

Bisimulation Lattice of Chi Processes^{*}

Yuxi Fu^{**}

Department of Computer Science
Shanghai Jiaotong University, Shanghai 200030, China

Abstract. Chi calculus was proposed as a process algebra that has a uniform treatment of names. The paper carries out a systematic study of bisimilarities for chi processes. The notion of L-bisimilarity is introduced to give a possible classification of bisimilarities on chi processes. It is shown that the set of L-bisimilarities forms a four element lattice and that well-known bisimilarities for chi processes fit into the lattice hierarchy. The four distinct L-bisimilarities give rise to four congruence relations. Complete axiomatization system is given for each of the four relations. The bisimulation lattice of asynchronous chi processes and that of asymmetric chi processes are also investigated. It turns out that the former consists of two elements while the latter twelve elements. Finally it is pointed out that the asynchronous asymmetric chi calculus has a bisimulation lattice of eight elements.

The χ -calculus ([4]) was introduced with two motivations in mind. One is to remove the *ad hoc* nature of prefix operation in π -calculus ([11]) by having a uniform treatment of names ([3]), thus arriving at a conceptually simpler language. The second is to materialize a communication-as-cut-elimination viewpoint ([5]), therefore taking up a proof theoretical approach to concurrency theory, an approach that has been proved very fruitful in the functional world. Independently Parrow and Victor have come up with essentially the same language, Update Calculus as they term it ([13]). They share, we believe, the first motivation but have quite a different second one originated from concurrent constraint programming. The difference between π and χ lies mainly in the way communications happen. The former adopts the familiar value-passing mechanism whereas the latter takes an information exchange or information update viewpoint. The algebraic theory of the language has been investigated in the above mentioned papers. Parrow and Victor have looked into strong bisimilarity and axiomatization of it for Update Calculus, while Fu has examined an observational bisimilarity for χ -processes. More recently, Parrow and Victor have proposed Fusion Calculus ([14, 17]), which is a polyadic version of χ -calculus. The authors have also studied an observational equivalence called weak hyperbisimilarity. What we know about the language, albeit little, tells us that it can practically do everything π can do and its algebraic properties are just as satisfactory. The studies carried out so far are however preliminary.

^{*} *ASIAN '98*, Lecture Notes in Computer Science 1538, 245-262, 1998.

^{**} Supported by the National Nature Science Foundation of China.

The objective of this paper is to continue our examination of the algebraic theory of χ -calculus. Section 1 reviews the operational semantics of χ . Section 2 defines L -bisimilarities and investigates their relationship. Section 3 gives alternative characterizations of L -bisimilarities. Section 4 presents a complete axiomatization system for each of the congruence relations induced by the four L -bisimilarities. The next three sections look into the L -bisimilarities of asynchronous, asymmetric, asynchronous asymmetric χ -calculi respectively.

1 Operational Semantics

In π -calculus there are two kinds of closed names, one has dummy names as x in $m(x).P$ and local names as x in $(x)\overline{m}x.P$. In simple words, the χ -calculus is obtained from π -calculus by unifying these names. This identification forces a unification of input and output prefix operations. The two π -processes just mentioned then turn into $(x)m[x].P$ and $(x)\overline{m}[x].P$ respectively. In the resulting calculus communications are completely symmetric as exemplified by the following reductions:

$$\begin{aligned} & \overline{m}[x].P|m[x].Q \xrightarrow{\tau} P|Q \\ (x)(R|(\overline{m}[y].P|m[x].Q)) & \xrightarrow{\tau} R[y/x]|(P[y/x]|Q[y/x]), \text{ where } y \neq x \\ (x)\overline{m}[x].P|(y)m[y].Q & \xrightarrow{\tau} (z)(P[z/x]|Q[z/y]), \text{ where } z \text{ is fresh} \end{aligned}$$

The reader is referred to [3–6, 13, 14, 17] for more explanations and examples.

Let \mathcal{N} be a set of names, ranged over by lower case letters. $\overline{\mathcal{N}}$, the set of conames, denotes $\{\overline{x} \mid x \in \mathcal{N}\}$. The following conventions will be used: α ranges over $\mathcal{N} \cup \overline{\mathcal{N}}$, μ over $\{\tau\} \cup \{\alpha[x], \alpha x \mid x \in \mathcal{N}\}$, and δ over $\{\tau\} \cup \{\alpha[x], \alpha x, [y/x] \mid x, y \in \mathcal{N}\}$. The set \mathcal{C} of χ -processes are defined by BNF as follows:

$$P := \mathbf{0} \mid \alpha[x].P \mid P|P \mid (x)P \mid [x=y]P \mid P+P$$

The process $\alpha[x].P$ is in prefix form. Here α or $\overline{\alpha}$ is the subject name, and x the object name, of the prefix. The composition operator “|” is standard. In $(x)P$ the name x is declared local; it cannot be seen from outside. The set of global names, or nonlocal names, in P is denoted by $gn(P)$. We will adopt the α -convention saying that a local name in a process can be replaced by a fresh name without changing the syntax of the process. The choice combinator ‘+’ is well-known. The process $P+Q$ acts either as P or as Q exclusively. In this paper we leave out the replication operator. The result of this paper would not be affected had it been included.

The operational semantics can be defined either by reduction semantics ([4]) or in terms of a labeled transition system ([3]). Here we opt for a pure transition semantics as it helps to present our results with clear-cut proofs. The labeled transition system given below defines an early semantics. The reason to use an early semantics is that the definition of weak bisimulation is more succinct in

early semantics than in late semantics. In the following formulation, symmetric rules are systematically omitted:

$$\begin{array}{c}
\frac{}{\alpha[x].P \xrightarrow{\alpha[x]} P} \text{Sqn} \quad \frac{P \xrightarrow{\delta} P'}{[x=x]P \xrightarrow{\delta} P'} \text{Cnd} \quad \frac{P \xrightarrow{\delta} P'}{P+Q \xrightarrow{\delta} P'} \text{Sum} \\
\\
\frac{P \xrightarrow{\mu} P'}{P|Q \xrightarrow{\mu} P'|Q} \text{Cmp}_0 \quad \frac{P \xrightarrow{[y/x]} P'}{P|Q \xrightarrow{[y/x]} P'|Q[y/x]} \text{Cmp}_1 \\
\\
\frac{P \xrightarrow{\alpha x} P' \quad Q \xrightarrow{\bar{\alpha}[x]} Q'}{P|Q \xrightarrow{\tau} P'|Q'} \text{Cmm}_0 \quad \frac{P \xrightarrow{\alpha x} P' \quad Q \xrightarrow{\bar{\alpha}x} Q' \quad x \notin \text{gn}(P|Q)}{P|Q \xrightarrow{\tau} (x)(P|Q')} \text{Cmm}_1 \\
\\
\frac{P \xrightarrow{\alpha[x]} P' \quad Q \xrightarrow{\bar{\alpha}[x]} Q'}{P|Q \xrightarrow{\tau} P'|Q'} \text{Cmm}_2 \quad \frac{P \xrightarrow{\alpha[x]} P' \quad Q \xrightarrow{\bar{\alpha}[y]} Q' \quad x \neq y}{P|Q \xrightarrow{[y/x]} P'[y/x]|Q'[y/x]} \text{Cmm}_3 \\
\\
\frac{P \xrightarrow{\delta} P' \quad x \notin n(\delta)}{(x)P \xrightarrow{\delta} (x)P'} \text{Loc}_0 \quad \frac{P \xrightarrow{\alpha[x]} P' \quad x \notin \{\alpha, \bar{\alpha}\}}{(x)P \xrightarrow{\alpha y} P'[y/x]} \text{Loc}_1 \quad \frac{P \xrightarrow{[y/x]} P'}{(x)P \xrightarrow{\tau} P'} \text{Loc}_2
\end{array}$$

Labeled transitions of the form $\xrightarrow{[y/x]}$, called update transitions, are first introduced in [3, 13] to help define communications in a transition semantics. In applying Loc_1 local names need be renamed if necessary to prevent y from being captured. In Loc_0 , $n(\delta)$ denotes the set of names appeared in δ . The notation $[y/x]$ occurred in $P[y/x]$ for example is an atomic substitution of y for x . A general substitution σ is the composition of atomic substitutions, whose effect is defined by $P[y_1/x_1] \dots [y_n/x_n] \stackrel{\text{def}}{=} (P[y_1/x_1] \dots [y_{n-1}/x_{n-1}])[y_n/x_n]$. The composition of zero atomic substitution is an empty substitution $[]$ whose effect is vacuous.

The next lemma collects some technical results whose proofs are simple inductions on derivation.

Lemma 1. (i) If $P \xrightarrow{\mu} P'$ then $P\sigma \xrightarrow{\mu\sigma} P'\sigma$.

(ii) If $P \xrightarrow{[y/x]} P'$ and $x\sigma \neq y\sigma$ then $P\sigma \xrightarrow{[y\sigma/x\sigma]} P'\sigma[y\sigma/x\sigma]$.

(iii) If $P \xrightarrow{[y/x]} P'$ and $x\sigma = y\sigma$ then $P\sigma \xrightarrow{\tau} P'\sigma$.

(iv) If $P \xrightarrow{[y/x]} P'$ then $P \xrightarrow{[x/y]} P'[x/y]$.

(v) $P \xrightarrow{\alpha x} P'$ if and only if $P \xrightarrow{\alpha z} P_1$ for some fresh z such that $P' \equiv P_1[x/z]$.

(vi) Suppose $a \notin \text{gn}(P)$. If $(x)(P|a[x]) \xrightarrow{\tau} \alpha y \rightarrow P'$ then $(x)(P|a[x]) \xrightarrow{\alpha y} \xrightarrow{\tau} P'$.

(vii) Suppose $a \notin \text{gn}(P)$. If $(x)(P|a[x]) \xrightarrow{\tau} P'|a[y]$ then $P \xrightarrow{[y/x]} P'$.

Let \Longrightarrow be the reflexive and transitive closure of $\xrightarrow{\tau}$. We will write $\xRightarrow{\mu}$ ($\xRightarrow{\delta}$) for $\Longrightarrow \xrightarrow{\mu} \Longrightarrow$ ($\Longrightarrow \xrightarrow{\delta} \Longrightarrow$). We will also write $\xRightarrow{\widehat{\mu}}$ ($\xRightarrow{\widehat{\delta}}$) for $\xRightarrow{\mu}$ ($\xRightarrow{\delta}$) if $\mu \neq \tau$ ($\delta \neq \tau$) and for \Longrightarrow otherwise. A sequence of names x_1, \dots, x_n will be abbreviated to \mathbf{x} ; and consequently $(x_1) \dots (x_n)P$ will be abbreviated to $(\mathbf{x})P$. When the length of \mathbf{x} is zero, $(\mathbf{x})P$ is just P .

2 Bisimulation Lattice

We introduce in this section L -bisimilarities, which are refinement of the local bisimilarity of [4]. The reason to study L -bisimilarities is that they provide a framework to understand bisimilarity relations of interest.

In a symmetric calculus such as χ , it does not make much sense to say that an action with positive, respectively negative, subject name is an input, respectively output, action. An action is an input or output, depending on if the object name is being received or being sent out. Let o denote the set $\{a[x] \mid a, x \in \mathcal{N}\}$ of output actions, \bar{o} the set $\{\bar{a}[x] \mid a, x \in \mathcal{N}\}$ of co-output actions, i the set $\{ax \mid a, x \in \mathcal{N}\}$ of input actions, \bar{i} the set $\{\bar{a}x \mid a, x \in \mathcal{N}\}$ of co-input actions and u the set $\{[y/x] \mid x, y \in \mathcal{N}\}$ of updates. Let \mathcal{L} stand for $\{\cup S \mid S \subseteq \{o, \bar{o}, i, \bar{i}, u\} \wedge S \neq \emptyset\}$.

Definition 2. Let \mathcal{R} be a binary symmetric relation on \mathcal{C} and let L be an element of \mathcal{L} . The relation \mathcal{R} is an L -bisimulation if whenever $P\mathcal{R}Q$ then for any process R and any sequence \mathbf{x} of names it holds that if $(\mathbf{x})(P|R) \xrightarrow{\phi} P'$ for $\phi \in L \cup \{\tau\}$ then there exists some Q' such that $(\mathbf{x})(Q|R) \xrightarrow{\hat{\phi}} Q'$ and $P'\mathcal{R}Q'$. The L -bisimilarity \approx_L is the largest L -bisimulation.

This is a uniform definition of 31 L -bisimilarities. The intuition behind is that \approx_L is what an observer recognizes if he/she is capable of observing actions in L and only in L . We will show that the L -bisimilarities collapse to four distinct relations. In the rest of this section let L be an arbitrarily fixed element of \mathcal{L} . First we establish a few technical lemmas. The next one follows directly from definition.

Lemma 3. If $P \Longrightarrow P_1 \approx_L Q$ and $Q \Longrightarrow Q_1 \approx_L P$ then $P \approx_L Q$.

For $\phi \in L$, let $\langle \phi \rangle$ be a process such that (i) $\langle \phi \rangle \xrightarrow{\phi} \mathbf{0}$ and (ii) if $\langle \phi \rangle \xrightarrow{\phi} A$ then $A \equiv \mathbf{0}$.

Lemma 4. Suppose $a \notin gn(P|Q)$. Then (i) $(x)(P|a[x]) \approx_L (x)(Q|a[x])$ implies $P \approx_L Q$; and (ii) $P|a[x] \approx_L Q|a[x]$ implies $P \approx_L Q$.

Proof. (i) Suppose $\phi \in L$ and $n(\phi) \cap gn(P|Q) = \emptyset$. As $(x)(P|a[x])|\bar{a}[x].\langle \phi \rangle \xrightarrow{\phi} (P|\mathbf{0})|\mathbf{0}$, Q_1 exists such that $(x)(Q|a[x])|\bar{a}[x].\langle \phi \rangle \xrightarrow{\phi} (Q_1|\mathbf{0})|\mathbf{0} \approx_L (P|\mathbf{0})|\mathbf{0}$, which implies $(x)(Q|a[x]) \xrightarrow{ax} Q_1|\mathbf{0} \approx_L P|\mathbf{0}$, which in turn implies $Q \Longrightarrow Q_1$. Similarly P_1 exists such that $P \Longrightarrow P_1 \approx_L Q$. By Lemma 3, $P \approx_L Q$. (ii) can be proved similarly. \square

Lemma 5. If $P \approx_L Q$ then $P\sigma \approx_L Q\sigma$ for an arbitrary substitution σ .

Proof. Suppose $P \approx_L Q$. We only have to show that for $x \in gn(P|Q)$ and $y \neq x$ one has that $P[y/x] \approx_L Q[y/x]$. Let b be a distinct fresh name. Suppose $\phi \in L$ and $n(\phi) \cap gn(P|Q) = \emptyset$. By definition the actions $(x)(P|(\bar{b}[y]|b[x].\langle \phi \rangle)) \xrightarrow{\phi} P[y/x]|(\mathbf{0}|\mathbf{0})$ must be matched up by

$$(x)(Q|(\bar{b}[y]|b[x].\langle \phi \rangle)) \xrightarrow{\phi} Q_1|(\mathbf{0}|\mathbf{0}). \quad (1)$$

If $(x)(Q|(\bar{b}[y]|b[x].\langle\phi\rangle)) \xrightarrow{\tau} (x')(Q_2|(\bar{b}[y]|b[x'].\langle\phi\rangle))$ then by symmetry and α -convention the reduction is the same as

$$(x)(Q|(\bar{b}[y]|b[x].\langle\phi\rangle)) \xrightarrow{\tau} (x)(Q_3|(\bar{b}[y]|b[x].\langle\phi\rangle))$$

such that $Q \xrightarrow{\tau} Q_3$. It follows that (1) can be factorized as follows

$$\begin{aligned} (x)(Q|(\bar{b}[y]|b[x].\langle\phi\rangle)) &\Longrightarrow (x)(Q'|(\bar{b}[y]|b[x].\langle\phi\rangle)) \\ &\xrightarrow{\tau} Q'[y/x]|(\mathbf{0}|\langle\phi\rangle) \\ &\xrightarrow{\phi} Q_1|(\mathbf{0}|\mathbf{0}) \end{aligned}$$

for some Q' and Q_1 such that $Q \Longrightarrow Q'$ and $Q'[y/x] \Longrightarrow Q_1 \approx_L P[y/x]$. By Lemma 1, $Q[y/x] \Longrightarrow Q'[y/x]$. Similarly P_1 exists such that $P[y/x] \Longrightarrow P_1 \approx_L Q[y/x]$. By Lemma 3, $P[y/x] \approx_L Q[y/x]$. \square

By definition the L -bisimilarity is closed under localization and composition. Using Lemma 5 it can be easily seen that \approx_L is closed under prefix operation.

Theorem 6. *If $P \approx_L Q$ and $O \in \mathcal{C}$ then (i) $\alpha[x].P \approx_L \alpha[x].Q$; (ii) $P|O \approx_L Q|O$; (iii) $(x)P \approx_L (x)Q$; and (iv) $[x=y]P \approx_L [x=y]Q$.*

We investigate next the order structure of L -bisimilarities.

Theorem 7. *The following properties hold of the L -bisimilarities:*

(i) $\approx_o \not\subseteq \approx_{\bar{o}}$; $\approx_{\bar{o}} \not\subseteq \approx_o$. (ii) $\approx_L \subseteq \approx_u$. (iii) $\approx_L \subseteq \approx_i = \approx_{\bar{i}}$.

Proof. (i) It is obvious that $(x)a[x].(b)(\bar{b}[x]|b[z]) \not\approx_o a[z]+(x)a[x].(b)(\bar{b}[x]|b[z])$. It takes a while to see that $(x)a[x].(b)(\bar{b}[x]|b[z]) \approx_{\bar{o}} a[z]+(x)a[x].(b)(\bar{b}[x]|b[z])$.

(ii) To prove $\approx_L \subseteq \approx_u$, one only has to show that if $P \approx_L Q$ and $P \xrightarrow{[y/x]} P'$ then Q' exists such that $Q \xrightarrow{[y/x]} Q'$ and $P' \approx_L Q'$. Now $P \xrightarrow{[y/x]} P'$ implies that $(x)(P|a[x]) \xrightarrow{\tau} P'|a[y]$ for a fresh a . So $(x)(Q|a[x]) \Longrightarrow Q'|a[y]$ for some Q' such that $P'|a[y] \approx_L Q'|a[y]$. It follows from Lemma 4 that $P' \approx_L Q'$. Clearly $(x)(Q|a[x]) \Longrightarrow Q'|a[y]$ can be factorized as

$$(x)(Q|a[x]) \Longrightarrow (x)(Q_1|a[x]) \xrightarrow{\tau} Q_2|a[y] \Longrightarrow Q'|a[y],$$

where $Q \Longrightarrow Q_1$. By (vii) of Lemma 1, $(x)(Q_1|a[x]) \xrightarrow{\tau} Q_2|a[y]$ implies $Q_1 \xrightarrow{[y/x]} Q_2$. Hence $Q \xrightarrow{[y/x]} Q'$.

(iii) Assume $P \approx_L Q$ and $P \xrightarrow{\alpha x} P'$. Suppose $\phi \in L$ and $n(\phi) \cap gn(P|Q) = \emptyset$. Now $P|(\bar{\alpha}[z]+\langle\phi\rangle) \xrightarrow{\tau} P_1|\mathbf{0}$ for some fresh z such that $P' \equiv P_1[x/z]$. There has to be some Q_1 such that $Q|(\bar{\alpha}[z]+\langle\phi\rangle) \Longrightarrow Q_1|\mathbf{0} \approx_L P_1|\mathbf{0}$. So $Q \xrightarrow{\alpha z} Q_1 \approx_L P_1$. Therefore $Q \xrightarrow{\alpha x} Q_1[x/z] \approx_L P_1[x/z] \equiv P'$ by Lemma 1 and Lemma 5. Hence $\approx_L \subseteq \approx_i = \approx_{\bar{i}}$. \square

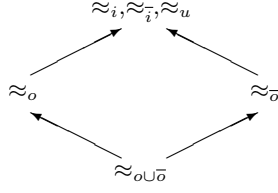


Fig. 1. The Bisimulation Lattice of Chi Processes

In the proof of (iii) of the above theorem, we need to use a fresh name z because we cannot conclude $R \xrightarrow{\alpha x} R'$ from $R[\bar{\alpha}[x] \xrightarrow{\tau} R'|\mathbf{0}$. It may well be that R participates in the communication by performing $R \xrightarrow{\alpha[x]} R'$.

Let $\mathcal{L}(\chi)$ be $\{\approx_L \mid L \in \mathcal{L}\}$, the set of all L -bisimilarities. $\mathcal{L}(\chi)$ is a partial order when equipped with \subseteq . For $\approx_{L_1}, \approx_{L_2} \in \mathcal{L}(\chi)$, $\approx_{L_1 \cup L_2}$ is the infimum. Theorem 7 says that $\mathcal{L}(\chi)$ is a four element lattice. The diagram in Fig. 1 is a pictorial representation of the lattice. In the diagram each node is the principal representative of a number of L -bisimilarities that boil down to a same relation. An arrow indicates a strict inclusion. The bottom element is represented by $\approx_{o\cup\bar{o}}$ while the top element is by $\approx_i = \approx_{\bar{i}} = \approx_u$. We will call $(\mathcal{L}(\chi), \subseteq)$ the bisimulation lattice of χ -processes. The lattice structure suggests that the ability to observe output actions is stronger than that to observe input actions. One way to understand this is that the effect of an output action is unknown whereas that of an input action has already been delimited.

3 Alternative Characterization

The definition of L -bisimilarity is natural but intractable. It contains universal quantifications over both processes and names. In this section alternative characterizations of the four distinct L -bisimilarities are presented. For each of the relations, an open style bisimilarity is shown to coincide with it. In the alternative definitions, one still has a universal quantification over substitutions. But it is clear that one only has to consider a finite number of them at each step. First we will see how barbed bisimilarity fits into the lattice hierarchy.

3.1 Barbed Bisimilarity

Barbed bisimilarity ([12]) seems to be the weakest bisimulation equivalence proposed so far. It applies to a whole range of process calculi and therefore acts as a convenient tool to study relationships between different calculi. The relationship of barbed bisimilarity to other bisimilarities is itself an interesting question. Usually it is easy to show that a given bisimilarity is included in the barbed one. The difficulty is in deciding if the inclusion is strict. This section answers the question for χ -processes.

Definition 8. A process P is strongly barbed at a , notation $P \downarrow a$, if $P \xrightarrow{\alpha x} P'$ or $P \xrightarrow{\alpha[x]} P'$ for some P' such that $a \in \{\alpha, \bar{\alpha}\}$. P is barbed at a , notation $P \Downarrow a$, if some P' exists such that $P \Longrightarrow P' \downarrow a$. A binary relation \mathcal{R} is barbed if $\forall a \in \mathcal{N}. P \Downarrow a \Leftrightarrow Q \Downarrow a$ whenever PRQ .

Definition 9. Let \mathcal{R} be a barbed symmetric relation on \mathcal{C} . It is called a barbed bisimulation if whenever PRQ then for any R and any sequence \mathbf{x} of names it holds that if $(\mathbf{x})(P|R) \xrightarrow{\tau} P'$ then Q' exists such that $(\mathbf{x})(Q|R) \Longrightarrow Q'$ and $P'\mathcal{R}Q'$. The barbed bisimilarity \approx_b is the largest barbed bisimulation.

The next result locates \approx_b in the bisimulation lattice.

Theorem 10. \approx_b is the same as \approx_i .

Proof. As \approx_i is clearly barbed, $\approx_i \subseteq \approx_b$. To prove the reverse inclusion, first notice that \approx_b is closed under substitution, the proof of which is similar to that of Lemma 5. Now suppose $P \approx_b Q$ and $P \xrightarrow{\alpha x} P'$. By (v) of Lemma 1 $P \xrightarrow{\alpha z} P_1$ for some fresh z such that $P' \equiv P_1[x/z]$. Let a be fresh. Now $P|(\bar{\alpha}[z].a[a]|\bar{a}[a]) \xrightarrow{\tau} P_1|(\mathbf{0}|\mathbf{0})$. This sequence of reductions must be matched up by $Q|(\bar{\alpha}[z].a[a]|\bar{a}[a]) \Longrightarrow Q_1|(\mathbf{0}|\mathbf{0}) \approx_b P_1|(\mathbf{0}|\mathbf{0})$. There are only two ways for Q to evolve into Q_1 : either $Q \xrightarrow{\alpha[z]} Q_1$ or $Q \xrightarrow{\alpha z} Q_1$. The former is impossible because z is fresh. Hence $Q \xrightarrow{\alpha z} Q_1 \approx_b P_1$. It follows from Lemma 1 that $Q \xrightarrow{\alpha x} Q_1[x/z] \approx_b P_1[x/z] \equiv P'$. Conclude that $\approx_b \subseteq \approx_i$. \square

3.2 Open Bisimilarity

Open bisimilarity is proposed by Sangiorgi in [16] for π -processes as a ‘‘correction of late bisimulation’’. This section defines open bisimilarity for χ -processes and relates it to one of the L -bisimilarities.

Definition 11. Let \mathcal{R} be a binary symmetric relation on \mathcal{C} . It is called an open bisimulation if whenever PRQ then for any substitution σ it holds that if $P\sigma \xrightarrow{\delta} P'$ then Q' exists such that $Q\sigma \xrightarrow{\hat{\delta}} Q'$ and $P'\mathcal{R}Q'$. The open bisimilarity \approx_{open} is the largest open bisimulation.

It is clear from the definition that \approx_{open} is closed under substitution.

Lemma 12. \approx_{open} is closed under localization and composition.

This easy lemma can be proved by constructing appropriate bisimulations.

Theorem 13. \approx_{open} coincides with $\approx_{o \cup \bar{o}}$.

Proof. By Lemma 5 and the proof of Theorem 7, $(o \cup \bar{o})$ -bisimilarity is an open bisimulation. By Lemma 12, open bisimilarity is an $(o \cup \bar{o})$ -bisimulation. \square

The definition of open bisimilarity makes it easy to give axiomatization system for finite χ -processes. Theorem 13 enables us to axiomatize the bottom element of the bisimulation lattice $\mathcal{L}(\chi)$. In order to do the same for the other three elements of the bisimulation lattice, one would like to characterize these three L -bisimilarities in an ‘open’ style, so to speak. This is precisely what we are going to do next. The intuition for the following definition comes from the example given in the proof of (i) of Theorem 7.

Definition 14. Let \mathcal{R} be a binary symmetric relation on \mathcal{C} . It is called an *open i -bisimulation* (open o -bisimulation, open \bar{o} -bisimulation) if whenever PRQ then for any substitution σ it holds that

(i) if $P\sigma \xrightarrow{\phi} P'$ for $\phi \in i \cup \bar{i} \cup u \cup \{\tau\}$ ($\phi \in o \cup i \cup \bar{i} \cup u \cup \{\tau\}$, $\phi \in \bar{o} \cup i \cup \bar{i} \cup u \cup \{\tau\}$)

then Q' exists such that $Q\sigma \xrightarrow{\hat{\phi}} Q'$ and $P'\mathcal{R}Q'$; and

(ii) if $P\sigma \xrightarrow{\alpha[x]} P'$ ($P\sigma \xrightarrow{\bar{\alpha}[x]} P'$, $P\sigma \xrightarrow{a[x]} P'$) then some Q' exists such that $P'\mathcal{R}Q'$ and either $Q\sigma \xrightarrow{\alpha[x]} Q'$ ($Q\sigma \xrightarrow{\bar{\alpha}[x]} Q'$, $Q\sigma \xrightarrow{a[x]} Q'$) or $Q\sigma \xrightarrow{\alpha z} \xrightarrow{[x/z]} Q'$ ($Q\sigma \xrightarrow{\bar{\alpha} z} \xrightarrow{[x/z]} Q'$, $Q\sigma \xrightarrow{a z} \xrightarrow{[x/z]} Q'$) for some fresh z .

The open i -bisimilarity (open o -bisimilarity, open \bar{o} -bisimilarity), denoted by \approx_{open}^i (\approx_{open}^o , $\approx_{open}^{\bar{o}}$), is the largest open i -bisimulation (open o -bisimulation, open \bar{o} -bisimulation).

The next theorem explains the reason to have Definition 14.

Theorem 15. (i) $\approx_{open}^o = \approx_o$; (ii) $\approx_{open}^{\bar{o}} = \approx_{\bar{o}}$; (iii) $\approx_{open}^i = \approx_i$.

Proof. It is enough to see how to establish (iii). Suppose $P \approx_i Q$ and $P \xrightarrow{\alpha[x]} P'$. For a fresh name z one has $P|\bar{\alpha}[z] \xrightarrow{[x/z]} P'|\mathbf{0}$. As $\approx_i = \approx_u$, Q' exists such that $Q|\bar{\alpha}[z] \xrightarrow{[x/z]} Q'|\mathbf{0} \approx_i P'|\mathbf{0}$. There are only two possibilities: either $Q \xrightarrow{\alpha[x]} Q' \approx_i P'$ or $Q \xrightarrow{\alpha z} \xrightarrow{[x/z]} Q' \approx_i P'$. Hence $\approx_i \subseteq \approx_{open}^i$ by Lemma 5.

To prove the reverse inclusion we use the fact that \approx_{open}^i is by definition closed under substitution. We only have to show that the composition and localization operators preserve the relation \approx_{open}^i . To prove that $\{((\mathbf{v})(P|R), (\mathbf{v})(Q|R)) \mid P \approx_{open}^i Q\}$ is an open i -bisimulation, it is sufficient to examine the cases where P performs an output action. We consider only one case:

- Suppose that $(\mathbf{v})(P|R) \xrightarrow{\alpha z} (\mathbf{u})(P'[z/y]|R[z/y])$, where $y \in \{\mathbf{v}\}$, is derived from $P \xrightarrow{\alpha[y]} P'$ by applying rule Loc_1 . Suppose $P \xrightarrow{\alpha[y]} P'$ is matched up by $Q \xrightarrow{\alpha w} Q_1 \xrightarrow{[y/w]} Q'$ for some Q_1 , Q' and fresh w . By Lemma 1, $Q \xrightarrow{\alpha z} Q_1[z/w] \xrightarrow{[y/z]} Q'[z/w][y/z] \equiv Q'[y/z]$ as $w \notin \text{gn}(Q')$. By (iv) of Lemma 1, $Q \xrightarrow{\alpha z} Q_1[z/w] \xrightarrow{[z/y]} Q'[y/z][z/y] \equiv Q'[z/y]$. It follows that

$$(\mathbf{v})(Q|R) \xrightarrow{\alpha z} (\mathbf{v})(Q_1[z/w]|R) \Longrightarrow (\mathbf{u})(Q'[z/y]|R[z/y])$$

matches up $(\mathbf{v})(P|R) \xrightarrow{\alpha z} (\mathbf{u})(P'[z/y]|R[z/y])$.

The closure property enables us to conclude that \approx_{open}^i is an i -bisimulation. \square

The open style bisimilarities are particularly suitable for χ -like process calculi in which names are uniform.

4 Axiomatization

The L -bisimilarities are not congruence relations. To get a congruence relation from \approx_L we use the standard approach.

Definition 16. $P =_L Q$ if and only if $P \approx_L Q$ and whenever $P \xrightarrow{\tau} P'$ ($Q \xrightarrow{\tau} Q'$) then Q' (P') exists such that $Q \xrightarrow{\tau} Q'$ ($P \xrightarrow{\tau} P'$) and $P' \approx_L Q'$.

We will call $=_L$ the L -congruence on χ -processes. According to the bisimulation lattice, there are only four distinct L -congruence relations. The aim of this section is to give complete axiomatization systems for all the four distinct L -congruence relations. In this section we omit most of the proofs since they are very much similar to the corresponding proofs in [7, 10]. In the axiomatization we need a prefix operation that generalizes the standard τ -operation. It is defined as follows:

$$[y|x].P \stackrel{\text{def}}{=} (a)(\bar{a}[y]|a[x].P) \text{ where } a \text{ is fresh.}$$

The next result is known as Hennessy Lemma. It is very useful in the study of axiomatization systems.

Lemma 17. $P \approx_L Q$ if and only if either $[x|x].P =_L Q$ or $P =_L Q$ or $P =_L [x|x].Q$.

For purpose of axiomatization, let π and γ range over $\{\alpha[x], [y|x] \mid x, y \in \mathcal{N}\}$. In the rest of this section M and N denote finite lists of match equalities $x=y$. Suppose M is $x_1=y_1, \dots, x_n=y_n$. Then $[M]P$ denotes $[x_1=y_1] \dots [x_n=y_n]P$. If M logically implies N , we write $M \Rightarrow N$; and if both $M \Rightarrow N$ and $N \Rightarrow M$ we write $M \Leftrightarrow N$. If M is an empty list, it plays the role of logical truth, in which case $[M]P$ is just P . Clearly a list M of match equalities defines an equivalence relation on the set $n(M)$ of names appeared in M . We use σ_M to denote an arbitrary substitution that replaces all members of an equivalence class by a representative of that class.

In [16] Sangiorgi presents a complete system for strong open congruence on π -processes. Parrow and Victor consider in their paper ([13]) two axiomatization systems for strong hyperbisimilarity, one with match operator and one without match operator. In Fig. 2 a conditional equational system AS is defined together with some derived rules and their justifications. The expansion law is the following equation:

$$P|Q = \sum_i [M_i](\mathbf{x}) \pi_i.(P_i|Q) + \sum_{\substack{\pi_i = \alpha_i[x_i] \\ \gamma_j = b_j[y_j]}} [M_i][N_j](\mathbf{x})(\mathbf{y})[a_i=b_j][x_i|y_j].(P_i|Q_j) + \sum_j [N_j](\mathbf{y}) \gamma_j.(P|Q_j) + \sum_{\substack{\pi_i = \alpha_i[x_i] \\ \gamma_j = b_j[y_j]}} [M_i][N_j](\mathbf{x})(\mathbf{y})[a_i=b_j][x_i|y_j].(P_i|Q_j)$$

L1	$(x)\mathbf{0} = \mathbf{0}$	
L2	$(x)\alpha[y].P = \mathbf{0}$	$x \in \{\alpha, \bar{\alpha}\}$
L3	$(x)\alpha[y].P = \alpha[y].(x)P$	$x \notin \{y, \alpha, \bar{\alpha}\}$
L4	$(x)(y)P = (y)(x)P$	
L5	$(x)[y=z]P = [y=z](x)P$	$x \notin \{y, z\}$
L6	$(x)(P+Q) = (x)P+(x)Q$	
L7	$(x)[x=y]P = \mathbf{0}$	
L8	$(x)[y x].P = [y y].P[y/x]$	$x \neq y$
L9	$(x)[y z].P = [y z].(x)P$	$x \notin \{y, z\}$
M1	$[M]P = [N]P$	if $M \Leftrightarrow N$
M2	$[x=y]P = [x=y]P[y/x]$	
M3	$[x=y](P+Q) = [x=y]P+[x=y]Q$	
S1	$P+\mathbf{0} = P$	
S2	$P+Q = Q+P$	
S3	$P+(Q+R) = (P+Q)+R$	
S4	$[x=y]P+P = P$	
U1	$[y x].P = [x y].P$	
U2	$[x x].P = [y y].P$	
U3	$[y x].P = [y x].[x=y]P$	
<i>Expansion Law</i>		
LD1	$(x)[x x].P = [y y].(x)P$	U2 and L9
MD1	$[x=y].\mathbf{0} = \mathbf{0}$	S1 and S4
MD2	$[x=x].P = P$	M1
MD3	$[M]P = [M](P\sigma_M)$	M2
SD1	$P+P = P$	MD2 and S4
SD2	$[M]P+P = P$	S-rules
UD1	$[y x].P = [y x].P[y/x]$	U3 and M2

Fig. 2. Axiom System AS and its Derived Rules

where P is $\sum_i [M_i](x)\pi_i.P_i$ and Q is $\sum_j [N_j](y)\gamma_j.Q_j$. Rules concerning equivalence and congruence relations have been omitted. This is Victor and Parrow's system without mismatch. The only difference is that we use a symmetric update prefix operator.

We will write $AS \cup \{R_1, \dots, R_n\} \vdash P = Q$ to mean that the equality $P = Q$ is derivable from the rules and axioms of AS together with rules R_1, \dots, R_n . When no confusion arises, we simply write $P = Q$. We will also write $P \stackrel{R}{=} Q$ to indicate that R is the major axiom applied to derive $P = Q$.

Definition 18. A process P is in normal form if $P \equiv \sum_{i \in I_1} [M_i]\alpha_i[x_i].P_i + \sum_{i \in I_2} [M_i](x)\alpha_i[x].P_i + \sum_{i \in I_3} [M_i][z_i|y_i].P_i$ such that x does not appear in P and P_i is in normal form for each $i \in I_1 \cup I_2 \cup I_3$. Here I_1, I_2 and I_3 are pairwise disjoint finite indexing sets.

Notice that if P is in normal form and σ is a substitution then $P\sigma$ is in normal form. The depth of a process measures the maximal length of nested prefixes in the process. The structural definition goes as follows: (i) $d(\mathbf{0}) = 0$;

T1	$\alpha[x].[y y].P = \alpha[x].P$
T2	$P + [y y].P = [y y].P$
T3	$\alpha[x].(P + [y y].Q) = \alpha[x].(P + [y y].Q) + \alpha[x].Q$
TD1	$[x z].[y y].P = [x z].P$
TD2	$[x z].(P + [y y].Q) = [x z].(P + [y y].Q) + [x z].Q$
O1	$(z)m[z].(P + [x z].Q) = (z)m[z].(P + [x z].Q) + m[x].Q \quad x \neq z$
O2	$(z)\bar{m}[z].(P + [x z].Q) = (z)\bar{m}[z].(P + [x z].Q) + \bar{m}[x].Q \quad x \neq z$
O3	$(z)\alpha[z].(P + [x z].Q) = (z)\alpha[z].(P + [x z].Q) + \alpha[x].Q \quad x \neq z$

Fig. 3. The Tau Laws and the Output Laws

(ii) $d(\alpha[x].P) = 1 + d(P)$; (iii) $d(P|Q) = d(P) + d(Q)$; (iv) $d((x)P) = d(P)$; (v) $d([x=y]P) = d(P)$; (vi) $d(P+Q) = \max\{d(P), d(Q)\}$; (vii) $d([x|y].P) = 1 + d(P)$.

Lemma 19. *In AS each process P is provably equal to some P' in normal form such that $d(P') \leq d(P)$.*

We need the χ -version of the well-known tau laws as given in Fig. 3. TD1 and TD2 are derivable from T1 and T3 respectively.

The tau laws are enough to characterize $=_{o\cup\bar{o}}$. To get complete systems for the other three L -congruence relations, we need what we call output laws O1, O2 and O3 given in Fig. 3.

The following lemma is crucial to the proof of the completeness theorem.

Lemma 20. *Suppose Q is in normal form. Then*

- (i) if $Q\sigma_M \xrightarrow{\tau} Q'$ then $AS \cup \{T1, T2, T3\} \vdash Q = Q + [M][x|x].Q'$ for each x ;
- (ii) if $Q\sigma_M \xrightarrow{\alpha[x]} Q'$ then $AS \cup \{T1, T2, T3\} \vdash Q = Q + [M]\alpha[x].Q'$;
- (iii) if $z \notin gn(Q) \cup n(M)$ and $Q\sigma_M \xrightarrow{\alpha z} Q'$ then $AS \cup \{T1, T2, T3\} \vdash Q = Q + [M](z)\alpha[z].Q'$;
- (iv) if $Q\sigma_M \xrightarrow{[x|y]} Q'$ then $AS \cup \{T1, T2, T3\} \vdash Q = Q + [M][x|y].Q'$;
- (v) if $z \notin gn(Q) \cup n(M)$ and $Q\sigma_M \xrightarrow{mz} \xrightarrow{[x/z]} Q'$ then $AS \cup \{T1, T2, T3\} \cup \{O1\} \vdash Q = Q + [M]m[x].Q'$;
- (vi) if $z \notin gn(Q) \cup n(M)$ and $Q\sigma_M \xrightarrow{\bar{m}z} \xrightarrow{[x/z]} Q'$ then $AS \cup \{T1, T2, T3\} \cup \{O2\} \vdash Q = Q + [M]\bar{m}[x].Q'$;
- (vii) if $z \notin gn(Q) \cup n(M)$ and $Q\sigma_M \xrightarrow{\alpha z} \xrightarrow{[x/z]} Q'$ then $AS \cup \{T1, T2, T3\} \cup \{O3\} \vdash Q = Q + [M]\alpha[x].Q'$.

Proof. (i) through (iv) are proved by inductions on derivation. (v) through (vii) are proved using (iii) and (iv). \square

Theorem 21. *We have the following completeness results:*

- (i) $AS \cup \{T1, T2, T3\}$ is sound and complete for $=_{o\cup\bar{o}}$;
- (ii) $AS \cup \{T1, T2, T3\} \cup \{O1\}$ is sound and complete for $=_{\bar{o}}$;
- (iii) $AS \cup \{T1, T2, T3\} \cup \{O2\}$ is sound and complete for $=_o$;
- (iv) $AS \cup \{T1, T2, T3\} \cup \{O3\}$ is sound and complete for $=_i$.

Proof. The soundness part is clear.

(i) Suppose P and Q are in normal form and $P =_{o\cup\bar{o}} Q$. We prove the completeness by induction on the sum of the depths of P and Q . Let $[M_i](x)\alpha_i[x].P_i$ be a summand of P . Now $([M_i](x)\alpha_i[x].P_i)\sigma_{M_i} \xrightarrow{\alpha_i\sigma_{M_i}x} P_i\sigma_{M_i}$ must be matched up by $Q\sigma_{M_i} \xrightarrow{\alpha_i\sigma_{M_i}x} Q'$ such that $P_i\sigma_{M_i} \approx_{o\cup\bar{o}} Q'$. By Lemma 17, we know that either $[y|y].P_i\sigma_{M_i} =_{o\cup\bar{o}} Q'$ or $P_i\sigma_{M_i} =_{o\cup\bar{o}} Q'$ or $P_i\sigma_{M_i} =_{o\cup\bar{o}} [x|x].Q'$. Suppose $P_i\sigma_{M_i} =_{o\cup\bar{o}} [y|y].Q'$ is the case. Both $P_i\sigma_{M_i}$ and $[y|y].Q'$ are in normal form and $d(P_i\sigma_{M_i}) + d([y|y].Q') < d(P) + d(Q)$. So by induction hypothesis $P_i\sigma_{M_i} = [y|y].Q'$. It follows that

$$\begin{aligned} [M_i](x)\alpha_i[x].P_i &\stackrel{MD3}{=} [M_i](x)\alpha_i\sigma_{M_i}[x].P_i\sigma_{M_i} \\ &\stackrel{I.H.}{=} [M_i](x)\alpha_i\sigma_{M_i}[x].[y|y].Q' \\ &\stackrel{T1}{=} [M_i](x)\alpha_i\sigma_{M_i}[x].Q'. \end{aligned}$$

Hence $[M_i](x)\alpha_i[x].P_i + Q = [M_i](x)\alpha_i\sigma_{M_i}[x].Q' + Q = Q$ by (iii) of Lemma 20. The situation is the same when $P' =_{o\cup\bar{o}} Q'$ or $P' =_{o\cup\bar{o}} [x|x].Q'$. It can be similarly proved that $[M_i]\alpha_i[x_i].P_i + Q = Q$, respectively $[M_i][x_i|y_i].P_i + Q = Q$, whenever $[M_i]\alpha_i[x_i].P_i$, respectively $[M_i][x_i|y_i].P_i$, is a summand of P . Conclude that $P + Q = Q$. Symmetrically $P + Q = P$. Hence $P = Q$.

(iv) Let P and Q be in normal form and $P =_i Q$. Suppose

$$([M_i]\alpha_i[x_i].P_i)\sigma_{M_i} \xrightarrow{\alpha_i\sigma_{M_i}[x_i\sigma_{M_i}]} P_i\sigma_{M_i}$$

is matched up by $Q\sigma_{M_i} \xrightarrow{\alpha_i\sigma_{M_i}z} Q_1 \xrightarrow{[x_i\sigma_{M_i}/z]} Q'$ such that $P_i\sigma_{M_i} \approx_i Q'$. By Theorem 13 this is the only situation not covered by the proof of (i). By Lemma 17, either $[x|x].P_i\sigma_{M_i} =_i Q'$ or $P_i\sigma_{M_i} =_i Q'$ or $P_i\sigma_{M_i} =_i [x|x].Q'$. Now suppose $[x|x].P_i\sigma_{M_i} =_i Q'$ is the case. Both $[x|x].P_i\sigma_{M_i}$ and Q' are in normal form. So by induction hypothesis $[x|x].P_i\sigma_{M_i} = Q'$. Therefore

$$\begin{aligned} [M_i]\alpha_i[x_i].P_i &= [M_i]\alpha_i\sigma_{M_i}[x_i\sigma_{M_i}].P_i\sigma_{M_i} \\ &= [M_i]\alpha_i\sigma_{M_i}[x_i\sigma_{M_i}].Q'. \end{aligned}$$

Thus $[M_i]\alpha_i[x_i].P_i + Q = [M_i]\alpha_i\sigma_{M_i}[x_i\sigma_{M_i}].Q' + Q = Q$ by (vii) of Lemma 20. The rest of the proof can be safely omitted. \square

5 Asynchronous Chi Calculus

In the world of π -calculus, attention has been paid to an asynchronous version of the language ([8, 2, 1]). It has been argued that it has the same expressive power as the synchronous one. There is also a case for asynchronous χ -calculus. Merro has recently given a fully abstract translation from the asynchronous χ to the asynchronous π ([9]). One hopes to establish that χ and π have the same expressive power by relating χ to its asynchronous cosine. As one would expect the asynchronous χ is defined as follows:

$$P := \mathbf{0} \mid a[x].P \mid \bar{a}[x] \mid P|P \mid (x)P \mid [x=y]P \mid P+P$$

The operational semantics remains unchanged. So does the definition of L -bisimilarity. Lemma 5 and Theorem 6 still hold. But the bisimulation lattice shrinks.

Theorem 22. $\approx_o \subseteq \approx_{\bar{o}} = \approx_u = \approx_i = \approx_{\bar{i}}$. *The inclusion is strict.*

Proof. The proofs of the equivalence $\approx_u = \approx_i = \approx_{\bar{i}}$ and the strictness of the inclusion are as before. It remains to show that $\approx_L \subseteq \approx_{\bar{o}}$ for all L -bisimilarities \approx_L . Now suppose $P \approx_L Q$ and $P \xrightarrow{\bar{a}[x]} P'$. Let R be $a[x] + \langle \phi \rangle$ such that $\phi \in L$ and $n(\phi) \cap gn(P|Q) = \emptyset$. Then $P|R \xrightarrow{\tau} P'|\mathbf{0}$. It follows that Q' exists such that $Q|R \xrightarrow{\tau} Q|\mathbf{0}$ and $P' \approx_L Q'$. There are two cases: Either $Q \xrightarrow{\bar{a}[x]} Q'$ or $Q \xrightarrow{\bar{a}z} Q_2 \xrightarrow{[x/z]} Q'$ for some fresh z . In the latter case, notice that no action is causally dependent on the action $\bar{a}z$. So all the rest of the actions can be performed before the action at that particular \bar{a} happens. But then the update $[x/z]$ becomes a communication in which a local name is substantiated by x . It follows that the delayed action at that particular \bar{a} must be $\bar{a}[x]$. That is $Q \xrightarrow{\bar{a}z} Q_2 \xrightarrow{[x/z]} Q'$ can be rearranged as $Q \xrightarrow{\bar{a}[x]} Q'$. So in either case $P \xrightarrow{\bar{a}[x]} P'$ is matchable by $Q \xrightarrow{\bar{a}[x]} Q'$. \square

Consequently the bisimulation lattice of the asynchronous χ contains only two elements.

6 Asymmetric Chi Calculus

The simplicity of the lattice $\mathcal{L}(\chi)$ is due essentially to the symmetry of the language. If we insist that there is a difference between an action with a positive subject name and an action with a negative subject name then an asymmetric version of χ -calculus results. This is the Update Calculus of Parrow and Victor.

The operational semantics of the asymmetric χ -calculus is defined similarly as is for symmetric χ . The rule **Sqn**, **Cnd** and **Sum** remain unchanged. Other rules are given below:

$$\begin{array}{c}
\frac{P \xrightarrow{\mu} P' \quad \text{if } \mu = \bar{m}(x) \text{ then } x \notin gn(Q)}{P|Q \xrightarrow{\mu} P'|Q} \text{Cmp}_0 \\
\frac{P \xrightarrow{[y/x]} P'}{P|Q \xrightarrow{[y/x]} P'|Q[y/x]} \text{Cmp}_1 \quad \frac{P \xrightarrow{(y/x)} P' \quad y \notin gn(Q)}{P|Q \xrightarrow{(y/x)} P'|Q[y/x]} \text{Cmp}_2 \\
\frac{P \xrightarrow{mx} P' \quad Q \xrightarrow{\bar{m}[x]} Q'}{P|Q \xrightarrow{\tau} P'|Q'} \text{Cmm}_0 \quad \frac{P \xrightarrow{mx} P' \quad Q \xrightarrow{\bar{m}(x)} Q' \quad x \notin gn(P)}{P|Q \xrightarrow{\tau} (x)(P'|Q')} \text{Cmm}_1 \\
\frac{P \xrightarrow{m[x]} P' \quad Q \xrightarrow{\bar{m}[x]} Q'}{P|Q \xrightarrow{\tau} P'|Q'} \text{Cmm}_2 \quad \frac{P \xrightarrow{m[x]} P' \quad Q \xrightarrow{\bar{m}[y]} Q' \quad x \neq y}{P|Q \xrightarrow{[y/x]} P'[y/x]|Q'[y/x]} \text{Cmm}_3
\end{array}$$

$$\begin{array}{c}
\frac{P \xrightarrow{m[x]} P' \quad Q \xrightarrow{\overline{m}(y)} Q' \quad y \notin gn(P)}{P|Q \xrightarrow{(y/x)} P'[y/x]|Q'[y/x]} \text{Cmm}_4 \\
\\
\frac{P \xrightarrow{\delta} P' \quad x \notin n(\delta)}{(x)P \xrightarrow{\delta} (x)P'} \text{Loc}_0 \quad \frac{P \xrightarrow{m[x]} P' \quad x \neq m}{(x)P \xrightarrow{m[y]} P'[y/x]} \text{Loc}_1 \quad \frac{P \xrightarrow{\overline{m}[x]} P' \quad x \neq m}{(x)P \xrightarrow{\overline{m}(x)} P'} \text{Loc}_2 \\
\\
\frac{P \xrightarrow{[y/x]} P'}{(x)P \xrightarrow{\tau} P'} \text{Loc}_3 \quad \frac{P \xrightarrow{[y/x]} P'}{(y)P \xrightarrow{(y/x)} P'} \text{Loc}_4 \quad \frac{P \xrightarrow{(y/x)} P'}{(x)P \xrightarrow{\tau} (y)P'} \text{Loc}_5
\end{array}$$

In the rules, μ ranges over $\{\tau\} \cup \{m[x], \overline{m}[x], mx, \overline{m}(x) \mid m, x \in \mathcal{N}\}$ and δ ranges over $\{\tau\} \cup \{m[x], \overline{m}[x], mx, \overline{m}(x), [y/x], (y/x) \mid m, x, y \in \mathcal{N}\}$.

The localization operator in asymmetric χ -calculus is fundamentally different from the restriction combinator in π -calculus. The next example is quite illuminating: $(x)([x=y]\overline{a}[a]|\overline{b}[x]) \approx (x)\overline{b}[x]$ in π -calculus but $(x)([x=y]\overline{a}[a]|\overline{b}[x]) \not\approx (x)\overline{b}[x]$ in the asymmetric χ -calculus. In asymmetric χ -calculus the effect of the consecutive communications

$$(z)(a[z].(b(b[z]|\overline{b}[y]))|(x)\overline{a}[x].Q \xrightarrow{\tau} (b)(\mathbf{0}|\mathbf{0})|Q[y/x] \approx Q[y/x]$$

is the same as the χ -communication

$$a[y]|(x)\overline{a}[x].Q \xrightarrow{\tau} \mathbf{0}|Q[y/x] \approx Q[y/x].$$

The asymmetric χ -calculus can be investigated in completely the same way as the χ -calculus has been. The proofs of the corresponding results are more or less the same. In this language, we think of an action with positive subject name as an input action whereas that with a negative subject name as an output action. Let fo denote the set $\{\overline{a}[x] \mid a, x \in \mathcal{N}\}$ of free output, fi the set $\{a[x] \mid x \in \mathcal{N}\}$ of free input, i the set $\{ax \mid x \in \mathcal{N}\}$ of input, ro the set $\{\overline{a}(x) \mid a, x \in \mathcal{N}\}$ of restricted output, u the set $\{[y/x] \mid x, y \in \mathcal{N}\}$ of updates and ru the set $\{(y/x) \mid x, y \in \mathcal{N}\}$ of restricted updates. We can define L -bisimilarities \approx_L , open bisimilarity and barbed bisimilarity for asymmetric χ -processes as we have for χ -processes. There are altogether 63 L -bisimilarities for the asymmetric χ -processes. The next theorem conveys all the necessary information to construct the bisimulation lattice of the asymmetric χ -calculus.

Theorem 23. *The following properties hold for asymmetric χ -calculus:*

- (i) $\approx_{ru \cup fi \cup u}$ and \approx_i coincide respectively with open and barbed bisimilarities.
- (ii) $\approx_L \subseteq \approx_i$.
- (iii) $\approx_L \subseteq \approx_{fo}$.
- (iv) $\approx_{L_1} \not\subseteq \approx_{L_2}$ if $fi \cap L_1 = \emptyset$ and $fi \subseteq L_2$.
- (v) $\approx_{L_1} \not\subseteq \approx_{L_2}$ if $ru \cap L_1 = \emptyset$ and $ru \subseteq L_2$.
- (vi) $\approx_{L_1} \not\subseteq \approx_{L_2}$ if $(ru \cup ro) \cap L_1 = \emptyset$ and $ro \subseteq L_2$.
- (vii) $\approx_{L_1} \not\subseteq \approx_{L_2}$ if $u \cap L_1 = \emptyset$ and $u \subseteq L_2$.
- (viii) If $L \cap i = \emptyset$ then the inclusion $\approx_L \subseteq \approx_i$ is strict.
- (ix) $\approx_{ru} \subseteq \approx_{ro}$. The inclusion is strict.

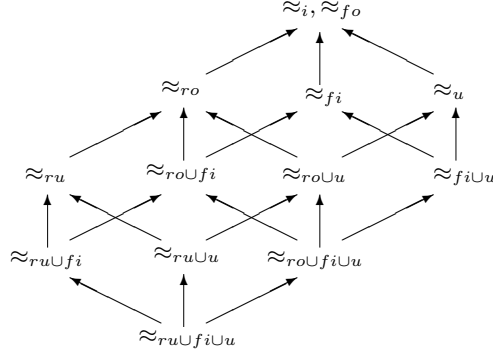


Fig. 4. The Bisimulation Lattice of Asymmetric Chi Processes

Proof. (i,ii) The proofs are similar to those in χ -calculus.

(iii) Suppose $P \approx_L Q$ and $P \xrightarrow{\bar{a}[x]} P'$. Let R be $a[x] + \langle \phi \rangle$ such that $\phi \in L$ and $n(\phi) \cap gn(P|Q) = \emptyset$. Then $P|R \xrightarrow{\tau} P'|\mathbf{0}$. It follows that Q' exists such that $Q|R \Longrightarrow Q'|\mathbf{0}$ and $P' \approx_L Q'$. Due to asymmetry, it must be the case that $Q \xrightarrow{\bar{a}[x]} Q'$.

(iv) Assume $fi \cap L_1 = \emptyset$ and $fi \subseteq L_2$. Then $(x)a[x].(b)(b[z]|\bar{b}[x]) \approx_{L_1} a[z] + (x)a[x].(b)(b[z]|\bar{b}[x])$ but $(x)a[x].(b)(b[z]|\bar{b}[x]) \not\approx_{L_2} a[z] + (x)a[x].(b)(b[z]|\bar{b}[x])$.

(v) One has $(a)((x)\bar{a}[x]|a[y]) \not\approx_{ru} \mathbf{0}$ but $(a)((x)\bar{a}[x]|a[y]) \approx_{L_1} \mathbf{0}$ whenever $ru \cap L_1 = \emptyset$.

(vi) Let R be $(x)\bar{a}[x].(b)(b[x]|(y)\bar{b}[y]).(w)c[w]$. Suppose $(ru \cup ro) \cap L_1 = \emptyset$ and $ro \subseteq L_2$. Then $R \approx_{L_1} (x)\bar{a}[x].(w)c[w] + R$ but $R \not\approx_{L_2} (x)\bar{a}[x].(w)c[w] + R$.

(vii) Let R be $(b)(z)(\bar{b}[z]|b[x].(a)(a[z]|\bar{a}[y] + a[y]|\bar{a}[z]))$. Suppose $u \cap L_1 = \emptyset$ and $u \subseteq L_2$. Then $R \approx_{L_1} (a)(a[x]|\bar{a}[y]) + R$ but $R \not\approx_{L_2} (a)(a[x]|\bar{a}[y]) + R$.

(viii) By (ii) through (vii) the inclusion $\approx_L \subseteq \approx_i$ is strict whenever $L \cap i = \emptyset$.

(ix) Suppose $P \approx_{ru} Q$ and $P \xrightarrow{\bar{a}(x)} P'$. Then $P|a[z] \xrightarrow{(x/z)} P'|\mathbf{0}$ for fresh z . So $Q|a[z] \xrightarrow{(x/z)} Q'|\mathbf{0}$ for some Q' . Hence $Q \xrightarrow{\bar{a}(x)} Q'$. Therefore $\approx_{ru} \subseteq \approx_{ro}$. The strictness follows from (v). \square

According to Theorem 23 the L -bisimilarities \approx_{fi} , \approx_i , \approx_{ro} , \approx_u and \approx_{ru} are all pairwise distinct. Using these five relations, the bisimulation lattice of asymmetric χ -processes can be generated. The pictorial description of the lattice is given by the diagram in Fig. 4. There are altogether 12 elements in the lattice. The bottom element is $\approx_{ru\cup fi\cup u}$, which characterizes the open bisimilarity. The top element is \approx_i , which coincides with the barbed bisimilarity.

7 The Asynchronous Asymmetric Chi

It seems reasonable to remove the symmetry of communications in asynchronous χ -calculus. The resulting language has the same grammar as the asynchronous χ

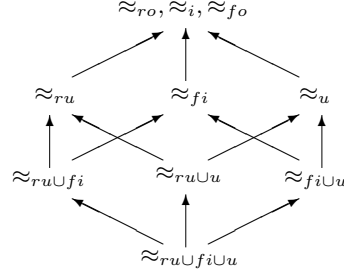


Fig. 5. The Bisimulation Lattice of Asynchronous Asymmetric Chi Processes

and the same operational semantics as the asymmetric χ . Without further ado, we come to describe the bisimulation lattice of the language.

Theorem 24. *In the asynchronous asymmetric χ -calculus, one has:*

- (i) $\approx_{fi}, \approx_{ru}, \approx_u \subseteq \approx_{ro} = \approx_i = \approx_{fo}$;
- (ii) $\approx_{fi}, \approx_{ru}, \approx_u$ are pairwise distinct.

Proof. Apart from the part to establish (vi), the proof of Theorem 23 works fine. The only extra work is to prove that $\approx_L \subseteq \approx_{ro}$ for all L -bisimilarities \approx_L . Suppose $P \approx_L Q$ and $P \xrightarrow{\bar{a}(x)} P'$. Let R be $(y)a[y].b[y]$ for some fresh b . Then $P|R \xrightarrow{bx} P'|0$. It follows that Q' exists such that $Q|R \xrightarrow{bx} Q'|0$ and $P' \approx_L Q'$. The rest of the argument is similar to that given in the proof of Theorem 22. In $Q|R \xrightarrow{bx} Q'|0$ only the reduction \xrightarrow{bx} causally depends on the communication involving R . So this particular communication can be delayed to happen just before the action involving b . It can then be easily seen that $Q \xrightarrow{\bar{a}(x)} Q'$. To help understand the idea. Let's see an example. Suppose Q is $(x)(\bar{a}[x]|Q_1)$. Then

$$\begin{aligned}
(x)(\bar{a}[x]|Q_1)|(y)a[y].b[y] &\Longrightarrow (x)(\bar{a}[x]|Q_2)|(y)a[y].b[y] \\
&\longrightarrow (x)((0|Q_2)|b[x]) \\
&\xrightarrow{bx} (0|Q_3)|0 \equiv Q'|0.
\end{aligned}$$

Clearly $(x)(\bar{a}[x]|Q_1) \Longrightarrow (x)(\bar{a}[x]|Q_3) \xrightarrow{\bar{a}(x)} 0|Q_3 \equiv Q'$. □

The bisimulation lattice of the asynchronous asymmetric χ -calculus is pictured by the diagram in Fig. 5.

8 Final Remark

The χ -calculus is meant to be a concurrent generalization of λ -calculus. In the latter the variables are uniform in the sense that both free and closed variables can be instantiated by any term. Free and closed variables become global and

local names in our model. In this way the names in χ -calculus behave very much like logical variables. This is not at all surprising as the calculus is designed with proof theory in mind. At the level of operational semantics, the computation-as-cut-elimination is a well accepted approach in the world of λ -calculus and its descendants but has attracted far less attention in process algebra. The χ -calculus advocates this approach and invites further study of it.

The L -bisimilarities are introduced as possible classification of bisimilarities on χ -processes. In both χ and its asymmetric version, open bisimilarity and barbed bisimilarity are respectively the bottom and the top elements of the bisimulation lattices. Other well-known bisimilarities also fit into the lattice hierarchy. For instance, ground bisimilarity coincides with $\approx_{o\cup\bar{o}}$.

The fact that the barbed bisimilarity on χ -processes is different from the obvious bisimilarity called open bisimilarity in this paper was first discovered in [6]. In present paper the difference is recast in the framework of L -bisimilarity. The barbed bisimilarity on χ -processes has some very interesting and unusual properties unknown from the study of π -processes. Our result implies that weak barbed congruence and weak hyperequivalence on Fusion processes are different.

The operational semantics of χ defined in this paper is slightly different from that given in [4]. In that paper the reduction

$$a[x].P|\bar{a}[x].Q \xrightarrow{\tau} P|Q$$

is not admissible although

$$(x)(a[x].P|\bar{a}[x].Q) \xrightarrow{\tau} (x)(P|Q)$$

is legal. It should be pointed out that if this restricted communication mechanism is adopted, the elements of the bisimulation lattices discussed in this paper would proliferate. For instance the bisimulation lattice of asymmetric χ -processes would have eighteen elements, the reason being that \approx_{fo} is different from \approx_i in this variant for the same reason that \approx_{fi} is different from \approx_i .

In our definition of L -bisimilarities, an update transition $P \xrightarrow{[y/x]} P'$ is required to be matched up by $Q \xrightarrow{[y/x]} Q'$ such that $P' \approx_L Q'$. This is correct because we use an early semantics. In late semantics we must replace $P' \approx_L Q'$ by $P'[y/x] \approx_L Q'[y/x]$. A simple example suffices to explain the situation. Let R be $[y|x].(x[x]|\bar{y}[y].c[c])$ where all names are assumed to be distinct. Then $[y|x].c[c] + R \approx_L R$. In late semantics there is no R' such that $R \xrightarrow{[y/x]} R' \approx_L c[c]$.

In a sense, axiomatization is internalization. Usually a meta-operation is internalized as choice operator and a meta-judgement is internalized as match combinator in calculi of mobile processes. The simplicity of the axiomatization systems for χ -processes is due to the fact that, unlike in π -calculus, one does not have the trouble of having to treat localization operator separately by resorting to distinction. In χ -calculus, if a local name can be ‘opened up’ it is open for all instantiations. The work reported in this paper was carried out independently with [15]. As far as axiomatization is concerned, the novelty of our result is a complete system for barbed congruence. We believe that by adding a modified

version of the output law O3 to Parrow and Victor's axiom system for weak hyperequivalence we get a complete system for the weak barbed congruence on finite Fusion processes.

An interesting avenue for further investigation is axiomatization of asymmetric χ -processes. We have already obtained complete systems for some of the twelve L -congruence relations. The rest appears more subtle.

References

1. Amadio, R., Castellani, I., Sangiorgi, D.: On Bisimulation for the Asynchronous π -Calculus. CONCUR'96, Lecture Notes in Computer Science 1119 (1996)
2. Boudol, G.: Asynchrony and the π -Calculus. Research Report 1702, INRIA, Sophia-Antipolis (1992)
3. Fu, Y.: The χ -Calculus. Proceedings of the International Conference on Advances in Parallel and Distributed Computing, March 19th-21th, Shanghai, IEEE Computer Society Press (1997) 74-81
4. Fu, Y.: A Proof Theoretical Approach to Communications. ICALP'97, July 7th-11th, Bologna, Italy, Lecture Notes in Computer Science 1256 (1997) 325-335
5. Fu, Y.: Reaction Graph. To appear in Journal of Computer Science and Technology (1998)
6. Fu, Y.: Variations on Mobile Processes. To appear in Theoretical Computer Science, Elsevier Science Publisher
7. Hennessy, M., Milner, R.: Algebraic Laws for Nondeterminism and Concurrency. Journal of ACM **67** (1985) 137-161
8. Honda, K., Tokoro, M.: An Object Calculus for Asynchronous Communication. ECOOP '91, Geneve, Lecture Notes in Computer Science (1991)
9. Merro, M.: On the Expressiveness of Chi, Update and Fusion Calculus. Express'98 (1998)
10. Milner, R.: Communication and Concurrency. Prentice Hall (1989)
11. Milner, R., Parrow, J., Walker, D.: A Calculus of Mobile Processes. Information and Computation **100** (1992) Part I:1-40, Part II:41-77
12. Milner, R., Sangiorgi, D.: Barbed Bisimulation. ICALP'92, Lecture Notes in Computer Science 623 (1992) 685-695
13. Parrow, J., Victor, B.: The Update Calculus. Proceedings of AMAST'97, Sydney, December 13-17 (1997)
14. Parrow, J., Victor, B.: The Fusion Calculus: Expressiveness and Symmetry in Mobile Processes. To appear in LICS'98 (1998)
15. Parrow, J., Victor, B.: The Tau-Laws of Fusion. To appear in CONCUR'98 (1998)
16. Sangiorgi, D.: A Theory of Bisimulation for π -Calculus. CONCUR'93, Lecture Notes in Computer Science 715 (1993)
17. Victor, B., Parrow, J.: Concurrent Constraints in the Fusion Calculus. To appear in ICALP'98 (1998)