The Ground Congruence for Chi Calculus*

Yuxi Fu** Zhenrong Yang

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

Abstract. The definition of open bisimilarity on the χ-processes does not give rise to a sensible relation on the χ-processes with the mismatch operator. The paper proposes ground open congruence as a principal open congruence on the χ-processes with the mismatch operator. The algebraic properties of the ground congruence is studied. The paper also takes a close look at barbed congruence. This relation is similar to the ground congruence. The precise relationship between the two is worked out. It is pointed out that the sound and complete system for the ground congruence can be obtained by removing one tau law from the complete system for the barbed congruence.

1 Introduction and χ-Calculus with Mismatch

The π-calculus ([6]) is a powerful process calculus. The expressiveness is partly supported by input processes of the form $a(x).P$ and output processes of the form $\overline{a}x.P$. The former may receive a name at channel name a before evolving as P with x replaced by the received name. The latter can emit x at a and then continues as P. The expressiveness is also supported by processes of the form $(x)P$. The localization operator (x) encapsulates the name x in P. In χ-calculus ([1–4]) the input and output processes are unified as $\alpha[x].P$, in which α stands for either a name or a coname.

Formally χ-processes are defined by the following abstract syntax:

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

where $\alpha \in \mathcal{N} \cup \overline{\mathcal{N}}$. Here $\mathcal{N}$ is the set of names ranged over by small case letters. The set $\{\overline{x} \mid x \in \mathcal{N}\}$ of conames is denoted by $\overline{\mathcal{N}}$. The name x in $(x)P$ is local. A name is global in P if it is not local in P. The global names, the local names and the names of a syntactical object, as well as the notations $gn(\_)$, $ln(\_)$ and $n(\_)$, are defined with their standard meanings. We adopt the α-convention widely used in the literature on process algebra. We do not consider replication or recursion operator since it does not affect the results of this paper.

* *FST&TCS 2000*, Lecture Notes in Computer Science 1974, 385-396, 2000.

** The author is funded by NNSFC (69873032) and 863 Hi-Tech Project (863-306-ZT06-02-2). He is also supported by BASICS, Center of Basic Studies in Computing Science, sponsored by Shanghai Education Committee. BASICS is affiliated to the Department of Computer Science at Shanghai Jiaotong University.

The following labeled transition system defines the operational semantics of χ-calculus, in which symmetric rules are systematically omitted. In the following rules the letter γ ranges over the set $\{\alpha(x), \alpha[x] \mid \alpha \in \mathcal{N} \cup \overline{\mathcal{N}}, x \in \mathcal{N}\} \cup \{\tau\}$ and the letter λ over the set $\{\alpha(x), \alpha[x], [y/x] \mid \alpha \in \mathcal{N} \cup \overline{\mathcal{N}}, x, y \in \mathcal{N}\} \cup \{\tau\}$. The symbols $\alpha(x), \alpha[x], [y/x]$ represent restricted action, free action and update action respectively. The x in the label $\alpha(x)$ is local.

Sequentialization

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

Composition

$$\frac{P \xrightarrow{\gamma} P' \quad ln(\gamma) \cap gn(Q) = \emptyset}{P|Q \xrightarrow{\gamma} P'|Q} Cmp_0 \qquad \frac{P \xrightarrow{[y/x]} P'}{P|Q \xrightarrow{[y/x]} P'|Q[y/x]} Cmp_1$$

Communication

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

$$\frac{P \xrightarrow{\alpha[x]} P' \quad Q \xrightarrow{\overline{\alpha}[y]} Q' \quad x \neq y}{P|Q \xrightarrow{[y/x]} P'[y/x]|Q'[y/x]} Cmm_2 \qquad \frac{P \xrightarrow{\alpha[x]} P' \quad Q \xrightarrow{\overline{\alpha}[x]} Q'}{P|Q \xrightarrow{\tau} P'|Q'} Cmm_3$$

Localization

$$\frac{P \xrightarrow{\lambda} P' \quad x \notin n(\lambda)}{(x)P \xrightarrow{\lambda} (x)P'} Loc_0 \qquad \frac{P \xrightarrow{\alpha[x]} P' \quad x \notin \{\alpha, \overline{\alpha}\}}{(x)P \xrightarrow{\alpha(x)} P'} Loc_1 \qquad \frac{P \xrightarrow{[y/x]} P'}{(x)P \xrightarrow{\tau} P'} Loc_2$$

Condition

$$\frac{P \xrightarrow{\lambda} P'}{[x{=}x]P \xrightarrow{\lambda} P'} Mtch$$

Summation

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

A substitution is a function from $\mathcal{N}$ to $\mathcal{N}$ that is identical on all but a finite number of names. Substitutions are usually denoted by $\sigma, \sigma', \ldots$. The notations $\Longrightarrow$ and $\stackrel{\widehat{\lambda}}{\Longrightarrow}$ are used in their standard meanings.

We will use two induced prefix operators, tau and update prefixes, defined as follows: $[y|x].P \stackrel{\text{def}}{=} (a)(\overline{a}[y]|a[x].P)$ and $\tau.P \stackrel{\text{def}}{=} (b)[b|b].P$ where a, b are fresh.

The subject language of this paper is $\chi^{\neq}$-calculus, the χ-calculus with the mismatch operator. The operational semantics of the mismatch combinator is defined as follows:

$$\frac{P \xrightarrow{\lambda} P' \quad x \neq y}{[x{\neq}y]P \xrightarrow{\lambda} P'} Mismtch$$

The set of $\chi^{\neq}$-processes is denoted by $\mathcal{C}$. Suppose Y is a finite set $\{y_1, \ldots, y_n\}$ of names. The notation $[y \notin Y]P$ will stand for $[y \neq y_1] \ldots [y \neq y_n]P$, where the order of mismatch operators is immaterial. We will write ϕ and ψ, called conditions, to stand for sequences of match and mismatch combinators concatenated one after another, μ for a sequence of match operators, and δ for a sequence of mismatch operators. Consequently we write ψP, μP and δP. When the length of ψ (μ, δ) is zero, ψP (μP, δP) is just P. The notation $\phi \Rightarrow \psi$ says that ϕ logically implies ψ and $\phi \Leftrightarrow \psi$ that ϕ and ψ are logically equivalent. A substitution σ agrees with ψ, and ψ agrees with σ, when $\psi \Rightarrow x{=}y$ if and only if $\sigma(x){=}\sigma(y)$.

Bisimulation equivalence relations on mobile processes are a lot more complex than those on CCS processes. The complication is mainly due to the dynamic aspect of mobile processes. The names in a process are subject to updates during the evolution of the process. These updates could be caused either by actions in which the process participates or by changes incurred by environments. A sensible observational equivalence for mobile processes must take that into account. To illustrate what kind of relations one would obtain if s/he ignored the mobility, we introduce the following definition for χ-calculus:

Definition 1. *Let $\mathcal{R}$ be a symmetric binary relation on the set of χ-processes. It is called a naked bisimulation if whenever $P\mathcal{R}Q$ and $P \xrightarrow{\lambda} P'$ then some Q' exists such that $Q \stackrel{\widehat{\lambda}}{\Longrightarrow} Q'\mathcal{R}P'$. The naked bisimilarity $\approx$ is the largest naked bisimulation.*

It is obvious that the definition of $\approx$ is simply a reiteration of the weak bisimilarity of CCS in terms of the operational semantic of χ-calculus. However the naked bisimilarity is not a good equivalence relation since it is not closed under the parallel composition. For instance one has $a[x]|\bar{b}[y] \approx a[x].\bar{b}[y]{+}\bar{b}[y].a[x]$ but not $(a[x]|\bar{b}[y])|(c[a]|\bar{c}[b]) \approx (a[x].\bar{b}[y]{+}\bar{b}[y].a[x])|(c[a]|\bar{c}[b])$. Process equivalence is observational equivalence. One of the defining properties for an observational equivalence is that the equivalence should be closed under parallel composition. In [1–4], it has been argued that bisimulation equivalences for χ-calculus are closed under substitution. This suggests to introduce the following definition:

> Let $\mathcal{R}$ be a symmetric binary relation on the set of χ-processes that is closed under substitution. It is called an open bisimulation if whenever $P\mathcal{R}Q$ and $P \xrightarrow{\lambda} P'$ then some Q' exists such that $Q \stackrel{\widehat{\lambda}}{\Longrightarrow} Q'\mathcal{R}P'$. The open bisimilarity $\approx_o$ is the largest open bisimulation.

The open bisimilarity $\approx_o$ has been studied in [1–4] in both the symmetric and the asymmetric frameworks. It must be pointed out that the investigations carried out in [1–4] are for the χ-calculus *without* the mismatch combinator. For the χ-calculus *with* the mismatch operator, one should ask the question whether the open bisimilarity $\approx_o$ is a sensible equivalence. In [5] the present authors have given a negative answer to the question. As it turned out the open bisimilarity defined above is not closed under parallel composition in $\chi^{\neq}$-calculus! One has $[x{\neq}y]a[x].P + a[x].[x{\neq}y]\tau.P \approx_o a[x].[x{\neq}y]\tau.P$ but it is clear that $\overline{a}[y]|([x{\neq}y]a[x].P + a[x].[x{\neq}y]\tau.P) \not\approx_o \overline{a}[y]|(a[x].[x{\neq}y]\tau.P)$. This is a serious problem because closure under parallel composition is an intrinsic property of observational equivalence. In [5] we have studied the problem and introduced two modified open congruences. These are early open congruence and late open congruence. Their relationship strongly recalls that between the weak early equivalence and the weak late equivalence ([6]). It should be said however that both the early open congruence and the late open congruence are the obvious modifications with motivation from π-calculus. They are not *the* open congruence for the χ-calculus with the mismatch operator. What is then the principal open congruence for χ-calculus with the mismatch combinator? We will give our answer to the question in this paper. The way to arrive to the definition of the open congruence is via a particular naked bisimulation. In order to define this relation we need the notion of contexts defined as follows: (i) $[]$ is a context; (ii) If $C[]$ is a context then $\alpha[x].C[]$, $C[]|P$, $P|C[]$, $(x)C[]$ and $[x{=}y]C[]$ are contexts.

Definition 2. *The ground bisimilarity $\approx_g$ is the largest naked bisimulation that is closed under context.*

In the above definition the requirement of closure under the prefix operator is reasonable since it is equivalent to that of closure under substitution. We will give an equivalent characterization of $\approx_g$ in the style of open semantics, which we argue is the principal open bisimilarity.

As it turns out the equivalence $\approx_g$ is very similar to the barbed bisimilarity of the χ-calculus with the mismatch operator. The difference is very subtle. The barbed bisimilarity also has an equivalent open characterization. The similarity and the difference between the ground bisimilarity and the barbed bisimilarity are revealed through their open characterizations.

This paper continues the work of [5] by studying the ground congruence and the barbed congruence for the $\chi^{\neq}$-calculus. The main contributions of this paper are as follows:

- We give an alternative characterization of the weak barbed bisimilarity. This characterization points out the complex nature of the weak barbed bisimilarity. Many unknown equalities are discovered. A complete system for the weak barbed congruence is provided. The new tau laws used to establish the completeness result are surprisingly complex.
- We study what we call ground open bisimilarity. A complete system for the ground open congruence is given. The relationship between the ground open congruence and the weak barbed congruence is revealed.

Due to space limitation, all proofs have been omitted.

2 Barbed Congruence

The barbed equivalence is often quoted as a universal equivalence relation for process algebras. For a specific process calculus barbed equivalence immediately gives rise to an observational equivalence. For two process calculi barbed equivalence can be used to compare the semantics of the two models. Despite the universal nature, barbed equivalence can have quite different displays in different process calculi. The barbed equivalence for the χ-calculus has brought some new insight into the calculi of mobile processes. In this section we demonstrate that the barbed equivalence for the $\chi^{\neq}$-calculus is even more different. A characterization theorem for the barbed bisimilarity on $\chi^{\neq}$-calculus is given. Some illustrating pairs of barbed equivalent processes are given. First we introduce the notion of barbedness.

Definition 3. *A process P is strongly barbed at a, notation $P{\downarrow}a$, if $P \stackrel{\alpha(x)}{\longrightarrow} P'$ or $P \stackrel{\alpha[x]}{\longrightarrow} P'$ for some P' such that $a \in \{\alpha, \overline{\alpha}\}$. 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 $P\mathcal{R}Q$.*

From the point of view of barbed equivalence an observer can not see the content of a communication. What an observer can detect is the ability of a process to communicate at particular channels. Two processes are identified if they can simulate each other in terms of this ability.

Definition 4. *Let $\mathcal{R}$ be a barbed symmetric relation on $\mathcal{C}$ closed under context. The relation $\mathcal{R}$ is a barbed bisimulation if whenever $P\mathcal{R}Q$ and $P \stackrel{\tau}{\longrightarrow} P'$ then $Q \Longrightarrow Q'\mathcal{R}P'$ for some Q'. The barbed bisimilarity $\approx_b$ is the largest barbed bisimulation.*

The trade-off of the simplicity of the above definition is that it provides little intuition about equivalent processes. We know that it is weaker than most bisimulation equivalences. But we want to know how much weaker it is. We first give some examples of barbed equivalent processes. To make the examples more readable, we will write $A \stackrel{\text{def}}{=} P\mathcal{R}(A{+}Q)$ for $P\mathcal{R}(P{+}Q)$, where $\mathcal{R}$ is a binary relation on processes. The first example of an equivalent pair is this:

$$A_1 \stackrel{\text{def}}{=} \alpha[x].(P_1{+}[x{=}y_1]\tau.Q){+}\alpha[x].(P_2{+}[x{\neq}y_1]\tau.Q) \approx_b A_1 + \alpha[x].Q$$

If $\alpha[x].Q$ on the right hand side is involved in a communication in which x is replaced by y_1 then $\alpha[x].(P_1{+}[x{=}y_1]\tau.Q)$ can simulate the action. Otherwise $\alpha[x].(P_2{+}[x{\neq}y_1]\tau.Q)$ would do the job. The second example is more interesting:

$$\begin{aligned} A_2 &\stackrel{\text{def}}{=} (z)\alpha[z].(P_1{+}[z{=}y_2][z|x].Q){+}\alpha[x].(P_2{+}[x{\neq}y_2]\tau.Q[x/z]) \\ &\approx_b A_2 + \alpha[x].Q[x/z] \end{aligned}$$

The communication $\overline{\alpha}[y_2]|(x)(A_2{+}\alpha[x].Q[x/z]) \stackrel{\tau}{\longrightarrow} \mathbf{0}|Q[x/z][y_2/x]$ for instance can be matched up by $\overline{\alpha}[y_2]|(x)A_2 \stackrel{\tau}{\longrightarrow} \mathbf{0}|(x)(P_1[y_2/z]{+}[y_2{=}y_2][y_2|x].Q[y_2/z]) \stackrel{\tau}{\longrightarrow} \mathbf{0}|Q[y_2/z][y_2/x]$. The third example is unusual:

$$A_3 \stackrel{\text{def}}{=} \alpha[y_3].(P_1{+}[y_3|x].Q){+}\alpha[x].(P_2{+}[x{\neq}y_3]\tau.Q) \approx_b A_3 + \alpha[x].Q$$

If $\alpha[x].Q$ participates in a communication in which x exchanges for y_3 then its role can be simulated by $\alpha[y_3].(P_1{+}[y_3|x].Q)$. The fourth is similar:

$$A_4 \stackrel{\text{def}}{=} [y_4|x].(P_1{+}\alpha[y_4].Q){+}\alpha[x].(P_2{+}[x{\neq}y_4]\tau.Q) \approx_b A_4 + \alpha[x].Q$$

If $(y_4)((A_4{+}\alpha[x].Q)|\overline{\alpha}[y_4].O) \stackrel{\tau}{\longrightarrow} Q[x/y_4]|O[x/y_4]$ then the simulation is:

$$(y_4)(A_4|\overline{\alpha}[y_4].O)\stackrel{\tau}{\longrightarrow}(P_1[x/y_4]{+}\alpha[x].Q[x/y_4])|\overline{\alpha}[x].O[x/y_4]\stackrel{\tau}{\longrightarrow}Q[x/y_4]|O[x/y_4]$$

The fifth example is the combination of the fourth and the second:

$$\begin{aligned} A_5 &\stackrel{\text{def}}{=} [y_5|x].(P_1{+}(z)\alpha[z].(P_1'{+}[z{=}y_5]\tau.Q)){+}\alpha[x].(P_2{+}[x{\neq}y_5]\tau.Q[x/z]) \\ &\approx_b A_5 + \alpha[x].Q[x/z] \end{aligned}$$

Notice that the component $[y_5|x].(P_1{+}(z)\alpha[z].(P_1'{+}[z{=}y_5]\tau.Q))$ is operationally the same as the process $[y_5|x].(P_1{+}(z)\alpha[z].(P_1'{+}[z{=}y_5][z|x].Q))$.

In the above examples, all the explicit mismatch operators contain the name x. In general there could be other conditions. The treatment of match operator is easy. The mismatch operator is however nontrivial. Suppose δ is a sequence of mismacth operators such that all names in δ are different from both x and z. An example more gerneral than A_1 is this:

$$A_1' \stackrel{\text{def}}{=} \alpha[x].(P_1{+}\delta[x{=}y_1]\tau.Q){+}\alpha[x].(P_2{+}\delta[x{\neq}y_1]\tau.Q) \approx_b A_1' + [x{\notin}n(\delta)]\delta\alpha[x].Q$$

We need to explain the mismatch sequence in $[x{\notin}n(\delta)]\delta\alpha[x].Q$. The δ before $\alpha[x].Q$ is necessary for otherwise an action of $([x{\notin}n(\delta)]\alpha[x].Q)\sigma$ may not be simulated by any action from $A_1'\sigma$ when σ invalidates δ. The $[x{\notin}n(\delta)]$ is necessary because otherwise it would not be closed under substitution. A counter example is the pair $\alpha[x].[y{\neq}z][x{=}y_1]\tau.Q{+}\alpha[x].[y{\neq}z][x{\neq}y_1]\tau.Q{+}[y{\neq}z]\alpha[x].Q$ and $\alpha[x].[y{\neq}z][x{=}y_1]\tau.Q{+}\alpha[x].[y{\neq}z][x{\neq}y_1]\tau.Q$. If we substitute x for z in the two processes we get two processes that are not barbed bisimilar. Similarly the example A_2 can be generalized to the following:

$$\begin{aligned} A_2' &\stackrel{\text{def}}{=} (z)\alpha[z].(P_1{+}[x{\notin}n(\delta)]\delta[z{=}y_2][z|x].Q){+}\alpha[x].(P_2{+}\delta[x{\neq}y_2]\tau.Q[x/z]) \\ &\approx_b A_2' + [x{\notin}n(\delta)]\delta\alpha[x].Q[x/z] \end{aligned}$$

The general form of A_3 is more delicate:

$$\begin{aligned} A_3' &\stackrel{\text{def}}{=} [x{\neq}y_3]\alpha[y_3].(P_1{+}[x{\notin}n(\delta)]\delta[y_3|x].Q){+}\alpha[x].(P_2{+}\delta[x{\neq}y_3]\tau.Q) \\ &\approx_b A_3' + [x{\neq}y_3][x{\notin}n(\delta)]\delta\alpha[x].Q \end{aligned}$$

In both $[x{\neq}y_3]\alpha[y_3].(P_1{+}[x{\notin}n(\delta)]\delta[y_3|x].Q)$ and $[x{\neq}y_3][x{\notin}n(\delta)]\delta\alpha[x].Q$ there is the mismatch $[x{\neq}y_3]$. If this operator is removed from A_3' one has

$$\begin{aligned} B_3' &\stackrel{\text{def}}{=} \alpha[y_3].(P_1{+}[x{\notin}n(\delta)]\delta[y_3|x].Q){+}\alpha[x].(P_2{+}\delta[x{\neq}y_3]\tau.Q) \\ &\not\approx_b B_3' + [x{\notin}n(\delta)]\delta\alpha[x].Q \end{aligned}$$

The inequality is clearer if one substitutes x for y_3 in the above:

$$\begin{aligned}C_3' &\stackrel{\text{def}}{=} \alpha[x].(P_1{+}[x{\notin}n(\delta)]\delta[x|x].Q){+}\alpha[x].(P_2{+}\delta[x{\neq}x]\tau.Q)\\ &\not\approx_b C_3' + [x{\notin}n(\delta)]\delta\alpha[x].Q\end{aligned}$$

The component $[x{\notin}n(\delta)]\delta\alpha[x].Q$ may be involved in a communication in which x is replaced by a name in δ. This action can not be simulated by C_3'. The general forms of A_4 and A_5 are as follows:

$$\begin{aligned}A_4' &\stackrel{\text{def}}{=} [y_4|x].(P_1{+}\delta\alpha[y_4].Q){+}\alpha[x].(P_2{+}\delta[x{\neq}y_4]\tau.Q) \approx_b A_4' + [x{\notin}n(\delta)]\delta\alpha[x].Q\\ A_5' &\stackrel{\text{def}}{=} [y_5|x].(P_1{+}(z)\alpha[z].(P_1'{+}\delta[z{=}y_5]\tau.Q)){+}\alpha[x].(P_2{+}\delta[x{\neq}y_5]\tau.Q[x/z])\\ &\approx_b A_5' + [x{\notin}n(\delta)]\delta\alpha[x].Q[x/z]\end{aligned}$$

If we replace the second summand $\alpha[x].(P_2{+}\delta[x{\neq}y_1]\tau.Q)$ of A_1' by $(z)\alpha[z].(P_2 + [x{\notin}n(\delta)]\delta[z{\neq}y_1][z|x].Q)$ and Q by $Q[x/z]$, we get an interesting variant of A_1' as follows:

$$\begin{aligned}A_1'' &\stackrel{\text{def}}{=} \alpha[x].(P_1{+}\delta[x{=}y_1]\tau.Q[x/z]){+}(z)\alpha[z].(P_2{+}[x{\notin}n(\delta)]\delta[z{\neq}y_1][z|x].Q)\\ &\approx_b A_1'' + [x{\notin}n(\delta)]\delta\alpha[x].Q[x/z]\end{aligned}$$

The bisimilar pairs A_2' through A_5' have similar variants:

$$\begin{aligned}A_2'' &\stackrel{\text{def}}{=} (z)\alpha[z].(P_1{+}[x{\notin}n(\delta)]\delta[z{=}y_2][z|x].Q){+}O_2\\ &\approx_b A_2'' + [x{\notin}n(\delta)]\delta\alpha[x].Q[x/z]\\ A_3'' &\stackrel{\text{def}}{=} [x{\neq}y_3]\alpha[y_3].(P_1{+}[x{\notin}n(\delta)]\delta[y_3|x].Q[x/z]){+}O_3\\ &\approx_b A_3'' + [x{\neq}y_3][x{\notin}n(\delta)]\delta\alpha[x].Q[x/z]\\ A_4'' &\stackrel{\text{def}}{=} [y_4|x].(P_1{+}\delta\alpha[y_4].Q[x/z]){+}O_4\\ &\approx_b A_4'' + [x{\notin}n(\delta)]\delta\alpha[x].Q[x/z]\\ A_5'' &\stackrel{\text{def}}{=} [y_5|x].(P_1{+}(z)\alpha[z].(P_1'{+}\delta[z{=}y_5]\tau.Q[z/x])){+}O_5\\ &\approx_b A_5'' + [x{\notin}n(\delta)]\delta\alpha[x].Q[x/z]\end{aligned}$$

where O_i is $(z)\alpha[z].(P_2{+}[x{\notin}n(\delta)]\delta[z{\neq}y_i][z|x].Q)$ for $i \in \{2,3,4,5\}$. The most complicated situation arises when all the five possibilities as described by A_1'' through A_5'' happen at one go:

$$\begin{aligned}A \stackrel{\text{def}}{=}\ & (z)\alpha[z].(P_2{+}[x{\notin}n(\delta)]\delta[z{\notin}\{y_1,y_2,y_3,y_4,y_5\}][z|x].Q)\\ & {+}\alpha[x].(P_1{+}\delta[x{=}y_1]\tau.Q[x/z])\\ & {+}(z)\alpha[z].(P_1{+}[x{\notin}n(\delta)]\delta[z{=}y_2][z|x].Q)\\ & {+}[x{\neq}y_3]\alpha[y_3].(P_1{+}[x{\notin}n(\delta)]\delta[y_3|x].Q[x/z])\\ & {+}[y_4|x].(P_1{+}\delta\alpha[y_4].Q[x/z])\\ & {+}[y_5|x].(P_1{+}(z)\alpha[z].(P_1'{+}\delta[z{=}y_5]\tau.Q[z/x]))\\ \approx_b\ & A + [x{\neq}y_3][x{\notin}n(\delta)]\delta\alpha[x].Q[x/z]\end{aligned}$$

Similarly the examples A'_1 through A'_5 can be combined into one as follows:

$$\begin{aligned} A' \stackrel{\text{def}}{=} & \ \alpha[x].(P_2{+}\delta[x{\notin}\{y_1,y_2,y_3,y_4,y_5\}]\tau.Q[x/z]) \\ & +\alpha[x].(P_1{+}\delta[x{=}y_1]\tau.Q[x/z]) \\ & +(z)\alpha[z].(P_1{+}[x{\notin}n(\delta)]\delta[z{=}y_2][z|x].Q) \\ & +[x{\neq}y_3]\alpha[y_3].(P_1{+}[x{\notin}n(\delta)]\delta[y_3|x].Q[x/z]) \\ & +[y_4|x].(P_1{+}\delta\alpha[y_4].Q[x/z]) \\ & +[y_5|x].(P_1{+}(z)\alpha[z].(P'_1{+}\delta[z{=}y_5]\tau.Q[z/x])) \\ \approx_b & \ A' + [x{\neq}y_3][x{\notin}n(\delta)]\delta\alpha[x].Q[x/z] \end{aligned}$$

Having seen so many bisimilar pairs of processes, the reader might wonder how we have discovered them. As a matter of fact these examples are all motivated by an alternative characterization of the barbed bisimilarity. This characterization is given by an open bisimilarity as defined below.

Definition 5. *Let $\mathcal{R}$ be a binary symmetric relation on $\mathcal{C}$ closed under substitution. The relation $\mathcal{R}$ is a barbed open bisimulation if the following properties hold for P and Q whenever $P\mathcal{R}Q$:*

(i) If λ is an update or a tau and $P \xrightarrow{\lambda} P'$ then Q' exists such that $Q \stackrel{\widehat{\lambda}}{\Longrightarrow} Q'\mathcal{R}P'$.
(ii) If $P \xrightarrow{\alpha[x]} P'$ then one of the following properties holds:

- *Q' exists such that $Q \stackrel{\alpha[x]}{\Longrightarrow} Q'\mathcal{R}P'$;*
- *Q' and Q'' exist such that $Q \Longrightarrow\xrightarrow{\alpha(z)} Q''$ and $Q''[x/z] \Longrightarrow Q'\mathcal{R}P'$;*

and, for each y different from x, one of the following properties holds:

- *Q' and Q'' exist such that $Q \Longrightarrow\xrightarrow{\alpha[x]} Q''$ and $Q''[y/x] \Longrightarrow Q'\mathcal{R}P'[y/x]$;*
- *Q' and Q'' exist such that $Q \Longrightarrow\xrightarrow{\alpha(z)} Q''$ and $Q''[y/z] \stackrel{[y/x]}{\Longrightarrow} Q'\mathcal{R}P'[y/x]$;*
- *Q' exists such that $Q \stackrel{\alpha[y]}{\Longrightarrow}\stackrel{[y/x]}{\Longrightarrow} Q'\mathcal{R}P'[y/x]$;*
- *Q' exists such that $Q \stackrel{[y/x]}{\Longrightarrow}\stackrel{\alpha[y]}{\Longrightarrow} Q'\mathcal{R}P'[y/x]$;*
- *Q' and Q'' exist such that $Q \stackrel{[y/x]}{\Longrightarrow}\xrightarrow{\alpha(z)} Q''$ and $Q''[y/z]{\Longrightarrow}Q'\mathcal{R}P'[y/x]$.*

(iii) If $P \xrightarrow{\alpha(x)} P'$ then, for each y, one of the following properties holds:

- *Q' and Q'' exist such that $Q \Longrightarrow\xrightarrow{\alpha(x)} Q''$ and $Q''[y/x]{\Longrightarrow}Q'\mathcal{R}P'[y/x]$;*
- *Q' exists such that $Q \stackrel{\alpha[y]}{\Longrightarrow} Q'\mathcal{R}P'[y/x]$.*

The barbed open bisimilarity $\approx^b_{open}$ is the largest barbed open bisimulation.

With a definition as complex as Definition 5, it is not very clear if the relation it introduces is well behaved. The next lemma gives one some confidence on the barbed open bisimilarity.

Lemma 6. *$\approx^b_{open}$ is closed under localization and composition.*

Since $\approx^b_{open}$ is closed under substitution, it must also be closed under prefix operation. It is also clear that $\approx^b_{open}$ is closed under match operation. However the relation is closed neither under the mismatch operation nor under the summation operation. For instance $[x{\neq}y]P \approx^b_{open} [x{\neq}y]\tau.P$ does not hold. To obtain the largest congruence contained in $\approx^b_{open}$ we use the standard approach.

Definition 7. *Two processes P and Q are barbed congruent, notation $P \simeq_b Q$, if $P \approx^b_{open} Q$ and for each substitution σ whenever $P\sigma \xrightarrow{\tau} P'$ then Q' exists such that $Q\sigma \overset{\tau}{\Longrightarrow} Q' \approx^b_{open} P'$ and vice versa.*

The notation $\simeq_b$ is not confusing because it is also the largest congruence contained in $\approx_b$. This is guaranteed by the next theorem.

Theorem 8. *$\approx^b_{open}$ and $\approx_b$ coincide.*

3 Axiomatic System

In this section we give a complete system for the barbed congruence on the finite $\chi^{\neq}$-processes. In order to prove the completeness theorem, we need some auxiliary definitions.

Definition 9. *Let V be a finite set of names. We say that ψ is complete on V if $n(\psi) \subseteq V$ and for each pair x, y of names in V it holds that either $\psi \Rightarrow x{=}y$ or $\psi \Rightarrow x{\neq}y$. A substitution σ is induced by ψ, and ψ induces σ, if σ agrees with ψ and $\sigma\sigma = \sigma$.*

We now begin to describe a system complete for the barbed congruence. Let AS denote the system consisting of the rules and laws in Figure 2 plus the following expansion law:

$$
\begin{aligned}
P|Q = \sum_i \phi_i(\tilde{x})\pi_i.(P_i|Q) + \sum_{\substack{\pi_i = a_i[x_i] \\ \gamma_j = \overline{b_j}[y_j]}} \phi_i\psi_j(\tilde{x})(\tilde{y})[a_i{=}b_j][x_i|y_j].(P_i|Q_j) + \\
\sum_j \psi_j(\tilde{y})\gamma_j.(P|Q_j) + \sum_{\substack{\pi_i = \overline{a_i}[x_i] \\ \gamma_j = b_j[y_j]}} \phi_i\psi_j(\tilde{x})(\tilde{y})[a_i{=}b_j][x_i|y_j].(P_i|Q_j)
\end{aligned}
$$

where P is $\sum_i \phi_i(\tilde{x})\pi_i.P_i$ and Q is $\sum_j \psi_j(\tilde{y})\gamma_j.Q_j$, π_i and γ_j range over $\{\alpha[x] \mid \alpha \in \mathcal{N} \cup \overline{\mathcal{N}}, x \in \mathcal{N}\}$.

Using axioms in AS, a process can be converted to a process that contains no occurrence of composition operator, the latter process is of special form as defined below.

Definition 10. *A process P is in normal form on $V \supseteq fn(P)$ if P is of the form $\sum_{i \in I_1} \phi_i\alpha_i[x_i].P_i + \sum_{i \in I_2} \phi_i(x)\alpha_i[x].P_i + \sum_{i \in I_3} \phi_i[z_i|y_i].P_i$ such that x does not appear in P, ϕ_i is complete on V for each $i \in I_1 \cup I_2 \cup I_3$, P_i is in normal form on V for $i \in I_1 \cup I_3$ and is in normal form on $V \cup \{x\}$ for $i \in I_2$. Here I_1, I_2 and I_3 are pairwise disjoint finite indexing sets.*

T1	$\lambda.\tau.P = \lambda.P$		
T2	$P{+}\tau.P = \tau.P$		
T3	$\lambda.(P{+}\tau.Q) = \lambda.(P{+}\tau.Q){+}\lambda.Q$		
T4	$\tau.P = \tau.(P{+}\psi\tau.P)$		
T5	$[y\|x].(P{+}\delta\tau.Q) = [y\|x].(P{+}\delta\tau.Q){+}\psi\delta[y\|x].Q$		$C(\psi,\delta)$
T6	$FF = FF{+}[x{\notin}Y_3][x{\notin}n(\delta)]\delta\alpha[x].Q[x/z]$		$z{\notin}n(\delta)$
T7	$FR = FR{+}[x{\notin}Y_3][x{\notin}n(\delta)]\delta\alpha[x].Q[x/z]$		$z{\notin}n(\delta)$
TD1	$RO = RO{+}\delta(x)\alpha[x].Q$		$x{\notin}n(\delta)$

Fig. 1. Tau Laws

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) $d([x{=}y]P) = d(P)$, $d([x{\neq}y]P) = d(P)$; (vi) $d(P{+}Q) = max\{d(P), d(Q)\}$.

Lemma 11. *For a process P and a finite set V of names such that $fn(P) \subseteq V$ there is a normal form Q on V such that $d(Q) \leq d(P)$ and $AS \vdash Q = P$.*

In order to obtain a complete system for the barbed congruence, we need some tau laws, some of which are new and complex. Figure 1 contains seven tau laws used in this paper. T4, introduced by the first author in previous publication, is a necessary law for open congruences. T5 holds under the condition $C(\psi, \delta)$:

If $\delta \Rightarrow [u{\neq}v]$ then either $\psi \Rightarrow [x{=}u][y{\neq}v]$ or $\psi \Rightarrow [x{=}v][y{\neq}u]$ or $\psi \Rightarrow [y{=}u][x{\neq}v]$ or $\psi \Rightarrow [y{=}v][x{\neq}u]$ or $\psi \Rightarrow [x{\neq}u][x{\neq}v][y{\neq}u][y{\neq}v]$.

This law was used for the first time in [5]. The laws T6 and T7 are equational formalization of the examples given in Section 2 in a more general form. In these axioms, FF (respectively FR) stands for

$$\begin{aligned}
&\alpha[x].(P{+}\delta[x{\notin}Y_1 \cup \ldots \cup Y_5]\tau.Q[x/z])\\
&(\text{respectively } (z)\alpha[z].(P{+}[x{\notin}n(\delta)]\delta[z{\notin}Y_1 \cup \ldots \cup Y_5][z|x].Q))\\
&+\Sigma_{y\in Y_1}\alpha[x].(P_y{+}\delta[x{=}y]\tau.Q[x/z]){+}\Sigma_{y\in Y_2}(z)\alpha[z].(P_y{+}[x{\notin}n(\delta)]\delta[z{=}y][z|x].Q)\\
&+\Sigma_{y\in Y_3}[x{\neq}y]\alpha[y].(P_y{+}[x{\notin}n(\delta)]\delta[y|x].Q[x/z])\\
&+\Sigma_{y\in Y_4}[y|x].(P_y{+}\delta\alpha[y].(P'_y{+}\delta\tau.Q[x/z]))\\
&+\Sigma_{y\in Y_5}[y|x].(P_y{+}\delta(z)\alpha[z].(P'_y{+}\delta[z{=}y]\tau.Q))
\end{aligned}$$

These two laws are new. In TD1, which is derivable from T6, RO is

$$\begin{aligned}
\Sigma_{y\in Y_1}\alpha[y].(P_y{+}\delta\tau.Q[y/x]) \,+\, &\Sigma_{y\in Y_2}(x)\alpha[x].(P_y{+}\delta[x{=}y]\tau.Q)\\
&+\,(x)\alpha[x].(P{+}\delta[x{\notin}Y_1 \cup Y_2]\tau.Q)
\end{aligned}$$

Let $AS \cup \{T1, T2, T3, T4, T5, T6, T7\}$ denote AS^b_o. Without further ado, we state the main result of this section.

Theorem 12. *AS^b_o is sound and complete for $\simeq_b$.*

4 Ground Congruence

In this section we sketch some main properties about $\approx_g$. First of all the ground bisimilarity can be characterized by an open bisimilarity called ground open bisimilarity, notation $\approx^g_{open}$. The definition of the ground open bisimilarity is obtained from Defintion 5 by replacing clause (ii) by

(ii') If $P \stackrel{\alpha[x]}{\longrightarrow} P'$ then Q' exists such that $Q \stackrel{\alpha[x]}{\Longrightarrow} Q'\mathcal{R}P'$.

It is easy to prove that $\approx^g_{open}$ is closed under localization and composition and that $\approx^g_{open}$ coincides with $\approx_g$. By definition the ground open bisimilarity is contained in the barbed one. The inclusion is strict because T7 is not valid for $\approx^g_{open}$.

Let $\simeq_g$ be the largest congruence contained in $\approx^g_{open}$. Its formal definition is completely similar to that of $\simeq_b$. Let AS^g_o stand for $AS \cup \{T1, T2, T3, T4, T5, T6\}$. It can be similarly proved that AS^g_o is sound and complete for $\simeq_g$.

5 Remark

Parrow and Victor have studied fusion calculus ([7]). It is a polyadic version of $\chi^{\neq}$-calculus. The main observational equivalence they have studied is what they call weak hyperequivalence. The weak hyperequivalence is essentially a polyadic version of the open bisimilarity $\approx_o$ we have defined in the introduction. Since $\chi^{\neq}$-calculus is a monadic version of the fusion calculus and therefore is a subcalculus of the latter, the counter example given in the introduction is valid in fusion calculus as well. One of the motivations of the ground bisimilarity is to rectify the weak hyperequivalence. Apart from its theoretical interest, the barbed bisimilarity is introduced partly to study the ground bisimilarity.

References

1. Y. Fu. A Proof Theoretical Approach to Communications. *ICALP'97*, Lecture Notes in Computer Science 1256, Springer, 325–335, 1997.
2. Y. Fu. Bisimulation Lattice of Chi Processes. *ASIAN'98*, Lecture Notes in Computer Science 1538, Springer, 245–262, 1998.
3. Y. Fu. Variations on Mobile Processes. *Theoretical Computer Science*, **221**: 327-368, 1999.
4. Y. Fu. Open Bisimulations of Chi Processes. *CONCUR'99*, Eindhoven, The Netherlands, August 24-27, Lecture Notes in Computer Science 1664, Springer, 304-319, 1999.
5. Y. Fu, Z. Yang. Chi Calculus with Mismatch. *CONCUR 2000*, Pennsylvania, USA, August 22-25, Lecture Notes in Computer Science 1877, Springer, 2000.
6. 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.
7. J. Parrow, B. Victor. The Fusion Calculus: Expressiveness and Symmetry in Mobile Processes.
8. J. Parrow, D. Sangiorgi. Algebraic Theories for Name-Passing Calculi. *Journal of Information and Computation*, **120**: 174-197, 1995.

E1	$P = P$	
E2	$P = Q$	if $Q = P$
E3	$P = R$	if $P = Q$ and $Q = R$
C1	$\alpha[x].P = \alpha[x].Q$	if $P = Q$
C2	$(x)P = (x)Q$	if $P = Q$
C3a	$[x{=}y]P = [x{=}y]Q$	if $P = Q$
C3b	$[x{\neq}y]P = [x{\neq}y]Q$	if $P = Q$
C4	$P{+}R = Q{+}R$	if $P = Q$
C5	$P_0 \vert P_1 = Q_0 \vert Q_1$	if $P_0 = Q_0$ and $P_1 = Q_1$
L1	$(x)\mathbf{0} = \mathbf{0}$	
L2	$(x)\alpha[y].P = \mathbf{0}$	$x \in \{\alpha, \overline{\alpha}\}$
L3	$(x)\alpha[y].P = \alpha[y].(x)P$	$x \notin \{y, \alpha, \overline{\alpha}\}$
L4	$(x)(y)P = (y)(x)P$	
L5	$(x)[y{=}z]P = [y{=}z](x)P$	$x \notin \{y, z\}$
L6	$(x)[x{=}y]P = \mathbf{0}$	$x{\neq}y$
L7	$(x)(P{+}Q) = (x)P{+}(x)Q$	
L8	$(x)[y \vert z].P = [y \vert z].(x)P$	$x \notin \{y, z\}$
L9	$(x)[y \vert x].P = \tau.P[y/x]$	$y \neq x$
L10	$(x)[x \vert x].P = \tau.(x)P$	
M1	$\phi P = \psi P$	if $\phi \Leftrightarrow \psi$
M2	$[x{=}y]P = [x{=}y]P[y/x]$	
M3a	$[x{=}y](P{+}Q) = [x{=}y]P{+}[x{=}y]Q$	
M3b	$[x{\neq}y](P{+}Q) = [x{\neq}y]P{+}[x{\neq}y]Q$	
M4	$P = [x{=}y]P{+}[x{\neq}y]P$	
M5	$[x{\neq}x]P = \mathbf{0}$	
S1	$P{+}\mathbf{0} = P$	
S2	$P{+}Q = Q{+}P$	
S3	$P{+}(Q{+}R) = (P{+}Q){+}R$	
S4	$P{+}P = P$	
U1	$[y \vert x].P = [x \vert y].P$	
U2	$[y \vert x].P = [y \vert x].[x{=}y]P$	
U3	$[x \vert x].P = \tau.P$	

Fig. 2. Axiomatic System AS

LD1	$(x)[x \vert x].P = [y \vert y].(x)P$	U3 and L8
LD2	$(x)[y{\neq}z]P = [y{\neq}z](x)P$	L5, L7 and M4
LD3	$(x)[x{\neq}y]P = (x)P$	L6, L7 and M4
MD1	$[x{=}y].\mathbf{0} = \mathbf{0}$	S1, S4 and M4
MD2	$[x{=}x].P = P$	M1
MD3	$\phi P = \phi(P\sigma)$ where σ is induced by ϕ	M2
SD1	$\phi P{+}P = P$	S-rules and M4
UD1	$[y \vert x].P = [y \vert x].P[y/x]$	U2 and M2

Fig. 3. Some Laws Derivable from AS