Q_i'))$.

第 2 组:有多种情况要考虑,与第 1 组中的情况很类似.作为示例,下面我们考虑其中一个典型情况:

- $\lambda_i \equiv (\tilde{y})\overline{x}_iA_i$. 我们将 $x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n$ 记为 $\tilde{x}_{-i}$,将 $c_1,\ldots,c_{i-1},c_{i+1},\ldots,c_n$ 记为 $\tilde{c}_{-i}$; 同时,设 d,e,f 是 fresh 名.关键的一个事实是,这里的动作须以下面的方式发生:首先,受限名 x_i 被发送给环境,因为只有 $\widetilde{\overline{c}x}$ 具有此种能力,因此必须先有一个类似于 $\overline{c}x_i$ 的动作将 x_i 发送出去;其次,在 $P\sigma$和接收到受限名 x_i 的进程之间发生一个通信.设进程 O 是这样的接收名 x_i 的进程;因为$\approx_{oh}$ 对并行复合封闭,我们可以将它和 $(\tilde{x})(\widetilde{\overline{c}x}\,|\,P\sigma)$ 及 $(\tilde{x})(\widetilde{\overline{c}x}\,|\,Q\sigma)$ 用并行复合操作复合.然后,我们考察这两个新的进程行为,从中发现需要的模拟成分来完成模拟步.

定义 O 为 $O \triangleq f + x_i(X).(\overline{d}.(X+e)\,|\,\overline{c}_ix_i)$,这样,我们有下面的推演步骤:

$$c_i(x_i).O\,|\,(\tilde{x})(\widetilde{\overline{c}x}\,|\,P\sigma) \xrightarrow{\tau} (x_i)((f+x_i(X).(\overline{d}.(X+e)\,|\,\overline{c}_ix_i))\,|\,(\tilde{x}_{-i})(\widetilde{\overline{c}_{-i}x_{-i}}\,|\,P\sigma))$$

$$\xrightarrow{\tau} (x_i)(\tilde{y})(\overline{d}.(A_i\sigma+e)\,|\,\overline{c}_ix_i\,|\,(\tilde{x}_{-i})(\widetilde{\overline{c}_{-i}x_{-i}}\,|\,P_i\sigma)) \sim_s (\tilde{x})(\widetilde{\overline{c}x}\,|\,(\tilde{y})(\overline{d}.(A_i\sigma+e)\,|\,P_i\sigma)).$$

前面提到,这些动作序列必须由 $c_i(x_i).O\,|\,(\tilde{x})(\widetilde{\overline{c}x}\,|\,Q\sigma)$ 来模拟,方式如下(我们假设 Q'是完备头范式,这由引理 39 保证):

$$c_i(x_i).O\,|\,(\tilde{x})(\widetilde{\overline{c}x}\,|\,Q\sigma) \overset{\tau}{\Rightarrow} (x_i)((f+x_i(X).(\overline{d}.(X+e)\,|\,\overline{c}_ix_i))\,|\,(\tilde{x}_{-i})(\widetilde{\overline{c}_{-i}x_{-i}}\,|\,Q'\sigma))$$

$$\overset{\tau}{\Rightarrow} (x_i)(\tilde{z})(\overline{d}.(B_i\sigma+e)\,|\,\overline{c}_ix_i\,|\,(\tilde{x}_{-i})(\widetilde{\overline{c}_{-i}x_{-i}}\,|\,Q_i'\sigma)) \sim_s (\tilde{x})(\widetilde{\overline{c}x}\,|\,(\tilde{z})(\overline{d}.(B_i\sigma+e)\,|\,Q_i'\sigma));$$

并且, $(\tilde{x})(\widetilde{\overline{c}x}\,|\,(\tilde{y})(\overline{d}.(A_i\sigma+e)\,|\,P_i\sigma)) \approx_{oh} (\tilde{x})(\widetilde{\overline{c}x}\,|\,(\tilde{z})(\overline{d}.(B_i\sigma+e)\,|\,Q_i'\sigma))$.

从上面的分析可发现,其中必然有下面的动作: $Q\sigma \overset{(\tilde{z})\overline{x}_i[B_i\sigma]}{\Rightarrow} Q_i'\sigma$.至此,剩下的部分与第 1 组中高阶输出场合的讨论类似,我们不进一步展开.

证明了上面的两个断言以后,我们现在进一步得出本引理的最终结论.将上述两个断言的结果合并起来,并使用 Choice Rule(CR),我们得到:

- 由断言 1(Claim 1):

$$\mathcal{AS}_{LM} \vdash (\tilde{x})(\widetilde{\overline{c}x} \mid (P+Q)) = (\tilde{x})\left(\widetilde{\overline{c}x}\left|\left(\sum_{i\in I_0}\varphi_i\tau.Q + \sum_{i\in I_1}\varphi_i\lambda_i.Q_i' + Q\right)\right.\right) = (\tilde{x})\left(\widetilde{\overline{c}x}\left|\left(\sum_{i\in I_0}\varphi_i\tau.Q + Q\right)\right.\right).$$

- 由断言 2(Claim 2):

$$\mathcal{AS}_{LM} \vdash (\tilde{x})(\widetilde{\overline{c}x} \mid (Q+P)) = (\tilde{x})\left(\widetilde{\overline{c}x}\left|\left(\sum_{j\in J_0}\varphi_j\tau.P + \sum_{j\in J_1}\varphi_j\lambda_j.P_j' + P\right)\right.\right) = (\tilde{x})\left(\widetilde{\overline{c}x}\left|\left(\sum_{j\in J_0}\varphi_j\tau.P + P\right)\right.\right).$$

因此,我们有 $\mathcal{AS}_{LM} \vdash (\tilde{x})\left(\widetilde{\overline{c}x}\left|\left(\sum_{i\in I_0}\varphi_i\tau.Q + Q\right)\right.\right) = (\tilde{x})\left(\widetilde{\overline{c}x}\left|\left(\sum_{j\in J_0}\varphi_j\tau.P + P\right)\right.\right)$.

- 应用 Prefix Rule(PR),我们有 $\mathcal{AS}_{LM} \vdash (\tilde{x})\left(\widetilde{\overline{c}x}\left|\tau.\left(\sum_{i\in I_0}\varphi_i\tau.Q + Q\right)\right.\right) = (\tilde{x})\left(\widetilde{\overline{c}x}\left|\tau.\left(\sum_{j\in J_0}\varphi_j\tau.P + P\right)\right.\right)$.
- 应用 T4,我们有 $\mathcal{AS}_{LM} \vdash (\tilde{x})(\widetilde{\overline{c}x} \mid \tau.Q) = (\tilde{x})(\widetilde{\overline{c}x} \mid \tau.P)$.

因此,我们得到 $\mathcal{AS}_{LM} \vdash (\tilde{x})(\widetilde{\overline{c}x} \mid \tau.P) = (\tilde{x})(\widetilde{\overline{c}x} \mid \tau.Q)$ 的结论. □

下面是 Promotion 引理(引理 41)的一个推论以及完备性定理.

推论 42. 设 P,Q 是有限线性 mHOPi 进程,如果 $P \approx_{oh} Q$,则 $\mathcal{AS}_{LM} \vdash \tau.P = \tau.Q$.

定理 43(completeness). 设 P,Q 是有限线性 mHOPi 进程,则 $P \simeq_{oh} Q$ 当且仅当 $\mathcal{AS}_{LM} \vdash P = Q$.

证明:我们只给出证明的要点,因为证明的内容和 Promotion 引理很类似[23,25]:

(1) 两个与 Promotion 引理(引理 41)中类似的断言;

(2) 在断言的证明中,在与原先 Promotion 引理中使用归纳假设的对应处转而使用 Promotion 引理;

(3) 将断言的结论综合起来,用类似 Promotion 引理中的方法证明 P 和 Q 均和 $P+Q$ 在公理系统中相等.

证毕. □

4.5 讨 论

我们知道,一阶π演算中的开互模拟[27]有对局部名相关的处理上鉴别能力过强的缺点,因此在拟开互模拟(quasi open bisimulation)[25,28]中,这一点被加以改进,其中,重点是对局部名的处理;通常在任何情况下,局部名不应该和上下文中的任何名(局部或自由)相同.之后,Fu 给出了对拟开互模拟的一些刻画,例如局部互模拟(local bisimulation)[25].在局部互模拟中,一阶受限输出子句的形式与高阶输出的形式有些类似.进一步,Fu 给出了一个可靠的完备的公理系统[25],其中扩充了对应于 quasi open 语义的等式.

在 mHOPi 中,我们也可以考虑拟开互模拟,称为拟开弱高阶互模拟(quasi open weak higher-order bisimulation).它把开弱高阶互模拟中的一部分设置为 quasi open 的风格.我们可以给出基于该互模拟的公理系统,其思路与开弱高阶互模拟的情形是类似的,不同的部分主要在于对局部名的处理采用了 quasi open 的想法(限于篇幅,可参见文献[32]中的细节展开).

5 总 结

本文在带 mismatch 的高阶π演算中完成了以下工作:

(1) 定义了带 mismatch 的高阶π演算及其线性片段;

(2) 定义了非 delayed 且开型的互模拟——开弱高阶互模拟;

(3) 证明了开弱高阶互模拟($\approx_{oh}$)的等价性及(局部)同余性等性质;

(4) 建立了有限线性进程上相对于开弱高阶同余的完备公理系统.

基于本文的公理系统,可以进一步设计检测有限线性进程关于开弱高阶同余($\simeq_{oh}$)等价性的判定算法.

基于文中的讨论,我们认为,判定算法可以扩展到上述互模拟的强版本.相应地,对于这些判定算法的复杂性分析则是一个有意义的后续工作.此外,另一个后续问题是:线性条件可以放松到什么程度,使得我们仍然能有类似的结论(如公理化)?线性具有较好的现实意义,现实应用(如网络计算)中,资源敏感性是一个比较常见的

现象.那么,线性条件放松之后的演算是否具有及具有怎样的现实意义,是值得探索的问题,其中的关键点在于控制进程变量的出现,因为正是该因素,在很大程度上调控着高阶进程的表达能力.我们相信:相关的理论体系的探讨是一个有意义的后续问题,因为高阶通信在各种网络计算中很常见,例如在网络计算中传递一个 Java 小应用程序或者一个函数对象以调用某个所需要的方法,相关研究一定程度上可为现实应用提供理论基础.此处为了展示本文的演算在实际应用建模中的可能形式,我们给出一个建模的原型.该原型描述了分布式应用中网上银行在线支付的若干步骤,其中有几个特征:首先,在线支付过程中,有些资源是受限的,例如一个支付会话,其中涉及到一些用以处理相关信息的 handler(例如,在服务计算中,私有数据传递的服务端口信息等),这些资源常常(在一定的会话范围内)只能使用 1 次,因而采用线性进程即可满足要求;其次,目前主流的网上银行一般都支持 RSA[37]等技术加密的(动态同步)密钥,内含动态口令(典型地,如,一串数字可抽象为一个名),在支付过程中需要进行比对,这里将使用 mismatch.下面给出该原型的定义:

$$Server \triangleq (e)a(x_1).\overline{x}_1(c).\overline{x}_1[Handler].e(x_2).e(x_3).Verifier,$$

$$Handler \triangleq c(z_1).\overline{e}z_1.c(z_2).\overline{e}z_2.0,$$

$$Verifier \triangleq (d)(Inquiry(x_2,d) \mid d(y).([y=x_3]\tau.Success + [y \neq x_3]\tau.Failure)),$$

$$Client \triangleq \overline{a}(f).f(z_3).f(X).(X \mid \overline{z}_3 n.\overline{z}_3 k.0).$$

在上面的原型中,*Server* 代表银行端的服务进程;*Client* 代表客户端的希望进行在线付款的客户启动的进程;*Handler* 代表一个 *Server* 发给 *Client* 并在客户端运行的进程,如前所述,该进程涉及一个特定会话并以线性方式使用,它的作用是在客户端接收信息(如客户的银行卡号、动态口令),并以客户端不知道的端口传回服务进程;*Verifier* 代表服务端的验证进程,在得到客户的相关信息(银行卡号、动态口令)后,进行查询比对(调用 *Inquiry* 进程).在比对成功的情况下予以扣款(表示为 *Success* 进程),否则拒绝请求(表示为 *Failure* 进程).为简化细节并突出主要步骤,这里没有给出 *Success*,*Failure*,*Inquiry* 进程的定义.对于 *Inquiry*,我们采用了非正式的表示 *Inquiry*(x_2,*d*),其含义是:*Inquiry* 将接受两个参数 x_2,*d*(来自客户端),利用 x_2 进行(数据库)查询,并通过 *d* 将查询结果(服务端的用于比对的动态口令)返回给 *Verifier*.以下是系统 *Server*|*Client* 主要运行过程的描述(注意,为简化,我们并没有考虑传输等过程中的数据加密):

(1) 通过公开的 *a* 通道,*Client* 将用于接收 *Server*(资源)信息的私有端口 *f* 发给 *Server*.

(2) *Server* 通过 *f*,发送两个对象给 *Client*:首先是一个私有名 *c*,其次是一个 *Handler* 实例.前者将用于 *Client* 向 *Handler* 传递相关数据(如银行卡号、动态口令).

(3) *Client* 接收到 *Handler* 后(注意,此处进程变量 *X* 满足线性要求),通过 *c* 向其发送卡号(*n*)和动态口令(*k*).它们将由 *Handler* 通过另一个私有通道 *e*(*Client* 并不知道该通道),依次传递给 *Server*.

(4) *Server* 接收到卡号(*n*)和动态口令(*k*)后,利用 *n* 调用 *Inquiry* 进程,通过私有通道 *d* 得到查询结果(*y*),然后将结果与 *k* 进行比对:若比对成功,则进入 *Success* 进程(如实际扣款操作等);否则,进入 *Failure* 进程(付款失败).

我们在后续工作中可以进一步地探究高阶进程的形式化体系的一般数学模式,尤其是其中的一些证明技术,如互模拟性质的证明、公理系统完备性的证明等.观察后不难发现,它们的规律都是有迹可寻的,比如:

- 互模拟的建立常利用一些较通用的 up-to 技术[30,35,38];
- 公理系统的正确性证明常需要的步骤是:范式、Normalization 引理、Saturation 性质、Promotion 引理等[22,26,36].

如果能够将证明的模式提取出来,得到一个较为通用的证明框架(如充分条件集合),将有效地简化证明的过程.进一步地,将这些证明的框架与当前主流的自动证明工具(例如 coq[39])结合起来,实现自动证明,则可以把诸多与高阶进程典型性质有关的证明的繁琐过程加以简化,使研究者可以将注意力放在更为关键的问题上.领域内已有的这方面的工作还不太多,如果要建立一个通用的数学框架,仍有不少工作需要展开.因此,在本文的后续工作中将这一方向的探索结合进来,不但有利于简化研究过程,而且有助于揭示高阶进程的更多本质特征.

致谢 感谢傅育熙教授在本文完成过程中给予的帮助,感谢BASICS实验室的所有老师和博士生对本文所提供的意见和建议.同时,向对本文的工作给予支持和建议的同行专家表示感谢.

References:

[1] Milner R, Parrow J, Walker D. A calculus of mobile processes (parts i and ii). Information and Computation, 1992,100(1):1−77. [doi: 10.1016/0890-5401(92)90008-4]

[2] Baeten JCM. A brief history of process algebra. Theoretical Computer Science, 2002,335(2-3):131−146. [doi: 10.1016/j.tcs.2004.07.036]

[3] Yang B, Huang J, Liu DY. Study on the theories and formalized methods of mobile agent computing. Journal of Computer Research and Development, 2006,43(Suppl.):274−278 (in Chinese with English abstract).

[4] Thomsen B. Calculi for higher order communicating systems [Ph.D. Thesis]. Department of Computing, Imperial College, 1990.

[5] Thomsen B. Plain chocs, a second generation calculus for higher-order processes. Acta Informatica, 1993,30(1):1−59. [doi: 10.1007/BF01200262]

[6] Thomsen B. A theory of higher order communication systems. Information and Computation, 1995,116(1):38−57. [doi: 10.1006/inco.1995.1004]

[7] Sangiorgi D. Bisimulation for higher-order process calculi. Information and Computation, 1996,131(2):141−178. [doi: 10.1006/inco.1996.0096]

[8] Thomsen B. A calculus of higher order communication systems. In: Proc. of the POPL'89. ACM Press, 1989. 143−154. [doi: 10.1145/75277.75290]

[9] Sangiorgi D. Expressing mobility in process algebras: First-Order and higher-order paradigms [Ph.D. Thesis]. University of Edinburgh, 1992.

[10] Cao ZN. More on bisimulations for higher order π-calculus. In: Aceto L, Inġolfsdóttir A, eds. Proc. of the FOSSACS 2006. LNCS 3921, Springer-Verlag, 2006. 63−78. [doi: 10.1007/11690634_5]

[11] 曹子宁.加标归约互模拟与高阶互模拟研究[博士学位论文].北京:中国科学院软件研究所,2003.

[12] 张严.一种高阶进程代数的弱互模拟研究[硕士学位论文].南京:南京航空航天大学,2008.

[13] Li YJ, Liu XX. Towards a theory of bisimulation for the higher-order process calculi. Journal of Computer Science and Technology, 2004,19(3):352−363. [doi: 10.1007/BF02944905]

[14] Deng YX. Symbolic open semantics for the full π-calculus [MS. Thesis]. Shanghai: Shanghai Jiaotong University, 2001 (in Chinese with English abstract).

[15] Xu X. Distinguishing and relating higher-order and first-order processes by expressiveness. Acta Informatica, 2012,49(7-8):445−484. [doi: 10.1007/s00236-012-0168-9]

[16] Yuan WJ, Ying S, Wu KJ, Yao JF. Formal description of the evolving reflective requirements specification with π-calculus. Computer Engineering and Science, 2010,32(6):146−154 (in Chinese with English abstract).

[17] You T, Du CL, Wang XW, Zheng W. A new component-based real-time system based on timed high-order (THO) π calculus. Journal of Northwestern Polytechnical University, 2009,27(6):6−11 (in Chinese with English abstract).

[18] Li CY, Li GS, He PJ. A fomal dynamic architecture description language. Ruan Jian Xue Bao/Journal of Software, 2006,17(6):1349−1359 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/17/1349.htm

[19] Gao J, Shen CL, Zheng MF, Li CY. Architecture description language of software oriented self-adaptive. Application Research of Computers, 2010,27(5):1796−1801 (in Chinese with English abstract).

[20] 詹乃军.高阶时段演算及其应用[博士学位论文].北京:中国科学院软件研究所,2000.

[21] 詹乃军.高阶时段演算及其完备性.中国科学(E 辑),2001,31(1):71−85.

[22] Parrow J, Sangiorgi D. Algebraic theories for name-passing calculi. Information and Computation, 1995,120(2):174−197. [doi: 10.1006/inco.1995.1108]

[23] Fu YX, Yang ZR. Understanding the mismatch combinator in chi-calculus. Theoretical Computer Science, 2003,290(1):779−830. [doi: 10.1016/S0304-3975(02)00373-0]

[24] Fu YX, Yang ZR. Tau laws for pi calculus. Theoretical Computer Science, 2003,308(1-3):55−130. [doi: 10.1016/S0304-3975(03)00202-0]

[25] Fu YX. On quasi open bisimulation. Theoretical Computer Science, 2005,338(1-3):96−126. [doi: 10.1016/j.tcs.2004.10.041]

[26] Fu YX. Checking equivalence for higher order processes. Technical Report, Shanghai: Shanghai Jiaotong University, 2005.

[27] Sangiorgi D. A theory of bisimulation for π-calculus. Acta Informatica, 1996,33(1):69−97. [doi: 10.1007/s002360050036]

[28] Sangiorgi D, Walker D. On barbed equivalences in π-calculus. In: Larsen KG, Nielsen M, eds. Proc. of the CONCUR 2001. LNCS 2154, Springer-Verlag, 2001. 292−304. [doi: 10.1007/3-540-44685-0_20]

[29] Milner R. Functions as processes. Journal of Mathematical Structures in Computer Science, 1992,2(2):119−141. [doi: 10.1017/S0960129500001407]

[30] Sangiorgi D, Walker D. The π-calculus: A Theory of Mobile Processes. Cambridge: Cambridge Universtity Press, 2001.

[31] Xu X. On bisimulation theory in linear higher-order π-calculus. Trans. on Petri Nets and Other Models of Concurrency III, 2009, 5800:244−274. [doi: 10.1007/978-3-642-04856-2_10]

[32] Xu X. On the bisimulation theory and axiomatization of higher-order process calculi [Ph.D Thesis]. Shanghai: Shanghai Jiaotong University, 2008.

[33] Frauenstein T, Baldamus M, Glas R. Congruence proofs for weak bisimulation on higher-order processes: Results for typed omega-order calculi. Technical Report, 96-19, Berlin University of Technology, 1996. 3−68.

[34] Baldamus M, Dingel J. Modal characterization of weak bisimulation for higher-order processes. In: Bidoit M, Dauchet M, eds. Proc. of the TAPSOFT'97. LNCS 1214, Springer-Verlag, 1997. 285−296. [doi: 10.1007/BFb0030604]

[35] Sangiorgi D, Kobayashi N, Sumii E. Environmental bisimulations for higher-order languages. ACM Trans. on Programming Languages and Systems, 2011,33(1):Article 5. [doi: 10.1145/1889997.1890002]

[36] Milner R. Communication and Concurrency. Prentice Hall, 1989.

[37] Rivest R, Shamir A, Adleman L. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 1978,21(2):120−126. [doi: 10.1145/359340.359342]

[38] Sangiorgi D, Milner R. The problem of weak bisimulation up-to. In: Cleaveland WR, ed. Proc. of the CONCUR'92. LNCS 630, Springer-Verlag, 1992. 32−46. [doi: 10.1007/BFb0084781]

[39] The Coq Development Team. The Coq Proof Assistant (Reference Manual). 2013.

附中文参考文献:

[3] 杨博,黄晶,刘大有.移动 agent 计算理论和形式化方法研究.计算机研究与发展,2006,43(增刊):274−278.

[14] 邓玉欣.完全π演算的符号化开语义[硕士学位论文].上海:上海交通大学,2001.

[16] 袁文杰,应时,吴可嘉,姚俊峰.基于π演算的反射式需求规约描述方法.计算机工程与科学,2010,32(6):146−154.

[17] 尤涛,杜承烈,王小伟,郑炜.基于高阶时间π演算的构件式实时软件研究.西北工业大学学报,2009,27(6):6−11.

[18] 李长云,李赣生,何频捷.一种形式化的动态体系结构描述语言.软件学报,2006,17(6):1349−1359. http://www.jos.org.cn/1000-9825/17/1349.htm

[19] 高俊,沈才梁,郑美芳,李长云.一种面向自适应软件系统的体系结构描述语言.计算机应用研究,2010,27(5):1796−1801.

徐贤(1979－),男,上海人,博士,副教授,主要研究领域为并发理论及其应用.

E-mail: xuxian@ecust.edu.cn