On the Bisimulation Congruence in χ-Calculus*
(Extended Abstract)

Taolue Chen[1,2]**, Tingting Han[2], and Jian Lu[2]

[1] CWI, Department of Software Engineering,
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
Taolue.Chen@cwi.nl
[2] State Key Laboratory of Novel Software Technology,
Nanjing University, Nanjing, Jiangsu, P.R.China, 210093

Abstract. In this paper, we study weak bisimulation congruences for the χ-calculus, a symmetric variant of the π-calculus. We distinguish two styles of such bisimulation definitions, i.e. "open" and "closed" bisimulation, the difference between which lies in that in open style the equivalence is closed under context in every bisimulation step whereas in closed style the equivalence is closed under context only at the very beginning. As a result, we show that both in labelled and barbed congruence, the open and closed style definitions coincide. Thus all bisimulation congruences collapse into two equivalences, that is, the well-known open congruence and open barbed congruence, which are the same in the strong case, while in the weak case their difference can be reflected by one axiom. The results of this paper close some conjectures in the literatures and shed light on the algebraic theory of a large class of mobile process calculi.

1 Introduction

Over the last decade, various calculi of mobile processes, notably the π-calculus [11], have been the focus of research in concurrency theory. Since 1997, several publications have focused on a class of new calculi of mobile process. These models include χ-calculus [5] due to Fu, update calculus [13] and fusion calculus [14] due to Parrow and Victor with its variants, such as explicit fusion [8], due to Gardner and Wischik. Roughly speaking, in a uniform terminology they are respectively χ-calculus, asymmetric χ-calculus and polyadic χ-calculus.

In the research of algebraic theory for mobile process, bisimulation equivalence is the standard paradigm for behavioral comparison. Comparing to the traditional process calculi, e.g. CCS [10], for mobile processes, there are often

* This work is partially supported by NNSFC (60233010, 60273034, 60403014) and 973 Program of China (2002CB312002).
** Corresponding author. The author is partially supported by the Dutch BSIK/ BRICKS Project (Basic Research in Informatics for Creating the Knowledge Society).

R. Ramanujam and S. Sen (Eds.): FSTTCS 2005, LNCS 3821, pp. 128–139, 2005.

many natural definitions of bisimilarity, which makes the theory much more involved. For example, in the π-calculus, the most well-known of them include late/early bisimulation [11], open bisimulation [15], barbed bisimulation [12], etc. It is widely recognized that a bisimulation equivalence is most useful when it is a *congruence*, i.e. is preserved by the syntactic constructions of the calculus. Unfortunately, in mobile process calculi, most of bisimulation equivalences are not congruences themselves! This gives rise to the problem on how to refine the bisimulation definition and thus obtain a congruence. It is well-known that congruence relations on mobile processes should be closed under *substitution* on names, which gives rise to 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. For example, open equivalence [15] is closed under substitution in each bisimulation step, while early, late [11] and barbed equivalences [12] are closed under substitution only in the first step of bisimulation. This distinction makes open bisimulation strictly stronger than the other three (please note that this is only the case in the π-calculus, as the results of this paper will suggest, in the χ-calculus, the situation is quite different).

In the light of the above discussion, we argue that "open" is indeed a *general* definition style while it is not only a *single* or *ad hoc* definition. Let us generalize the above mentioned "substitution" to a broader notion of *context*. Remarkably, there are at least two reasonable ways of ensuring the congruence property:

- Either take the largest congruence that is a bisimulation; this is the "reduction based" equivalence chosen for the ν-calculus in [9] and Abadi and Fournet's work, e.g. [1]. In this paper, we will call it "open" style, following Sangiorgi [15]. This models the situation where environments change during execution (the norm in distributed computation).
- Or take the largest congruence included in the bisimulation; this is the two-stage definition traditionally chosen for CCS and the π-calculus; for symmetry, in this paper, we will call it "closed" style. This models the situation where a sub-program's context is fixed at compile time. They are generally just called "congruence" in the literature on π-calculus, e.g. [11].

We also can study the bisimulation relations in mobile process calculi from another perspective, that is, we can distinguish the labelled and the barbed style. Since we believe this distinction is much more familiar to the readers, we will not explain it further. To summarize, in our opinion, there are actually four key ways to define behavioral equivalences in mobile process calculi: depending on whether the relation is closed under initial contexts ("closed" style) or under subsequently-changing contexts as well (reduction-closed congruence, open style); and orthogonally whether we just observe the channels over which messages are sent (barbed style) or also record the message and the resulting state (labelled style). Clearly, the combinations give rise to (at least) four sensible bisimulation equivalences. Three of them are familiar in the community and have been mentioned above. The remaining one, that is, the combination of "open" and "barbed", is open barbed bisimulation, which is also extensively studied by Sangiorgi and Walker [16] and the first author in [2] in recent days.

So far, most of the bisimulation congruences for χ-calculus, notably [6], are defined in the open style, while in the research on mobile process calculi, the closed style definitions seem to be more standard. In our opinion, this is not very ideal for our understanding of the χ-calculus, the representative of a large class of mobile process calculi. Under such a situation, one of the contributions of this paper is to show how the four standard definitions of bisimulation congruence, familiar from the π-calculus, can be applied to the χ-calculus. Based on it, we study the relationships of these bisimulation equivalences. Intuitively, one might expect the closed and open congruences to generate the same relation, since one could presumably write an initial environment sophisticated enough to model a subsequently-changing environment. Unfortunately, this result does not hold for the (synchronous) π-calculus. Interestingly, if we restrict to the asynchronous π-calculus, the analogous results hold! See Fournet and Gonthier's work [4] for more details and we will discuss it further in Section 5. Now, a natural question is: What's the situation in the χ-calculus? The main contribution of this paper is to provide a systematic investigation on this. We compare the open and closed style bisimulations in the setting of χ-calculus, and study both labelled and barbed versions. Moreover, we focus on the weak version of bisimulation, since it is much more general and difficult. Our results show that for both labelled and barbed bisimulation, the closed and open style definitions coincide. This is mainly, in our point of view, due to the fact that in χ-calculus, we can simulate the notion of substitution through the parallel operator. It is worth pointing out that Fu says ([6], pp. 225): "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." One of the results in this paper confirms Fu's conjecture (see Theorem 2 in Section 4), thus we close this open problem. Moreover, our results also support Fu, Parrow, Victor's general arguments that in χ-like process algebra, open bisimulation is more natural.

We note that in [18] Wischik and Gardner have performed closely related research. However, modulo the differences on the underlying process calculi, there are some dramatic differences: (1) We discuss the weak case of bisimulation, while [18] only considers the strong case. They claim that weak bisimulation congruences have been studied by Fu for the χ-calculus. However, as we have pointed out, this is not the case, since Fu only considers the "open" style relations. Under such a situation, our results can be regarded as the weak counterpart of [18]. Moreover, [18] mentions that "An interesting open problem is to explore such congruences[3] for the explicit fusion calculus ...". We believe the results of the current paper will, at least, shed light on such issues, due to the similarity of χ-calculus and explicit fusion. (2) [18] only proves the coincide of "open" and "closed" (they call them reduction-based and shaped) style for labelled bisimulation. As for the barbed case, they apply Sangiorgi's "stratification" technique [17] to show the coincidence of ground equivalence and barbed equivalence, then they show (trivially) the coincidence of open barbed congruence and open con-

[3] Here, they are essentially referring to the weak open congruences studied by Fu.

gruence, thus the result is obtained. This is feasible in the strong case. However, in the weak case, as we will see, labelled bisimulation and barbed bisimulation do not coincide at all! To deal with this, we give a direct proof for the coincide of "open" and "closed" style for barbed bisimulation, which is much more complicated.

We should point out that the bisimulation congruences studied in this paper are all "partial" congruences since they are not closed under summation (technically speaking, they are closed under *context* but fail under *full context*, see Definition 1). Since this problem is common and Milner [10] has provided an elegant way to deal with this, it is not a true drawback. We also note that this is only an extended abstract, since due to space restriction, all the proofs have to be omitted. For more detailed proofs, explanations, remarks, we refer the interested readers to our technical report [3].

The structure of this paper is as follows: Section 2 summarizes some background material on χ-calculus. Section 3 presents the results on label style bisimulation while Section 4 discusses barbed style bisimulations. This paper is concluded in Section 5, where some remarks are also given.

2 Background

In this section, we will review some background material for χ-calculus, we refer the reader to [5] [6] for more details. Let $\mathcal{N}$ be a set of names, ranged over by lower case letters. $\bar{\mathcal{N}}$, the set of conames, denotes $\{\bar{x} \mid x \in \mathcal{N}\}$. The set $\mathcal{N} \cup \bar{\mathcal{N}}$ will be ranged over by α. Let $\bar{\alpha}$ be a if $\alpha = \bar{a}$ and $\bar{a}$ if $\alpha = a$.

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

$$P := 0 \mid \alpha x.P \mid P|P \mid (\nu x)P \mid [x = y]P \mid P + P \mid !P$$

The intuitional sense is standard. The name x in $(\nu x)P$ is bound. A name is free in P if it is not bound in P. The free names, the bound names and names of P, as well as the notations $fn(P)$, $bn(P)$ and $n(P)$, are used in their standard meanings. In the sequel we will use the functions fn(-),bn(-) and n(-) without explanation. 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. And in any discussion we assume that the bound names of any processes or actions under consideration are chosen to be different from the names free in any other entities under consideration, such as processes, actions, substitutions, and set of names. As a convention, we often abbreviate $\alpha x.0$ simply as αx. Moreover, sometimes a communication needs to carry no parameter when the passed names are unimportant. To model this, we will usually write $\alpha.P$ for $\alpha x.P$ where $x \notin fn(P)$.

A *context* is a process with a hole. Now, we give a formal definition as follows.

Definition 1. Contexts *are defined inductively as follows:*

(i) $[]$ *is a context.*
(ii) If $C[]$ *is a context then* $\alpha x.C[], P|C[], C[]|P, (\nu x)C[], [x = y]C[]$ *are contexts.*

Full contexts *are those contexts that satisfy additionally:*
(iii) If $C[]$ *is a context then* $P + C[], C[] + P$ *are contexts.*

The operational semantics is defined by the following labelled transition system:

$$\text{Sqn} \quad \frac{}{\alpha x.P \xrightarrow{\alpha x} P} \qquad \text{Sum} \quad \frac{P \xrightarrow{\lambda} P'}{P + Q \xrightarrow{\lambda} P'}$$

$$\text{Cmp0} \quad \frac{P \xrightarrow{\gamma} P' \quad bn(\gamma) \cap fn(Q) = \emptyset}{P|Q \xrightarrow{\gamma} P'|Q} \qquad \text{Cmp1} \quad \frac{P \xrightarrow{y/x} P'}{P|Q \xrightarrow{y/x} P'|Q\{y/x\}}$$

$$\text{Cmm0} \quad \frac{P \xrightarrow{\alpha(x)} P' \quad Q \xrightarrow{\bar{\alpha}y} Q'}{P|Q \xrightarrow{\tau} P'\{y/x\}|Q'} \qquad \text{Cmm1} \quad \frac{P \xrightarrow{\alpha(x)} P' \quad Q \xrightarrow{\bar{\alpha}(x)} Q'}{P|Q \xrightarrow{\tau} (\nu x)(P'|Q')}$$

$$\text{Cmm2} \quad \frac{P \xrightarrow{\alpha x} P' \quad Q \xrightarrow{\bar{\alpha}y} Q' \quad x \neq y}{P|Q \xrightarrow{y/x} P'\{y/x\}|Q'\{y/x\}} \qquad \text{Cmm3} \quad \frac{P \xrightarrow{\alpha x} P' \quad Q \xrightarrow{\bar{\alpha}x} Q'}{P|Q \xrightarrow{\tau} P'|Q'}$$

$$\text{Loc0} \quad \frac{P \xrightarrow{\lambda} P' \quad x \notin n(\lambda)}{(\nu x)P \xrightarrow{\lambda} (\nu x)P'} \qquad \text{Loc1} \quad \frac{P \xrightarrow{\alpha x} P' \quad x \notin \{\alpha, \bar{\alpha}\}}{(\nu x)P \xrightarrow{\alpha(x)} P'}$$

$$\text{Loc2} \quad \frac{P \xrightarrow{y/x} P'}{(\nu x)P \xrightarrow{\tau} P'} \qquad \text{Match} \quad \frac{P \xrightarrow{\lambda} P'}{[x = x]P \xrightarrow{\lambda} P'}$$

$$\text{Rep} \quad \frac{!P|P \xrightarrow{\lambda} P'}{!P \xrightarrow{\lambda} P'}$$

Note that we have omitted all the symmetric rules. In the above rules the letter γ ranges over the set $\{\alpha(x), \alpha x \mid \alpha \in \mathcal{N} \cup \bar{\mathcal{N}},\ x \in \mathcal{N}\} \cup \{\tau\}$ of non-update actions and the letter λ over the set $\{\alpha(x), \alpha x, y/x \mid \alpha \in \mathcal{N} \cup \bar{\mathcal{N}},\ x \in \mathcal{N}\} \cup \{\tau\}$ of all actions. The symbols $\alpha(x), \alpha x, y/x$ represent restricted action, free action and update action respectively. The x in $\alpha(x)$ is bounded whereas the other names are all free. We refer to [6] for more detailed explanations.

The process $P\{y/x\}$ appearing in the above structured operational semantics is obtained by substituting y for x throughout P. The notion $\{y/x\}$ is an atomic substitution of y for x. A general substitution denoted by σ, σ' etc, is the composition of atomic substitutions. The composition of zero atomic substitutions is an empty substitution, written as $\{\}$ whose effect on a process is vacuous. The result of applying σ to P is denoted by $P\sigma$.

As usual, let $\Rightarrow$ be the reflexive and transitive closure of $\xrightarrow{\tau}$, and $\overset{\tau}{\Rightarrow}$ be the composition $\Rightarrow\xrightarrow{\tau}\Rightarrow$. The relation $\overset{\hat{\lambda}}{\Rightarrow}$ is the same as $\overset{\lambda}{\Rightarrow}$ if $\lambda \neq \tau$ and is $\Rightarrow$ otherwise. A sequence of names $x_1, \ldots, x_n$ will be abbreviated as $\tilde{x}$; and consequently $(\nu x_1) \ldots (\nu x_n)P$ will be abbreviated to $(\nu \tilde{x})P$. Moreover, we will abuse the notation a little since for a finite name set $N = \{x_1, \ldots, x_n\}$, we will write $(\nu \tilde{N})P$ for $(\nu \tilde{x})P$.

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

Lemma 1. *The following two properties hold:*

(i) If $P \xrightarrow{\lambda} P'$, then $fn(P') \subseteq fn(P) \cup bn(\lambda)$.
(ii) If $P \xrightarrow{y/x} P'$, then $x \notin fn(P')$.
(iii) If $P \Rightarrow P'$, then $P\sigma \Rightarrow P'\sigma$.
(iv) If $P \stackrel{y/x}{\Rightarrow} P'$, then $P \stackrel{x/y}{\Rightarrow} P'\{x/y\}$.

3 Labelled Bisimulation

In this section, we discuss bisimulation for χ-calculus in the labelled semantics.

3.1 Closed Style Definitions

First, let us see how late bisimulation in π-calculus can be adapted to χ-calculus.

Definition 2. *Let $\mathcal{R}$ be a binary symmetric relation on $\mathcal{C}$. It is called a* late bisimulation *if whenever $P\mathcal{R}Q$ then the following properties hold:*

(i) If $P \xrightarrow{\lambda} P'$, where $\lambda = y/x, \tau$ then Q' exists such that $Q \stackrel{\hat{\lambda}}{\Rightarrow} Q'$ with $P'\mathcal{R}Q'$.
(ii) If $P \xrightarrow{\alpha x} P'$ then Q', Q'' exist such that $Q \Rightarrow\xrightarrow{\alpha x} Q''$, and for every y, $Q''\{y/x\} \Rightarrow Q'$ with $P'\{y/x\}\mathcal{R}Q'$.
(iii) If $P \xrightarrow{\alpha(x)} P'$ then Q', Q'' exist such that $Q \Rightarrow\xrightarrow{\alpha(x)} Q''$, and for every y, $Q''\{y/x\} \Rightarrow Q'$ with $P'\{y/x\}\mathcal{R}Q'$.

Late bisimilarity $\dot{\approx}_l$ *is the largest* late bisimulation.

Since we are interested in bisimulation congruence, we consider the finer equivalence obtained as bisimilarity.

Definition 3. *P and Q are* late equivalent*, written $P \approx_{cl} Q$, if for any* context *$C[]$, $C[P] \dot{\approx}_l C[Q]$.*

In a similar way, we also can adapt early bisimulation to χ-calculus.

Definition 4. *Let $\mathcal{R}$ be a binary symmetric relation on $\mathcal{C}$. It is called an* early bisimulation *if whenever $P\mathcal{R}Q$ then the following properties hold:*

(i) If $P \xrightarrow{\lambda} P'$, where $\lambda = y/x, \tau$ then Q' exists such that $Q \stackrel{\hat{\lambda}}{\Rightarrow} Q'$ with $P'\mathcal{R}Q'$.
(ii) If $P \xrightarrow{\alpha x} P'$ then for every y, Q', Q'' exist such that $Q \Rightarrow\xrightarrow{\alpha x} Q''$, and $Q''\{y/x\} \Rightarrow Q'$ with $P'\{y/x\}\mathcal{R}Q'$.
(iv) If $P \xrightarrow{\alpha(x)} P'$ then for every y, Q', Q'' exist such that $Q \Rightarrow\xrightarrow{\alpha(x)} Q''$, and $Q''\{y/x\} \Rightarrow Q'$ with $P'\{y/x\}\mathcal{R}Q'$.

Early bisimilarity $\dot{\approx}_e$ *is the largest* early bisimulation.

P and Q are early equivalent*, written $P \approx_{ce} Q$, if for any* context *$C[]$, $C[P] \dot{\approx}_e C[Q]$.*

Definition 5. *Let $\mathcal{R}$ be a binary symmetric relation on $\mathcal{C}$. It is called a* ground bisimulation *if whenever $P\mathcal{R}Q$ and $P \xrightarrow{\lambda} P'$ then Q' exists such that $Q \stackrel{\hat{\lambda}}{\Rightarrow} Q'$ with $P'\mathcal{R}Q'$.*

Ground bisimilarity $\dot{\approx}_g$ is the largest ground bisimulation.

Definition 6. *P and Q are* ground equivalent*, written $P \approx_{cg} Q$, if for any* context *$C[]$, $C[P] \dot{\approx}_g C[Q]$.*

Remark 1. In the π-calculus, late and early bisimulation, which appeared in the original paper [11], are well-known. In order to obtain a congruence, it is often required that bisimulation should be closed under substitutions, see the corresponding definitions in [11]. Here, in order to reflect our understanding on the "closed" style equivalence, we choose to present it by the notion of *context*. However, this can be simplified in the sense that we provide the following *Context Lemma*. Note that ground bisimulation is not common in research on π-calculus. The main reason is that it is not even closed under the parallel operator (thus it can not be refined to congruence only by requiring closure under substitution), so that is of little sense. Therefore, to obtain a congruence, one has to require closure both under substitution and the parallel operator, which actually will lead to early congruence. Moreover, it is worth pointing out that in the setting of χ-calculus, only requiring closure under substitution is not sufficient! We provide a counterexample to illustrate this. Suppose $P = \bar{a}.\bar{a}+\bar{a}+\bar{a}.[x = y]\bar{a}$, and $Q = \bar{a}.\bar{a}+\bar{a}$. It is not difficult to observe that for any substitution σ, $P\sigma \dot{\approx}_l Q\sigma$. However, if $R = \langle x|y\rangle$, then $P|R \xrightarrow{\bar{a}} [x = y]\tau|R$. How can $Q|R$ match this transition? Clearly we have only two choices, $Q|R \xrightarrow{\bar{a}} 0|\langle x|y\rangle$ or $Q|R \xrightarrow{\bar{a}} \tau|\langle x|y\rangle$, but in both cases, the bisimulation game fails. Thus, we can conclude that $P|R \dot{\not\approx}_l Q|R$, so $P \not\approx_l Q$! That is, if only requiring closure under substitution is required, we would obtain an ill-defined bisimulation relation, since it would not be closed under the parallel operator. Instead, it is interesting and surprising that to require closure under the parallel operator is enough to give rise to a (partial) congruence, since it turns out that the parallel operator can exert a similar effect as substitution.

Lemma 2. *For any processes P and Q, substitution σ, if $P \approx_\Box Q$, then $P\sigma \approx_\Box Q\sigma$, where $\Box \in \{cg, cl, ce\}$.*

Lemma 3. *(Context Lemma for Labelled Bisimulation) $P \approx_{c\Box} Q$, iff for any process $R \in \mathcal{C}$, $P|R \dot{\approx}_\Box Q|R$, where $\Box \in \{g, l, e\}$.*

Lemma 4. $\approx_{cl} \subseteq \approx_{ce} \subseteq \approx_{cg}$.

3.2 Open Style Definitions

Definition 7. *Let $\mathcal{R}$ be a binary symmetric relation on $\mathcal{C}$. It is called an* open congruence *if the following two properties hold:*

(i) *$\mathcal{R}$ is a* ground bisimulation.
(ii) *For any* context *$C[]$, $(P, Q) \in \mathcal{R}$ implies $(C[P], C[Q]) \in \mathcal{R}$.*

P and Q are open congruent*, notation $P \approx_o Q$, if there exists some* open congruence $\mathcal{R}$ *such that* $(P, Q) \in \mathcal{R}$.

We present a different form of open congruence.

Definition 8. *([6], Definition 17) Let $\mathcal{R}$ be a binary symmetric relation on $\mathcal{C}$. It is called an* open bisimulation *if whenever $P\mathcal{R}Q$ and $P\sigma \xrightarrow{\lambda} P'$, then Q' exists such that $Q\sigma \stackrel{\hat{\lambda}}{\Rightarrow} Q'$ and $(P', Q') \in \mathcal{R}$.*

Open bisimilarity $\approx_{open}$ is the largest open bisimulation.

Clearly, according to Theorem 19 and Theorem 21 of [6], the above definition ($\approx_{open}$) is a rephrase of open bisimulation defined in Definition 7. Precisely, it can be regarded as a characterization of Definition 7, which will smooth the proofs of the results presented in the next section.

3.3 Relationships

As in [6], we first establish a technical lemma about the following general property, which will simplify the proof greatly, though it is very simple and obvious itself.

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

Lemma 5. $\dot{\approx}_g$ *satisfies the *-property.*

Lemma 6. $\approx_o \subseteq \approx_{cl}$.

The following lemma is devoted to stating that ground equivalent is not weaker than open congruence. The main proof idea is to construct a bisimulation relation $\mathcal{S}$ such that $\approx_{cg} \subseteq \mathcal{S}$, and prove that $\mathcal{S}$ is an open congruence. To this end, we need to construct a sophisticated context. Unfortunately, its proof is rather long and can not be presented here because of the space restriction. For more details, see [3].

Lemma 7. $\approx_{cg} \subseteq \approx_o$.

Theorem 1. $\approx_{cl} = \approx_{ce} = \approx_{cg} = \approx_o$.

4 Barbed Bisimulation

In this section, we turn to reduction semantics and barbed style bisimulation whose idea lies in that two processes are regarded as equal if they can simulate each other's communication while maintaining the same ability to communicate through any particular name. As in the previous section, we start from the closed style definition, and then treat the open style one.

Definition 9. *(Barb) A process P is* strongly barbed *at a, notion $P \downarrow_a$, if $P \stackrel{\alpha(x)}{\rightarrow} P'$ or $P \stackrel{\alpha x}{\rightarrow} P'$ 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 $P \Rightarrow P' \downarrow_a$.*

Definition 10. *Let $\mathcal{R}$ be a binary symmetric relation on $\mathcal{C}$. It is called a* barbed bisimulation *if whenever $P\mathcal{R}Q$ then the following two properties hold:*

- *For any name a, if $P \downarrow_a$, then $Q \Downarrow_a$.*
- *If $P \stackrel{\tau}{\rightarrow} P'$ then Q' exists such that $Q \Rightarrow Q'$ with $P'\mathcal{R}Q'$.*

The barbed bisimilarity $\dot{\approx}_b$ *is the largest barbed bisimulation.*

For barbed bisimilarity, we have the following properties.

Lemma 8. *$\dot{\approx}_b$ satisfies the *-property.*

Lemma 9. *For any processes P, Q, R, and name $s \notin fn(P, Q, R)$, if $P|(R + s) \dot{\approx}_b Q|(R + s)$, then $P|R \dot{\approx}_b Q|R$.*

Definition 11. *P and Q are* barbed equivalent*, written $P \approx_{cb} Q$, if for any* context *$C[]$, $C[P] \dot{\approx}_{cb} C[Q]$.*

We also provide a *Context Lemma* to simplify "any context" in the above definition.

Lemma 10. *(Context Lemma for Barbed Bisimulation)*

$$P \ \approx_{cb} \ Q \text{ iff } (\nu\tilde{x})(P|R) \dot{\approx}_b (\nu\tilde{x})(Q|R) \text{ for any } \tilde{x} \in \mathcal{N}, \text{ process } R \in \mathcal{C}.$$

Remark 2. The context in Lemma 10 is essential. We can not only require that it is closed by the parallel operator as in Lemma 3, because this is not discriminate enough. The following is a counterexample. Suppose $P = ax|\bar{a}y$ and $Q = ax.\bar{a}y + \bar{a}y.ax$. We can prove that for any process R, $P|R \dot{\approx}_b Q|R$ (here please note that in our semantics, $ax|\bar{a}y$ has no interaction, i.e. $ax|\bar{a}y \stackrel{\tau}{\not\rightarrow}$). However, $P \not\approx_b Q$, because when they are put the context $(\nu x)[]$, we can distinguish them.

Now, we turn to the "open" style definition.

Definition 12. *Let $\mathcal{R}$ be a binary symmetric relation on $\mathcal{C}$. It is called an* open barbed congruence *if the following two properties hold:*

(i) $\mathcal{R}$ is a barbed bisimulation.
(ii) For any context C, $(P, Q) \in \mathcal{R}$ implies $(C[P], C[Q]) \in \mathcal{R}$.

P and Q are open barbed congruent, notation $P \approx_{ob} Q$, if there exists some barbed congruence $\mathcal{R}$ such that $(P, Q) \in \mathcal{R}$.

Note 1. Barbed equivalence is studied in [12], and *open barbed congruence* here is essentially the *barbed congruence* in [6]. Here for a uniform terminology, we follow Sangiorgi and Walker [16].

To characterize open barbed congruence, we borrow an alternative definition from [6].

Definition 13. *([6], Definition 20) Let $\mathcal{R}$ be a binary symmetric relation on $\mathcal{C}$. It is called an* open ba-bisimulation *if whenever $P\mathcal{R}Q$ then for any substitution σ it holds that:*

(i) If $P\sigma \stackrel{\lambda}{\rightarrow} P'$, where $\lambda = y/x, \tau, \alpha(x)$ then Q' exists such that $Q\sigma \stackrel{\hat{\lambda}}{\Rightarrow} Q'$ and $P'\mathcal{R}Q'$.

(ii) If $P\sigma \stackrel{\alpha x}{\rightarrow} P'$ then Q' exists such that $P'\mathcal{R}Q'$ and either $Q\sigma \Rightarrow\stackrel{\alpha x}{\rightarrow} Q'$, or $Q\sigma \stackrel{\alpha(z)x/z}{\Rightarrow\Rightarrow} Q'$ for some fresh z.

Open ba-bisimilarity*, denoted $\approx^{ba}_{open}$, is the largest* open ba-bisimulation*.*

Actually, we have the following lemma. And in the sequel, when we mention open barbed congruence, we will use the form in Definition 13.

Lemma 11. *([6], Theorem 21 (iii)) $\approx^{ba}_{open}=\approx_o$.*

Now, we sketch the proof of the main result of this section, that is, the relationship of barbed equivalent and open barbed congruence. First we present a simple result.

Lemma 12. $\approx_{ob}\subseteq\approx_{cb}$.

The following lemma is the most important result of this paper. The main idea of proof is similar to Lemma 7. For more details, see [3].

Lemma 13. $\approx_{cb}\subseteq\approx_{ob}$.

Now, we have the following theorem:

Theorem 2. $\approx_{cb}=\approx_{ob}$.

Naturally, this raises the following problem: does ground congruence ($\approx_{cg}$) coincide with barbed equivalence ($\approx_{cb}$)? This question has also been studied extensively in π-calculus and it turned out to be a very difficult problem. Please see [2][3], among others, for more detailed discussion. Fortunately, now we can solve this problem easily in the setting of χ-calculus. Thanks to Theorem 1 and Theorem 2, clearly we can reduce this problem to the similar problem of open congruence and open barbed congruence, which is much simpler. As [6] shows:

- For the strong case, the two bisimulation equivalences coincide.
- For the weak case, the differences can be characterized by the following axiom, which holds for open barbed congruence while not for open congruence.

Prefix Law: $\alpha(z).(P + \langle x|z\rangle.Q) = \alpha(z).(P + \langle x|z\rangle.Q) + \alpha x.Q\{x/z\} \quad x \neq z$

Now, we can claim that we solve this problem in the χ-calculus completely: In the strong case, barbed equivalence and late equivalence (thus early and ground equivalence) coincide while in the weak case barbed equivalence is *strictly weaker* than the other three.

5 Conclusion

In this paper, we study the bisimilation congruences in χ-calculus. The main contributions and results are as follows:

- We adapt the bisimulation definitions for π-calculus to χ-calculus in a natural way. We find that in χ-calculus, the "open" and "closed" distinction common in π-calculus disappears. Thus we close the conjecture proposed by Fu in [6].
- We show that there are essentially two different bisimulation congruences, i.e. open congruence and open barbed congruence. In the weak case, the difference can be characterized by one axiom. Moreover, if we only consider the strong case, all sensible bisimulation equivalences collapse to just one.
- As a byproduct, we solve the problem of characterizing weak barbed equivalent in χ-calculus. It is essentially Definition 13. Moreover, according to our results, [6] actually gives an axiomatization for this relation.

In short, the key results of this paper can be reflected by the following diagram.

$$\begin{array}{ccccc} \approx_{cl} = \approx_{ce} = & \approx_{cg} & \subset & \approx_{cb} \\ & \| & & \| \\ & \approx_{o} & \subset & \approx_{ob} \end{array}$$

We now present some concluding remarks.

- In this paper, we only consider χ-calculus without mismatch operator [7]. This is not a very serious disadvantage, since besides some technical details, the main results of this paper can be adapted. Here, we would like to mention Fu and Yang's paper [7]. In their paper, open barbed congruence is also studied, and they also mention barbed equivalence ([7], Definition 34). However, they argue that open barbed congruence is contained in barbed equivalence and the inclusion is *strict*. To support this, they invite an example: $P_1 = [x \neq y]\tau.(P + \tau.[x \neq y]\tau.(P + \tau))$ and $P_2 = [x \neq y]\tau.(P + \tau)$, for which they "show" that they are open barbed congruent but not barbed equivalent. In our point of view, this is definitely incorrect since $P_1|\langle x|y\rangle$ and $P_2|\langle x|y\rangle$ are not open barbed congruent. Thus P_1 and P_2 are also not open barbed congruent! Currently we are performing a similar systematic study of χ-calculus with mismatch.
- It is worth emphasizing that the results in this paper have strong implications to other calculi falling into the family of fusion-style mobile calculi. Since all of these calculi share a similar communication mechanism, we believe the results in this paper generally also hold in, among others, update-calculus, fusion calculus, explicit fusion calculus.
- As we have said in Section 1, [4] discusses a similar problem in the setting of asynchronous π-calculus. To prove the analogous result for weak barbed congruences, Fournet and Gonthier actually have to use a *Universal Pi-calculus Machine* for their initial environment, and they use it to simulate the execution of a Gödelized version of a program. This leads to a very long

technical proof. Our proof technique, like that of Fournet and Gonthier, also involves creating an initial sophisticated environment. However, thanks to the mechanism of χ-calculus, our environment is much simpler.

Acknowledgement. We are grateful to Wan Fokkink for his careful reading of a draft of this paper and valuable comments. We also would like to thank the anonymous referees for their excellent criticisms.

References

1. M. Abadi and C. Fournet. Mobile values, new names, and secure communication. In *POPL*, pages 104–115, 2001.
2. T. Chen. *Research on the Theory and Application of Mobile Process Calculi.* Master's thesis, Nanjing University, Nanjing, P.R.China, 2005.
3. T. Chen, T. Han, and J. Lu. On bisimulation congruence in χ-calculus. Technical report, CWI, Amsterdam, The Netherlands. 2005.
4. C. Fournet and G. Gonthier. A hierarchy of equivalences for asynchronous calculi. *J. Log. Algebr. Program.*, 63:131–1739, 2005.
5. Y. Fu. A proof theoretical approach to communication. In P. Degano, R. Gorrieri, and A. Marchetti-Spaccamela, editors, *ICALP*, volume 1256 of *Lecture Notes in Computer Science*, pages 325–335. Springer, 1997.
6. Y. Fu. Bisimulation congruence of chi calculus. *Inf. Comput.*, 184(1):201–226, 2003.
7. Y. Fu and Z. Yang. Understanding the mismatch combinator in chi calculus. *Theor. Comput. Sci.*, 290(1):779–830, 2003.
8. P. Gardner and L. Wischik. Explicit fusions. In M. Nielsen and B. Rovan, editors, *MFCS*, volume 1893 of *Lecture Notes in Computer Science*, pages 373–382. Springer, 2000.
9. K. Honda and N. Yoshida. On reduction-based process semantics. *Theor. Comput. Sci.*, 151(2):437–486, 1995.
10. R. Milner. *Communication and Concurrency.* Prentice Hall, 1989.
11. R. Milner, J. Parrow, and D. Walker. A calculus of mobile process, part I/II. *Inf. Comput.*, 100:1–77, 1992.
12. R. Milner and D. Sangiorgi. Barbed bisimulation. In W. Kuich, editor, *ICALP*, volume 623 of *Lecture Notes in Computer Science*, pages 685–695. Springer, 1992.
13. J. Parrow and B. Victor. The update calculus (extended abstract). In M. Johnson, editor, *AMAST*, volume 1349 of *Lecture Notes in Computer Science*, pages 409–423, 1997.
14. J. Parrow and B. Victor. The fusion calculus: Expressiveness and symmetry in mobile processes. In *LICS*, pages 176–185, 1998.
15. D. Sangiorgi. A theory of bisimulation for the π-calculus. *Acta Inf.*, 33(1):69–97, 1996.
16. D. Sangiorgi and D. Walker. On barbed equivalences in pi-calculus. In *CONCUR*, volume 2154 of *Lecture Notes in Computer Science*, pages 292–304, 2001.
17. D. Sangiorgi and D. Walker. *The π-calculus: a Theory of Mobile Processes.* Cambridge University Press, 2001.
18. L. Wischik and P. Gardner. Strong bisimulation for the explicit fusion calculus. In I. Walukiewicz, editor, *FoSSaCS*, volume 2987 of *Lecture Notes in Computer Science*, pages 484–498. Springer, 2004.