

Bisimulation Congruence of Chi Calculus

Yuxi Fu *

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

Abstract

Chi calculus was proposed as a process calculus that has a uniform treatment of names. Preliminary properties of chi calculus have been examined in literature. In this paper a more systematic study of bisimilarities for chi processes is carried out. 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 form 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 congruences.

Key Words: Process Algebra, Mobile Process, Bisimulation, Axiomatization

1 Introduction

The theory of mobile processes has been one of the focuses in concurrency theory over the nineties. The prime example of calculi of mobile processes is of course the π -calculus ([17]) of Milner, Parrow and Walker. A mobile process differs from a CCS process ([16]) in that the communication topology of the former can change during the evolution of the process while that of the latter is static. A communication between two mobile processes passes a channel name from one process to another; and therefore may establish new link between them. As it turns out, the ability to send and receive channel names gives a considerable expressive power to calculi of mobile processes. For one thing, π -calculus can simulate higher order process calculi in which processes can emit or receive processes as well as channel names ([27, 28, 29, 23]). For another thing, the π -calculus has been shown to be capable of modeling a wide range of computing phenomena familiar in the computing world ([31, 32]).

Mobility also renders the algebraic theory of process algebra difficult. One of the difficulties is to do with prefix operations. Semantically an input action can instantiate the parameter name by any other name. The congruence then requires that substituting a name for a free name in two equivalent π -processes should result in two equivalent processes. This seems to suggest that equivalence relations on π -processes should be closed under substitution on names. In bisimulation equivalences however there is a choice of whether the requirement of closure under substitution is placed on the first step of a bisimulation or on each step of the bisimulation. On the other hand, a restricted output action is quite different. A plain fact about π -calculus is that a communication never changes a bound name. A bound name remains so during process evolution. But in bisimulations, a restricted output action opens up, so to speak, a bound name. This raises the question of if it is sensible to even consider restricted output actions in bisimulation argument. Consequently there is a diversity as regards to observational equalities on π -processes. Among the bisimulation equivalence relations, the early, late, open and barbed equivalences are the most well known ([17, 24, 18]). The open equivalence is closed under substitution in each bisimulation step. The other three are closed under substitution only in the first step of bisimulation. This distinction makes open equivalence strictly stronger than the other three. The barbed equivalence differs from the other three in another sense: It ignores restricted output actions. As a consequence, it is weaker than both open and late equivalences and is at least as weak as the early equivalence. Another difficulty is to do with axiomatization. To start with, what operators are necessary in order to have

*The author is supported by the NNSFC (69873032), the 863 Hi-Tech Project (863-306-ZT06-02-2), the Young Scientist Research Fund and the University Scholar Funding Scheme. He is also supported by BASICS, Center of Basic Studies in Computing Science, sponsored by Shanghai Education Commission. BASICS is affiliated to the Department of Computer Science of Shanghai Jiaotong University.

an expansion law? For example how does one expand $\bar{a}x|b(y)$? The law must take into account the possibility that b might be equated to a after some substitution. For that purpose the match operator is introduced. Now $\bar{a}x.P|b(y).Q$ can be expanded as follows:

$$\bar{a}x.(P|b(y).Q) + b(y).(\bar{a}x.P|Q) + [a=b]\tau.(P|Q\{x/y\})$$

The component $[a=b]\tau.(P|Q\{x/y\})$ retains the possibility that a communication can happen between $\bar{a}x.P$ and $b(y).Q$ whenever $a = b$. Using match operator, complete axiomatization systems for various strong congruences have been given. Lin has given for the first time a successful attempt to axiomatize both the weak early congruence and the weak late congruence, using the symbolic approach ([13, 14, 15]). For some time a complete system for the weak open congruence was not available although it was generally believed that Milner's tau laws are sufficient to lift a complete system for the strong open congruence to a complete system for the weak open congruence. Recently the present author discovered in [10] that the popular view was wrong. He has proved that Milner's tau laws fail to achieve a completeness result for weak open congruence introduced by Sangiorgi. A fourth tau law

$$\tau.P = \tau.(P + [x=y]\tau.P) \tag{1}$$

is called for. It is obvious that (1) is valid for weak early congruence, weak late congruence and weak barbed congruence. One hopes that the discovery of (1) would lead to successful investigation of non-symbolic approach to axiomatization for weak congruences on mobile processes.

Many variants of π -calculus have been proposed and studied. The internal π -calculus, the πI of Sangiorgi ([25, 2]), removes the free output prefix operation from the π -calculus. The calculus becomes symmetric. Another variant is obtained from π -calculus by omitting continuations of output operations. This is the so called asynchronous π -calculus ([3, 12, 1]). Both variants place restrictions on the use of output prefix operations. A third variant on the other hand places restrictions on input prefix operations. The local π -calculus prohibit the use of a parameter name as the subject name of an input prefix in the continuation of a process in input prefix form ([19]). Other variants can be obtained by combining the above features. In [6, 7] a symmetric presentation of π -calculus is given in which a communication amounts to exchange of names. As for the π -calculus, the algebraic theories of these variants have been investigated along two lines. One is to find sensible bisimulation equivalences for these languages. The other is to figure out complete systems for the corresponding bisimulation congruences.

Another variant that departs from π -calculus more significantly is the χ -calculus ([4, 5, 8, 9]). It was introduced with two motivations in mind. One is to remove the *ad hoc* nature of prefix operation in π -calculus by having a uniform treatment of names, thus arriving at a conceptually simpler language. The second is to materialize a communication-as-cut-elimination viewpoint, therefore taking up a proof theoretical approach to concurrency theory ([7]), 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 ([20]). 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. Later on, Parrow and Victor have proposed Fusion Calculus ([21, 30]), which is a polyadic version of χ -calculus. The authors have also studied an observational equivalence, called weak hyperbisimilarity ([22]). 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 objective of this paper is to continue our examination of the algebraic theory of χ -calculus. The focus will be on bisimulation congruences and their axiomatizations. We will introduce a uniform definition of weak bisimilarities called L -bisimilarities. These bisimilarities differ from one another in the extent actions are admitted. There are by definition thirty one L -bisimilarities, most of which coincide. It turns out that there are only four distinct L -bisimilarities. Ordered by inclusion, they form the diamond lattice of four elements. The bottom element of the lattice coincides with open bisimilarity and the top element with barbed bisimilarity. We then investigate alternative characterizations of the L -bisimilarities. For each of the four L -bisimilarities, we introduce an open style bisimilarity, called open L -bisimilarity, and prove its coincidence with L -bisimilarity. We then investigate axiomatization problems for the largest congruence relations contained in the four open L -bisimilarities respectively. A

sound and complete system for the smallest congruence is obtained by extending the complete system for the strong bisimilarity with the tau laws. This system can be further extended to complete systems for the other three congruence relations by adding additional axioms.

The contribution of the paper is two fold. First it introduces the L -bisimilarities, leading to the discovery of bisimulation lattice. The notion of bisimulation lattice is general enough for it to be applied to other calculi of mobile processes. For Fusion Calculus it points out that weak barbed bisimilarity and weak open bisimilarity are distinct. For Update Calculus it leads to an explicit definition of barbed congruence and axiomatization of it. Second it gives for the first time correct axiomatizations of all the L -congruence relations. More importantly it explains how such axiomatizations can be achieved: First one must work out an open characterization of an L -bisimilarity and then makes use of the fact that open congruence on mobile processes is more easily axiomatized than any other congruence relations. This approach can be readily applied to both Fusion Calculus and Update Calculus.

The paper is organized as follows: Section 2 reviews the operational semantics of χ . Section 3 defines L -bisimilarities and investigates their relationship. Section 4 gives alternative characterizations of L -bisimilarities. Section 5 presents a complete axiomatization system for each of the congruence relations induced by the four L -bisimilarities. Some conclusions are made in Section 6.

An extended abstract of this paper has been published ([8]). A mistake in [8] is corrected in this full paper.

2 Operational Semantics

In π -calculus there are two kinds of bound names. One has dummy names as x in $m(x).P$ and local names as x in $(x)\overline{m}x.P$. Semantically they are very different. The former can be instantiated by any other name received through communication whereas the latter remain unchanged in whatever environment the process is placed. In simple words, the χ -calculus is obtained from π -calculus by unifying these two kinds of names. This identification forces a unification of input and output prefix operations. The two π -processes just mentioned then turn into $(x)mx.P$ and $(x)\overline{m}x.P$ respectively. In the resulting calculus communications are completely symmetric as exemplified by the following instances of reductions:

$$\overline{m}x.P|m.x.Q \xrightarrow{\tau} P|Q \quad (2)$$

$$(x)(R|(\overline{m}y.P|m.x.Q)) \xrightarrow{\tau} R\{y/x\}|(P\{y/x\}|Q\{y/x\}), \text{ where } y \neq x \quad (3)$$

$$(x)\overline{m}x.P|(y)my.Q \xrightarrow{\tau} (z)(P\{z/x\}|Q\{z/y\}), \text{ where } z \text{ is fresh} \quad (4)$$

In (3) a communication through channel m happens. The communication replaces the bound name x by the free name y throughout the body the restriction operator (x) applies. In (4) the communication identifies two bound names. The identification is achieved by replacing both the bound x and y by a fresh name z and then binds it. In these two communications at least one bound name is affected. This is not true of the other communication.

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 set $\mathcal{N} \cup \overline{\mathcal{N}}$ will be ranged over by α . Let $\overline{\alpha}$ be α if $\alpha = \overline{a}$ and be \overline{a} if $\alpha = a$.

We will write \mathcal{C} for the set of χ -processes defined by the following abstract syntax:

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

As usual $\mathbf{0}$ denotes the inactive process. The process $\alpha x.P$ is in prefix form. It must first participate in a communication through channel α or $\overline{\alpha}$ before P can be activated. Here α or $\overline{\alpha}$ is the subject name, and x the object name, of the prefix. The composition operator “ $|$ ” is standard. Two components of $|$ can either evolve independently or communicate at a common channel. In $(x)P$ the name x is declared local; it is a bound name. Let $bn(P)$ denote the set of bound names in P . The set of free names, or nonbound names, in P is denoted by $fn(P)$. We will adopt the α -convention saying that a bound name in a process can be replaced by a fresh name without changing the syntax of the process. The conditional process $[x=y]P$ behaves like either P or $\mathbf{0}$, depending on whether $x = y$ or not. The constructor $[x=y]$ is often referred to as a match operator. 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 ([5]) or in terms of a labeled transition system ([4]). Here we opt for a pure transition semantics as it helps to present our results with

clear-cut proofs. The following labeled transition system defines the operational semantics of χ -calculus, in which symmetric rules are systematically omitted. In the rules the following conventions will be used: μ ranges over $\{\tau\} \cup \{\alpha x, \alpha(x) \mid x \in \mathcal{N}\}$ and δ over $\{\tau\} \cup \{\alpha x, \alpha(x), \{y/x\} \mid x, y \in \mathcal{N}\}$. In αx and $\{y/x\}$ all names are free. In $\alpha(x)$ the name x is bound whereas the other name is free. Let $fn(\mu)$, respectively $bn(\mu)$ denote the set of free names, respectively the set of bound names, in μ . The set of names appeared in μ is denoted by $n(\mu)$. The notations $fn(\delta)$, $bn(\delta)$ and $n(\delta)$ are defined similarly.

Sequentialization

$$\frac{}{\alpha x.P \xrightarrow{\alpha x} P} Sqn$$

Composition

$$\frac{P \xrightarrow{\mu} P' \quad bn(\mu) \cap fn(Q) = \emptyset}{P|Q \xrightarrow{\mu} P'|Q} Cmp_0 \quad \frac{P \xrightarrow{y/x} P'}{P|Q \xrightarrow{y/x} P'|Q\{y/x\}} Cmp_1$$

Communication

$$\frac{P \xrightarrow{\alpha(x)} P' \quad Q \xrightarrow{\bar{\alpha}y} Q'}{P|Q \xrightarrow{\tau} P'\{y/x\}|Q'} Cmm_0 \quad \frac{P \xrightarrow{\alpha(x)} P' \quad Q \xrightarrow{\bar{\alpha}(x)} Q'}{P|Q \xrightarrow{\tau} (x)(P'|Q')} Cmm_1$$

$$\frac{P \xrightarrow{\alpha x} P' \quad Q \xrightarrow{\bar{\alpha}y} Q' \quad x \neq y}{P|Q \xrightarrow{y/x} P'\{y/x\}|Q'\{y/x\}} Cmm_2 \quad \frac{P \xrightarrow{\alpha x} P' \quad Q \xrightarrow{\bar{\alpha}x} Q'}{P|Q \xrightarrow{\tau} P'|Q'} Cmm_3$$

Localization

$$\frac{P \xrightarrow{\delta} P' \quad x \notin n(\delta)}{(x)P \xrightarrow{\delta} (x)P'} Loc_0 \quad \frac{P \xrightarrow{\alpha x} P' \quad x \notin \{\alpha, \bar{\alpha}\}}{(x)P \xrightarrow{\alpha(x)} P'} Loc_1 \quad \frac{P \xrightarrow{y/x} P'}{(x)P \xrightarrow{\tau} P'} Loc_2$$

Condition

$$\frac{P \xrightarrow{\delta} P'}{[x=x]P \xrightarrow{\delta} P'} Cnd$$

Summation

$$\frac{P \xrightarrow{\delta} P'}{P+Q \xrightarrow{\delta} P'} Sum$$

The labeled transition system uses four kinds of labels, the free action labels of the form αx , the bound action labels of the form $\alpha(x)$, the internal communication label τ and the update labels of the form $\{y/x\}$. The update transition was first introduced in [4, 20] to help define communications in a transition semantics. The second composition rule shows that an update action affects the neighbouring process Q by substituting y for x in Q . The notation $\{y/x\}$ occurred in $Q\{y/x\}$ is an atomic substitution of y for x . A general substitution σ is the composition of atomic substitutions, whose effect on a process P 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 on a process is vacuous. The labeled transition system also contains four communication rules. The first is the common one. The second describes the situation in which a communication identifies two bound names. The side condition is to guarantee that the restriction operator (x) does not capture any free names already in $P|Q$. The third introduces updates. The fourth communication rule admits the kind of communications like that in (2). The side condition makes sure that Cmm_2 and Cmm_3 deal with exclusive situations. There are altogether three localization rules. The first and the second are standard. The last localization rule closes up an update. So an update is an incomplete communication. The condition rule Cnd and the summation rule Sum are standard.

The following three derivations might help the reader understand some unusual rules of the semantics. The first

$$\frac{\frac{}{\bar{m}x.P \xrightarrow{\bar{m}x} P} Sqn \quad \frac{}{mx.Q \xrightarrow{mx} Q} Sqn}{\bar{m}x.P|mx.Q \xrightarrow{\tau} P|Q} Cmm_3$$

establishes (2); the second

$$\frac{\frac{\overline{\overline{m}y.P} \xrightarrow{\overline{m}y} P} Sqn \quad \frac{\overline{m}x.Q \xrightarrow{m} Q} Sqn}{\overline{\overline{m}y.P|m}x.Q \xrightarrow{y/x} P\{y/x\}|Q\{y/x\}} Cmm_2} \frac{R|(\overline{\overline{m}y.P|m}x.Q) \xrightarrow{y/x} R\{y/x\}|(P\{y/x\}|Q\{y/x\})} Cmp_1}{(x)(R|(\overline{\overline{m}y.P|m}x.Q)) \xrightarrow{\tau} R\{y/x\}|(P\{y/x\}|Q\{y/x\})} Loc_2$$

justifies (3); and the third

$$\frac{\frac{\overline{\overline{m}z.P\{z/x\}} \xrightarrow{\overline{m}z} P\{z/x\}} Sqn}{(z)\overline{\overline{m}z.P\{z/x\}} \xrightarrow{\overline{m}(z)} P\{z/x\}} Loc_1 \quad \frac{\overline{m}z.Q\{z/y\} \xrightarrow{m} Q\{z/y\}} Sqn}{(z)mz.Q\{z/y\} \xrightarrow{m(z)} Q\{z/y\}} Loc_1}{(x)\overline{m}x.P|(y)m}y.Q \xrightarrow{\tau} (z)(P\{z/x\}|Q\{z/y\})} Cmm_1$$

legalizes (4). In the third derivation the α -convention has been applied.

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 $\xRightarrow{\hat{\mu}}$ ($\xRightarrow{\hat{\delta}}$) 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 as \vec{x} ; and consequently $(x_1) \dots (x_n)P$ will be abbreviated to $(\vec{x})P$. When the length of \vec{x} is zero, $(\vec{x})P$ is just P .

In the rest of this section we state some technical lemmas whose proofs are simple inductions on derivation.

Lemma 1. *If $P \xrightarrow{y/x} P'$ then $x \notin bn(P')$.*

Lemma 2. *The following properties hold:*

- (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 $P\sigma \xrightarrow{\tau} P'\sigma$.*

Proof. The proof is carried out by mutual induction.

(i) Suppose the last rule applied is Loc_2 . By induction hypothesis on (ii), $P\sigma \xrightarrow{y\sigma/x\sigma} P'\sigma\{y\sigma/x\sigma\}$. By Lemma 1 $x \notin bn(P')$. So $P\sigma \xrightarrow{y\sigma/x\sigma} P'\sigma$. Conclude that $(x)P\sigma \xrightarrow{\tau} P'\sigma$.

(ii) Suppose the last rule applied is Cmm_2 . By induction hypothesis on (i), $P\sigma \xrightarrow{\alpha\sigma x\sigma} P'\sigma$ and $Q\sigma \xrightarrow{\alpha\sigma y\sigma} Q'\sigma$. By Cmm_2 , $P\sigma|Q\sigma \xrightarrow{y\sigma/x\sigma} P'\sigma\{y\sigma/x\sigma\}|Q'\sigma\{y\sigma/x\sigma\}$. So $(P|Q)\sigma \xrightarrow{y\sigma/x\sigma} (P'|Q')\sigma\{y\sigma/x\sigma\}$.

(iii) The last rule applied must be either Cmp_1 or Cmm_2 . In both cases the proof is simple. \square

Lemma 3. *If $P \xrightarrow{y/x} P'$ then $P \xrightarrow{x/y} P'\{x/y\}$.*

Lemma 4. *If $P \xrightarrow{\alpha(x)} P'$ and $y \notin n(P)$ then $P \xrightarrow{\alpha(y)} P'\{y/x\}$.*

Lemma 5. *Suppose $a \notin bn(P)$. If $(x)(P|ax) \xrightarrow{\tau} \xrightarrow{\alpha(y)} P'$ then $(x)(P|ax) \xrightarrow{\alpha(y)} \xrightarrow{\tau} P'$.*

Proof. Suppose $(x)(P|ax) \xrightarrow{\tau} P_1 \xrightarrow{\alpha(y)} P'$. Then P_1 must be of the form $(z)(P''|az)$. The last rule applied to derive $(x)(P|ax) \xrightarrow{\tau} P_1$ must be Loc_0 . Therefore $P\{z/x\} \xrightarrow{\tau} P''$ for a fresh z and $P_1 \equiv (z)(P''|az)$. By Lemma 2, $P\{z/x\}\{y/z\} \xrightarrow{\tau} P''\{y/z\}$. It follows that

$$\begin{aligned} (x)(P|ax) &\xrightarrow{\alpha(y)} P\{y/x\}|0 \equiv P\{z/x\}\{y/z\}|0 \\ &\xrightarrow{\tau} P''\{y/z\}|0 \equiv P_1 \end{aligned}$$

We are done. \square

Lemma 6. *Suppose $a \notin bn(P)$. If $(x)(P|ax) \xrightarrow{\tau} P'|ay$ then $P \xrightarrow{y/x} P'$.*

Proof. The last rule applied must be Loc_2 . So $P|ax \xrightarrow{y/x} P'|ay$, which in turn implies that $P \xrightarrow{y/x} P'$ by Cmp_1 . \square

3 Bisimulation Lattice

The main focus of the algebraic theory of process algebras is on extensional equalities on processes. The extensional equalities are observational in the sense that two processes are deemed equal if no process can detect their differences by communicating with them. The finest observational equalities are bisimulation equalities. Two processes are bisimilar if they can simulate each other's observational actions up to tau actions, evolving to processes that are still bisimilar. In the standard approach, all non-tau actions are observable. In χ -calculus however there are good reasons to ignore certain types of non-tau actions. First the updates are incomplete communications; it is better to regard them as auxiliary. Second the χ -calculus is intended as target model for concurrent computation. When interpreting a process calculus in χ -calculus, not all non-tau actions of χ are used. An observable equality of the former is most likely to be related to an observable equality on χ in which these non-tau actions are regarded as auxiliary. We remark that there are fundamental difference between unobservable actions and auxiliary actions. Unobservable actions as proper actions whereas auxiliary actions are not. In a bisimulation equivalence, observable actions are required to match up each other up to unobservable actions. In such an equivalence auxiliary actions are not even considered.

In this section we study systematically bisimulation equalities on χ -processes, whereby some non-tau actions are seen as auxiliary actions. For that purpose, we introduce L -bisimilarities, which are refinement of the local bisimilarity of [5]. The L -bisimilarities provide a framework to understand bisimilarity relations of interest.

Before formally defining L -bisimilarities, some notations need be fixed. Let fa denote the set $\{ax \mid a, x \in \mathcal{N}\}$ of free actions with positive subject name, \overline{fa} the set $\{\overline{ax} \mid a, x \in \mathcal{N}\}$ of free actions with negative subject name, ba the set $\{a(x) \mid a, x \in \mathcal{N}\}$ of bound actions with positive subject name, \overline{ba} the set $\{\overline{a(x)} \mid a, x \in \mathcal{N}\}$ of bound actions with negative subject name and u the set $\{y/x \mid x, y \in \mathcal{N}\}$ of updates. Let \mathcal{L} stand for $\{\cup S \mid S \subseteq \{fa, \overline{fa}, ba, \overline{ba}, u\} \wedge S \neq \emptyset\}$. Let L range over \mathcal{L} .

Definition 7. 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 PRQ then for any process R and any sequence \vec{x} of names it holds that if $(\vec{x})(P|R) \xrightarrow{\phi} P'$ for $\phi \in L \cup \{\tau\}$ such that $bn(\phi) \cap fn((\vec{x})(Q|R)) = \emptyset$ then there exists some Q' such that $(\vec{x})(Q|R) \xrightarrow{\hat{\phi}} Q'$ and $P'RQ'$. 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 order to guarantee that \approx_L is closed under all but the choice combinator it is required that an L -bisimulation is closed under parallel composition and restriction operation under every bisimulation step. First we establish a few technical lemmas. The first lemma is about the following general property:

A weak observational equivalence \approx is said to satisfy the *-property if $P \Longrightarrow P_1 \approx Q$ and $Q \Longrightarrow Q_1 \approx P$ implies $P \approx Q$.

Lemma 8. \approx_L satisfies the *-property.

Proof. Suppose $P \Longrightarrow P_1 \approx Q$ and $Q \Longrightarrow Q_1 \approx P$. Then the relation $\{(P, Q)\} \cup \approx_L$ is an L -bisimulation. Suppose for instance $(\vec{x})(P|R) \xrightarrow{\phi} P'$ for some $\phi \in L \cup \{\tau\}$. Then by the fact that $Q_1 \approx_L P$ one has some Q' such that $(\vec{x})(Q_1|R) \xrightarrow{\hat{\phi}} Q' \approx_L P'$. It follows that $(\vec{x})(Q|R) \Longrightarrow (\vec{x})(Q_1|R) \xrightarrow{\hat{\phi}} Q' \approx_L P'$. \square

Remark: The above lemma holds not only for the L -bisimilarities but also for a wide range of weak observational equivalences. The barbed and open bisimilarities to be discussed later on enjoy the *-property. The proofs of these facts are almost word for word iteration of the above proof.

For $\phi \in L$, let $\langle \phi \rangle$ be the 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 9. Suppose $a \notin fn(P|Q)$. Then

- (i) $(x)(P|ax) \approx_L (x)(Q|ax)$ implies $P \approx_L Q$;
- (ii) $P|ax \approx_L Q|ax$ implies $P \approx_L Q$.

Proof. (i) Suppose $\phi \in L$ and $n(\phi) \cap fn(P|Q) = \emptyset$. As $(x)(P|ax)|\bar{a}x.\langle\phi\rangle \xrightarrow{\phi} (P|\mathbf{0})|\mathbf{0}$, Q_1 exists such that $(x)(Q|ax)|\bar{a}x.\langle\phi\rangle \xrightarrow{\phi} (Q_1|\mathbf{0})|\mathbf{0} \approx_L (P|\mathbf{0})|\mathbf{0}$, which implies $(x)(Q|ax) \xrightarrow{a(x)} Q_1|\mathbf{0} \approx_L P|\mathbf{0}$, which in turn implies $Q \Longrightarrow Q_1 \approx_L P$. Similarly P_1 exists such that $P \Longrightarrow P_1 \approx_L Q$. By Lemma 8, $P \approx_L Q$.

(ii) can be proved similarly. \square

Lemma 10. *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 $x \in fn(P|Q)$ 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 $n(\phi) \cap fn(P|Q) = \emptyset$. By definition the actions

$$(x)(P|(\bar{b}y|bx).\langle\phi\rangle) \xrightarrow{\phi} P\{y/x\}|(\mathbf{0}|\mathbf{0})$$

must be matched up by

$$(x)(Q|(\bar{b}y|bx).\langle\phi\rangle) \xrightarrow{\phi} Q_1|(\mathbf{0}|\mathbf{0}) \quad (5)$$

for some Q_1 . If $(x)(Q|(\bar{b}y|bx).\langle\phi\rangle) \xrightarrow{\tau} (x')(Q_2|(\bar{b}y|bx').\langle\phi\rangle)$ then it follows from the proof of Lemma 5 that the reduction is the same as $(x)(Q|(\bar{b}y|bx).\langle\phi\rangle) \xrightarrow{\tau} (x)(Q_3|(\bar{b}y|bx).\langle\phi\rangle)$ such that $Q \xrightarrow{\tau} Q_3$ and $Q_3 \equiv Q_2\{x/x'\}$. It follows that (5) can be factorized as follows

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

for some Q' such that $Q \Longrightarrow Q'$ and $Q'\{y/x\} \Longrightarrow Q_1 \approx_L P\{y/x\}$. By Lemma 2, $Q\{y/x\} \Longrightarrow Q'\{y/x\}$. Similarly P_1 exists such that $P\{y/x\} \Longrightarrow P_1 \approx_L Q\{y/x\}$. By Lemma 8, $P\{y/x\} \approx_L Q\{y/x\}$. \square

By definition the L -bisimilarity is closed under restriction and composition. Using Lemma 10 it can be easily seen that \approx_L is also closed under prefix operation.

Theorem 11. *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) $(x)P \approx_L (x)Q$;
- (iv) $[x=y]P \approx_L [x=y]Q$.

Proof. (ii) and (iii) follow easily from definition.

(i) Using (ii) and (iii), one can easily prove that $\{((\bar{y})(R|\alpha x.P), (\bar{y})(R|\alpha x.Q)) \mid \bar{y} \text{ names, } R \in \mathcal{C}, P \approx_L Q\} \cup \approx_L$ is an L -bisimulation.

(iv) Similarly $\{((\bar{z})(R|[x=y]P), (\bar{z})(R|[x=y]Q)) \mid x, y, \bar{z} \text{ names, } R \in \mathcal{C}, P \approx_L Q\} \cup \approx_L$ is an L -bisimulation. \square

We investigate next the order structure of L -bisimilarities. The following theorem unveils the full picture.

Theorem 12 (bisimulation lattice). *The following properties hold of the L -bisimilarities:*

- (i) $\approx_{fa} \subseteq \approx_{\bar{f}a} \approx_{fa}$;
- (ii) $\approx_L \subseteq \approx_u$;
- (iii) $\approx_L \subseteq \approx_{ba} \approx_{\bar{b}a}$.

Proof. (i) It is immediate that $(x)ax.(b)(\bar{b}x|bz) \not\approx_{fa} az+(x)ax.(b)(\bar{b}x|bz)$. It takes a little while to see that $(x)ax.(b)(\bar{b}x|bz) \approx_{\bar{f}a} az+(x)ax.(b)(\bar{b}x|bz)$.

(ii) To prove $\approx_L \subseteq \approx_u$, one only has to show, by Theorem 11, 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 $(x)(P|ax) \xrightarrow{\tau} P'|ay$ for a fresh a .

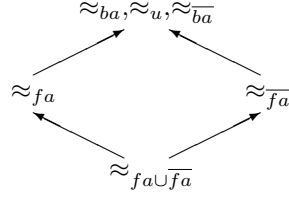


Figure 1: The Bisimulation Lattice of Chi Processes

So $(x)(Q|ax) \Longrightarrow Q'|ay$ for some Q' such that $P'|ay \approx_L Q'|ay$. It follows from Lemma 9 that $P' \approx_L Q'$. Clearly $(x)(Q|ax) \Longrightarrow Q'|ay$ can be factorized as follows

$$\begin{aligned} (x)(Q|ax) &\Longrightarrow (x)(Q_1|ax) \\ &\xrightarrow{\tau} Q_2|ay \\ &\Longrightarrow Q'|ay \end{aligned}$$

where $Q \Longrightarrow Q_1$. By Lemma 6, $(x)(Q_1|ax) \xrightarrow{\tau} Q_2|ay$ 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 $n(\phi) \cap fn(P|Q) = \emptyset$. 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 $Q \xrightarrow{\alpha(y)} Q_1$ such that $Q_1\{z/y\} \approx_L P_1$. Then $Q \xrightarrow{\alpha(x)} Q_1\{x/y\}$ by Lemma 4. On the other hand $Q_1\{x/y\} \equiv Q_1\{z/y\}\{x/z\} \approx_L P_1\{x/z\} \equiv P'$ by Lemma 10. Hence $\approx_L \subseteq \approx_{ba} = \approx_{\bar{b}a}$. \square

In the proof of (iii) of the above theorem, we need to use a fresh name z because we cannot conclude $R \xrightarrow{\alpha(y)} R'$ 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 the subset relation \subseteq . For $\approx_{L_1}, \approx_{L_2} \in \mathcal{L}(\chi)$, $\approx_{L_1 \cup L_2}$ is the infimum. Theorem 12 says that $\mathcal{L}(\chi)$ is a four element lattice as drawn in Figure 1. In the diagram each node is the principal representative of a number of L -bisimilarities that boil down to a same relation. For instance \approx_{fa} represents all the elements of $\{\approx_L \mid fa \subseteq L \subseteq fa \cup ba \cup \bar{b}a \cup u, L \in \mathcal{L}\}$. An arrow indicates a strict inclusion. The bottom element is represented by $\approx_{fa \cup \bar{f}a}$ while the top element is by $\approx_{ba} = \approx_{\bar{b}a} = \approx_u$. We will call $(\mathcal{L}(\chi), \subseteq)$ the bisimulation lattice of χ -processes. The lattice structure suggests that the ability to observe free actions is stronger than that to observe bound actions.

4 Alternative Characterization

The definition of L -bisimilarities is natural but not very tractable. 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.

4.1 Barbed Bisimilarity

Barbed bisimilarity ([18]) 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 process 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 often in deciding if the inclusion is strict or not. This section takes a look at barbed bisimilarity on χ -processes.

Definition 13. 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 , written $P \Downarrow a$, 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 .

The idea of barbed bisimilarity is that two processes are regarded as equal if they can simulate each other's communication while maintaining same ability to communicate through any particular name.

Definition 14. Let \mathcal{R} be a barbed symmetric relation on \mathcal{C} . It is called a barbed bisimulation if whenever $P\mathcal{R}Q$ then for any R and any sequence \vec{x} of names it holds that if $(\vec{x})(P|R) \xrightarrow{\tau} P'$ then Q' exists such that $(\vec{x})(Q|R) \Longrightarrow Q'$ and $P'\mathcal{R}Q'$. The barbed bisimilarity \approx_b is the largest barbed bisimulation.

The barbed bisimilarity \approx_b can be investigated in a similar manner the L -bisimilarities have been investigated. We will avoid such a rerun by establishing the coincidence of \approx_b with one of the L -bisimilarities. The proof of the coincidence employs the following lemma.

Lemma 15. \approx_b is closed under substitution.

Proof. Suppose $P \approx_b Q$ and a is a distinct fresh name. Now

$$(x)(P|(ax|\bar{a}y)) \xrightarrow{\tau} P\{y/x\}|(\mathbf{0}|\mathbf{0})$$

which by definition must be matched up by

$$(x)(Q|(ax|\bar{a}y)) \Longrightarrow Q_1|(\mathbf{0}|\mathbf{0}) \approx_b P\{y/x\}|(\mathbf{0}|\mathbf{0})$$

By the proof of Lemma 5 the above reduction can be factorized as

$$\begin{aligned} (x)(Q|(ax|\bar{a}y)) &\Longrightarrow (x)(Q_2|(ax|\bar{a}y)) \\ &\longrightarrow Q_2\{y/x\}|(\mathbf{0}|\mathbf{0}) \\ &\Longrightarrow Q_1|(\mathbf{0}|\mathbf{0}) \end{aligned}$$

such that $Q \Longrightarrow Q_2$. By Lemma 2, $Q\{y/x\} \Longrightarrow Q_2\{y/x\} \Longrightarrow Q_1 \approx_b P\{y/x\}$. Similarly P_1 exists such that $P\{y/x\} \Longrightarrow P_1 \approx_b Q\{y/x\}$. To show that \approx_b satisfy the *-property we need to verify that $\forall c \in \mathcal{N}. P\{y/x\} \downarrow c \Leftrightarrow Q\{y/x\} \downarrow c$. But this is obvious for if $P\{y/x\} \Longrightarrow P_1 \downarrow c$ then $(x)(P|(ax|\bar{a}y)) \Longrightarrow (P_1|(\mathbf{0}|\mathbf{0})) \downarrow c$, which implies that $(x)(Q|(ax|\bar{a}y)) \Longrightarrow (Q'|(\mathbf{0}|\mathbf{0})) \downarrow c$ for some Q' by barbedness. It follows that $Q\{y/x\} \Longrightarrow Q' \downarrow c$. By the remark right after Lemma 8 one conclude that $P\{y/x\} \approx_b Q\{y/x\}$. \square

We can now locate \approx_b in the bisimulation lattice.

Theorem 16. \approx_b is the same as \approx_{ba} .

Proof. As \approx_{ba} is clearly barbed, $\approx_{ba} \subseteq \approx_b$. Now suppose $P \approx_b Q$ and $P \xrightarrow{\alpha(x)} P'$. By Lemma 4, $P \xrightarrow{\alpha(z)} P_1$ for some fresh z such that $P' \equiv P_1\{x/z\}$. Let a be fresh. Now

$$P|(\bar{\alpha}z.a|\bar{a}a) \xrightarrow{\tau} \xrightarrow{\tau} P_1|(\mathbf{0}|\mathbf{0})$$

which 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 $Q \xrightarrow{\alpha(y)} Q'$ and $Q'\{z/y\} \Longrightarrow Q_1$. The former is impossible because z is fresh. By Lemma 2 and Lemma 4

$$Q \xrightarrow{\alpha(x)} Q'\{x/y\}$$

and

$$Q'\{z/y\}\{x/z\} \Longrightarrow Q_1\{x/z\}$$

By Lemma 15

$$Q \xrightarrow{\alpha(x)} Q_1\{x/z\} \approx_b P_1\{x/z\} \equiv P'.$$

Conclude that $\approx_b \subseteq \approx_{ba}$. \square

4.2 Open Bisimilarity

In calculi of mobile processes the names in a process are “open” for instantiations by other names. It is therefore natural to require that bisimulations on mobile processes should be closed under substitution. Such bisimulations have been studied by Sangiorgi in [24] for π -processes. In this section we define open bisimilarities for χ -processes and relate them to the L -bisimilarities.

Definition 17. 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 18. \approx_{open} is closed under restriction and composition.

Proof. Using the fact that \approx_{open} is closed under substitution, one sees immediately that $\{(x)P, (x)Q \mid P \approx_{open} Q\}$ is an open bisimulation. For instance if $(x)P \xrightarrow{\tau} P'$ is caused by $P \xrightarrow{y/x} P'$ then $Q \xrightarrow{y/x} Q' \approx_{open} P'$ and therefore $(x)Q \xrightarrow{\tau} Q'$. Other cases are equally simple. \square

Theorem 19. \approx_{open} coincides with $\approx_{fa \cup \bar{fa}}$.

Proof. By Lemma 10 and the proof of Theorem 12, $(fa \cup \bar{fa})$ -bisimilarity is an open bisimulation. By Lemma 18, open bisimilarity is an $(fa \cup \bar{fa})$ -bisimulation. \square

The definition of open bisimilarity makes it easy to give axiomatization system for finite χ -processes. Theorem 19 enables us to axiomatize the largest congruence contained in the bottom element of the bisimulation lattice $\mathcal{L}(\chi)$, as we shall see in next section. 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 12.

Definition 20. Let \mathcal{R} be a binary symmetric relation on \mathcal{C} . It is called an open ba -bisimulation (open fa -bisimulation, open \bar{fa} -bisimulation) if whenever PRQ then for any substitution σ it holds that

- (i) if $P\sigma \xrightarrow{\phi} P'$ for $\phi \in ba \cup \bar{ba} \cup u \cup \{\tau\}$ ($\phi \in fa \cup ba \cup \bar{ba} \cup u \cup \{\tau\}$, $\phi \in \bar{fa} \cup ba \cup \bar{ba} \cup u \cup \{\tau\}$) then Q' exists such that $Q\sigma \xrightarrow{\hat{\phi}} Q'$ and $P'\mathcal{R}Q'$; and
- (ii) if $P\sigma \xrightarrow{\alpha x} P'$ ($P\sigma \xrightarrow{\bar{\alpha} x} P'$, $P\sigma \xrightarrow{\alpha x} P'$) then some Q' exists such that $P'\mathcal{R}Q'$ and either $Q\sigma \xrightarrow{\alpha x} Q'$ ($Q\sigma \xrightarrow{\bar{\alpha} x} Q'$, $Q\sigma \xrightarrow{\alpha x} Q'$) or $Q\sigma \xrightarrow{\alpha(z) x/z} Q'$ ($Q\sigma \xrightarrow{\bar{\alpha}(z) x/z} Q'$, $Q\sigma \xrightarrow{\alpha(z) x/z} Q'$) for some fresh z .

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

The next theorem explains the reason to introduce Definition 20.

Theorem 21. (i) $\approx_{open}^{fa} = \approx_{fa}$; (ii) $\approx_{open}^{\bar{fa}} = \approx_{\bar{fa}}$; (iii) $\approx_{open}^{ba} = \approx_{ba}$.

Proof. It is enough to see how to establish (iii). Suppose $P \approx_{ba} Q$ and $P \xrightarrow{\alpha x} P'$. For a fresh name z one has $P|\bar{\alpha}z \xrightarrow{x/z} P'|\mathbf{0}$. As $\approx_{ba} = \approx_u$, Q' exists such that $Q|\bar{\alpha}z \xrightarrow{x/z} Q'|\mathbf{0} \approx_{ba} P'|\mathbf{0}$. There are only two possibilities: either $Q \xrightarrow{\alpha x} Q' \approx_{ba} P'$ or $Q \xrightarrow{\alpha(y) x/y} Q' \approx_{ba} P'$. If the latter holds then $Q \xrightarrow{\alpha(z) x/z} Q' \approx_{ba} P'$ by Lemma 4 and Lemma 10. Hence $\approx_{ba} \subseteq \approx_{open}^{ba}$ by Lemma 10.

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

- $(\bar{v})(P|R) \xrightarrow{x/y} (\bar{v})(P'\{x/y\}|R'\{x/y\})$ is derived from $P \xrightarrow{\alpha x} P'$ and $R \xrightarrow{\bar{\alpha} y} R'$. If $P \xrightarrow{\alpha x} P'$ can be matched up by $Q \xrightarrow{\alpha x} Q' \approx_{open}^{ba} P'$ we are done. Otherwise it must be matched up by $Q \xrightarrow{\alpha(z)} Q_1 \xrightarrow{\{x/z\}} Q'$ for some Q_1 , Q' and fresh z . By Lemma 2

$$(\bar{v})(Q|R) \xrightarrow{x/y} (\bar{v})(Q_1\{y/z\}|R') \xrightarrow{\{x/z\}} (\bar{v})(Q'\{y/z\}\{x/y\}|R'\{x/y\}) \equiv (\bar{v})(Q'\{x/y\}|R'\{x/y\})$$

- $(\vec{v})(P|R) \xrightarrow{y/x} (\vec{v})(P'\{y/x\}|R'\{y/x\})$ is derived from $P \xrightarrow{\alpha x} P'$ and $R \xrightarrow{\bar{\alpha}y} R'$. Suppose $P \xrightarrow{\alpha x} P'$ is matched up by $Q \xrightarrow{\alpha(z)} Q_1 \xrightarrow{x/z} Q'$ for some Q_1, Q' and fresh z . Then $Q_1 \xrightarrow{z/x} Q'\{z/x\}$. It follows from Lemma 2 that $Q_1\{y/z\} \xrightarrow{y/x} Q'\{y/x\}$. Hence

$$(\vec{v})(Q|R) \Longrightarrow (\vec{v})(Q_1\{y/z\}|R') \xrightarrow{y/x} (\vec{v})(Q'\{y/x\}|R'\{y/x\})$$

- $(\vec{v})(P|R) \xrightarrow{\tau} (\vec{v})(P'|R')$ is derived from $P \xrightarrow{\alpha x} P'$ and $R \xrightarrow{\bar{\alpha}(y)} R_1$ such that $R' \equiv R_1\{x/y\}$. Suppose $P \xrightarrow{\alpha x} P'$ is matched up by $Q \xrightarrow{\alpha(z)} Q_1 \xrightarrow{\{x/z\}} Q'$ for some Q_1, Q' and fresh z . Therefore

$$(\vec{v})(Q|R) \Longrightarrow (\vec{v})(z)(Q_1|R_1\{z/y\}) \Longrightarrow (\vec{v})(Q'|R_1\{z/y\}\{x/z\}) \equiv (\vec{v})(Q'|R')$$

- Suppose that $(\vec{v})(P|R) \xrightarrow{\alpha(x)} (\vec{v}')(P'|R)$, where $x \in \{\vec{v}\}$, is derived from $P \xrightarrow{\alpha x} P'$ by applying rule *Loc*₁. Suppose $P \xrightarrow{\alpha x} P'$ is matched up by $Q \xrightarrow{\alpha(z)} Q_1 \xrightarrow{x/z} Q'$ for some Q_1, Q' and fresh z . For a fresh w one has that

$$Q\{w/x\} \xrightarrow{\alpha(z)} Q_1\{w/x\} \xrightarrow{w/z} Q'\{w/x\}$$

By Lemma 4 and Lemma 10

$$Q\{w/x\} \xrightarrow{\alpha(x)} Q_1\{w/x\}\{x/z\} \xrightarrow{w/x} Q'\{w/x\}$$

since $z \notin fn(Q'\{w/x\})$. By Lemma 3

$$Q\{w/x\} \xrightarrow{\alpha(x)} Q_1\{w/x\}\{x/z\} \xrightarrow{x/w} Q'$$

It follows that

$$\begin{aligned} (\vec{v}'')(Q\{w/x\}|R\{w/x\}) &\xrightarrow{\alpha(x)} (\vec{v}'')(Q_1\{w/x\}\{x/z\}|R\{w/x\}) \\ &\Longrightarrow (\vec{v}'')(Q'|R) \end{aligned}$$

where w is a fresh name and \vec{v}'' is obtained from \vec{v} by substituting w for x .

The closure property allows us to conclude that \approx_{open}^{ba} is a *ba*-bisimulation. \square

Using the above theorem it is easy to see that the following propositions hold:

$$\begin{aligned} (z)mz.(P + \langle x|z \rangle.Q) &=_{\bar{f}a} (z)mz.(P + \langle x|z \rangle.Q) + mx.Q\{x/z\}, \text{ where } x \neq z \\ (z)\bar{m}z.(P + \langle x|z \rangle.Q) &=_{fa} (z)\bar{m}z.(P + \langle x|z \rangle.Q) + \bar{m}x.Q\{x/z\}, \text{ where } x \neq z \\ (z)\alpha z.(P + \langle x|z \rangle.Q) &=_{ba} (z)\alpha z.(P + \langle x|z \rangle.Q) + \alpha x.Q\{x/z\}, \text{ where } x \neq z \end{aligned}$$

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

5 Axiomatization

For each of the congruence relation on processes, there is a question of how to give an inference system that is sound and complete for the relation on finite processes. This is an important question in two aspects. First a complete system represents a crucial step in our understanding of the congruence relation. Second a complete system is usually a basis for automatic reasoning of the relation. The question is relatively simple for congruence relations in CCS ([11, 16]). It is a difficult one in π -calculus.

The *L*-bisimilarities are not congruence relations. To get a congruence relation from \approx_L we use the standard approach: Let $=_L$ be the largest congruence contained in \approx_L . It can be formally defined as follows:

Definition 22. $P =_L Q$ if and only if $P \approx_L Q$ and for each substitution σ the following property holds:

- (i) whenever $P\sigma \xrightarrow{\tau} P'$ then $Q'\sigma$ exists such that $Q\sigma \xrightarrow{\tau} Q'\sigma$ and $P' \approx_L Q'\sigma$;
- (ii) whenever $Q\sigma \xrightarrow{\tau} Q'$ then $P'\sigma$ exists such that $P\sigma \xrightarrow{\tau} P'\sigma$ and $P' \approx_L Q'\sigma$.

The fact that $=_L$ is congruent is established by exploiting Theorem 19 and Theorem 21 since the standard argument can be applied to the open bisimilarities in an obvious way. We will call $=_L$ the L -congruence on χ -processes. The bisimulation lattice says that 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.

5.1 Strong Case

Before tackling the problem of axiomatizing L -congruences, we introduce strong bisimilarity on χ -processes and its complete axiomatization system. The results in this section are not original, most of which are already in [22]. They are included for completeness reason.

Definition 23. Let \mathcal{R} be a binary symmetric relation on \mathcal{C} . The relation \mathcal{R} is a strong bisimulation if, for each substitution σ , whenever $P\sigma \xrightarrow{\delta} P'$ then Q' exists such that $Q\sigma \xrightarrow{\delta} Q'$ and $P'\mathcal{R}Q'$. The largest strong bisimulation, called the strong bisimilarity, is denoted by \sim .

The relation is what one would have called strong open bisimilarity. It is routine to show that \sim is an equivalence and congruence relation and is closed under substitution. Strong versions of L -bisimilarity, barbed bisimilarity and others can be defined in very much the standard manner. All of them coincide with \sim . It seems better to leave out any adjective and simply call \sim the strong bisimilarity.

For purpose of axiomatization, we introduce update and tau prefix operations. They are defined as follows, where a and b are fresh:

$$\begin{aligned} \langle y|x \rangle.P &\stackrel{\text{def}}{=} (a)(\bar{a}y|ax.P) \\ \tau.P &\stackrel{\text{def}}{=} (b)\langle b|b \rangle.P \end{aligned}$$

Notice that $\langle x|x \rangle.P \xrightarrow{\tau} P$ and $\langle x|x \rangle.P =_L \tau.P$. So $\langle x|x \rangle$ also plays the role of τ prefix operation. It is clear that $\langle x|y \rangle.P =_L \langle y|x \rangle.P$ due to the symmetry of communication.

We will need to deal with a sequence of match constructs concatenated one after another. It is convenient to have a meta symbol for such a sequence. In the rest of the paper M and N denote finite lists of match equalities. 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 appearing in M . We use σ_M to denote an arbitrary substitution that sends all members of an equivalence class of that relation to a representative of that class. Let π and γ range over $\{\alpha x, \langle y|x \rangle \mid x, y \in \mathcal{N}\}$.

In [24] Sangiorgi presents a complete system for strong open congruence on π -processes. Parrow and Victor consider in [20] two axiomatization systems for strong bisimilarity on Update processes, one with mismatch operator and one without mismatch operator. The latter is of similar style to Sangiorgi's system. The strong bisimilarity of Fusion processes has also been axiomatized ([22]). In Figure 2 a conditional equational system is defined, which is a monadic version of Parrow and Victor's systems. Let AS denote the system consisting of the rules and laws in Figure 2 plus the following expansion law:

$$\begin{aligned} P|Q &= \sum_i [M_i](\vec{x})\pi_i.(P_i|Q) + \sum_{\substack{\pi_i = a_i x_i \\ \gamma_j = \bar{b}_j y_j}} [M_i][N_j](\vec{x})(\vec{y})[a_i=b_j]\langle x_i|y_j \rangle.(P_i|Q_j) + \\ &\quad \sum_j [N_j](\vec{y})\gamma_j.(P|Q_j) + \sum_{\substack{\pi_i = \bar{a}_i x_i \\ \gamma_j = b_j y_j}} [M_i][N_j](\vec{x})(\vec{y})[a_i=b_j]\langle x_i|y_j \rangle.(P_i|Q_j) \end{aligned}$$

where P is $\sum_i [M_i](\vec{x})\pi_i.P_i$ and Q is $\sum_j [N_j](\vec{y})\gamma_j.Q_j$. In the expansion law the summand

$$\sum_{\substack{\pi_i = a_i x_i \\ \gamma_j = \bar{b}_j y_j}} [M_i][N_j](\vec{x})(\vec{y})[a_i=b_j]\langle x_i|y_j \rangle.(P_i|Q_j)$$

contains $[M_i][N_j](\vec{x})(\vec{y})[a_i=b_j]\langle x_i|y_j \rangle.(P_i|Q_j)$ as a summand whenever $\pi_i = a_i x_i$ and $\gamma_j = \bar{b}_j y_j$. AS is Victor and Parrow's system without mismatch. The only difference is that we use a symmetric update prefix operator. Some derived rules of AS , together with their justifications, are given in Figure 3.

<i>Equivalence</i>		
E1	$P = P$	
E2	$P = Q$	if $Q = P$
E3	$P = R$	if $P = Q$ and $Q = R$
<i>Congruence</i>		
C1	$\alpha x.P = \alpha x.Q$	if $P = Q$
C2	$(x)P = (x)Q$	if $P = Q$
C3	$[x=y]P = [x=y]Q$	if $P = Q$
C4	$P+R = Q+R$	if $P = Q$
C5	$P_0 P_1 = Q_0 Q_1$	if $P_0 = Q_0$ and $P_1 = Q_1$
<i>Localization</i>		
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	$(x)[x=y]P = \mathbf{0}$	
L7	$(x)(P+Q) = (x)P+(x)Q$	
L8	$(x)\langle y z \rangle.P = \langle y z \rangle.(x)P$	$x \notin \{y, z\}$
L9	$(x)\langle y x \rangle.P = \tau.P\{y/x\}$	$y \neq x$
L10	$(x)\langle x x \rangle.P = \tau.(x)P$	
<i>Match</i>		
M1	$[M]P = [N]P$	if $M \Leftrightarrow N$
M2	$[x=y]P = [x=y]P\{y/x\}$	
M3	$[x=y](P+Q) = [x=y]P+[x=y]Q$	
<i>Selection</i>		
S1	$P+\mathbf{0} = P$	
S2	$P+Q = Q+P$	
S3	$P+(Q+R) = (P+Q)+R$	
S4	$[x=y]P+P = P$	
<i>Update</i>		
U1	$\langle y x \rangle.P = \langle x y \rangle.P$	
U2	$\langle y x \rangle.P = \langle y x \rangle.[x=y]P$	
U3	$\langle x x \rangle.P = \tau.P$	

Figure 2: Axioms for Strong Bisimilarity on Chi Processes

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 $P = Q$.

Using axioms in AS , a process can be converted to a process that contains no occurrence of composition operator.

Definition 24. A process P is in normal form if it is of the form

$$\sum_{i \in I_1} [M_i] \alpha_i x_i . P_i + \sum_{i \in I_2} [M_i] (x) \alpha_i x . P_i + \sum_{i \in I_3} [M_i] \langle z_i | y_i \rangle . P_i$$

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) $d(\mathbf{0}) = 0$; (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) $d(P+Q) = \max\{d(P), d(Q)\}$.

Lemma 25. In AS every process P is provably equal to some P' in normal form such that $d(P') \leq d(P)$.

LD1	$(x)\langle x x \rangle.P = \langle y y \rangle.(x)P$	U3 and L8
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	$\langle y x \rangle.P = \langle y x \rangle.P\{y/x\}$	U2 and M2

Figure 3: Some Laws Derivable from AS

Proof. The proof is carried out by structural induction. If for example the outer most combinator of a process is a restriction operator then the process is either equal to $\mathbf{0}$ or equal to a process of the form $(x)\alpha x.P$ such that $x \notin \{\alpha, \bar{\alpha}\}$ or the restriction operator can be pushed inside. It is obvious that this conversion procedure does not increase the depth of the process. \square

Theorem 26. AS is sound and complete for \sim .

Proof. By Lemma 25 every process is provably equal to a normal form process. So we can focus on normal form processes exclusively. Let P be

$$\sum_{i \in I_1} [M_i] \alpha_i x_i . P_i + \sum_{i \in I_2} [M_i] (x) \alpha_i x . P_i + \sum_{i \in I_3} [M_i] \langle z_i | y_i \rangle . P_i$$

and Q be

$$\sum_{j \in J_1} [N_j] \alpha_j x_j . Q_j + \sum_{j \in J_2} [N_j] (x) \alpha_j x . Q_j + \sum_{j \in J_3} [N_j] \langle z_j | y_j \rangle . Q_j$$

Suppose $P \sim Q$. We show that $P = Q$ is provable in AS . The proof is carried out by induction on the sum of depths of P and Q . Consider $[M_i] \langle z_i | y_i \rangle . P_i$ for some $i \in I_3$. Now either $([M_i] \langle z_i | y_i \rangle . P_i) \sigma_{M_i} \xrightarrow{z_i \sigma_{M_i} / y_i \sigma_{M_i}} P_i \sigma_{M_i} \{z_i \sigma_{M_i} / y_i \sigma_{M_i}\}$ or $([M_i] \langle z_i | y_i \rangle . P_i) \sigma_{M_i} \xrightarrow{\tau} P_i \sigma_{M_i}$. In the first case, some $j \in J_3$ must exist such that $([N_j] \langle z_j | y_j \rangle . Q_j) \sigma_{M_i} \xrightarrow{z_j \sigma_{M_i} / y_j \sigma_{M_i}} Q_j \sigma_{M_i} \{z_j \sigma_{M_i} / y_j \sigma_{M_i}\}$, $M_i \Rightarrow N_j$, $z_i \sigma_{M_i} = z_j \sigma_{M_i}$, $y_i \sigma_{M_i} = y_j \sigma_{M_i}$ and $P_i \sigma_{M_i} \{z_i \sigma_{M_i} / y_i \sigma_{M_i}\} \sim Q_j \sigma_{M_i} \{z_j \sigma_{M_i} / y_j \sigma_{M_i}\}$. By induction hypothesis $P_i \sigma_{M_i} \{z_i \sigma_{M_i} / y_i \sigma_{M_i}\} = Q_j \sigma_{M_i} \{z_j \sigma_{M_i} / y_j \sigma_{M_i}\}$ is provable in AS . It follows that

$$\begin{aligned} Q + [M_i] \langle z_i | y_i \rangle . P_i &\stackrel{MD3}{=} Q + [M_i] \langle z_i \sigma_{M_i} | y_i \sigma_{M_i} \rangle . P_i \sigma_{M_i} \\ &\stackrel{UD1}{=} Q + [M_i] \langle z_i \sigma_{M_i} | y_i \sigma_{M_i} \rangle . P_i \sigma_{M_i} \{z_i \sigma_{M_i} / y_i \sigma_{M_i}\} \\ &\stackrel{I.H.}{=} Q + [M_i] \langle z_j \sigma_{M_i} | y_j \sigma_{M_i} \rangle . Q_j \sigma_{M_i} \{z_j \sigma_{M_i} / y_j \sigma_{M_i}\} \\ &\stackrel{UD1}{=} Q + [M_i] \langle z_j \sigma_{M_i} | y_j \sigma_{M_i} \rangle . Q_j \sigma_{M_i} \\ &\stackrel{MD3}{=} Q + [M_i] \langle z_j | y_j \rangle . Q_j \\ &\stackrel{M1}{=} Q + [M_i] [N_j] \langle z_j | y_j \rangle . Q_j \\ &\stackrel{SD2}{=} Q \end{aligned}$$

In the second case it can also be proved that $Q + [M_i] \langle z_i | y_i \rangle . P_i = Q$.

Similarly one shows that $AS \vdash Q + [M_i] \alpha_i x_i . P_i = Q$ for every $i \in I_1$ and $AS \vdash Q + [M_i] (x) \alpha_i x . P_i = Q$ for every $i \in I_2$. Conclude $AS \vdash P + Q = P$. Similarly $AS \vdash P + Q = Q$. Hence $AS \vdash P = Q$. \square

5.2 Weak Case

Complete systems for L -congruence relations are obtained by extending the system for strong bisimilarity. We need the χ -version of the well-known tau laws as given in Figure 4. The simple forms of T1 through T3 are enough since TD1 through TD4 are derivable from T1 and T3. As we mentioned in the introduction, we need a new tau law to deal with mobility. The following lemma demonstrates the power of this new tau law.

T1	$\alpha x.\tau.P$	$=$	$\alpha x.P$
T2	$P + \tau.P$	$=$	$\tau.P$
T3	$\alpha x.(P + \tau.Q)$	$=$	$\alpha x.(P + \tau.Q) + \alpha x.Q$
T4	$\tau.P$	$=$	$\tau.(P + [x=y]\tau.P)$
TD1	$\langle x y \rangle.\tau.P$	$=$	$\langle x y \rangle.P$
TD2	$\tau.\tau.P$	$=$	$\tau.P$
TD3	$\langle x y \rangle.(P + \tau.Q)$	$=$	$\langle x y \rangle.(P + \tau.Q) + \langle x y \rangle.Q$
TD4	$\tau.(P + \tau.Q)$	$=$	$\tau.(P + \tau.Q) + \tau.Q$

Figure 4: The Tau Laws

P1	$(z)az.(P + \langle x z \rangle.Q)$	$=$	$(z)az.(P + \langle x z \rangle.Q) + ax.Q\{x/z\}$	$x \neq z$
P2	$(z)\bar{a}z.(P + \langle x z \rangle.Q)$	$=$	$(z)\bar{a}z.(P + \langle x z \rangle.Q) + \bar{a}x.Q\{x/z\}$	$x \neq z$

Figure 5: The Prefix Laws

Lemma 27. $AS \cup \{T1, T2, T3, T4\} \vdash \tau.P = \tau.(P + \sum_{i \in I} [M_i]\tau.P)$ for a finite nonempty indexing set I .

Proof. First we prove that $AS \cup \{T1, T2, T3, T4\} \vdash \tau.P = \tau.(P + [M]\tau.P)$. When M is empty, the result follows from T1 and T2. So the base case holds. Now observe that

$$\begin{aligned}
\tau.P &\stackrel{T4}{=} \tau.(P + [x=y]\tau.P) \\
&\stackrel{SD2}{=} \tau.(P + [M][x=y]\tau.P + [x=y]\tau.P) \\
&\stackrel{I.H.}{=} \tau.(P + [M][x=y]\tau.P + [x=y]\tau.(P + [M]\tau.P)) \\
&\stackrel{M2}{=} \tau.(P + [M][x=y]\tau.P + [x=y]\tau.(P + [M]\tau.P)\{y/x\}) \\
&\stackrel{M1}{=} \tau.(P + [M][x=y]\tau.P + [x=y]\tau.(P + [M][x=y]\tau.P)\{y/x\}) \\
&\stackrel{M2}{=} \tau.(P + [M][x=y]\tau.P + [x=y]\tau.(P + [M][x=y]\tau.P)) \\
&\stackrel{T4}{=} \tau.(P + [M][x=y]\tau.P)
\end{aligned}$$

So the induction step can be established.

The above result provides the base case of the lemma, and the following derivation provides the induction step:

$$\begin{aligned}
\tau.(P + \sum_{i \in I} [M_i]\tau.P + [M]\tau.P) &\stackrel{I.H.}{=} \tau.(P + \sum_{i \in I} [M_i]\tau.P + [M]\tau.(P + \sum_{i \in I} [M_i]\tau.P)) \\
&= \tau.(P + \sum_{i \in I} [M_i]\tau.P) \\
&\stackrel{I.H.}{=} \tau.P
\end{aligned}$$

where the second equality holds by the above result. \square

The tau laws are enough to characterize $=_{fa \cup \bar{f}a}$. To get complete systems for the other three L -congruence relations, we need what we call prefix laws given in Figure 5. Notice that

$$\begin{aligned}
(z)az.(P + \langle x|z \rangle.Q) &\neq_{fa} (z)az.(P + \langle x|z \rangle.Q) + ax.Q\{x/z\} \\
(z)az.(P + \langle x|z \rangle.Q) &=_{\bar{f}a} (z)az.(P + \langle x|z \rangle.Q) + ax.Q\{x/z\} \\
(z)\bar{a}z.(P + \langle x|z \rangle.Q) &\neq_{\bar{f}a} (z)\bar{a}z.(P + \langle x|z \rangle.Q) + \bar{a}x.Q\{x/z\} \\
(z)\bar{a}z.(P + \langle x|z \rangle.Q) &=_{fa} (z)\bar{a}z.(P + \langle x|z \rangle.Q) + \bar{a}x.Q\{x/z\}
\end{aligned}$$

The counter examples given in the proof of (i) of Theorem 12 are special instances of P1 and P2.

The following lemma is crucial to the simple proof of completeness theorem. The properties it describes are usually called saturation properties.

Lemma 28. *Suppose Q is in normal form and M is a sequence of match equalities. Then*

- (i) if $Q\sigma_M \xrightarrow{\tau} Q'$ then $AS \cup \{T1, T2, T3\} \vdash Q = Q + [M]\tau.Q'$;
- (ii) if $Q\sigma_M \xrightarrow{\alpha x} Q'$ then $AS \cup \{T1, T2, T3\} \vdash Q = Q + [M]\alpha x.Q'$;
- (iii) if $Q\sigma_M \xrightarrow{\alpha(z)} Q'$ then $AS \cup \{T1, T2, T3\} \vdash Q = Q + [M](z)\alpha z.Q'$;
- (iv) if $Q\sigma_M \xrightarrow{x/y} Q'$ then $AS \cup \{T1, T2, T3\} \vdash Q = Q + [M]\langle x|y \rangle.Q'$;
- (v) if $Q\sigma_M \xrightarrow{m(z) x/z} Q'$ then $AS \cup \{T1, T2, T3\} \cup \{P1\} \vdash Q = Q + [M]mx.Q'$;
- (vi) if $Q\sigma_M \xrightarrow{\bar{m}(z) x/z} Q'$ then $AS \cup \{T1, T2, T3\} \cup \{P2\} \vdash Q = Q + [M]\bar{m}x.Q'$;
- (vii) if $Q\sigma_M \xrightarrow{\alpha(z) x/z} Q'$ then $AS \cup \{T1, T2, T3\} \cup \{P1, P2\} \vdash Q = Q + [M]\alpha x.Q'$.

Proof. (i) through (iii) are proved by induction.

(iv) Suppose $[N_j]\langle x_j|y_j \rangle.Q_j$ is a summand of Q such that $M \Rightarrow N_j$ and $Q\sigma_M \xrightarrow{x/y} Q'$ is caused by $([N_j]\langle x_j|y_j \rangle.Q_j)\sigma_M \xrightarrow{x_j\sigma_M/y_j\sigma_M} Q_j\sigma_M\{x_j\sigma_M/y_j\sigma_M\}$. Then $Q' \equiv Q_j\sigma_M\{x_j\sigma_M/y_j\sigma_M\}$, $x = x_j\sigma_M$ and $y = y_j\sigma_M$. Therefore

$$\begin{aligned}
AS \cup \{T1, T2, T3\} \vdash Q + [M]\langle x|y \rangle.Q' &= Q + [M]\langle x_j\sigma_M|y_j\sigma_M \rangle.Q_j\sigma_M\{x_j\sigma_M/y_j\sigma_M\} \\
&\stackrel{UD1}{=} Q + [M]\langle x_j\sigma_M|y_j\sigma_M \rangle.Q_j\sigma_M \\
&\stackrel{MD3}{=} Q + [M]\langle x_j|y_j \rangle.Q_j \\
&\stackrel{M1}{=} Q + [M][N_j]\langle x_j|y_j \rangle.Q_j \\
&\stackrel{SD2}{=} Q
\end{aligned}$$

If for example $Q\sigma_M \xrightarrow{\tau} Q_1 \xrightarrow{x/y} Q_2 \xrightarrow{\tau} Q'$ then by (i)

$$\begin{aligned}
AS \cup \{T1, T2, T3\} \vdash Q &= Q + [M]\tau.Q_1 \\
&= Q + [M]\tau.(Q_1 + \langle x|y \rangle.Q_2) \\
&\stackrel{T2}{=} Q + [M](\tau.(Q_1 + \langle x|y \rangle.Q_2) + \langle x|y \rangle.Q_2) \\
&\stackrel{M3}{=} Q + [M]\tau.(Q_1 + \langle x|y \rangle.Q_2) + [M]\langle x|y \rangle.Q_2 \\
&= Q + [M]\langle x|y \rangle.Q_2 \\
&= Q + [M]\langle x|y \rangle.(Q_2 + \tau.Q') \\
&\stackrel{TD3}{=} Q + [M](\langle x|y \rangle.(Q_2 + \tau.Q') + \langle x|y \rangle.Q') \\
&\stackrel{M3}{=} Q + [M]\langle x|y \rangle.(Q_2 + \tau.Q') + [M]\langle x|y \rangle.Q' \\
&= Q + [M]\langle x|y \rangle.Q'
\end{aligned}$$

This completes the proof of (iv).

(v) Suppose $Q\sigma_M \xrightarrow{m(z)} Q_1 \xrightarrow{x/z} Q'$. Then by (iii) and (iv)

$$AS \cup \{T1, T2, T3\} \vdash Q = Q + [M](z)mz.Q_1$$

and

$$AS \cup \{T1, T2, T3\} \vdash Q_1 = Q_1 + \langle x|z \rangle.Q'$$

Therefore

$$\begin{aligned}
Q &= Q + [M](z)mz.(Q_1 + \langle x|z \rangle.Q') \\
&\stackrel{P1}{=} Q + [M]((z)mz.(Q_1 + \langle x|z \rangle.Q') + mx.Q') \\
&\stackrel{M3}{=} Q + [M](z)mz.(Q_1 + \langle x|z \rangle.Q') + [M]mx.Q' \\
&= Q + [M]mx.Q'
\end{aligned}$$

This completes the proof of (v).

(vi) through (vii) are proved similarly. □

Using the saturation properties, we can establish the promotion properties, so called because the properties allows one to lift a pair, P and Q , of semantically equivalent processes to a pair, $\tau.P$ and $\tau.Q$, of proof theoretically equal terms.

Lemma 29. *Suppose both P and Q are in normal form. The following properties hold:*

- (i) *If $P \approx_{fa\cup\bar{fa}} Q$ then $AS \cup \{T1, T2, T3, T4\} \vdash \tau.P = \tau.Q$;*
- (ii) *If $P \approx_{\bar{fa}} Q$ then $AS \cup \{T1, T2, T3, T4\} \cup \{P1\} \vdash \tau.P = \tau.Q$;*
- (iii) *If $P \approx_{fa} Q$ then $AS \cup \{T1, T2, T3, T4\} \cup \{P2\} \vdash \tau.P = \tau.Q$;*
- (iv) *If $P \approx_{ba} Q$ then $AS \cup \{T1, T2, T3, T4\} \cup \{P1, P2\} \vdash \tau.P = \tau.Q$.*

Proof. (i) The proof is done by induction on the sum of the depths of P and Q . Suppose P and Q are in normal form. Let P be

$$\sum_{i \in I_1} [M_i] \alpha_i x_i . P_i + \sum_{i \in I_2} [M_i] (x) \alpha_i x . P_i + \sum_{i \in I_3} [M_i] \langle z_i | y_i \rangle . P_i$$

and Q be

$$\sum_{j \in J_1} [N_j] \alpha_j x_j . Q_j + \sum_{j \in J_2} [N_j] (x) \alpha_j x . Q_j + \sum_{j \in J_3} [N_j] \langle z_j | y_j \rangle . Q_j$$

Suppose further that $P \approx_{fa\cup\bar{fa}} Q$ and $[M_i](x) \alpha_i x . P_i$ is 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 $P_i \sigma_{M_i} \approx_{fa\cup\bar{fa}} Q'$. Both $P_i \sigma_{M_i}$ and Q' are in normal form and $d(P_i \sigma_{M_i}) + d(Q') \leq d(P) + d(Q)$. So $AS \cup \{T1, T2, T3, T4\} \vdash \tau.P_i \sigma_{M_i} = \tau.Q'$ by induction hypothesis. It follows that

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

Therefore $[M_i](x) \alpha_i x . P_i + Q = [M_i](x) \alpha_i \sigma_{M_i} x . Q' + Q = Q$ by (iii) of Lemma 28. Similarly one proves that $[M_i] \alpha_i x_i . P_i + Q = Q$ whenever $[M_i] \alpha_i x_i . P_i$ is a summand of P .

Now suppose $[M_i] \langle x_i | y_i \rangle . P_i$ is a summand of P . Then there are two cases. Either

$$([M_i] \langle x_i | y_i \rangle . P_i) \sigma_{M_i} \xrightarrow{x_i \sigma_{M_i} / y_i \sigma_{M_i}} P_i \sigma_{M_i} \{x_i \sigma_{M_i} / y_i \sigma_{M_i}\}$$

or

$$([M_i] \langle x_i | y_i \rangle . P_i) \sigma_{M_i} \xrightarrow{\tau} P_i \sigma_{M_i}$$

In the first case one has some Q' such that $Q \sigma_{M_i} \xrightarrow{x_i \sigma_{M_i} / y_i \sigma_{M_i}} Q'$ and $P_i \sigma_{M_i} \{x_i \sigma_{M_i} / y_i \sigma_{M_i}\} \approx_{fa\cup\bar{fa}} Q'$. By induction hypothesis $AS \cup \{T1, T2, T3, T4\} \vdash \tau.P_i \sigma_{M_i} \{x_i \sigma_{M_i} / y_i \sigma_{M_i}\} = \tau.Q'$. So

$$\begin{aligned} [M_i] \langle x_i | y_i \rangle . P_i &\stackrel{MD3}{=} [M_i] \langle x_i \sigma_{M_i} | y_i \sigma_{M_i} \rangle . P_i \sigma_{M_i} \\ &\stackrel{UD1}{=} [M_i] \langle x_i \sigma_{M_i} | y_i \sigma_{M_i} \rangle . P_i \sigma_{M_i} \{x_i \sigma_{M_i} / y_i \sigma_{M_i}\} \\ &\stackrel{T1}{=} [M_i] \langle x_i \sigma_{M_i} | y_i \sigma_{M_i} \rangle . \tau . P_i \sigma_{M_i} \{x_i \sigma_{M_i} / y_i \sigma_{M_i}\} \\ &\stackrel{I.H.}{=} [M_i] \langle x_i \sigma_{M_i} | y_i \sigma_{M_i} \rangle . \tau . Q' \\ &\stackrel{T1}{=} [M_i] \langle x_i \sigma_{M_i} | y_i \sigma_{M_i} \rangle . Q' \end{aligned}$$

Therefore $[M_i] \langle x_i | y_i \rangle . P_i + Q = [M_i] \langle x_i \sigma_{M_i} | y_i \sigma_{M_i} \rangle . Q' + Q = Q$ by (iv) of Lemma 28. In the second case some Q' exists such that $P_i \sigma_{M_i} \{x_i \sigma_{M_i} / y_i \sigma_{M_i}\} \approx_{fa\cup\bar{fa}} Q'$ and either $Q \sigma_{M_i} \xrightarrow{\tau} Q'$ or $Q \sigma_{M_i} \equiv Q'$. By

induction hypothesis $\tau.P_i\sigma_{M_i}\{x_i\sigma_{M_i}/y_i\sigma_{M_i}\} = \tau.Q'$. In the first subcase it can be easily shown that $[M_i]\langle x_i|y_i\rangle.P_i + Q = Q$. In the second subcase one has

$$\begin{aligned}
[M_i]\langle x_i|y_i\rangle.P_i &\stackrel{MD3}{=} [M_i]\langle x_i\sigma_{M_i}|y_i\sigma_{M_i}\rangle.P_i\sigma_{M_i} \\
&\stackrel{UD1}{=} [M_i]\langle x_i\sigma_{M_i}|y_i\sigma_{M_i}\rangle.P_i\sigma_{M_i}\{x_i\sigma_{M_i}/y_i\sigma_{M_i}\} \\
&\stackrel{TD1}{=} [M_i]\langle x_i\sigma_{M_i}|y_i\sigma_{M_i}\rangle.\tau.P_i\sigma_{M_i}\{x_i\sigma_{M_i}/y_i\sigma_{M_i}\} \\
&\stackrel{I.H.}{=} [M_i]\langle x_i\sigma_{M_i}|y_i\sigma_{M_i}\rangle.\tau.Q' \\
&\stackrel{TD1}{=} [M_i]\langle x_i\sigma_{M_i}|y_i\sigma_{M_i}\rangle.Q\sigma_{M_i} \\
&\stackrel{U3}{=} [M_i]\tau.Q\sigma_{M_i} \\
&\stackrel{MD3}{=} [M_i]\tau.Q
\end{aligned}$$

It follows that $AS \cup \{T1, T2, T3, T4\} \vdash P+Q = Q + \sum_{i \in I} [M_i]\tau.Q$ for some finite indexing sets I . By Lemma 27, $AS \cup \{T1, T2, T3, T4\} \vdash \tau.(P+Q) = \tau.(Q + \sum_{i \in I} [M_i]\tau.Q) = \tau.Q$. Similarly one can prove that $AS \cup \{T1, T2, T3, T4\} \vdash \tau.(P+Q) = \tau.P$. Hence $AS \cup \{T1, T2, T3, T4\} \vdash \tau.P = \tau.Q$. \square

Theorem 30 (completeness). *We have the following completeness results:*

- (i) $AS \cup \{T1, T2, T3, T4\}$ is sound and complete for $=_{fa \cup \overline{fa}}$;
- (ii) $AS \cup \{T1, T2, T3, T4\} \cup \{P1\}$ is sound and complete for $=_{\overline{fa}}$;
- (iii) $AS \cup \{T1, T2, T3, T4\} \cup \{P2\}$ is sound and complete for $=_{fa}$;
- (iv) $AS \cup \{T1, T2, T3, T4\} \cup \{P1, P2\}$ is sound and complete for $=_{ba}$.

Proof. The proof of completeness is almost the same as that of Lemma 29. The only difference is that in the inductive step, one uses Lemma 29 rather than induction hypothesis! For illustration, let's assume that $P =_{fa \cup \overline{fa}} Q$ for normal form processes P and Q . Suppose $[M_i]\langle z_i|y_i\rangle.P_i$ is a summand of P and $([M_i]\langle z_i|y_i\rangle.P_i)\sigma_{M_i} \xrightarrow{\tau} P_i\sigma_{M_i}$. Then $Q\sigma_{M_i} \xrightarrow{\tau} Q' \approx_{fa \cup \overline{fa}} P'$. By Lemma 29, $AS \cup \{T1, T2, T3, T4\} \vdash \tau.P' = \tau.Q'$, from which it is routine to show that $AS \cup \{T1, T2, T3, T4\} \vdash [M_i]\langle z_i|y_i\rangle.P_i + Q = Q$. This should provide enough information about how to establish $AS \cup \{T1, T2, T3, T4\} \vdash P+Q = Q$. Similarly $AS \cup \{T1, T2, T3, T4\} \vdash P+Q = P$. Hence $AS \cup \{T1, T2, T3, T4\} \vdash P = Q$. \square

6 Conclusion

The paper has achieved two things. First a general framework is introduced to give a possible classification of bisimulation congruence relations. The L -bisimilarity covers all familiar bisimulation equivalences. For instance the bottom element of the bisimulation lattice is the open bisimilarity and the top element of the lattice coincides with the barbed bisimilarity. This interesting phenomenon suggests that open congruence is usually the strongest congruence whereas the barbed congruence the weakest. An explanation of the phenomenon is this: In the open approach processes are supposed to simulate each other's actions to the full extent. The actions not only reveal the channels at which the actions occurs but also the messages carried by the actions. On the other extreme, the barbed approach only requires bisimilar processes to simulate each other's ability to communicate at particular channels. The contents of communications are irrelevant. The barbed bisimilarity on χ -processes satisfies some unusual equalities not known in the studies of π -calculus. It departs from the 'standard bisimilarity' significantly. The second achievement is a complete system for each of the L -congruences. This is the first time non-symbolic complete systems are derived for weak observational congruences on finite χ -like processes. There have been other attempts to axiomatize weak equivalences using non-symbolic approach. Unfortunately all of them have been unsuccessful.

The approach used in this paper has strong implications to other calculi of mobile processes. The asymmetric χ -calculus, the Update Calculus without the mismatch combinator, can be investigated in similar line. The situation is complicated by the presence of restricted updates in the asymmetric language. Preliminary results are reported in [10]. The study of polyadic χ -calculus, the Fusion Calculus without the mismatch operator, can also benefit from the idea of bisimulation lattice. The top element of the bisimulation lattice for Fusion processes is the barbed bisimilarity, which is strictly weaker than the weak hyperbisimilarity, the latter being the same as the open bisimilarity in the fragment of Fusion Calculus without the mismatch operator. Complete axiomatization systems for barbed equivalence and

hyperequivalence on Fusion processes without the mismatch combinator can be given using T4 and some versions of P-rules as well as the rules given in [22].

The result of this paper should be of help to the investigation of a non-symbolic approach to axiomatization of π -calculus. The introduction of T4 allows us to promote the complete system for the strong open congruence on π -processes to one for the weak open congruence. However this system still has to deal with distinctions. This use of distinction is unsatisfactory because it is a disguised form of symbolic treatment. So one avenue for further study is to see how to eliminate distinctions from the system.

The relationship of this piece of work to the work of Parrow and Victor is as follows:

- Parrow and Victor have studied weak hyperbisimilarity ([30]), which is basically the weak open bisimilarity. We have studied in this paper L -bisimilarities that subsume the weak hyperbisimilarity, the latter being the smallest L -bisimilarity under the subset inclusion. The largest L -bisimilarity is the barbed bisimilarity. The definition of barbed bisimilarity considered in this paper is different from that of the barbed equivalence of Milner and Sangiorgi. The latter is required to be closed under context only at the very beginning of bisimulation. The definition of the barbed bisimilarity makes it easy to compare to the L -bisimilarities. Our intuition strongly suggests that the barbed bisimilarity and the barbed equivalence, which Parrow and Victor have studied, coincide in the absence of the mismatch operator. But so far we have not been able to formally prove this conjecture. In the setting of π -calculus, Sangiorgi and Walker have introduced open barbed bisimilarity which is closed under substitution in every bisimulation step. They have proved that the open barbed bisimilarity coincides with quasi open bisimilarity that is slightly different from the well known open bisimilarity. For detail confer [26]. It is clear that the barbed bisimilarity is contained in the barbed equivalence. Therefore the barbed equivalence is strictly weaker than the weak hyperequivalence. This result rectifies a mistaken result in [30].
- Parrow and Victor have studied complete axiomatic system for the strong hyperequivalence on fusion processes ([22]). The results in Section 5.1 are special cases of their results. In the same paper they have tried to give a complete axiomatic system for the weak hyperequivalence. But the system they came up with is incorrect because it lacks of the tau law T4. The fourth tau law plays an essential role in the semantics of open style congruences.

Acknowledgement

The work reported in this paper has been supported by the National Nature Science Foundation of China (69873032) and by the National 863 High-Tech Project (863-306-ZT06-02-2).

The author wishes to thank two anonymous referees for their criticism and suggestions. He also wishes to thank Zhenrong Yang for proof-reading the paper.

References

- [1] R. Amadio, I. Castellani, D. Sangiorgi. On Bisimulation for the Asynchronous π -Calculus. *CONCUR'96*, Lecture Notes in Computer Science 1119, Springer, 1996.
- [2] M. Boreale. On the Expressiveness of Internal Mobility in Name-Passing Calculi. *CONCUR '96*, Lecture Notes in Computer Science 1119, Springer, 161–178, 1996.
- [3] G. Boudol. Asynchrony and the π -calculus. Research Report 1702, INRIA, Sophia-Antipolis, 1992.
- [4] Y. Fu. The χ -Calculus. *Proceedings of the International Conference on Advances in Parallel and Distributed Computing*, 74-81, March 19th-21th, Shanghai, IEEE Computer Society Press, 1997.
- [5] Y. Fu. A Proof Theoretical Approach to Communications. *ICALP'97*, 325–335, July 7th-11th, Bologna, Italy, Lecture Notes in Computer Science 1256, Springer, 325-335, 1997.
- [6] Y. Fu. Symmetric π -Calculus. *Journal of Computer Science and Technology*, **13**: 202–208, 1998.
- [7] Y. Fu. Reaction Graphs. *Journal of Computer Science and Technology*, **13**: 510–530, 1998.
- [8] Y. Fu. Bisimulation Lattice of Chi Processes. *ASIAN'98*, December 8-10, Manila, The Philippines, Lecture Notes in Computer Science 1538, Springer, 245–262, 1998.

- [9] Y. Fu. Variations on Mobile Processes. *Theoretical Computer Science*, **221**: 327-368, 1999.
- [10] Y. Fu. Open Bisimulations of Chi Processes. *CONCUR'99*, Eindhoven, August 24-27, Lecture Notes in Computer Science 1664, Springer, 304-319, 1999.
- [11] M. Hennessy, R. Milner. Algebraic Laws for Nondeterminism and Concurrency. *Journal of ACM*, **67**: 137-161, 1985.
- [12] K. Honda, M. Tokoro. An Object Calculus for Asynchronous Communication. *Proc. ECOOP '91*, Geneva, 1991.
- [13] H. Lin. Complete Inference Systems for Weak Bisimulation Equivalences in the π -Calculus. *Proceedings of Sixth International Joint Conference on the Theory and Practice of Software Development*, Lecture Notes in Computer Science 915, Springer, 187-201, 1995.
- [14] H. Lin. Unique Fixpoint Induction for Mobile Processes. *CONCUR '95*, Lecture Notes in Computer Science 962, Springer, 88-102, 1995.
- [15] H. Lin. Complete Proof Systems for Observation Congruences in Finite-Control π -Calculus. *ICALP '98*, Lecture Notes in Computer Science 1443, Springer, 443-454, 1998.
- [16] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.
- [17] R. Milner, J. Parrow, D. Walker. A Calculus of Mobile Processes. *Information and Computation*, **100**: 1-40 (Part I), 41-77 (Part II), Academic Press.
- [18] R. Milner, D. Sangiorgi. Barbed Bisimulation. *19th ICALP*, Lecture Notes in Computer Science 623, Springer, 685-695, 1992.
- [19] M. Merro, D. Sangiorgi. On Asynchrony in Name-Passing Calculi. *ICALP'98*, Lecture Notes in Computer Science 1443, Springer, 1998.
- [20] J. Parrow, B. Victor. The Update Calculus. *AMAST'97*, Sydney, December 13-17, 1997.
- [21] J. Parrow, B. Victor. The Fusion Calculus: Expressive and Symmetry in Mobile Processes. *LICS'98*, 1998.
- [22] J. Parrow, B. Victor. The Tau-Laws of Fusion. *CONCUR'98*, Lecture Notes in Computer Science, 1998.
- [23] D. Sangiorgi. *Expressing Mobility in Process Algebras: First-Order and Higher-Order Paradigms*, PhD thesis, Department of Computer Science, University of Edinburgh, 1993.
- [24] D. Sangiorgi. A Theory of Bisimulation for π -Calculus. *CONCUR 93*, Lecture Notes in Computer Science 715, Springer, 1993.
- [25] D. Sangiorgi. π -Calculus, Internal Mobility and Agent-Passing Calculi. *Theoretical Computer Science*, 235-274, 1996
- [26] D. Sangiorgi, D. Walker. On Barbed Equivalence in π -Calculus. *CONCUR'01*, 292-304, Lecture Notes in Computer Science, 2001.
- [27] B. Thomsen. A Calculus of Higher Order Communicating Systems. *POPL '89*, ACM, 143-154, 1989.
- [28] B. Thomsen. Plain CHOCS—A Second Generation Calculus for Higher Order Processes. *Acta Informatica*, **30**: 1-59, 1993.
- [29] B. Thomsen. A Theory of Higher Order Communicating Systems. *Information and Computation*, **116**: 38-57, 1995.
- [30] B. Victor, J. Parrow. Concurrent Constraints in the Fusion Calculus. *ICALP'98*, Lecture Notes in Computer Science, 1998.
- [31] D. Walker. π -Calculus Semantics for Object-Oriented Programming Languages, *TACS '91*, Lecture Notes in Computer Science 526, Springer, 532-547, 1991.
- [32] D. Walker. Objects in the π -Calculus, *Information and Computation*, **116**: 253-271, 1995.