TCS 4447
pp: 1-52 (col.fig.: nil)
PROD. TYPE: COM
ED: SG
PAGN: LAK – SCAN: Vinod
ARTICLE IN PRESS
ELSEVIER
Theoretical Computer Science ▮▮▮ (▮▮▮▮) ▮▮▮–▮▮▮
Theoretical Computer Science
www.elsevier.com/locate/tcs

Understanding the mismatch combinator in chi calculus

Yuxi Fu*[,1], Zhenrong Yang

Department of Computer Science, Shanghai Jiaotong University, 1954 Hua Shan Road, Shanghai 200030, China

Received 7 February 2001; received in revised form 22 April 2002; accepted 22 April 2002
Communicated by P.-L. Curien

Abstract

The theory of chi processes with the mismatch operator is studied. Four congruence relations are investigated. These are late open congruence, early open congruence, ground congruence and barbed congruence. The late and early open congruence relations are the chi calculus counterparts of the weak late and early congruence relations of pi calculus. Both turn out to be special cases of the ground congruence and the barbed congruence. The ground congruence is essentially the open congruence. Complete systems are given for all the four congruence relations. These systems use some interesting tau laws unknown from previous studies of the chi calculus without the mismatch combinator. The results of this paper point out that the mismatch operator changes the algebraic semantics of chi calculus dramatically. They also correct some common mistakes in literature.

Keywords: Process algebra; Chi process; Bisimulation; Axiomatization

1. Introduction

In recent years several publications have focused on a class of new calculi of mobile processes. These models include χ-calculus [4–8,12], update calculus [27] and fusion

* Corresponding author.
E-mail address: fu-yx@cs.sjtu.edu.cn (Y. Fu).
[1] 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.

0304-3975/02/$ - see front matter © 2002 Published by Elsevier Science B.V.
PII: S0304-3975(02)00373-0

calculus [28]. In a uniform terminology they are respectively χ-calculus, asymmetric χ-calculus and polyadic χ-calculus. The χ-calculus has its motivation from proof theory. In process algebraic model of classical proofs there has been no application of the mismatch operator. The χ-calculus studied so far contains no mismatch operator. On the other hand the update and fusion calculi get the motivation from concurrent constraint programming. When applying process calculi to model real programming problems one finds very handy the mismatch operator. For that reason the full update and fusion calculi always come with the mismatch combinator. Strong bisimulation congruence has been investigated for each of the three models. It is basically the strong open congruence. A fundamental difference between χ-like calculi and π-like calculi [24] is that all names in the former are subject to update whereas bound names in the latter are never changed. In terms of the algebraic semantics, it says that open style congruence relations are particularly suitable to χ-like process calculi. Several weak observational equivalence relations have been examined. Fu studied in [5] weak open congruence and weak barbed congruence. It was shown that a sensible bisimulation equivalence on χ-processes must be closed under substitution in every bisimulation step. In χ-like calculi closure under substitution amounts to the same thing as closure under parallel composition and restriction. This is the property that led Fu to introduce L-congruences [6]. These congruence relations form a lattice under inclusion order. It has been demonstrated that L-congruences are general enough so as to subsume familiar bisimulation congruences. The open congruence and the barbed congruence for instance are respectively the bottom and the top elements of the lattice. This is also true for the asymmetric χ-calculus [8]. Complete systems have been discovered for L-congruences on both finite χ-processes [6] and finite asymmetric χ-processes [8]. An important discovery in the work of axiomatizing χ-processes is that Milner's tau laws are insufficient for open congruences. Another basic tau law called T4

$$\tau.P = \tau.(P + [x = y]\tau.P)$$

is necessary to deal with the dynamic aspect of name update. Parrow and Victor have worked on completeness problems for fusion calculus [29]. The system they provide for the weak hypercongruence for sub-fusion calculus *without* the mismatch operator is deficient because it lacks of the axiom T4. However their main effort in the above mentioned paper is on the full fusion calculus *with* the mismatch operator. This part of work is unfortunately more problematic. To explain what we mean by that we need to take a closer look at hyperequivalence.

Process equivalence is observational in the sense that two processes are deemed to be equal unless an environment can detect a difference between the two processes. It follows that process equivalences must be closed under, among other things, parallel composition. Weak hyperequivalence is basically an open equivalence. This relation is fine with the sub-fusion calculus without the mismatch combinator. It is however a bad equivalence for the full fusion calculus for the reason that it is not closed under parallel composition. A simple counter example is as follows: Let $\approx_h$ be the hyperequivalence. Now for distinct names x, y it holds that

$$(x)ax.[x \neq y]\tau.P \approx_h (x)ax.[x \neq y]\tau.P + (x)ax.P.$$

This is because the transition

$$(x)ax.[x \neq y]\tau.P + (x)ax.P \stackrel{a(x)}{\longrightarrow} P.$$

can be simulated by

$$(x)ax.[x \neq y]\tau.P \stackrel{a(x)}{\longrightarrow}\stackrel{\tau}{\longrightarrow} P.$$

However

$$\bar{a}y \,|\, (x)ax.[x \neq y]\tau.P \not\approx_h \bar{a}y \,|\, ((x)ax.[x \neq y]\tau.P + (x)ax.P)$$

for the transition $\bar{a}y \,|\, ((x)ax.[x \neq y]\tau.P + (x)ax.P) \stackrel{\tau}{\longrightarrow} \mathbf{0} \,|\, P\{y/x\}$ cannot be matched up by any transitions from $\bar{a}y \,|\, (x)ax.[x \neq y]\tau.P$. For similar reason

$$ax.[x \neq y]\tau.P \approx_h ax.[x \neq y]\tau.P + [x \neq y]ax.P$$

but

$$\bar{a}y \,|\, ax.[x \neq y]\tau.P \not\approx_h \bar{a}y \,|\, (ax.[x \neq y]\tau.P + [x \neq y]ax.P).$$

So the theory of weak equivalence of fusion calculus need be overhauled.

It should be pointed out that the failure of the weak hyperequivalence has nothing to do with the property of closure under substitution or the lack of it. Although the weak hyperequivalence is by definition closed under substitution, it still admits the counter examples. A well prepared reader should realize immediately that neither counter example is affected by substitution. On the other hand, the failure does have a lot to do with the mismatch operator. From a programming point of view the role of the mismatch combinator is to terminate a process at run time. This is a useful function in practice and yet realizable in neither CCS nor calculi of mobile processes without the mismatch combinator. The effect of the mismatch operator on the operational semantics is well-known: Transitions are no longer stable under name instantiations. It is also well-known that this phenomenon renders the algebraic theory difficult. The mismatch operator often creates a ‘now-or-never’ situation in which if an action does not happen right now it might never be allowed to happen. In the calculi with the mismatch operator processes are more sensible to the timing of actions. This reminds one of the difference between the early and late semantics.

The early/late dichotomy is well known in the semantic theory of π-calculus [24]. The weak late congruence is strictly contained in the weak early congruence in π-calculus whether the mismatch combinator is present or not. For some time it was taken for granted that there is no early and late distinction in weak open congruence. At least this is true for the calculus without the mismatch combinator. Very recently the present authors discovered to their surprise that early and late approaches give rise to two different weak open congruences in the π-calculus in the presence of the mismatch combinator [13]. This has led them to realize the problem with the weak hyperequivalence.

We are therefore forced to reexamine the algebraic theory of χ-like calculi with the mismatch combinator. In this paper we study the bisimulation congruence relations for χ-calculus with the mismatch operator. Our main focus will be on the barbed congruence and the ground congruence. The barbed approach is a widely applicable tool to give an observational equivalence relation for a process calculus. When applied to the χ-calculus with the mismatch operator, it gives rise to a very subtle equivalence. The ground congruence can be seen as a rectification of the hyperequivalence. It is equal to the largest congruence relation contained in the hyperequivalence. As it turns out the ground congruence is very similar to the barbed congruence. In order to give complete systems for the barbed congruence and the ground congruence, one need to have a 'complete' understanding of the intrinsic properties of the two relations. We do this by providing alternative characterizations of the two relations. These characterizations have the virtue that they are given purely in terms of the actions a process can perform without any reference to context. It is these alternative characterizations that pinpoint the precise relationship between the two congruences. We will also take a look at the late and early congruence relations. The attention we pay to them serves two purposes. Firstly, since the weak late congruence and the weak early congruence are two main equivalence relations for the π-calculus, the corresponding relations in the χ-calculus should be studied and compared to the barbed congruence and the ground congruence. Secondly the late and the early congruence relations are much simpler than the barbed and ground congruence relations. A warming up exercise with the former two would make smooth the transition to the study of the latter two.

The main contributions of this paper are as follows:

- We initiate a study of χ-like calculi with the mismatch combinator. We point out that the algebraic theory of the χ-calculus with the mismatch combinator is very different from that of the χ-calculus without the mismatch operator. All previous works on the algebraic theory of χ-like calculi with the mismatch operator have fundamental mistakes. Even the very definition of hyperequivalence has to be abandoned.
- We study the counterparts of the weak early congruence and the weak late congruence of π-calculus in the framework of χ-calculus with the mismatch combinator. Complete systems are given for both the relations. At the same time we points out that these two equivalence relations do not play as much important role in the χ-calculus as in the π-calculus.
- We study the barbed congruence. 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 congruence. A complete system for the ground congruence is given. The relationship between the ground congruence and the barbed congruence is revealed.

The structure of the paper is as follows: Section 2 summarizes some background material on the calculus. Section 3 defines two weak open congruences: weak early and late open congruences. Section 4 gives an equivalent account of the weak barbed congruence. Examples are provided to give the reader a glimpse of the complexity of the relation. Section 5 studies a rectification of the hyperequivalence: the ground open congruence. The difference between the ground open congruence and the weak barbed congruence is pointed out. Section 6 discusses some basic equational laws for the calculus. Section 7 proposes four new tau laws to handle tau prefixes under other prefix combinators. Section 8 establishes all the completeness results. Section 9 locates the barbed bisimilarity and the ground bisimilarity in bisimulation lattice and shows that, to a certain extent, they are the only bisimilarities for the calculus. Some comments are made in the final section.

Extended abstracts of this work have been published in [11,12].

2. The full χ-calculus with mismatch

The π-calculus has been shown to be a powerful language for concurrent computation. From the algebraic point of view, the model is slightly inconvenient due to the presence of two classes of bound names. The input prefix operator $a(x)$ introduces the bound name x to be instantiated by an action induced by the prefix operator. On the other hand the restriction operator (y) in $(y)P$ forces the name y to be bound in P, which will never be instantiated. Semantically these two bound names are very different. The following two examples suffice to make the point clear:

$$a(x).(P \,|\, [x=y]Q) \,|\, \bar{a}y.R \xrightarrow{\tau} (P\{y/x\} \,|\, [y=y]Q\{y/x\}) \,|\, R, \tag{1}$$

$$a(x).(P \,|\, [x=y]Q) \,|\, (z)\bar{a}z.R \xrightarrow{\tau} (z)((P\{z/x\} \,|\, [z=y]Q\{z/x\}) \,|\, R). \tag{2}$$

In (1) the subprocess $Q\{y/x\}$ can be fired after the internal communication whereas in (2) the component $[z=y]Q\{z/x\}$ will remain inactive forever since the bound name z will never be identified with any other name.

The χ-calculus can be seen as obtained from the π-calculus by unifying the two classes of bound names. The approach is to unify the input prefix and the output prefix. In χ-calculus a prefix takes the form of $\alpha x.P$, where α stands for either a name a or a coname $\bar{a}$. The most important thing is that the explicit x in $\alpha x.P$ is a free name. In χ-calculus the above two reductions become the following ones:

$$(x)ax.(P \,|\, [x=y]Q) \,|\, \bar{a}y.R \xrightarrow{\tau} (P\{y/x\} \,|\, [y=y]Q\{y/x\}) \,|\, R, \tag{3}$$

$$(x)ax.(P \,|\, [x=y]Q) \,|\, (z)\bar{a}z.R \xrightarrow{\tau} (z)((P\{z/x\} \,|\, [z=y]Q\{z/x\}) \,|\, R). \tag{4}$$

In (3) the effect of the communication is to substitute the free name y for the bound name x throughout the process over which the restriction operator (x) applies. In (4) the communication identifies two bound names. The difference between (2) and (4) is that in the latter the component $[z=y]Q\{z/x\}$ could be activated since further communication might replace the bound name z by y. This is because in χ-calculus the effect of a communication is delimited not by prefix operations, as in the π-calculus, but by the restriction operator. This is clear from the following examples:

$$(x)(ax.P \,|\, [x=y]Q) \,|\, \bar{a}y.R \xrightarrow{\tau} (P\{y/x\} \,|\, [y=y]Q\{y/x\}) \,|\, R, \tag{5}$$

$$(x)(ax.P \,|\, [x=y]Q) \,|\, (z)\bar{a}z.R \xrightarrow{\tau} (z)((P\{z/x\} \,|\, [z=y]Q\{z/x\}) \,|\, R). \tag{6}$$

Another distinguished property of the χ-calculus is that communications are symmetric. This can already been seen from (6) since we could equally have substituted x for z as in

$$(x)(ax.P \mid [x=y]Q) \mid (z)\bar{a}z.R \xrightarrow{\tau} (x)((P \mid [x=y]Q) \mid R\{x/z\}).$$

A symmetric version of (5) is

$$(x)(\bar{a}x.P \mid [x=y]Q) \mid ay.R \xrightarrow{\tau} (P\{y/x\} \mid [y=y]Q\{y/x\}) \mid R.$$

So the restriction operator in the χ-calculus plays a more important role than in the π-calculus.

There is also a polyadic version of the χ-calculus of course [28]. It is difficult to describe the operational semantics of this calculus using a labeled transition system. The following examples of communication should help to explain why

$$(x)(b)(axy.P \mid \bar{a}ab.Q) \xrightarrow{\tau} P\{a/x\}\{y/b\} \mid Q\{a/x\}\{y/b\}, \quad (7)$$

$$(x)(b)(axx.P \mid \bar{a}ab.Q) \xrightarrow{\tau} P\{a/x\}\{a/b\} \mid Q\{a/x\}\{a/b\}. \quad (8)$$

In (7) either of the two prefix operators induces both an input action and an output action in the traditional sense. In (8) the communication instantiates the bound name x by the free name a and at the same time the bound name b should be identified with x. But since the bound name to which b is identified is to be replaced by a, it might as well to replace b by a too.

The algebraic theory of χ-calculus has been systematically studied. In [6] a class of bisimulation equalities called L-bisimilarities were proposed and investigated. In [8] similar study has been carried out for the asymmetric χ-calculus. When restricted to finite processes, the congruence relations derived from these bisimilarities have all been axiomatized. So the initial work about the χ-calculus has all been done. However our knowledge about the χ-calculus with the mismatch operator is almost nil.

The calculus studied in this paper is the χ-calculus extended with the mismatch operator. This language will be referred to as the $\chi^{\neq}$-calculus in the rest of the paper. We will write $\mathscr{C}$ for the set of $\chi^{\neq}$-processes defined by the following grammar:

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

where $\alpha \in \mathscr{N} \cup \bar{\mathscr{N}}$. Here $\mathscr{N}$ is the set of names ranged over by small case letters. The set $\{\bar{x} \mid x \in \mathscr{N}\}$ of conames is denoted by $\bar{\mathscr{N}}$. We have left out replication processes since we will be focusing on axiomatization of equivalences on finite processes. The name x in $(x)P$ is bound. A name is free in P if it is not bound in P. The free names, the bound names and the names of P, as well as the notations $fn(P)$, $bn(P)$ and $n(P)$, are used in their standard meanings. In sequel we will use the functions $fn(\_)$, $bn(\_)$ and $n(\_)$ without explanation. We write $\bar{\alpha}$ for $\bar{a}$ if $\alpha = a$ and for a if $\alpha = \bar{a}$.

The following labeled transition system defines the operational semantics:

Sequentialization:

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

Composition:

$$\frac{P \xrightarrow{\gamma} P' \quad bn(\gamma) \cap fn(Q) = \emptyset}{P \mid Q \xrightarrow{\gamma} P' \mid Q} Cmp_0, \qquad \frac{P \xrightarrow{y/x} P'}{P \mid Q \xrightarrow{y/x} P' \mid Q\{y/x\}} Cmp_1.$$

Communication:

$$\frac{P \xrightarrow{\alpha(x)} P' \qquad Q \xrightarrow{\bar{\alpha}y} Q'}{P \mid Q \xrightarrow{\tau} P'\{y/x\} \mid Q'} Cmm_0, \qquad \frac{P \xrightarrow{\alpha(x)} P' \qquad Q \xrightarrow{\bar{\alpha}(x)} Q'}{P \mid Q \xrightarrow{\tau} (x)(P' \mid Q')} Cmm_1,$$

$$\frac{P \xrightarrow{\alpha x} P' \qquad Q \xrightarrow{\bar{\alpha}y} Q' \qquad x \neq y}{P \mid Q \xrightarrow{y/x} P'\{y/x\} \mid Q'\{y/x\}} Cmm_2, \qquad \frac{P \xrightarrow{\alpha x} P' \qquad Q \xrightarrow{\bar{\alpha}x} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'} Cmm_3.$$

Restriction:

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

Condition:

$$\frac{P \xrightarrow{\lambda} P'}{[x = x]P \xrightarrow{\lambda} P'} Mtch, \qquad \frac{P \xrightarrow{\lambda} P' \qquad x \neq y}{[x \neq y]P \xrightarrow{\lambda} P'} Mismtch.$$

Summation:

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

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 $\{\alpha(x), \alpha x, y/x \mid \alpha \in \mathcal{N} \cup \bar{\mathcal{N}},\ x, y \in \mathcal{N}\} \cup \{\tau\}$ of all actions. There are four kinds of actions:

- The label $\alpha(x)$ represents a bound action that exchanges the bound name x at channel α. The x in $\alpha(x)$ is bound.
- The label αx stands for a free action that exchanges the free name x at channel α. The x in αx is free.
- The label y/x indicates an update action, an incomplete communication so to speak, that replaces x by y half way through a communication. The x, y in y/x are free.
- The label τ as usual stands for a communication.

The process $Q\{y/x\}$ appeared in the above labeled transitional system is obtained by substituting y for x throughout Q. A substitution $\{y_1/x_1, \ldots, y_n/x_n\}$ is a function from $\mathcal{N}$ to $\mathcal{N}$ that maps x_i onto y_i for $i \in \{1, \ldots, n\}$ and x onto itself for $x \notin \{x_1, \ldots, x_n\}$. Substitutions are usually denoted by σ, σ' etc. The empty substitution, that is the identity function on $\mathcal{N}$, is written as $\{\,\}$. The result of applying σ to P is denoted by $P\sigma$.

Notice that a substitution σ may disable an action of P. So $P \xrightarrow{ax} P'$, say, does not imply $P\sigma \xrightarrow{\sigma(a)\sigma(x)} P'\sigma$. But if σ does not disable the action ax, then $P\sigma \xrightarrow{\sigma(a)\sigma(x)} P'\sigma$ follows.

Most of the operational rules are straightforward. For someone familiar with the π-calculus, only the rules Cmp_1, Cmm_2, Cmm_3 and Loc_2 need explanation. The rule Cmm_2 introduces an update action. An update action can be seen as an incomplete communication. In a structural labeled transitional semantics for χ-calculus, update actions have to be introduced. An update action has side effect on neighboring processes. This explains the rule Cmp_1. The rule Cmm_3 is a matter of design decision. It permits the following communication

$$ax.P \mid \bar{a}x.Q \xrightarrow{\tau} P \mid Q.$$

We could have worked with the χ-calculus that bans the above reduction. But the present version slightly simplifies the algebraic theory.

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 the 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 $(\mu P, \ \delta P)$ 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. In what follows we will often use a substitution that draws a particular relationship with a condition. Some of these relationships are made precise in the following definition.

Definition 1. A substitution σ respects ψ if $\psi \Rightarrow x = y$ implies $\sigma(x) = \sigma(y)$ and $\psi \Rightarrow x \neq y$ implies $\sigma(x) \neq \sigma(y)$. Dually ψ respects σ if $\sigma(x) = \sigma(y)$ implies $\psi \Rightarrow x = y$ and $\sigma(x) \neq \sigma(y)$ implies $\psi \Rightarrow x \neq y$. The substitution σ agrees with ψ, and ψ agrees with σ, if they respect each other. The substitution σ is induced by ψ if it agrees with ψ and $n(\sigma) \subseteq n(\psi)$.

Intuitively a substitution σ is induced by ψ if it maps all the elements of an equivalence class induced by ψ onto a representative of the class.

The notation $\Rightarrow$ stands for the reflexive and transitive closure of $\xrightarrow{\tau}$ and $\overset{\lambda}{\Rightarrow}$ for the composition $\Rightarrow \xrightarrow{\lambda} \Rightarrow$. The relation $\overset{\hat{\lambda}}{\Rightarrow}$ is the same as $\overset{\lambda}{\Rightarrow}$ if $\lambda \neq \tau$ and is $\Rightarrow$ otherwise. A sequence $x_1, \ldots, x_n$ of names will be abbreviated to $\tilde{x}$. So $(\tilde{x})P$ stands for $(x_1) \ldots (x_n)P$. When the length of $\tilde{x}$ is zero $(\tilde{x})P$ is simply P. We will use three induced prefix operators, update prefix, tau prefix and bound prefix, defined as follows:

$$\langle y|x \rangle . P \overset{\text{def}}{=} (a)(\bar{a}y \mid ax.P),$$

$$\tau.P \overset{\text{def}}{=} (b)\langle b|b \rangle . P,$$

$$\alpha(x).P \overset{\text{def}}{=} (x)\alpha x.P,$$

where a, b are fresh. Notice that the update prefix can perform two symmetric update actions:

$$\langle y|x\rangle.P \xrightarrow{y/x} P\{y/x\},$$

$$\langle y|x\rangle.P \xrightarrow{x/y} P\{x/y\}.$$

In what follows we will overload the use of λ by letting it also range over the set $\{\alpha(x), \alpha x, \langle y|x\rangle \mid \alpha \in \mathcal{N} \cup \bar{\mathcal{N}},\ x, y \in \mathcal{N}\} \cup \{\tau\}$ of extended prefixes.

The notion of context is very important to the algebraic theory of process calculus. So we give a formal definition as follows.

Definition 2. Contexts are defined inductively as follows:

(i) $[\,]$ is a context;

(ii) If $C[\,]$ is a context then $\alpha x.C[\,]$, $C[\,] \mid P$, $P \mid C[\,]$, $(x)C[\,]$ and $[x=y]C[\,]$ are contexts.

Full contexts are those contexts that satisfy additionally:

(iii) If $C[\,]$ is a context then $C[\,]+P$, $P+C[\,]$ and $[x \neq y]C[\,]$ are contexts.

3. Early and late bisimilarities

Many observational equivalence relations have been proposed for calculi of mobile processes. The most well-known of them include the early equivalence [24], the late equivalence [24], the barbed equivalence [25,30], the open equivalence [31] and the testing equivalence [3]. The first four are bisimulation equivalence relations whereas the last one is not. All these relations are closed under substitutions in order for the relations to be closed under prefix operations. But there is a notable difference. The open equivalence is a bisimulation equivalence closed under substitution of names in every simulating step. On the other hand the other four equivalence relations are closed under substitution only on the very beginning of simulations. In our view the most natural bisimulation equivalence for mobile processes is the open equivalence introduced by Sangiorgi [31]. The open approach assumes that the environments are dynamic in the sense that after each computation step, the environment might be totally different. As a matter of fact the very idea of bisimulation is to ensure that no operational difference can be detected by any dynamic environment. So closure under substitution is a reasonable requirement.

A naive definition of weak open bisimulation for $\chi^{\neq}$-calculus would go as follows:

> A binary relation $\mathcal{R}$ on $\mathcal{C}$ is a weak open bisimulation if it is symmetric and closed under substitution such that whenever $P\mathcal{R}Q$ and $P \xrightarrow{\lambda} P'$ then $Q \stackrel{\hat{\lambda}}{\Rightarrow} Q'\mathcal{R}P'$ for some Q'.

This definition is good for the χ-calculus without the mismatch operator. But as it turns out it is not closed under the parallel composition operation for processes with the mismatch operator. Counter examples are given in the introduction. The problem here is that the instantiation of names is delayed for any period of time. This is not always possible in $\chi^{\neq}$-calculus since the instantiation might falsify an inequality condition. This problem does not occur in the χ-calculus because an instantiation only validates, but never invalidates, an equality condition.

There could be many ways to rectify the above definition. In this section we seek to correct it in a most straightforward manner. Since the problem is caused by the delay of instantiation of names, we insist that name instantiations should take place in the earliest possible occasion. This brings us to the familiar early and late frameworks.

Definition 3. Let $\mathcal{R}$ be a binary symmetric relation on $\mathcal{C}$. It is called an early open bisimulation if it is closed under substitution and whenever $P\mathcal{R}Q$ then the following properties hold:

(i) If $P \stackrel{\tau}{\rightarrow} P'$ then Q' exists such that $Q \Rightarrow Q'\mathcal{R}P'$.
(ii) If $P \stackrel{y/x}{\rightarrow} P'$ then Q' exists such that $Q \stackrel{y/x}{\Rightarrow} Q'\mathcal{R}P'$.
(iii) If $P \stackrel{\alpha x}{\rightarrow} P'$ then for every y some Q', Q'' exist such that $Q \Rightarrow \stackrel{\alpha x}{\rightarrow} Q''$ and $Q''\{y/x\} \Rightarrow Q'\mathcal{R}P'\{y/x\}$.
(iv) If $P \stackrel{\alpha(x)}{\rightarrow} P'$ then for every y some Q', Q'' exist such that $Q \Rightarrow \stackrel{\alpha(x)}{\rightarrow} Q''$ and $Q''\{y/x\} \Rightarrow Q'\mathcal{R}P'\{y/x\}$.

The early open bisimilarity $\approx_o^e$ is the largest early open bisimulation.

Clause (iv) is easy to understand. Its counterpart for weak bisimilarity of π-calculus is familiar. Clause (iii) calls for some explanation. In $\chi^{\neq}$-calculus free actions can also incur name updates in suitable contexts. Suppose $P \stackrel{\alpha x}{\rightarrow} P''$. Then $(x)(P \mid \bar{\alpha} y.Q) \stackrel{\tau}{\rightarrow} P''\{y/x\} \mid Q\{y/x\}$. Even if $P'' \Rightarrow P'$, one does not necessarily have $P''\{y/x\} \Rightarrow P'\{y/x\}$. Had we replaced clause (iii) by (iii′) If $P \stackrel{\alpha x}{\rightarrow} P'$ then some Q' exists such that $Q \stackrel{\alpha x}{\Rightarrow} Q'\mathcal{R}P'$ then we would have obtained the problematic equation involving mismatch in Section 1. The similarity of clause (iii) and clause (iv) exhibits once again the uniformity of the names in χ-like calculi.

Analogously we can introduce late open bisimilarity.

Definition 4. Let $\mathcal{R}$ be a binary symmetric relation on $\mathcal{C}$. It is called a late open bisimulation if it is closed under substitution and whenever $P\mathcal{R}Q$ then the following properties hold:

(i) If $P \stackrel{\tau}{\rightarrow} P'$ then Q' exists such that $Q \Rightarrow Q'\mathcal{R}P'$.
(ii) If $P \stackrel{y/x}{\rightarrow} P'$ then Q' exists such that $Q \stackrel{y/x}{\Rightarrow} Q'\mathcal{R}P'$.
(iii) If $P \stackrel{\alpha x}{\rightarrow} P'$ then Q'' exists such that $Q \Rightarrow \stackrel{\alpha x}{\rightarrow} Q''$ and for every y some Q' exists such that $Q''\{y/x\} \Rightarrow Q'\mathcal{R}P'\{y/x\}$.
(iv) If $P \stackrel{\alpha(x)}{\rightarrow} P'$ then Q'' exists such that $Q \Rightarrow \stackrel{\alpha(x)}{\rightarrow} Q''$ and for every y some Q' exists such that $Q''\{y/x\} \Rightarrow Q'\mathcal{R}P'\{y/x\}$.

The late open bisimilarity $\approx_o^l$ is the largest late open bisimulation.

It is clear that $\approx_o^l \subseteq \approx_o^e$. The following example shows that inclusion is strict:

$$ax.[x = y]\tau.P + ax.[x \neq y]\tau.P \approx_o^e ax.[x = y]\tau.P + ax.[x \neq y]\tau.P + ax.P$$

but not

$$ax.[x=y]\tau.P + ax.[x \neq y]\tau.P \approx_o^l ax.[x=y]\tau.P + ax.[x \neq y]\tau.P + ax.P$$

since the action

$$ax.[x=y]\tau.P + ax.[x \neq y]\tau.P + ax.P \xrightarrow{ax} P$$

cannot be matched by any action from $ax.[x=y]\tau.P + ax.[x \neq y]\tau.P$ in the late approach.

The next lemma assures that both the early open bisimilarity and the late open bisimilarity are closed under parallel operation.

Lemma 5. *Both $\approx_o^e$ and $\approx_o^l$ are closed under the restriction and composition operations.*

Proof. We prove the lemma for $\approx_o^e$. The proof for $\approx_o^l$ is similar. Let $\mathscr{R}$ be the following relation:

$$\{((\tilde{x})(P \mid R), (\tilde{x})(Q \mid R)) \mid P \approx_o^e Q\}.$$

We prove that $\mathscr{R}$ is an early open bisimulation. It is clear that $\mathscr{R}$ is closed under substitution. Now suppose $P \approx_o^e Q$. Consider two cases:

- Suppose that $(\tilde{x})(P \mid R) \xrightarrow{\alpha(x)} (\tilde{x'})(P' \mid R')$ is caused by $P \xrightarrow{\alpha x} P'$ such that $x \in \{\tilde{x}\}$. By definition we get that, for every y, there exist Q', Q'' such that $Q \Rightarrow \xrightarrow{\alpha x} Q''$ and $Q''\{y/x\} \Rightarrow Q' \approx_o^e P'\{y/x\}$. So $(\tilde{x})(Q \mid R) \xrightarrow{\alpha(x)} (\tilde{x'})(Q'' \mid R')$ and

$$\begin{aligned}(\tilde{x'})(Q''\{y/x\} \mid R'\{y/x\}) &\Rightarrow (\tilde{x'})(Q' \mid R'\{y/x\}) \\ &\mathrel{\mathscr{R}} (\tilde{x'})(P'\{y/x\} \mid R'\{y/x\}).\end{aligned}$$

- Suppose that $(\tilde{x})(P \mid R) \xrightarrow{\tau} (\tilde{x'})(y)(P'\{y/x\} \mid R'\{y/x\})$ is caused by $P \xrightarrow{\alpha x} P'$, for some $x \in \{\tilde{x}\}$, and $R \xrightarrow{\overline{\alpha}(y)} R'$. Then there exist Q', Q'' such that $Q \Rightarrow \xrightarrow{\alpha x} Q''$ and $q''\{Y/X\} \Rightarrow Q' \approx_O^E p'\{Y/X\}$. It follows that

$$\begin{aligned}(\tilde{x})(Q \mid R) &\Rightarrow (\tilde{x'})(y)(Q''\{y/x\} \mid R'\{y/x\}) \\ &\Rightarrow (\tilde{x'})(y)(Q' \mid R'\{y/x\}) \\ &\mathrel{\mathscr{R}} (\tilde{x'})(y)(P'\{y/x\} \mid R'\{y/x\}).\end{aligned}$$

Conclude that $\approx_o^e$ is closed under restriction and parallel composition. □

Neither $\approx_o^e$ nor $\approx_o^l$ is closed under the choice combinator and the mismatch operator. For instance neither $P + Q \approx_o^e \tau.P + Q$ nor $[x \neq y]P \approx_o^e [x \neq y]\tau.P$ necessarily holds, although $P \approx_o^e \tau.P$ is valid. To obtain the largest congruence relation contained in $\approx_o^e$ ($\approx_o^l$), we apply the standard approach [22]. This standard approach was originally proposed to close up a weak equivalence under the choice combinator. As it turns out the relation thus obtained is also closed under the mismatch operator.

Definition 6. Two processes P and Q are early open congruent, notation $P \simeq_o^e Q$, if $P \approx_o^e Q$ and, for each substitution σ, the following conditions are satisfied:
(i) If $P\sigma \stackrel{\tau}{\rightarrow} P'$ then Q' exists such that $Q\sigma \stackrel{\tau}{\Rightarrow} Q'$ and $P' \approx_o^e Q'$.
(ii) If $Q\sigma \stackrel{\tau}{\rightarrow} Q'$ then P' exists such that $P\sigma \stackrel{\tau}{\Rightarrow} P'$ and $P' \approx_o^e Q'$.
The late open congruence $\simeq_o^l$ is defined similarly.

It is clear that $\simeq_o^e$ is defined in terms of $\approx_o^e$. The difference between the two is that the former requires a first tau action of P be simulated by a *nonempty* sequence of tau moves from Q whereas the latter requires a first tau action of P be simulated by a sequence, possibly an empty sequence, of tau moves from Q. The stronger requirement of $\simeq_o^e$ is to make sure that things are closed under the choice operator as well as the mismatch operator. For early open bisimilarity it could be that $Q+P \not\approx_o^e Q+\tau.P$, although $P \approx_o^e \tau.P$. The Definition 6 rules out situation like this for $\simeq_o^e$. The properties (i) and (ii) should hold for every substitution σ for otherwise $\simeq_o^e$ would not be closed under prefix operation. For instance a first tau action of $P \,|\, [x=y]\tau.Q$, where x and y are distinct, can be simulated by a first tau action of $P \,|\, [x=y]Q$ since no actions from either $[x=y]\tau.Q$ or $[x=y]Q$ are allowed. But $a(x).(R+P \,|\, [x=y]\tau.Q)$ is not early open bisimilar to $a(x).(R+P \,|\, [x=y]Q)$ because after the first computation step the name x might be identified to y.

Lemma 7. *Both $\simeq_o^e$ and $\simeq_o^l$ are congruence relations.*

Proof. The proof that $\simeq_o^e$ and $\simeq_o^l$ are equivalent is routine.

Suppose $P \simeq_o^e Q$. Let $C[\,]$ be a context and σ be a substitution. If $C[P\sigma] \stackrel{\lambda}{\rightarrow} C'[P\sigma']$ is caused by an action induced by the context $C[\,]$ then $C[Q\sigma] \stackrel{\lambda}{\rightarrow} C'[Q\sigma']$ matches up the action. If $C[P\sigma] \stackrel{a(x)}{\rightarrow} C'[P']$ is caused by an action induced by $P\sigma \stackrel{a(x)}{\rightarrow} P'$ then for each y some Q' and Q'' exist such that $Q\sigma \Rightarrow \stackrel{a(x)}{\rightarrow} Q''$ and $Q''\{y/x\} \Rightarrow Q' \simeq_o^e P'\{y/x\}$. Consequently $C[Q\sigma] \Rightarrow \stackrel{a(x)}{\rightarrow} C'[Q'']$ and $C'\{y/x\}[Q''\{y/x\}] \Rightarrow C'\{y/x\}[Q'] \simeq_o^e C'\{y/x\}[P'\{y/x\}]$.

The reader can easily check out the rest of the cases. □

It is easy to check that $P \simeq_o^e Q$ ($P \simeq_o^l Q$) if and only if $C[P] \approx_o^e C[Q]$ ($C[P] \approx_o^l C[Q]$) for every full context $C[\,]$. As a matter of fact the implication from the left to the right is given by Lemma 7. For the reverse implication, simply let $C[\,]$ be $\_ + \bar{a}a$ for a fresh name a.

4. Barbed bisimilarity

The barbed equivalence, introduced by Milner and Sangiorgi in [25], is often quoted as a universal equivalence relation for process algebra. The definition of barbed equivalence is often based on a reduction semantics introduced by Berry and Boudol [2] and Milner [23]. The reduction based algebraic semantics has also been studied by Honda and Yoshida [16]. 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. The barbed approach has been quite successful in the study of a number of offsprings of the π-calculus. Examples include the higher order π-calculus [30], the asynchronous π-calculus [1,19] and object calculus [14,15].

Despite the universal nature, barbed equivalence may enjoy quite different properties in different process calculi. 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 $\chi^{\neq}$-processes is provided. Some illustrating pairs of barbed equivalent processes are given. First we introduce the notion of barbedness.

Definition 8. 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 $a\in\{\alpha,\bar{\alpha}\}$. P is barbed at a, written $P\Downarrow a$, if some P' exists such that $P\Rightarrow P'\downarrow a$. A binary relation $\mathscr{R}$ is barbed if $\forall a\in\mathcal{N}.P\Downarrow a\Leftrightarrow Q\Downarrow a$ whenever $P\mathscr{R}Q$.

From the point of view of barbed equivalence an observer cannot 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 9. Let $\mathscr{R}$ be a barbed symmetric relation on $\mathscr{C}$. The relation $\mathscr{R}$ is a barbed bisimulation if whenever $P\mathscr{R}Q$ and $P\stackrel{\tau}{\longrightarrow}P'$ then $Q\Rightarrow Q'\mathscr{R}P'$ for some Q'. The barbed bisimilarity $\approx_b$ is the largest barbed bisimulation closed under context.

Some explanation is called for. The barbed bisimilarity defined above is different from the popular one since $\approx_b$ is required to be closed under context. We have adopted this definition since barbed bisimilarity is a *bisimulation* equivalence. Like any other bisimulation equivalence, it should be tested against dynamic environments. In other words, it must be closed under all contexts at each bisimulation step. This version of barbed congruence was initially studied by Honda and Yoshida in [16]. Recent works on barbed congruence are increasingly using this version. For example Sangiorgi and Walker have related this version of barbed congruence to quasi-open congruence [32].

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. One way to understand the barbed bisimilarity is that the observing power of environments are so weak that they can only detect the *effects* of the actions performed by the observees. In other words, two processes are equivalent if they can always exert same effects on environments. Now suppose that $P\approx_b Q$ and that P wants to exchange the name x for the name y from an environment at channel α by performing $P\stackrel{\alpha x}{\longrightarrow}P'$. For the exchange to happen the environment must be able to perform an observing action which is $\bar{\alpha}y$. In this case in order for Q to deliver the same effect as the action αx could, Q can do one of the following things:

- $Q \Rightarrow \xrightarrow{\alpha x} Q''$ and $Q''\{y/x\} \Rightarrow Q' \approx_b P'\{y/x\}$. This means that Q does absolutely the same thing.
- $Q \Rightarrow \xrightarrow{\alpha(z)} Q''$ and $Q''\{y/z\} \stackrel{y/x}{\Rightarrow} Q' \approx_b P'\{y/x\}$. In this case Q receives the name y at channel α and then cheats on the environment by delivering the effect of the exchange of x for y through incurring the exchange on its own.
- $Q \stackrel{\alpha y}{\Rightarrow} \stackrel{y/x}{\Rightarrow} Q' \approx_b P'\{y/x\}$. Here the first thing Q does is to exchange y for y at channel α, which is nothing but to take in as it were the observing action of the environment. Then it incurs an exchange of x for y on its own, achieving the same effect on the environment.
- $Q \stackrel{y/x}{\Rightarrow} \stackrel{\alpha y}{\Rightarrow} Q' \approx_b P'\{y/x\}$. The situation is similar to the one in previous case. But now the cheating happens even before the consumption of the observing action.
- $Q \stackrel{y/x}{\Rightarrow} \xrightarrow{\alpha(z)} Q''$ and $Q''\{y/z\} \Rightarrow Q' \approx_b P'\{y/x\}$. Like in previous case, here the observee delivers the effect first, but then finds another way to consume the observing action.

From the point of view of a mobile process, its interaction with an environment consists of two ingredients: One is the consumption of the observing move of the environment; the other is the delivery of the effect of the interaction. In π-calculus the two things always go together. In χ-calculus however they may happen at different points of the interaction.

With above observation in mind, we now give some examples of barbed equivalent processes that substantiate our intuition. Most of the examples in this paper involve long expressions. To make things 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 equivalent pair is this:

$$A_1 \stackrel{\text{def}}{=} \alpha x.(P_1 + [x = y_1]\tau.Q) + \alpha x.(P_2 + [x \neq y_1]\tau.Q) \approx_b A_1 + \alpha x.Q. \tag{9}$$

If the component $\alpha x.Q$ 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 role of the tau prefix is to remove the match or the mismatch operator. The second example is more interesting:

$$\begin{aligned} A_2 &\stackrel{\text{def}}{=} \alpha(z).(P_1 + [z = y_2]\langle z|x\rangle.Q) + \alpha x.(P_2 + [x \neq y_2]\tau.Q\{x/z\}) \\ &\approx_b A_2 + \alpha x.Q\{x/z\}. \end{aligned} \tag{10}$$

The communication $\bar{\alpha} y_2 \,|\, (x)(A_2 + \alpha x.Q\{x/z\}) \xrightarrow{\tau} \mathbf{0} \,|\, Q\{x/z\}\{y_2/x\}$ for instance can be matched up by the following communication:

$$\bar{\alpha} y_2 \,|\, (x)A_2 \xrightarrow{\tau} \mathbf{0} \,|\, (x)(P_1\{y_2/z\} + [y_2 = y_2]\langle y_2|x\rangle.Q\{y_2/z\}) \xrightarrow{\tau} \mathbf{0} \,|\, Q\{y_2/z\}\{y_2/x\}.$$

For a name w distinct from y_2 the action

$$\bar{\alpha} w \,|\, (x)(A_2 + \alpha x.Q\{x/z\}) \xrightarrow{\tau} \mathbf{0} \,|\, Q\{x/z\}\{w/x\}$$

can be matched by

$$\bar{\alpha}w \,|\, (x)A_2 \xrightarrow{\tau} \mathbf{0} \,|\, P_2\{w/x\} + [w \neq y_2]\tau.Q\{x/z\}\{w/x\} \xrightarrow{\tau} \mathbf{0} \,|\, Q\{w/z\}\{w/x\}.$$

Another possibility arises when the name y_2 is bound. In this case the communication

$$(y_2)(\bar{\alpha}y_2 \,|\, (A_2 + \alpha x.Q\{x/z\})) \xrightarrow{\tau} \mathbf{0} \,|\, Q\{x/z\}\{x/y_2\}$$

for instance is matched by

$$(y_2)(\bar{\alpha}y_2 \,|\, A_2) \xrightarrow{\tau} (y_2)(\mathbf{0} \,|\, (P_1 + [y_2 = y_2]\langle y_2|x\rangle.Q\{y_2/z\})) \xrightarrow{\tau} \mathbf{0} \,|\, Q\{y_2/z\}\{x/y_2\}.$$

The third example is unusual:

$$A_3 \overset{\text{def}}{=} \alpha y_3.(P_1 + \langle y_3|x\rangle.Q) + \alpha x.(P_2 + [x \neq y_3]\tau.Q) \approx_b A_3 + \alpha x.Q. \tag{11}$$

If the component $\alpha x.Q$ participates in a communication in which x exchanges for y_3 then its role can be simulated by $\alpha y_3.(P_1 + \langle y_3|x\rangle.Q)$. For instance

$$(x)((A_3 + \alpha x.Q) \,|\, \bar{\alpha}y_3) \xrightarrow{\tau} Q\{y_3/x\} \,|\, \mathbf{0}$$

is simulated by the following reduction:

$$(x)(\alpha y_3.(P_1 + \langle y_3|x\rangle.Q) \,|\, \bar{\alpha}y_3) \xrightarrow{\tau} (x)((P_1 + \langle y_3|x\rangle.Q) \,|\, \mathbf{0}) \xrightarrow{\tau} Q\{y_3/x\} \,|\, \mathbf{0}$$

The fourth example, given below, is similar to the third one:

$$A_4 \overset{\text{def}}{=} \langle y_4|x\rangle.(P_1 + \alpha y_4.Q) + \alpha x.(P_2 + [x \neq y_4]\tau.Q) \approx_b A_4 + \alpha x.Q. \tag{12}$$

If for example $(x)((A_4 + \alpha x.Q) \,|\, \bar{\alpha}y_4.O) \xrightarrow{\tau} Q\{y_4/x\} \,|\, O\{y_4/x\}$ then the simulation is as follows:

$$(x)(A_4 \,|\, \bar{\alpha}y_4.O) \xrightarrow{\tau} (P_1\{y_4/x\} + \alpha y_4.Q\{y_4/x\}) \,|\, \bar{\alpha}y_4.O\{y_4/x\} \xrightarrow{\tau} Q\{y_4/x\} \,|\, O\{y_4/x\}$$

The fifth example is the combination of (10) and (12):

$$A_5 \overset{\text{def}}{=} \langle y_5|x\rangle.(P_1 + \alpha(z).(P_1' + [z = y_5]\tau.Q)) + \alpha x.(P_2 + [x \neq y_5]\tau.Q\{x/z\})$$
$$\approx_b A_5 + \alpha x.Q\{x/z\}. \tag{13}$$

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

In the above examples, all the explicit mismatch operators contain the name x. In general there could be other conditions. The treatment of the match operator is easy. The mismatch operator is however nontrivial. Suppose δ is a sequence of mismatch operators such that all names in δ are different from both x and z. An example more general than (9) is this:

$$A_1' \overset{\text{def}}{=} \alpha x.(P_1 + \delta[x = y_1]\tau.Q) + \alpha x.(P_2 + \delta[x \neq y_1]\tau.Q)$$
$$\approx_b A_1' + [x \notin n(\delta)]\delta\alpha x.Q. \tag{14}$$

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 condition $[x\notin n(\delta)]$ is necessary because otherwise (14) would not be closed under substitution. A counter example is given by the pair of processes:

$$\alpha x.[y\neq z][x=y_1]\tau.Q+\alpha x.[y\neq z][x\neq y_1]\tau.Q$$

and

$$\alpha x.[y\neq z][x=y_1]\tau.Q+\alpha x.[y\neq z][x\neq y_1]\tau.Q+[y\neq z]\alpha x.Q.$$

If we substitute x for z in the two processes we get the following two processes:

$$\alpha x.[y\neq x][x=y_1]\tau.Q\{z/x\}+\alpha x.[y\neq x][x\neq y_1]\tau.Q\{z/x\}$$

and

$$\alpha x.[y\neq x][x=y_1]\tau.Q\{z/x\}+\alpha x.[y\neq x][x\neq y_1]\tau.Q\{z/x\}+[y\neq x]\alpha x.Q\{z/x\}$$

which are not barbed bisimilar. This is because the communication

$$(D+[y\neq x]\alpha x.Q\{z/x\})\,|\,\bar{\alpha}y\xrightarrow{\tau}Q\{z/x\}\{y/x\},$$

where D is $\alpha x.[y\neq x][x=y_1]\tau.Q\{z/x\}+\alpha x.[y\neq x][x\neq y_1]\tau.Q\{z/x\}$, cannot be simulated by $D\,|\,\bar{\alpha}y$ in general. Similarly example (10) can be generalized to the following:

$$\begin{aligned}A_2' &\stackrel{\text{def}}{=} \alpha(z).(P_1+[x\notin n(\delta)]\delta[z=y_2]\langle z|x\rangle.Q)+\alpha x.(P_2+\delta[x\neq y_2]\tau.Q\{x/z\})\\ &\approx_b A_2'+[x\notin n(\delta)]\delta\alpha x.Q\{x/z\},\end{aligned} \tag{15}$$

where $z\notin n(\delta)\cup\{x\}$. Here the mismatch sequence $[x\notin n(\delta)]$ in the first summand of A_2' can be removed without affecting the validity of (15). But (15) as it stands is more general. The general form of (11) is more delicate:

$$\begin{aligned}A_3' &\stackrel{\text{def}}{=} [x\neq y_3]\alpha y_3.(P_1+[x\notin n(\delta)]\delta\langle y_3|x\rangle.Q)+\alpha x.(P_2+\delta[x\neq y_3]\tau.Q)\\ &\approx_b A_3'+[x\neq y_3][x\notin n(\delta)]\delta\alpha x.Q.\end{aligned} \tag{16}$$

In both $[x\neq y_3]\alpha y_3.(P_1+[x\notin n(\delta)]\delta\langle y_3|x\rangle.Q)$ 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 (16) one has

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

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

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

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 cannot be simulated by C_3'. The general forms of (12) and (13) are as follows:

$$A_4' \stackrel{\text{def}}{=} \langle y_4|x\rangle.(P_1 + \delta\alpha y_4.Q) + \alpha x.(P_2 + \delta[x \neq y_4]\tau.Q)$$
$$\approx_b A_4' + [x \notin n(\delta)]\delta\alpha x.Q, \tag{17}$$

$$A_5' \stackrel{\text{def}}{=} \langle y_5|x\rangle.(P_1 + \alpha(z).(P_1' + \delta[z = y_5]\tau.Q)) + \alpha x.(P_2 + \delta[x \neq y_5]\tau.Q\{x/z\})$$
$$\approx_b A_5' + [x \notin n(\delta)]\delta\alpha x.Q\{x/z\}. \tag{18}$$

If we replace in (14) the second summand $\alpha x.(P_2 + \delta[x \neq y_1]\tau.Q)$ of A_1' by

$$\alpha(z).(P_2 + [x \notin n(\delta)]\delta[z \neq y_1]\langle z|x\rangle.Q)$$

and Q by $Q\{x/z\}$, where $z \notin n(\delta) \cup \{x\}$, we get an interesting variant of (14) as follows:

$$A_1'' \stackrel{\text{def}}{=} \alpha x.(P_1 + \delta[x = y_1]\tau.Q\{x/z\}) + \alpha(z).(P_2 + [x \notin n(\delta)]\delta[z \neq y_1]\langle z|x\rangle.Q)$$
$$\approx_b A_1'' + [x \notin n(\delta)]\delta\alpha x.Q\{x/z\}. \tag{19}$$

If for instance w is distinct from y_1 then $\bar{\alpha}w \,|\, (x)(A_1'' + [x \notin n(\delta)]\delta\alpha x.Q\{x/z\}) \xrightarrow{\tau} \mathbf{0} \,|\, Q\{x/z\}\{w/x\}$ is matched by $\bar{\alpha}w \,|\, (x)A_1'' \xrightarrow{\tau} \mathbf{0} \,|\, (P_2 + [x \notin n(\delta)]\delta[w \neq y_1]\langle w|x\rangle.Q\{w/z\}) \xrightarrow{\tau} \mathbf{0} \,|\, Q\{w/z\}\{w/x\}$. The bisimilar pairs (15)–(18) have similar variants:

$$A_2'' \stackrel{\text{def}}{=} \alpha(z).(P_1 + [x \notin n(\delta)]\delta[z = y_2]\langle z|x\rangle.Q)$$
$$+ \alpha(z).(P_2 + [x \notin n(\delta)]\delta[z \neq y_2]\langle z|x\rangle.Q)$$
$$\approx_b A_2'' + [x \notin n(\delta)]\delta\alpha x.Q\{x/z\}, \tag{20}$$

$$A_3'' \stackrel{\text{def}}{=} [x \neq y_3]\alpha y_3.(P_1 + [x \notin n(\delta)]\delta\langle y_3|x\rangle.Q\{x/z\})$$
$$+ \alpha(z).(P_2 + [x \notin n(\delta)]\delta[z \neq y_3]\langle z|x\rangle.Q)$$
$$\approx_b A_3'' + [x \neq y_3][x \notin n(\delta)]\delta\alpha x.Q\{x/z\}, \tag{21}$$

$$A_4'' \stackrel{\text{def}}{=} \langle y_4|x\rangle.(P_1 + \delta\alpha y_4.Q\{x/z\}) + \alpha(z).(P_2 + [x \notin n(\delta)]\delta[z \neq y_4]\langle z|x\rangle.Q)$$
$$\approx_b A_4'' + [x \notin n(\delta)]\delta\alpha x.Q\{x/z\}, \tag{22}$$

$$A_5'' \stackrel{\text{def}}{=} \langle y_5|x\rangle.(P_1 + \alpha(z).(P_1' + \delta[z = y_5]\tau.Q\{z/x\}))$$
$$+ \alpha(z).(P_2 + [x \notin n(\delta)]\delta[z \neq y_5]\langle z|x\rangle.Q)$$
$$\approx_b A_5'' + [x \notin n(\delta)]\delta\alpha x.Q\{x/z\}. \tag{23}$$

The most complicated situation arises when all the five possibilities as described by (19) through (23) happen at one go:

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

Similarly examples (14)–(18) can be combined into one as follows:

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

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 10. 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 $Q \stackrel{\hat{\lambda}}{\Rightarrow} Q'\mathcal{R}P'$.

(ii) If $P \xrightarrow{\alpha x} P'$ then one of the following properties holds:

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

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

- Q' and Q'' exist such that $Q \Rightarrow \xrightarrow{\alpha x} Q''$ and $Q''\{y/x\} \Rightarrow Q'\mathcal{R}P'\{y/x\}$;

- Q' and Q'' exist such that $Q \Rightarrow \stackrel{\alpha(z)}{\longrightarrow} Q''$ and $Q''\{y/z\} \stackrel{y/x}{\Rightarrow} Q' \mathcal{R} P'\{y/x\}$;
- Q' exists such that $Q \stackrel{\alpha y}{\Rightarrow} \stackrel{y/x}{\Rightarrow} Q' \mathcal{R} P'\{y/x\}$;
- Q' exists such that $Q \stackrel{y/x}{\Rightarrow} \stackrel{\alpha y}{\Rightarrow} Q' \mathcal{R} P'\{y/x\}$;
- Q' and Q'' exist such that $Q \stackrel{y/x}{\Rightarrow} \stackrel{\alpha(z)}{\longrightarrow} Q''$ and $Q''\{y/z\} \Rightarrow Q' \mathcal{R} P'\{y/x\}$.

(iii) If $P \stackrel{\alpha(x)}{\longrightarrow} P'$ then Q' exists such that $Q \stackrel{\alpha(x)}{\Rightarrow} Q' \approx_o^b P'$ and, for each y distinct from x, one of the following properties holds:

- Q' and Q'' exist such that $Q \Rightarrow \stackrel{\alpha(x)}{\longrightarrow} Q''$ and $Q''\{y/x\} \Rightarrow Q' \mathcal{R} P'\{y/x\}$;
- Q' exists such that $Q \stackrel{\alpha y}{\Rightarrow} Q' \mathcal{R} P'\{y/x\}$.

The barbed open bisimilarity $\approx_o^b$ is the largest barbed open bisimulation.

With a definition as complex as Definition 10, it is not very clear that the relation it introduces is well behaved. The next lemma gives one some confidence on the barbed open bisimilarity.

Lemma 11. *$\approx_o^b$ is closed under context.*

Proof. Let $\mathcal{R}$ be $\{((\tilde{x})(P\,|\,R), (\tilde{x})(Q\,|\,R)) \mid P \approx_o^b Q\} \cup \approx_o^b$. We prove that $\mathcal{R}$ is a barbed open bisimulation. Suppose $(x)(P\,|\,R) \stackrel{\alpha(x)}{\longrightarrow} P'\,|\,R$ is induced by $P \stackrel{\alpha x}{\longrightarrow} P'$. Since $P \approx_o^b Q$, one has, for each y, the following cases:

- If Q' and Q'' exist such that $Q \Rightarrow \stackrel{\alpha x}{\longrightarrow} Q''$ and $Q''\{y/x\} \Rightarrow Q' \approx_o^b P'\{y/x\}$ then $(x)(Q\,|\,R) \Rightarrow \stackrel{\alpha(x)}{\longrightarrow} Q''\,|\,R$ and $Q''\{y/x\}\,|\,R\{y/x\} \Rightarrow (Q'\,|\,R\{y/x\}) \mathcal{R} (P'\{y/x\}\,|\,R\{y/x\})$.
- If Q' and Q'' exist such that $Q \Rightarrow \stackrel{\alpha(z)}{\longrightarrow} Q''$ and $Q''\{y/z\} \stackrel{y/x}{\Rightarrow} Q' \approx_o^b P'\{y/x\}$ then $(x)(Q\,|\,R) \Rightarrow \stackrel{\alpha(z)}{\longrightarrow} (x)(Q''\,|\,R)$ and $(x)(Q''\{y/z\}\,|\,R) \Rightarrow (Q'\,|\,R\{y/x\}) \mathcal{R} (P'\{y/x\}\,|\,R\{y/x\})$.
- If Q' exists such that $Q \stackrel{\alpha y}{\Rightarrow} \stackrel{y/x}{\Rightarrow} Q' \approx_o^b P'\{y/x\}$ then
$$(x)(Q\,|\,R) \stackrel{\alpha y}{\Rightarrow} (Q'\,|\,R\{y/x\}) \mathcal{R} (P'\{y/x\}\,|\,R\{y/x\}).$$
- If Q' exists such that $Q \stackrel{y/x}{\Rightarrow} \stackrel{\alpha y}{\Rightarrow} Q' \approx_o^b P'\{y/x\}$ then
$$(x)(Q\,|\,R) \stackrel{\alpha y}{\Rightarrow} (Q'\,|\,R\{y/x\}) \mathcal{R} (P'\{y/x\}\,|\,R\{y/x\}).$$
- If Q' and Q'' exist such that $Q \stackrel{y/x}{\Rightarrow} \stackrel{\alpha(z)}{\longrightarrow} Q''$ and $Q''\{y/z\} \Rightarrow Q' \mathcal{R} P'\{y/x\}$ then $(x)(Q\,|\,R) \stackrel{\tau}{\Rightarrow} \stackrel{\alpha(z)}{\longrightarrow} Q''\,|\,R\{y/x\}$ and $Q''\{y/z\}\,|\,R\{y/x\} \Rightarrow (Q'\,|\,R\{y/x\}) \mathcal{R} (P'\{y/x\}\,|\,R\{y/x\})$.

The proofs of other cases are similar.

The fact that $\approx_o^b$ is closed under match and prefix operations follow immediately from the closure under substitution. □

Now we need to show that $\approx_o^b$ and $\approx_b$ coincide. First we state and prove a technical lemma.

Lemma 12. *If $P \approx_b Q$ and $P \stackrel{y/x}{\longrightarrow} P'$, then there exists Q' such that $Q \stackrel{y/x}{\Rightarrow} Q' \approx_b P'$.*

Proof. Suppose $P \approx_b Q$ and $P \xrightarrow{y/x} P'$. Let c be a fresh name. Then $(x)(P \,|\, ([x=y]cc \,|\, \bar{c}c)) \xrightarrow{\tau}\xrightarrow{\tau} P' \,|\, (\mathbf{0} \,|\, \mathbf{0})$. It follows from $P \approx_b Q$ that $(x)(Q \,|\, ([x=y]cc \,|\, \bar{c}c)) \Rightarrow Q' \,|\, (\mathbf{0} \,|\, \mathbf{0}) \approx_b P' \,|\, (\mathbf{0} \,|\, \mathbf{0})$ for some Q'. This is possible only if $Q \stackrel{y/x}{\Rightarrow} Q' \approx_b P'$. □

Theorem 13. *$\approx_o^b$ and $\approx_b$ coincide.*

Proof. The inclusion $\approx_o^b \subseteq \approx_b$ holds by Lemma 11 and the fact that $\approx_o^b$ is barbed. Now we show that $\approx_b \subseteq \approx_o^b$. By definition $\approx_b$ is symmetric and closed under substitution. Suppose $P \approx_b Q$ and $P \xrightarrow{\lambda} P'$.

- If λ is a tau then it is matched up by $Q \Rightarrow Q' \approx_b P'$ by definition.
- If λ is an update action y/x then it is matched up by $Q \stackrel{\widehat{y/x}}{\Rightarrow} Q' \approx_b P'$ according to Lemma 12.
- If λ is a free action αx then $P \,|\, (\bar{\alpha} y + \langle a|b\rangle) \xrightarrow{y/x} P'\{y/x\} \,|\, \mathbf{0}$ for fresh a and b. It follows from $P \approx_b Q$ and Lemma 12 that

$$Q \,|\, (\bar{\alpha} y + \langle a|b\rangle) \stackrel{y/x}{\Rightarrow} Q' \,|\, \mathbf{0} \approx_b P'\{y/x\} \,|\, \mathbf{0}$$

  for some Q'. There are the following cases:
  - ∘ Q'' exists such that $Q \Rightarrow\xrightarrow{\alpha x} Q''$ and $Q''\{y/x\} \Rightarrow Q'$.
  - ∘ Q'' exists such that $Q \Rightarrow\xrightarrow{\alpha(z)} Q''$ and $Q''\{y/z\} \stackrel{y/x}{\Rightarrow} Q'$.
  - ∘ $Q \stackrel{\alpha y}{\Rightarrow}\stackrel{y/x}{\Rightarrow} Q'$.
  - ∘ $Q \stackrel{y/x}{\Rightarrow}\stackrel{\alpha y}{\Rightarrow} Q'$.
  - ∘ Q'' exists such that $Q \stackrel{y/x}{\Rightarrow}\xrightarrow{\alpha(z)} Q''$ and $Q''\{y/z\} \Rightarrow Q'$.

  Therefore the evolution from Q to Q' must take one of the five forms laid down in the definition of barbed open bisimulation. When y is x only the first two cases are possible. They can be restated as follows:
  - ∘ $Q \stackrel{\alpha x}{\Rightarrow} Q'$.
  - ∘ Q'' exists such that $Q \Rightarrow\xrightarrow{\alpha(z)} Q''$ and $Q''\{x/z\} \Rightarrow Q'$.
- If λ is a bound action $\alpha(x)$ then $P \,|\, (\bar{\alpha} y + \langle a|b\rangle) \xrightarrow{\tau} P'\{y/x\} \,|\, \mathbf{0}$ for fresh a and b. It follows from $P \approx_b Q$ that

$$Q \,|\, (\bar{\alpha} y + \langle a|b\rangle) \Rightarrow Q' \,|\, \mathbf{0} \approx_b P'\{y/x\} \,|\, \mathbf{0}$$

  for some Q'. So either $Q \stackrel{\alpha y}{\Rightarrow} Q'$ or $Q \Rightarrow\xrightarrow{\alpha(x)} Q''$ and $Q''\{y/x\} \Rightarrow Q'$ for some Q''. If y does not appear in Q then the only possibility is that $Q \stackrel{\alpha(x)}{\Rightarrow} Q'$.

Therefore $\approx_b$ is a barbed open bisimulation. We conclude that $\approx_o^b$ and $\approx_b$ coincide. □

The congruence $\simeq_o^b$ is defined from $\approx_o^b$ in the manner of Definition 6.

5. Ground bisimilarity

The most straightforward bisimulation equivalence for mobile calculi is the open bisimilarity. As we have seen in the introduction the standard definition of open bisimilarity gives rise to a bad relation. In order to modify the definition to obtain a sensible equivalence relation, we have provided in Section 3 two alternative definitions. Although these two definitions give rise to two reasonable equivalence relations, they appear to be *ad hoc*. In some sense the early and the late congruence relations are too strong. In Section 4 we have seen many barbed equivalent pairs of processes. Most of these pairs are identified neither by the early congruence nor by the late congruence. We need a canonical version, so to speak, of the open congruence for $\chi^{\neq}$. One way to achieve this is to take the largest subrelation of the hyperequivalence that is closed under the parallel composition operator. It is then easy to get a congruence in the manner of Definition 6. However we can arrive at the same relation in a cleaner way.

Definition 14. Let $\mathscr{R}$ be a symmetric binary relation on $\mathscr{C}$. It is called a bisimulation if whenever $P\mathscr{R}Q$ and $P \xrightarrow{\lambda} P'$ then some Q' exists such that $Q \stackrel{\hat{\lambda}}{\Rightarrow} Q' \mathscr{R} P'$.

Using this auxiliary relation, we can get the desired bisimilarity.

Definition 15. The ground bisimilarity $\approx_g$ is the largest bisimulation closed under context.

As in the barbed case, we will now give an equivalent characterization of $\approx_g$ in the style of open semantics.

Definition 16. Let $\mathscr{R}$ be a binary symmetric relation on $\mathscr{C}$ closed under substitution. The relation $\mathscr{R}$ is a ground open bisimulation if the following properties hold for P and Q whenever $P\mathscr{R}Q$:

(i) If λ is an update or a tau and $P \xrightarrow{\lambda} P'$ then Q' exists such that $Q \stackrel{\hat{\lambda}}{\Rightarrow} Q' \mathscr{R} P'$.

(ii) If $P \xrightarrow{\alpha x} P'$ then Q' exists such that $Q \stackrel{\alpha x}{\Rightarrow} Q' \mathscr{R} P'$ and, for each y different from x, one of the following properties holds:
- Q' and Q'' exist such that $Q \Rightarrow \xrightarrow{\alpha x} Q''$ and $Q''\{y/x\} \Rightarrow Q' \mathscr{R} P'\{y/x\}$;
- Q' and Q'' exist such that $Q \Rightarrow \xrightarrow{\alpha(z)} Q''$ and $Q''\{y/z\} \stackrel{y/x}{\Rightarrow} Q' \mathscr{R} P'\{y/x\}$;
- Q' exists such that $Q \stackrel{\alpha y}{\Rightarrow} \stackrel{y/x}{\Rightarrow} Q' \mathscr{R} P'\{y/x\}$;
- Q' exists such that $Q \stackrel{y/x}{\Rightarrow} \stackrel{\alpha y}{\Rightarrow} Q' \mathscr{R} P'\{y/x\}$;
- Q' and Q'' exist such that $Q \stackrel{y/x}{\Rightarrow} \xrightarrow{\alpha(z)} Q''$ and $Q''\{y/z\} \Rightarrow Q' \mathscr{R} P'\{y/x\}$.

(iii) If $P \xrightarrow{\alpha(x)} P'$ then Q' exists such that $Q \stackrel{\alpha(x)}{\Rightarrow} Q' \approx_o^b P'$ and, for each y distinct from x, one of the following properties holds:
- Q' and Q'' exist such that $Q \Rightarrow \xrightarrow{\alpha(x)} Q''$ and $Q''\{y/x\} \Rightarrow Q' \mathscr{R} P'\{y/x\}$;
- Q' exists such that $Q \stackrel{\alpha y}{\Rightarrow} Q' \mathscr{R} P'\{y/x\}$.

The ground open bisimilarity $\approx_o^g$ is the largest ground open bisimulation.

According to the definition, the ground open bisimilarity is very similar to that of the barbed open bisimilarity. There is only a subtle difference in the treatment of free actions. In the barbed case the action $P \xrightarrow{\alpha x} P'$ can be matched by either $Q \stackrel{\alpha x}{\Rightarrow} Q'$ for some Q' or by $Q \Rightarrow \xrightarrow{\alpha(z)} Q''$ and $Q''\{x/z\} \Rightarrow Q'$ for some Q'' and Q'. In the ground case however the action $P \xrightarrow{\alpha x} P'$ can always be matched by $Q \stackrel{\alpha x}{\Rightarrow} Q'$ for some Q' since this is declared in Definition 15.

The proof of Lemma 11 is sufficient to establish the following lemma.

Lemma 17. $\approx_o^g$ *is closed under context.*

Using Lemma 17 the proof of Theorem 13 can be reiterated to prove the following theorem.

Theorem 18. $\approx_o^g$ *and* $\approx_g$ *coincide.*

By definition the ground open bisimilarity is contained in the barbed open bisimilarity. The inclusion is strict because equality (10) is not valid for $\approx_o^g$. It follows from Theorem 13 and Theorem 18 that the inclusion $\approx_g \subseteq \approx_b$ is also strict. It is also easy to see that the early open bisimilarity is strictly contained in the ground open bisimilarity.

The ground congruence $\simeq_o^g$ is defined in the fashion of Definition 6.

6. Basic laws

An interesting question about a congruence relation is if there is a finite set of sound equation schemes and inference rules such that all congruent pairs can be derived from these equation schemes and rules. Sometimes one has to be less ambitious and be contented with a recursively enumerable set of those. Such a set is called a complete equational system for the congruence. The procedure of finding such a complete system is called axiomatization. A complete system for an observational equivalence on the finite processes of a process calculus represents a milestone in our understanding of the equivalence.

In [6] completeness theorems are proved for L-bisimilarities on χ-processes without the mismatch operator. The proofs of these completeness results use essentially the inductive definitions of L-bisimilarities. In the presence of the mismatch operator, the method used in [6] should be modified. The modification is done by incorporating ideas from [26]. In this section we give a complete axiomatic system for each of the four congruence relations using the modified approach.

The proofs of completeness theorems use the fact that all processes can be transferred to those in normal forms. The definition of normal form for the $\chi^{\neq}$-calculus is different from that of normal form for the χ-calculus. The former definition makes use of the fact that the mismatch operator makes it possible for us to deal exclusively with complete conditions in the following sense.

Definition 19. Let V be a finite set of names. We say that ψ is complete on V if $n(\psi) = V$ and for each pair x, y of names in V it holds that either $\psi \Rightarrow x = y$ or $\psi \Rightarrow x \neq y$.

Suppose ψ is complete on V and $n(\phi) \subseteq V$. Then it should be clear that either $\psi\phi \Leftrightarrow \psi$ or $\psi\phi \Leftrightarrow \bot$. In sequel this fact will be used implicitly.

Lemma 20. *If ϕ and ψ are complete on V and both agree with σ then $\phi \Leftrightarrow \psi$.*

We now begin to describe four axiomatic systems that are complete for the respective congruence relations. Let AS denote the system consisting of the rules and laws in Fig. 1 plus the following expansion law:

$$P \mid Q = \sum_i \phi_i(\tilde{x})\pi_i.(P_i \mid Q) + \sum_{\pi_j = \overline{b_j} y_j}^{\pi_i = a_i x_i} \phi_i \psi_j(\tilde{x})(\tilde{y})[a_i = b_j]\langle x_i | y_j \rangle.(P_i \mid Q_j)$$

$$+ \sum_j \psi_j(\tilde{y})\pi_j.(P \mid Q_j) + \sum_{\pi_j = b_j y_j}^{\pi_i = \overline{a_i} x_i} \phi_i \psi_j(\tilde{x})(\tilde{y})[a_i = b_j]\langle x_i | y_j \rangle.(P_i \mid Q_j)$$

where P is $\sum_i \phi_i(\tilde{x})\pi_i.P_i$, Q is $\sum_j \psi_j(\tilde{y})\pi_j.Q_j$, and π_i and π_j range over $\{\alpha x \mid \alpha \in \mathcal{N} \cup \bar{\mathcal{N}},\ x \in \mathcal{N}\}$. In the expansion law, the summand

$$\sum_{\pi_j = \overline{b_j} y_j}^{\pi_i = a_i x_i} \phi_i \psi_j(\tilde{x})(\tilde{y})[a_i = b_j]\langle x_i | y_j \rangle.(P_i \mid Q_j)$$

contains $\phi_i \psi_j(\tilde{x})(\tilde{y})[a_i = b_j]\langle x_i | y_j \rangle.(P_i \mid Q_j)$ as a summand whenever $\pi_i = a_i x_i$ and $\pi_j = \overline{b_j} y_j$.

The system AS is essentially the complete system of Parrow and Victor [28] for the strong hyperequivalence. AS is complete for the strong open bisimilarity of $\chi^{\neq}$-calculus. The strong open bisimilarity is equal to the strong hyperequivalence. So the completeness follows from Parrow and Victor's result.

We write $AS \vdash P = Q$ to indicate that the equality $P = Q$ can be inferred from AS. When $R1, \ldots, Rn$ are the major axioms used to derive $P = Q$, we write $P \overset{R1,\ldots,Rn}{=} Q$. Some important derived laws of AS are given in Fig. 2.

It can be shown that AS is complete for the strong open bisimilarity on $\chi^{\neq}$-processes. This fact will not be proved here. Our attention will be confined to the completeness problems of the four weak open congruence relations.

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 21. A process P is in normal form on $V \supseteq fn(P)$ if P is of the form

$$\sum_{i \in I_1} \phi_i \alpha_i x_i.P_i + \sum_{i \in I_2} \phi_i \alpha_i(x).P_i + \sum_{i \in I_3} \phi_i \langle z_i \mid y_i \rangle.P_i$$

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	$P_0 \mid P_1 = Q_0 \mid Q_1$	if $P_0 = Q_0$ and $P_1 = Q_1$
L1	$(x)\mathbf{0} = \mathbf{0}$	
L2	$(x)\alpha y.P = \mathbf{0}$	$x \in \{\alpha, \bar{\alpha}\}$
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	$(x)[x = y]P = \mathbf{0}$	$x \neq y$
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	$(x)\langle y\|x\rangle.P = \tau.P\{y/x\}$	$y \neq x$
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	$\langle y\|x\rangle.P = \langle x\|y\rangle.P$	
U2	$\langle y\|x\rangle.P = \langle y\|x\rangle.[x = y]P$	
U3	$\langle x\|x\rangle.P = \tau.P$	

Fig. 1. The axiomatic system AS.

such that the following conditions are satisfied:

1. ϕ_i is complete on V for each $i \in I_1 \cup I_2 \cup I_3$;
2. P_i is in normal form on V for $i \in I_1 \cup I_3$;
3. P_i is in normal form on $V \cup \{x\}$ for $i \in I_2$ and x does not appear in P.

Here I_1, I_2 and I_3 are pairwise disjoint finite indexing sets.

Suppose P is in normal form and σ is a substitution. The following observations about $P\sigma$ are useful:

- If $P\sigma \xrightarrow{\alpha x} Q$ then there is a summand $\phi\alpha' x'.P'$ of P such that $P'\sigma \equiv Q$, $\alpha'\sigma = \alpha$, $\sigma(x') = x$ and that σ validates ϕ.

LD1	$(x)\langle x\|x\rangle.P = \langle y\|y\rangle.(x)P$	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-laws and M4
UD1	$\langle y\|x\rangle.P = \langle y\|x\rangle.P\{y/x\}$	U2 and M2

Fig. 2. Some laws derivable from AS.

- If $P\sigma \xrightarrow{\alpha(x)} Q$ then there is a summand $\phi\alpha'(x).P'$ of P such that $P'\sigma \equiv Q$, $\alpha'\sigma = \alpha$ and that σ validates ϕ.
- If $P\sigma \xrightarrow{y/x} Q$ then there is a summand $\phi\langle y'|x'\rangle.P'$ of P such that $P'\sigma\{y/x\} \equiv Q$, $\sigma(x') = x$, $\sigma(y') = y$ and that σ validates ϕ.
- If $P\sigma \xrightarrow{\tau} Q$ then there is a summand $\phi\langle y'|x'\rangle.P'$ of P such that $P'\sigma \equiv Q$, $\sigma(x') = \sigma(y')$ and that σ validates ϕ.

The depth of a process measures the maximal length of nested extended prefixes in the expansion of the process. The structural definition goes as follows:

$$\begin{aligned}
d(\mathbf{0}) &\stackrel{\text{def}}{=} 0 \\
d(\alpha x.P) &\stackrel{\text{def}}{=} 1 + d(P) \\
d(P \mid Q) &\stackrel{\text{def}}{=} d(P) + d(Q) \\
d((x)P) &\stackrel{\text{def}}{=} d(P) \\
d([x = y]P) &\stackrel{\text{def}}{=} d(P) \\
d([x \neq y]P) &\stackrel{\text{def}}{=} d(P) \\
d(P + Q) &\stackrel{\text{def}}{=} \max\{d(P), d(Q)\}.
\end{aligned}$$

Lemma 22. *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 $d(Q) \leqslant d(P)$ and $AS \vdash Q = P$.*

Proof. The proof is carried out by structural induction. If for example the outer most combinator of a process is a restriction operator then there are three cases: (i) The process is equal to $\mathbf{0}$; (ii) It is equal to a process of the form $\alpha(x).P$ such that $x \notin \{\alpha, \bar{\alpha}\}$; (iii) Otherwise the restriction operator can be pushed inside. Use $M4$ if necessary to expand the outer most condition operators so that they are complete on V. It is obvious that this conversion procedure does not increase the depth of the process. □

T1	$P + \tau.P = \tau.(P + \psi\tau.P)$	
T2	$\alpha x.\tau.P = \alpha x.P$	
T3	$\alpha x.(P + \delta\tau.Q) = \alpha x.(P + \delta\tau.Q) + [x \notin n(\delta)]\delta\alpha x.Q$	
T4	$TT4 = TT4 + [x \notin n(\delta)]\delta\alpha x.Q$	
T5	$TT5 = TT5 + [x \notin Y_3][x \notin n(\delta)]\delta\alpha x.Q\{x/z\}$	$z \notin n(\delta)$
T6	$TT6 = TT6 + [x \notin Y_3][x \notin n(\delta)]\delta\alpha x.Q\{x/z\}$	$z \notin n(\delta)$

Fig. 3. The tau laws.

7. Tau laws

The complete systems for the weak relations are obtained from AS by adding some tau laws. The tau laws used in this paper are given in Fig. 3. Some explanations of these tau laws are as follows:

- T1 is different from the other tau laws in that it is purely for tau prefix. If we let ψ be false, say $[x \neq x]$, then it becomes

$$P + \tau.P = \tau.P$$

which is Milner's second tau laws. It follows immediately that

$$\tau.P = \tau.(P + \psi\tau.P)$$

which was proposed by the first author to axiomatize weak open congruences. The necessity of this law has been established in [9], see also [13]. Observe that T1 has an equivalent formulation as follows:

$$\tau.P = P + \tau.(P + \psi\tau.P).$$

By induction the law also implies

$$\tau.P = \tau.\left(P + \sum_{i=1}^{n} \psi_i \tau.P\right)$$

See [13] for a proof.

- T2 is Milner's first tau law.
- T3 is a nontrivial extension of Milner's third tau law. The condition $[x \notin n(\delta)]$ is important for otherwise the prefix αx in the summand $[x \notin n(\delta)]\delta\alpha x.Q$ could incur an action that invalidates δ, which makes it impossible for $\alpha x.(P + \delta\tau.Q)$ to simulate the action.
- In T4 the summand $TT4$ is $\sum_{y \in Y} \alpha x.(P_y + \delta[x = y]\tau.Q) + \alpha x.(P + \delta[x \notin Y]\tau.Q)$. Here P_y and $P_{y'}$ could be different for distinct y and y' in Y. More explicitly T4 can be formulated as follows:

$$\sum_{i=1,\ldots,n} \alpha x.(P_i + \delta[x = y_i]\tau.Q) + \alpha x.(P + \delta[x \notin Y]\tau.Q)$$

$$= \sum_{i=1,\ldots,n} \alpha x.(P_i + \delta[x = y_i]\tau.Q) + \alpha x.(P + \delta[x \notin Y]\tau.Q) + [x \notin n(\delta)]\delta\alpha x.Q,$$

where Y is $\{y_1, \ldots, y_n\}$. When Y is a singleton set the law becomes

$$\alpha x.(P_1 + \delta[x = y]\tau.Q) + \alpha x.(P_2 + \delta[x \neq y]\tau.Q)$$
$$= \alpha x.(P_1 + \delta[x = y]\tau.Q) + \alpha x.(P_2 + \delta[x \neq y]\tau.Q) + [x \notin n(\delta)]\delta\alpha x.Q.$$

It does not seem possible to derive the general form of T4 from this simple equality. It should be pointed out that the condition $x \notin n(\delta)$ has been internalized in the law. It cannot be placed as a side condition, like in T5 and T6, because x appears free in the law. A substitution may well invalidates the condition.

- The laws T5 and T6 are equational formalizations of the last two examples given in Section 4. In T5, $TT5$ abbreviates

$$\begin{aligned}\alpha x.(P + \delta[x \notin Y]\tau.Q\{x/z\}) &+ \sum_{y\in Y_1} \alpha x.(P_y + \delta[x = y]\tau.Q\{x/z\})\\ &+ \sum_{y\in Y_2} \alpha(z).(P_y + [x \notin n(\delta)]\delta[z = y]\langle z|x\rangle.Q)\\ &+ \sum_{y\in Y_3} [x \neq y]\alpha y.(P_y + [x \notin n(\delta)]\delta\langle y|x\rangle.Q\{x/z\})\\ &+ \sum_{y\in Y_4} \langle y|x\rangle.(P_y + \delta\alpha y.(P'_y + \delta\tau.Q\{x/z\}))\\ &+ \sum_{y\in Y_5} \langle y|x\rangle.(P_y + \delta\alpha(z).(P'_y + \delta[z = y]\tau.Q)),\end{aligned}$$

where Y is $Y_1 \cup Y_2 \cup Y_3 \cup Y_4 \cup Y_5$. In T5 the side condition $z \notin n(\delta)$ is safe because z appears as a bound name in the law. No substitution could invalidate the side condition because no substitution could identify a bound name to a free name. The same remark can be made to T6.

- In T6, $TT6$ stands for

$$\begin{aligned}&\alpha(z).(P + [x \notin n(\delta)]\delta[z \notin Y]\langle z|x\rangle.Q)\\ &+ \sum_{y\in Y_1} \alpha x.(P_y + \delta[x = y]\tau.Q\{x/z\})\\ &+ \sum_{y\in Y_2} \alpha(z).(P_y + [x \notin n(\delta)]\delta[z = y]\langle z|x\rangle.Q)\\ &+ \sum_{y\in Y_3} [x \neq y]\alpha y.(P_y + [x \notin n(\delta)]\delta\langle y|x\rangle.Q\{x/z\})\\ &+ \sum_{y\in Y_4} \langle y|x\rangle.(P_y + \delta\alpha y.(P'_y + \delta\tau.Q\{x/z\}))\\ &+ \sum_{y\in Y_5} \langle y|x\rangle.(P_y + \delta\alpha(z).(P'_y + \delta[z = y]\tau.Q)),\end{aligned}$$

T1a	$\tau.P = P + \tau.P$	
T1b	$\tau.P = \tau.(P + \sum_{i=1}^{n} \psi_i \tau.P)$	
T2a	$\lambda.\tau.P = \lambda.P$	
T3a	$\alpha(x).(P + \delta\tau.Q) = \alpha(x).(P + \delta\tau.Q) + \delta\alpha(x).Q$	$x \notin n(\delta)$
T3b	$\langle y\|x\rangle.(P + \delta\tau.Q) = \langle y\|x\rangle.(P + \delta\tau.Q) + [x \notin n(\delta)]\delta\langle y\|x\rangle.Q$	
T4a	$TT4a = TT4a + \delta\alpha(x).Q$	$x \notin n(\delta)$
T5a	$TT5a = TT5a + \delta\alpha(x).Q$	$x \notin n(\delta)$

Fig. 4. Some derived tau laws.

where Y is $Y_1 \cup Y_2 \cup Y_3 \cup Y_4 \cup Y_5$. Notice that

$$\alpha(z).(P + \langle z|x\rangle.Q) = \alpha(z).(P + \langle z|x\rangle.Q) + \alpha x.Q\{x/z\} \tag{24}$$

is a special case of T6, which does not hold for the ground congruence since the right hand can perform a free action but the left hand cannot. In χ-calculus (24) distinguishes the barbed congruence from the open congruence [6].

The tau laws T1, T3, T4, T5 and T6 are new. It is worth remarking that T3, T4, T5 and T6 are of the same type. They all deal with tau prefixes under the prefix αx. Notice that all of them trivialize to

$$\alpha x.(P + \tau.Q) = \alpha x.(P + \tau.Q) + \alpha x.Q$$

which is Milner's third tau law when we remove the mismatch operators. Notice also that T2, T3, T4, T5 and T6 are only formulated for free prefixes. The bound prefix and update prefix versions of these laws are derivable.

Lemma 23. (i) $AS \cup \{T4\} \vdash T3$. (ii) $AS \cup \{T5\} \vdash T4$.

Proof. T3 is the special case of T4 when Y is empty. T4 is the special case of T4 when Y_2, Y_3, Y_4 and Y_5 are all empty. □

Some derived tau laws are given in Fig. 4. In T4a of Fig. 4 the shorthand notation $TT4a$ is for

$$\sum_{y \in Y} \alpha(x).(P_y + \delta[x = y]\tau.Q) + \alpha(x).(P + \delta[x \notin Y]\tau.Q)$$

and in T5a the abbreviation $TT5a$ stands for

$$\sum_{y \in Y_1} \alpha y.(P_y + \delta\tau.Q\{y/x\}) + \sum_{y \in Y_2} \alpha(x).(P_y + \delta[x = y]\tau.Q)$$
$$+ \alpha(x).(P + \delta[x \notin Y_1 \cup Y_2]\tau.Q)$$

Lemma 24. *The following properties hold*:
(i) $AS \cup \{T1\} \vdash T1a$; $AS \cup \{T1\} \vdash T1b$.
(ii) $AS \cup \{T2\} \vdash T2a$
(iii) $AS \cup \{T3\} \vdash T3a$; $AS \cup \{T3\} \vdash T3b$.
(iv) $AS \cup \{T4\} \vdash T4a$.
(v) $AS \cup \{T5\} \vdash T5a$.

Proof. (iii) T3a is derived by using the L-laws. For T3b, observe that

$$\begin{aligned} \langle y|x\rangle.(P + \delta\tau.Q) &= (a)(\bar{a}y \mid ax.(P + \delta\tau.Q)) \\ &\overset{T3}{=} (a)(\bar{a}y \mid (\alpha x.(P + \delta\tau.Q) + [x \notin n(\delta)]\delta\alpha x.Q)) \\ &= \langle y|x\rangle.(P + \delta\tau.Q) + [x \notin n(\delta)]\delta\langle y|x\rangle.Q, \end{aligned}$$

where the third equality holds by the expansion law.
(iv) By T4 and C2 we get that

$$\begin{aligned} AS \cup \{T4\} \vdash TT4a &= (x)(TT4 + [x \notin n(\delta)]\delta ax.Q) \\ &= TT4a + (x)[x \notin n(\delta)]\delta ax.Q \\ &\overset{LD3}{=} TT4a + (x)\delta ax.Q \\ &\overset{LD2}{=} TT4a + \delta a(x).Q \end{aligned}$$

(v) Let Y_2, Y_4, Y_5 in $TT5$ be empty. Then by $T5$ one gets that

$$\begin{aligned} AS \cup \{T5\} \vdash (x)TT5 &= (x)(TT5 + [x \notin Y_3][x \notin n(\delta)]\delta\alpha x.Q) \\ &= (x)TT5 + (x)[x \notin Y_3][x \notin n(\delta)]\delta\alpha x.Q \\ &\overset{LD3}{=} (x)TT5 + (x)\delta\alpha x.Q \\ &\overset{LD2}{=} (x)TT5 + \delta\alpha(x).Q. \end{aligned}$$

It is routine to show that $AS \vdash (x)TT5 = TT5a$ using L9. □

There are two major uses of tau laws. One is to equate processes with related operational behaviours. The other is to identify processes indistinguishable by an algebraic semantics. The former is to do with saturation properties whereas the latter with promotion properties. As a matter of fact one can classify the tau laws according to whether they are used to establish saturation and/or promotion properties. For instance T1 is typically related to promotion properties whereas T3, T4, T5, T6 with saturation properties. The classification is not so clear-cut because a law may be crucial in the proofs of both properties.

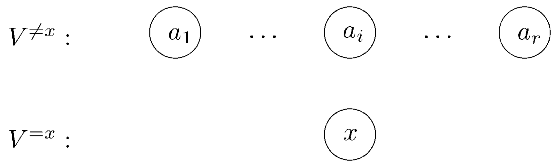

Fig. 5. Classification of V by ϕ.

8. Completeness

In the proofs of this section, we need a careful analysis of the requirement 'ϕ is complete on V'. Here are some observations about the requirement and notations used to analyse the requirement:

- Since ϕ is complete on V, it groups the elements of V into several disjoint classes. Assume that these classes are $[x], [a_1], \ldots, [a_r]$.
  - Let $\phi^{=}$ be the sequence of match operators induced by the equivalence classes $[a_1], \ldots, [a_r]$.
  - Let $\phi^{=x}$ be the sequence of match operators induced by the equivalence class $[x]$.
  - Let $\phi^{\neq}$ be the sequence of mismatch combinators constructed as follows:
    $a \neq b$ is in $\phi^{\neq}$ if and only if $a \in [a_p]$ and $b \in [a_q]$ for some $1 \leqslant p, q \leqslant r$ such that $p \neq q$.
  - Let $\phi^{\neq x}$ be the sequence of mismatch combinators constructed as follows:
    $a \neq x$ is in $\phi^{\neq x}$ if and only if $a \in [a_1] \cup \cdots \cup [a_r]$.

  Clearly $\phi \Leftrightarrow \phi^{=}\phi^{=x}\phi^{\neq}\phi^{\neq x}$. The set V can be divided into two subsets:

$$V^{\neq x} \stackrel{\text{def}}{=} \{y \mid y \in V,\ \phi \Rightarrow y \neq x\} = [a_1] \cup \cdots \cup [a_r],$$

$$V^{=x} \stackrel{\text{def}}{=} \{y \mid y \in V,\ \phi \Rightarrow y = x\} = [x].$$

  Fig. 5 helps one understand the induced conditions.
- If $y \in V^{\neq x}$ then we define $\phi^{\neq x}_{\setminus [y]}$ as follows:
  $a \neq x$ is in $\phi^{\neq x}_{\setminus [y]}$ if and only if $a \in V^{\neq x} \setminus [y]$.

  It is important to observe that

$$\phi^{=}\phi^{=x}\phi^{\neq}\phi^{\neq x}_{\setminus [y]}[x = y]$$

  is complete on V and induces $\sigma\{y/x\}$. Also notice that

$$\phi^{\neq} \Rightarrow \phi^{\neq x}_{\setminus [y]}\{y/x\}, \tag{25}$$

$$\phi^{\neq x} \Leftrightarrow [x \notin n(\phi^{\neq})] \Leftrightarrow [x \notin V^{\neq x}]. \tag{26}$$

system	rules and laws
AS_o^l	$AS \cup \{T1, T2, T3\}$
AS_o^e	$AS \cup \{T1, T2, T4\}$
AS_o^g	$AS \cup \{T1, T2, T5\}$
AS_o^b	$AS \cup \{T1, T2, T5, T6\}$

Fig. 6. Four axiomatic systems in $\chi^{\neq}$-calculus.

In the proofs of the saturation and promotion lemmas, we need to use the following equality schemes:

$$\phi(\ldots \phi^{=} \phi^{=x} \phi^{\neq} \phi^{\neq x}_{\backslash [y]} [x = y] \ldots) = \phi(\ldots \phi^{\neq} [x = y] \ldots) \tag{27}$$

$$\phi(\ldots \phi \ldots) = \phi(\ldots [x \notin n(\phi^{\neq})] \phi^{\neq} \ldots). \tag{28}$$

These can be proved as follows:

$$\begin{aligned} \phi(\ldots \phi^{=} \phi^{=x} \phi^{\neq} \phi^{\neq x}_{\backslash [y]} [x = y] \ldots) &\stackrel{M2}{=} \phi(\ldots \phi^{\neq} \phi^{\neq x}_{\backslash [y]} [x = y] \ldots) \\ &\stackrel{M1}{=} \phi(\ldots \phi^{\neq} (\phi^{\neq x}_{\backslash [y]} \{y/x\}) [x = y] \ldots) \\ &\stackrel{(25)}{=} \phi(\ldots \phi^{\neq} [x = y] \ldots) \end{aligned}$$

and

$$\begin{aligned} \phi(\ldots \phi \ldots) &= \phi(\ldots \phi^{=} \phi^{=x} \phi^{\neq} \phi^{\neq x} \ldots) \\ &\stackrel{M2}{=} \phi(\ldots \phi^{\neq} \phi^{\neq x} \ldots) \\ &\stackrel{(26)}{=} \phi(\ldots [x \notin n(\phi^{\neq})] \phi^{\neq} \ldots). \end{aligned}$$

Now we have enough machinery to do proofs.

We have provided enough laws to construct the required axiomatic systems. Fig. 6 defines four such systems. For instance AS_o^b is defined to be the system $AS \cup \{T1, T2, T5, T6\}$. These systems will be shown to be complete respectively for the four congruence relations.

We will follow by now the standard strategy to prove the completeness. First we establish two lemmas stating the saturation properties. The first is a general property held by all the four systems. The second is for individual systems. Based upon these lemmas, a promotion lemma is proved that lifts a pair of observational equivalent processes to a pair of proof theoretical equal processes. The promotion lemma plays the role of Hennessy Lemma which does not hold for mobile processes. A proof of the promotion lemma *is* a proof of the completeness theorem if all the references to induction hypothesis in the proof is replaced by references to the promotion lemma!

Lemma 25 (saturation 1). *Suppose Q is in normal form on V, ϕ is complete on V, and σ is a substitution induced by ϕ. Then the following properties hold*:

(i) *If* $Q\sigma \stackrel{\tau}{\Rightarrow} Q'$ *then* $AS \cup \{T1, T2, T3\} \vdash Q = Q + \phi\tau.Q'$.
(ii) *If* $Q\sigma \Rightarrow \stackrel{\alpha x}{\rightarrow} Q'$ *then* $AS \cup \{T1, T2, T3\} \vdash Q = Q + \phi\alpha x.Q'$.
(iii) *If* $Q\sigma \Rightarrow \stackrel{\alpha(x)}{\rightarrow} Q'$ *then* $AS \cup \{T1, T2, T3\} \vdash Q = Q + \phi\alpha(x).Q'$.
(iv) *If* $Q\sigma \stackrel{y/x}{\Rightarrow} Q'$ *then* $AS \cup \{T1, T2, T3\} \vdash Q = Q + \phi\langle y|x\rangle.Q'$.

Proof. The proof of (i)–(iii) are routine using induction. Here we only give a proof of (iv). If $Q\sigma \stackrel{y/x}{\rightarrow} Q'$ then it is routine to show that $AS \cup \{T1, T2, T3\} \vdash Q = Q + \phi\langle y|x\rangle.Q'$. By the observations made right after Definition 21, we may assume without loss of generality that $Q\sigma \stackrel{y/x}{\Rightarrow} Q'$ is of the following form:

$$Q\sigma \stackrel{\tau}{\Rightarrow} Q_1\sigma \stackrel{y/x}{\rightarrow} Q_2\sigma\{y/x\} \stackrel{\tau}{\Rightarrow} Q'.$$

Since $x \neq y$ the equivalence induced by ϕ contains at least two distinct elements $[x]$ and $[y]$. Using the notations just defined, one has the following equality reasoning:

$$\begin{aligned}
Q &\stackrel{(i)}{=} Q + \phi\tau.Q_1\sigma \\
&\stackrel{MD3}{=} Q + \phi\tau.Q_1 \\
&\stackrel{IH}{=} Q + \phi\tau.(Q_1 + \phi\langle y|x\rangle.Q_2\sigma\{y/x\}) \\
&\stackrel{T1a}{=} Q + \phi\langle y|x\rangle.Q_2\sigma\{y/x\} \\
&\stackrel{MD3}{=} Q + \phi\langle y|x\rangle.Q_2\{y/x\} \\
&\stackrel{UD1}{=} Q + \phi\langle y|x\rangle.Q_2 \\
&\stackrel{(i)}{=} Q + \phi\langle y|x\rangle.(Q_2 + \phi^{=}\phi^{=x}\phi^{\neq}\phi^{\neq x}_{\backslash[y]}[x = y]\tau.Q') \\
&\stackrel{(27)}{=} Q + \phi\langle y|x\rangle.(Q_2 + \phi^{\neq}[x = y]\tau.Q') \\
&\stackrel{UD1}{=} Q + \phi\langle y|x\rangle.(Q_2 + \phi^{\neq}\tau.Q') \\
&\stackrel{T3b}{=} Q + \phi\langle y|x\rangle.(Q_2 + \phi^{\neq}\tau.Q') + \phi[x \notin n(\phi^{\neq})]\phi^{\neq}\langle y|x\rangle.Q' \\
&= Q + \phi[x \notin n(\phi^{\neq})]\phi^{\neq}\langle y|x\rangle.Q' \\
&\stackrel{(26)}{=} Q + \phi\phi^{\neq x}\phi^{\neq}\langle y|x\rangle.Q' \\
&= Q + \phi\langle y|x\rangle.Q'.
\end{aligned}$$

The last equality holds because $\phi \Rightarrow \phi^{\neq x}\phi^{\neq}$. □

Notice that unlike the situation in the χ-calculus, the second and the third clauses of the above lemma cannot be strengthened to the following:

(ii′) If $Q\sigma \stackrel{\alpha x}{\Rightarrow} Q'$ then $AS \cup \{T1, T2, T3\} \vdash Q = Q + \phi\alpha x.Q'$.

(iii′) If $Q\,\sigma \stackrel{\alpha(x)}{\Rightarrow} Q'$ then $AS \cup \{T1, T2, T3\} \vdash Q = Q + \phi\alpha(x).Q'$.

For instance $ax.[x \neq a]\tau.aa \stackrel{ax}{\Rightarrow} aa$ but not $AS \cup \{T1, T2, T3\} \vdash ax.[x \neq a]\tau.aa = ax.[x \neq a]\tau.aa + ax.aa$.

The next lemma describes some additional saturation properties for input actions.

Lemma 26 (saturation 2). *Suppose Q is a normal form on some $V = \{y_1, \dots, y_k\} \supseteq fn(Q)$, ψ is complete on V, and σ is a substitution induced by ψ. If*

$$Q\sigma \Rightarrow \stackrel{\alpha(x)}{\rightarrow} Q_1'\sigma,\ Q_1'\sigma\{y_1/x\} \Rightarrow Q_1,$$
$$Q\sigma \Rightarrow \stackrel{\alpha(x)}{\rightarrow} Q_2'\sigma,\ Q_2'\sigma\{y_2/x\} \Rightarrow Q_2,$$
$$\vdots$$
$$Q\sigma \Rightarrow \stackrel{\alpha(x)}{\rightarrow} Q_k'\sigma,\ Q_k'\sigma\{y_k/x\} \Rightarrow Q_k,$$
$$Q\sigma \Rightarrow \stackrel{\alpha(x)}{\rightarrow} Q_{k+1}'\sigma \Rightarrow Q_{k+1}$$

then the following properties hold:

1. $Q + \psi\sum_{j=1}^{k} \alpha(x).(\tau.Q_j' + \psi[x = y_j]\tau.Q_j) + \psi\alpha(x).(\tau.Q_{k+1}' + \psi[x \notin V]\tau.Q_{k+1})$ *is provably equal to Q in $AS \cup \{T1, T2, T3\}$.*
2. *If $Q_1' \equiv Q', \dots, Q_{k+1}' \equiv Q'$ then $Q + \psi\alpha(x).(\tau.Q' + \psi\sum_{j=1}^{k}[x = y_j]\tau.Q_j + \psi[x \notin V]\tau.Q_{k+1})$ is provably equal to Q in $AS \cup \{T1, T2, T3\}$.*

Proof. We only prove the first equality. For every $l \in \{1, \dots, k\}$ there are two cases for $Q_l'\sigma\{y_l/x\} \Rightarrow Q_l$:

- $Q_l'\sigma\{y_l/x\} \equiv Q_l$. Then

$$\begin{aligned}\tau.Q_l' &= \tau.Q_l' + \psi[x = y_l]\tau.Q_l' \\ &\stackrel{MD3}{=} \tau.Q_l' + \psi[x = y_l]\tau.Q_l'\sigma\{y_l/x\} \\ &= \tau.Q_l' + \psi[x = y_l]\tau.Q_l.\end{aligned}$$

- $Q_l'\sigma\{y_l/x\} \stackrel{\tau}{\Rightarrow} Q_l$. Clearly $\sigma\{y_l/x\}$ agrees with $\psi[y_l = x]$ and $\psi[y_l = x]$ is complete on $V \cup \{x\}$. Hence

$$\begin{aligned}\tau.Q_l' &= \tau.(Q_l' + \psi[x = y_l]\tau.Q_l) \\ &= \tau.(Q_l' + \psi[x = y_l]\tau.Q_l) + \psi[x = y_l]\tau.Q_l \\ &= \tau.Q_l' + \psi[x = y_l]\tau.Q_l.\end{aligned}$$

By similar argument we get that $\tau.Q_{k+1}' = \tau.Q_{k+1}' + \psi[x \notin V]\tau.Q_{k+1}$. We are done by using (iii) of Lemma 25. □

Now we come to the promotion lemma.

Lemma 27 (promotion). *In $\chi^{\neq}$-calculus the following properties hold*:
(i) *If* $P \approx_o^l Q$ *then* $AS_o^l \vdash \tau.P = \tau.Q$.
(ii) *If* $P \approx_o^e Q$ *then* $AS_o^e \vdash \tau.P = \tau.Q$.
(iii) *If* $P \approx_o^g Q$ *then* $AS_o^g \vdash \tau.P = \tau.Q$.
(iv) *If* $P \approx_o^b Q$ *then* $AS_o^b \vdash \tau.P = \tau.Q$.

Proof. By Lemma 22 we may assume that P, Q are in normal form on $V = fn(P \mid Q) = \{y_1, y_2, \ldots, y_k\}$. Let P be

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

and Q be

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

We prove this lemma by induction on the depth of $P \mid Q$. Suppose $\phi_i \pi_i.P_i$ is a summand of P and σ is induced by ϕ_i.

(ii) $P \approx_o^e Q$. There are several cases:

- $\pi_i \sigma$ is an update prefix $\langle y|x \rangle$. It follows from $P \approx_o^e Q$ that $Q\sigma \stackrel{y/x}{\Rightarrow} Q' \approx_o^e P_i\{y/x\}\sigma$ for some Q'. By induction hypothesis we have that $AS_o^e \vdash \tau.Q' = \tau.P_i\sigma\{y/x\}$. By (iv) of Lemma 25

$$\begin{aligned} Q &= Q + \phi_i \langle y|x \rangle.Q' \\ &= Q + \phi_i \langle y|x \rangle.\tau.Q' \\ &= Q + \phi_i \langle y|x \rangle.\tau.P_i\sigma\{y/x\} \\ &= Q + \phi_i \langle y|x \rangle.P_i\sigma\{y/x\} \\ &= Q + \phi_i \langle y|x \rangle.P_i\sigma \\ &= Q + \phi_i \pi_i \sigma.P_i\sigma \\ &= Q + \phi_i \pi_i.P_i. \end{aligned}$$

- $\pi_i \sigma$ is a bound action $\alpha(x)$. According to the definition of early open bisimilarity there are the following cases:
  - For each $l \in \{1, \ldots, k\}$, Q'_{i_l} and Q_{i_l} exist such that $Q\sigma \Rightarrow \stackrel{\alpha(x)}{\rightarrow} Q'_{i_l}\sigma$ and $Q'_{i_l}\sigma\{y_l/x\} \Rightarrow Q_{i_l} \approx_o^e P_i\sigma\{y_l/x\}$.
  - $Q'_{i_{k+1}}$ and $Q_{i_{k+1}}$ exist such that $Q\sigma \Rightarrow \stackrel{\alpha(x)}{\rightarrow} Q'_{i_{k+1}}\sigma \Rightarrow Q_{i_{k+1}} \approx_o^e P_i\sigma$.

  By induction hypothesis

$$AS_o^e \vdash \tau.Q_{i_l} = \tau.P_i\sigma\{y_l/x\}$$

  for $l \in \{1, \ldots, k\}$ and

$$AS_o^e \vdash \tau.Q_{i_{k+1}} = \tau.P_i\sigma.$$

By Lemma 26

$$\begin{aligned}
Q &= Q + \sum_{l=1}^{k} \phi_i a(x).(\tau.Q'_{i_l} + \phi_i[x = y_l]\tau.Q_{i_l}) \\
&\quad + \phi_i a(x).(\tau.Q'_{i_{k+1}} + \phi_i[x \notin V]\tau.Q_{i_{k+1}}) \\
&= Q + \sum_{l=1}^{k} \phi_i a(x).(\tau.Q'_{i_l} + \phi_i[x = y_l]\tau.P_i\sigma\{y_l/x\}) \\
&\quad + \phi_i a(x).(\tau.Q'_{i_{k+1}} + \phi_i[x \notin V]\tau.P_i\sigma) \\
&= Q + \sum_{l=1}^{k} \phi_i a(x).(\tau.Q'_{i_l} + \phi_i[x = y_l]\tau.P_i) + \phi_i a(x).(\tau.Q'_{i_{k+1}} + \phi_i[x \notin V]\tau.P_i) \\
&\stackrel{T4a}{=} Q + \phi_i a(x).P_i \\
&= Q + \phi_i \pi_i.P_i.
\end{aligned}$$

- $\pi_i\sigma$ is a free action αx. Similarly there are two cases:
  - For each $l \in \{1,\ldots,k\}$, Q'_{i_l} and Q_{i_l} exist such that $Q\sigma \Rightarrow\stackrel{\alpha x}{\rightarrow} Q'_{i_l}\sigma$ and $Q'_{i_l}\sigma\{y_l/x\} \Rightarrow Q_{i_l} \approx_o^e P_i\sigma\{y_l/x\}$.
  - $Q'_{i_{k+1}}$ and $Q_{i_{k+1}}$ exist such that $Q\sigma \Rightarrow\stackrel{\alpha x}{\rightarrow} Q'_{i_{k+1}}\sigma \Rightarrow Q_{i_{k+1}} \approx_o^e P_i\sigma$.

  By induction hypothesis

$$AS_o^e \vdash \tau.Q_{i_l} = \tau.P_i\sigma\{y_l/x\}$$

for $l \in \{1,\ldots,k\}$ and

$$AS_o^e \vdash \tau.Q_{i_{k+1}} = \tau.P_i\sigma.$$

Since ϕ_i is complete on V, it groups the elements of V into several disjoint classes. Assume that these classes are $[x],[a_1],\ldots,[a_r]$.

- If $y_l \in V^{\neq x}$ then $\phi_i^{=}\phi_i^{=x}\phi_i^{\neq}\phi_{i\backslash[y_l]}^{\neq x}[x = y_l]$ is complete on V and induces $\sigma\{y_l/x\}$. By Lemma 25

$$\begin{aligned}
Q &= Q + \phi_i\alpha x.Q'_{i_l} \\
&= Q + \phi_i\alpha x.\tau.Q'_{i_l} \\
&= Q + \phi_i\alpha x.(\tau.Q'_{i_l} + \phi_i^{=}\phi_i^{=x}\phi_i^{\neq}\phi_{i\backslash[y_l]}^{\neq x}[x = y_l]\tau.Q_{i_l}) \\
&\stackrel{(27)}{=} Q + \phi_i\alpha x.(\tau.Q'_{i_l} + \phi_i^{\neq}[x = y_l]\tau.Q_{i_l}).
\end{aligned}$$

- ∘ Since $\phi_i^{=}\phi_i^{=x}\phi_i^{\neq}\phi_i^{\neq x}$ is complete on V and induces σ, one has by Lemma 25 that

$$\begin{aligned}
Q &= Q + \phi_i \alpha x.Q'_{i_k}\\
&= Q + \phi_i \alpha x.\tau.Q'_{i_k}\\
&= Q + \phi_i \alpha x.(\tau.Q'_{i_k} + \phi_i^{=}\phi_i^{=x}\phi_i^{\neq}\phi_i^{\neq x}\tau.Q_{i_{k+1}})\\
&\overset{(28)}{=} Q + \phi_i \alpha x.(\tau.Q'_{i_k} + [x \notin n(\phi_i^{\neq})]\phi_i^{\neq}\tau.Q_{i_{k+1}})\\
&\overset{(26)}{=} Q + \phi_i \alpha x.(\tau.Q'_{i_k} + \phi_i^{\neq}[x \notin V^{\neq x}]\tau.Q_{i_{k+1}}).
\end{aligned}$$

Now

$$\begin{aligned}
Q &= Q + \sum_{y_l \in V^{\neq x}} \phi_i \alpha x.(\tau.Q'_{i_l}\\
&\quad + \phi_i^{\neq}[x = y_l]\tau.Q_{i_l}) + \phi_i \alpha x.(\tau.Q'_{i_{k+1}} + \phi_i^{\neq}[x \notin V^{\neq x}]\tau.Q_{i_{k+1}})\\
&= Q + \sum_{y_l \in V^{\neq x}} \phi_i \alpha x.(\tau.Q'_{i_l} + \phi_i^{\neq}[x = y_l]\tau.P_i\sigma\{y_l/x\})\\
&\quad + \phi_i \alpha x.(\tau.Q'_{i_{k+1}} + \phi_i^{\neq}[x \notin V^{\neq x}]\tau.P_i\sigma)\\
&= Q + \sum_{y_l \in V^{\neq x}} \phi_i \alpha x.(\tau.Q'_{i_l} + \phi_i^{\neq}[x = y_l]\tau.P_i) + \phi_i \alpha x.(\tau.Q'_{i_{k+1}} + \phi_i^{\neq}[x \notin V^{\neq x}]\tau.P_i)\\
&\overset{T4}{=} Q + \phi_i[x \notin n(\phi_i^{\neq})]\phi_i^{\neq}\alpha x.P_i\\
&= Q + \phi_i \alpha x.P_i.
\end{aligned}$$

- $\pi_i\sigma$ is a tau action. If the tau action is matched by $Q\sigma \overset{\tau}{\Rightarrow} Q'$ then it is easy to prove that $AS_o^e \vdash Q = Q + \phi_i\pi_i.P_i$. If the tau action is matched vacuously then $AS_o^e \vdash Q + \phi_i\pi_i.P_i = Q + \phi_i\tau.Q$.

In summary we have $AS_o^e \vdash P + Q = Q + \sum_{i\in I'} \phi_i\tau.Q$ for some $I' \subseteq I$. So by T1b we get

$$AS_o^e \vdash \tau.(P + Q) = \tau.\left(Q + \sum_{i\in I'} \phi_i\tau.Q\right) = \tau.Q$$

Symmetrically we can prove $AS_o^e \vdash \tau.(P + Q) = \tau.P$. Hence $AS_o^e \vdash \tau.P = \tau.Q$.

(i) $P \approx_o^l Q$. The proof is similar to that for $\approx_o^e$. We consider only one case:

- $\pi_i\sigma$ is a bound action $\alpha(x)$. It follows from $P \approx_o^l Q$ that some Q' exists such that the following hold:
  - ∘ For each $l \in \{1,\ldots,k\}$, Q' and Q_{i_l} exist such that $Q\sigma \Rightarrow \overset{\alpha(x)}{\to} Q'\sigma$ and $Q'\sigma\{y_l/x\} \Rightarrow Q_{i_l} \approx_o^l P_i\sigma\{y_l/x\}$.
  - ∘ $Q_{i_{k+1}}$ exists such that $Q\sigma \Rightarrow \overset{\alpha(x)}{\to} Q'\sigma \Rightarrow Q_{i_{k+1}} \approx_o^l P_i\sigma$.

By induction hypothesis,

$$AS_o^l \vdash \tau.Q_{i_l} = \tau.P_i\sigma\{y_l/x\}$$

for $l \in \{1,\ldots,k\}$ and

$$AS_o^l \vdash \tau.Q_{i_{k+1}} = \tau.P_i\sigma.$$

By (ii) of Lemma 26 we get

$$\begin{aligned}
Q &= Q + \phi_i\alpha(x).\left(\tau.Q' + \phi_i \sum_{l=1}^{k}[x = y_l]\tau.Q_{i_l} + \phi_i[x \notin V]\tau.Q_{i_{k+1}}\right)\\
&= Q + \phi_i\alpha(x).\left(\tau.Q' + \phi_i \sum_{l=1}^{k}[x = y_l]\tau.P_i\sigma\{y_l/x\} + \phi_i[x \notin V]\tau.P_i\sigma\right)\\
&= Q + \phi_i\alpha(x).\left(\tau.Q' + \phi_i \sum_{l=1}^{k}[x = y_l]\tau.P_i + \phi_i[x \notin V]\tau.P_i\right)\\
&= Q + \phi_i\alpha(x).(\tau.Q' + \phi_i\tau.P_i)\\
&\overset{T3a}{=} Q + \phi_i\alpha(x).P_i.
\end{aligned}$$

Then by a similar argument as in (i) we get that $AS_o^l \vdash \tau.P = \tau.Q$.

(iii) The proof is similar to that of (iv).

(iv) Suppose $P \approx_o^b Q$ and $P\sigma \xrightarrow{\pi_i\sigma} P_i\sigma$. By assumption Q must be able to match this action. There are several cases:

- $\pi_i\sigma$ is a bound action $\alpha(x) = \alpha_i\sigma(x)$. In this case V could be divided into two parts V_1 and V_2.
  - For each $y \in V_1 \subseteq V$, Q''_y and Q' exist such that $Q\sigma \Rightarrow \xrightarrow{\alpha y} Q''_y\sigma \Rightarrow Q' \approx_o^b P_i\sigma\{y/x\}$. By induction hypothesis $AS_o^b \vdash \tau.Q' = \tau.P_i\sigma\{y/x\}$. Then by (i) and (ii) of Lemma 25,

$$\begin{aligned}
Q &= Q + \phi_i\alpha y.Q''_y\\
&= Q + \phi_i\alpha y.(Q''_y + \phi_i\tau.Q')\\
&= Q + \phi_i\alpha y.(Q''_y + \phi_i\tau.P_i\sigma\{y/x\})\\
&\overset{MD3}{=} Q + \phi_i\alpha y.(Q''_y + \phi_i\tau.P_i\{y/x\})\\
&= Q + \phi_i\alpha y.(Q''_y + \phi_i^{=}\phi_i^{=x}\phi_i^{\neq}\phi_i^{\neq x}\tau.P_i\{y/x\})\\
&\overset{(28)}{=} Q + \phi_i\alpha y.(Q''_y + [x \notin n(\phi_i^{=})]\phi_i^{=}\tau.P_i\{y/x\}).
\end{aligned}$$

- For each $y \in V_2$, Q''_y and Q' exist such that $Q\sigma \Rightarrow \stackrel{\alpha(x)}{\rightarrow} Q''_y\sigma$ and

$$Q''_y\sigma\{y/x\} \Rightarrow Q' \approx_o^b P_i\sigma\{y/x\}.$$

By induction hypothesis $AS_o^b \vdash \tau.Q' = \tau.P_i\sigma\{y/x\}$. Then by (iii) of Lemma 25 and the fact that $[x=y]\phi_i$ induces $\sigma\{y/x\}$ and is complete on $V \cup \{x\}$, one has

$$\begin{aligned} Q &= Q + \phi_i\alpha(x).Q''_y \\ &= Q + \phi_i\alpha(x).(Q''_y + \phi_i[x=y]\tau.Q') \\ &= Q + \phi_i\alpha(x).(Q''_y + \phi_i[x=y]\tau.P_i\sigma\{y/x\}) \\ &= Q + \phi_i\alpha(x).(Q''_y + \phi_i[x=y]\tau.P_i) \\ &= Q + \phi_i\alpha(x).(Q''_y + \phi_i^{=}\phi_i^{=x}\phi_i^{\neq}\phi_i^{\neq x}[x=y]\tau.P_i) \\ &\stackrel{(28)}{=} Q + \phi_i\alpha(x).(Q''_y + [x \notin n(\phi_i^{=})]\phi_i^{=}[x=y]\tau.P_i). \end{aligned}$$

- Q''_y and Q' exist such that $Q\sigma \Rightarrow \stackrel{\alpha(x)}{\rightarrow} Q''_y\sigma$ and $Q''_y\sigma \Rightarrow Q' \approx_o^b P_i\sigma$. By induction hypothesis $AS_o^b \vdash \tau.Q' = \tau.P_i\sigma$. Then by (iii) of Lemma 25 and the fact that $[x \notin V]\phi_i$ induces σ and is complete on $V \cup \{x\}$, we can prove in similar manner that

$$AS_o^b \vdash Q = Q + \phi_i\alpha(x).(Q''_y + [x \notin n(\phi_i^{=})]\phi_i^{=}[x \notin V]\tau.P_i).$$

Putting together all the equalities we have obtained, one has

$$\begin{aligned} Q &= Q + \phi_i\alpha(x).(Q'' + [x \notin n(\phi_i^{=})]\phi_i^{=}[x \notin V]\tau.P_i) \\ &\quad + \phi_i \sum_{y \in V_1} \alpha y.(Q''_y + [x \notin n(\phi_i^{=})]\phi_i^{=}\tau.P_i\{y/x\}) \\ &\quad + \phi_i \sum_{y \in V_2} \alpha(x).(Q''_y + [x \notin n(\phi_i^{=})]\phi_i^{=}[x=y]\tau.P_i) \\ &\stackrel{T5a}{=} Q + \phi_i[x \notin n(\phi_i^{=})]\phi_i^{=}\alpha(x).P_i \\ &= Q + \phi_i\alpha(x).P_i \\ &= Q + \phi_i\alpha_i\sigma(x).P_i \\ &= Q + \phi_i\alpha_i(x).P_i. \end{aligned}$$

- $\pi_i\sigma$ is a free action $\alpha x = \sigma(\alpha_i)\sigma(x_i)$. As in the proof of (ii) we define the equivalent classes $[x], [a_1], \ldots, [a_r]$ and the notations $\phi_i^{=}$, $\phi_i^{=x}$, $\phi_i^{\neq}$, $\phi_i^{\neq x}$, $V^{=x}$ and $V^{\neq x}$. Now $V^{\neq x}$ could be divided into at most five disjoint subsets V_1, V_2, V_3, V_4, V_5 according to how $Q\sigma$ simulates the free action.

- For each $y \in V_1$, Q''_y and Q' exist such that $Q\sigma \Rightarrow \stackrel{\alpha x}{\rightarrow} Q''_y\sigma$ and $Q''_y\sigma\{y/x\} \Rightarrow Q' \approx_o^b P_i\sigma\{y/x\}$. By induction hypothesis $AS_o^b \vdash \tau.Q' = \tau.P_i\sigma\{y/x\}$. Now $\phi_i^= \phi_i^{=x} \phi_i^{\neq} \phi_{i\backslash[y]}^{\neq x}[x=y]$ is complete on V and induces $\sigma\{y/x\}$. Therefore

$$\begin{aligned} Q &= Q + \phi_i \alpha x.Q''_y \\ &= Q + \phi_i \alpha x.(Q''_y + \phi_i^= \phi_i^{=x} \phi_i^{\neq} \phi_{i\backslash[y]}^{\neq x}[x = y]\tau.Q') \\ &\stackrel{(27)}{=} Q + \phi_i \alpha x.(Q''_y + \phi_i^{\neq}[x = y]\tau.P_i). \end{aligned}$$

- For each $y \in V_2$, Q''_y and Q' exist such that $Q\sigma \Rightarrow \stackrel{\alpha(z)}{\rightarrow} Q''_y\sigma$ and $Q''_y\sigma\{y/z\} \stackrel{y/x}{\Rightarrow} Q' \approx_o^b P_i\sigma\{y/x\}$. By induction hypothesis $AS_o^b \vdash \tau.Q' = \tau.P_i\sigma\{y/x\}$. Since $\phi_i[z=y]$ is complete on $V \cup \{z\}$ and induces $\sigma\{y/z\}$, one has

$$\begin{aligned} Q &= Q + \phi_i \alpha(z).Q''_y \\ &= Q + \phi_i \alpha(z).(Q''_y + \phi_i[z = y]\langle y|x\rangle.Q') \\ &= Q + \phi_i \alpha(z).(Q''_y + \phi_i[z = y]\langle y|x\rangle.\tau.Q') \\ &= Q + \phi_i \alpha(z).(Q''_y + \phi_i[z = y]\langle y|x\rangle.\tau.P_i\sigma\{y/x\}) \\ &= Q + \phi_i \alpha(z).(Q''_y + \phi_i[z = y]\langle y|x\rangle.P_i\sigma\{y/x\}) \\ &= Q + \phi_i \alpha(z).(Q''_y + \phi_i[z = y]\langle y|x\rangle.P_i) \\ &\stackrel{(28)}{=} Q + \phi_i \alpha(z).(Q''_y + [x \notin n(\phi_i^{\neq})]\phi_i^{\neq}[z = y]\langle z|x\rangle.P_i). \end{aligned}$$

- For each $y \in V_3$, Q''_y and Q' exist such that $Q\sigma \Rightarrow \stackrel{ay}{\rightarrow} Q''_y\sigma \stackrel{y/x}{\Rightarrow} Q' \approx_o^b P_i\sigma\{y/x\}$. By induction hypothesis $AS_o^b \vdash \tau.Q' = \tau.P_i\sigma\{y/x\}$.

$$\begin{aligned} Q &= Q + \phi_i \alpha y.Q''_y \\ &= Q + \phi_i \alpha y.(Q''_y + \phi_i\langle y|x\rangle.Q') \\ &= Q + \phi_i \alpha y.(Q''_y + \phi_i\langle y|x\rangle.\tau.Q') \\ &= Q + \phi_i \alpha y.(Q''_y + \phi_i\langle y|x\rangle.\tau.P_i\sigma\{y/x\}) \\ &= Q + \phi_i \alpha y.(Q''_y + \phi_i\langle y|x\rangle.P_i\sigma\{y/x\}) \\ &= Q + \phi_i \alpha y.(Q''_y + \phi_i\langle y|x\rangle.P_i) \\ &\stackrel{(28)}{=} Q + \phi_i \alpha y.(Q''_y + [x \notin n(\phi_i^{\neq})]\phi_i^{\neq}\langle y|x\rangle.P_i) \\ &= Q + \phi_i[x \neq y]\alpha y.(Q''_y + [x \notin n(\phi_i^{\neq})]\phi_i^{\neq}\langle y|x\rangle.P_i). \end{aligned}$$

- For each $y \in V_4$, Q'''_y, Q''_y and Q' exist such that $Q\sigma \stackrel{y/x}{\Rightarrow} Q''_y\sigma\{y/x\} \stackrel{\alpha y}{\to} Q'''_y\sigma\{y/x\} \Rightarrow Q' \approx_o^b P_i\sigma\{y/x\}$. By induction hypothesis $AS_o^b \vdash \tau.Q' = \tau.P_i\sigma\{y/x\}$. Now $\phi_i^= \phi_i^{=x} \phi_i^{\neq} \phi_{i\setminus[y]}^{\neq x}[x=y]$ is complete on V and induces $\sigma\{y/x\}$. Therefore

$$
\begin{aligned}
Q &= Q + \phi_i\langle y|x\rangle.Q''_y \\
&= Q + \phi_i\langle y|x\rangle.(Q''_y + \phi_i^= \phi_i^{=x} \phi_i^{\neq} \phi_{i\setminus[y]}^{\neq x}[x=y]\alpha y.Q'''_y) \\
&\stackrel{(27)}{=} Q + \phi_i\langle y|x\rangle.(Q''_y + \phi_i^{\neq}\alpha y.Q'''_y) \\
&= Q + \phi_i\langle y|x\rangle.(Q''_y + \phi_i^{\neq}\alpha y.(Q'''_y + \phi_i^= \phi_i^{=x} \phi_i^{\neq} \phi_{i\setminus[y]}^{\neq x}[x=y]\tau.Q')) \\
&\stackrel{(27)}{=} Q + \phi_i\langle y|x\rangle.(Q''_y + \phi_i^{\neq}\alpha y.(Q'''_y + \phi_i^{\neq}\tau.Q')) \\
&= Q + \phi_i\langle y|x\rangle.(Q''_y + \phi_i^{\neq}\alpha y.(Q'''_y + \phi_i^{\neq}\tau.P_i\sigma\{y/x\})) \\
&= Q + \phi_i\langle y|x\rangle.(Q''_y + \phi_i^{\neq}\alpha y.(Q'''_y + \phi_i^{\neq}\tau.P_i)).
\end{aligned}
$$

- For each $y \in V_5$ there exist Q''_y, Q'''_y and Q' such that $Q\sigma \stackrel{y/x}{\Rightarrow} Q''_y\sigma\{y/x\} \Rightarrow \stackrel{\alpha(z)}{\to} Q'''_y\sigma\{y/x\}$ and $Q'''_y\sigma\{y/x\}\{y/z\} \Rightarrow Q' \approx_o^b P_i\{y/x\}$. In similar manner one shows that

$$Q = Q + \phi_i\langle y|x\rangle.(Q''_y + \phi_i^{\neq}\alpha(z).(Q'''_y + \phi_i^{\neq}[z=y]\tau.P_i)).$$

- For name $z \notin V^{\neq x}$, it is clear that σ substitutes z for x. There are two possibilities:

  1. Q'' and Q' exist such that $Q\sigma \Rightarrow \stackrel{\alpha x}{\to} Q''\sigma$ and $Q''\sigma \Rightarrow Q' \approx_o^b P_i\sigma$. Then we get

$$
\begin{aligned}
Q &= Q + \phi_i\alpha x.Q'' \\
&= Q + \phi_i\alpha x.(Q'' + \phi_i^= \phi_i^{=x} \phi_i^{\neq x} \phi_i^{\neq}\tau.Q') \\
&\stackrel{(28)}{=} Q + \phi_i\alpha x.(Q'' + [x \notin n(\phi_i^{\neq})]\phi_i^{\neq}\tau.Q') \\
&= Q + \phi_i\alpha x.(Q'' + [x \notin n(\phi_i^{\neq})]\phi_i^{\neq}\tau.P_i),
\end{aligned}
$$

  where the last equality holds by induction hypothesis.

  2. Q'' and Q' exist such that $Q\sigma \Rightarrow \stackrel{\alpha(z)}{\to} Q''\sigma$ and $Q''\sigma\{x/z\} \Rightarrow Q' \approx_o^b P_i\sigma$. Then

$$
\begin{aligned}
Q &= Q + \phi_i\alpha(z).Q'' \\
&= Q + \phi_i\alpha(z).(Q'' + \phi_i[z=x]\tau.Q') \\
&= Q + \phi_i\alpha(z).(Q'' + \phi_i[z=x]\tau.P_i\sigma)
\end{aligned}
$$

$$
\begin{aligned}
&= Q + \phi_i\alpha(z).(Q'' + \phi_i[z = x]\tau.P_i)\\
&\overset{(28)}{=} Q + \phi_i\alpha(z).(Q'' + [x \notin n(\phi_i^{\neq})]\phi_i^{\neq}[z = x]\tau.P_i)\\
&= Q + \phi_i\alpha(z).(Q'' + [x \notin n(\phi_i^{\neq})]\phi_i^{\neq}[z = x]\langle z|x\rangle.P_i),
\end{aligned}
$$

where the third equality holds by induction hypothesis.

If the subcase 1 holds, we can use T5 to conclude that

$$
\begin{aligned}
Q &= Q + \phi_i\alpha x.(Q'' + [x \notin n(\phi_i^{\neq})]\phi_i^{\neq}\tau.P_i)\\
&\quad + \phi_i \sum_{y\in V_1} \alpha y.(Q''_y + \phi_i^{\neq}[x = y]\tau.P_i\{y/x\})\\
&\quad + \phi_i \sum_{y\in V_2} \alpha(z).(Q''_y + [x \notin n(\phi_i^{\neq})]\phi_i^{\neq}[z = y]\langle z|x\rangle.P_i)\\
&\quad + \phi_i \sum_{y\in V_3} [x \neq y]\alpha y.(Q''_y + [x \notin n(\phi_i^{\neq})]\phi_i^{\neq}\langle y|x\rangle.P_i)\\
&\quad + \phi_i \sum_{y\in V_4} \langle y|x\rangle.(Q''_y + \phi_i^{\neq}\alpha y.(Q'''_y + \phi_i^{\neq}\tau.P_i))\\
&\quad + \phi_i \sum_{y\in V_5} \langle y|x\rangle.(Q''_y + \phi_i^{\neq}\alpha(z).(Q'''_y + \phi_i^{\neq}[z = y]\tau.P_i))\\
&= Q + \phi_i[x \notin V_3][x \notin n(\phi_i^{\neq})]\phi_i^{\neq}\alpha x.P_i\\
&= Q + \phi_i[x \notin V_3][x \notin V^{\neq x}]\phi_i^{\neq}\alpha x.P_i\\
&= Q + \phi_i\alpha x.P_i\\
&= Q + \phi_i\sigma(\alpha_i)\sigma(x_i).P_i\\
&= Q + \phi_i\alpha_i x_i.P_i.
\end{aligned}
$$

Otherwise subcase 2 must hold. Then we can use T6 to get $Q = Q + \phi_i\alpha_i x_i.P_i$ using a similar derivation.

- $\pi_i\sigma$ is an update action $\langle y|x\rangle = \langle y_i\sigma|x_i\sigma\rangle$. Then Q' exists such that $Q\sigma \overset{y/x}{\Rightarrow} Q' \approx_o^b P_i\sigma\{y/x\}$. By induction hypothesis one has $AS_o^b \vdash \tau.Q' = \tau.P_i\sigma\{y/x\}$. By (iv) of Lemma 25 one gets

$$
\begin{aligned}
Q &= Q + \phi_i\langle y|x\rangle.Q'\\
&= Q + \phi_i\langle y|x\rangle.\tau.Q'\\
&= Q + \phi_i\langle y|x\rangle.\tau.P_i\sigma\{y/x\}\\
&= Q + \phi_i\langle y|x\rangle.P_i\sigma\{y/x\}\\
&= Q + \phi_i\langle y|x\rangle.P_i\sigma
\end{aligned}
$$

$$= Q + \phi_i \langle y_i\sigma | x_i\sigma \rangle . P_i\sigma$$

$$= Q + \phi_i \langle y_i | x_i \rangle . P_i$$

- $\pi_i\sigma$ is a tau action. If the tau action is matched by $Q\sigma \stackrel{\tau}{\Rightarrow} Q'$ then it is easy to prove that $AS_o^b \vdash Q = Q + \phi_i \pi_i . P_i$. If the tau action is matched up by Q vacuously then we can prove that $AS_o^b \vdash Q + \phi_i \pi_i . P_i = Q + \phi_i \tau . Q$.

In summary we have $AS_o^b \vdash P + Q = Q + \sum_{i \in I'} \phi_i \tau . Q$ for some $I' \subseteq I_3$. By Lemma 24 we get $AS_o^b \vdash \tau.(P + Q) = \tau.(Q + \sum_{i \in I'} \phi_i \tau . Q) = \tau . Q$. Symmetrically we can prove $AS_o^b \vdash \tau.(P + Q) = \tau . P$. Hence $AS_o^b \vdash \tau . P = \tau . Q$. □

The promotion lemma can now be used to prove the main result of this section.

Theorem 28 (Completeness). *In $\chi^{\neq}$-calculus the following completeness results hold*:
(i) $P \simeq_o^l Q$ *if and only if* $AS_o^l \vdash P = Q$.
(ii) $P \simeq_o^e Q$ *if and only if* $AS_o^e \vdash P = Q$.
(iii) $P \simeq_o^g Q$ *if and only if* $AS_o^g \vdash P = Q$.
(iv) $P \simeq_o^b Q$ *if and only if* $AS_o^b \vdash P = Q$.

Proof. The implications from the right to the left are about soundness. The soundness of *AS* is subsumed by the soundness of Parrow and Victor's system for the strong hyperequivalence [29]. The verifications of the validity of the tau laws are routine and simple.

The implications from the left to the right are about completeness. By Lemmas 25 and 27 one can prove the completeness in very much the same way the proof of Lemma 27 is done. □

9. Bisimulation lattice

By definition observational equivalences place a lot of emphasis on observers. In process algebra, the role of the observers are played by the contexts. Two processes are tested for equality by putting them in same contexts and then observing the consequences. This approach actually calls for a careful study of contexts. However working with contexts are not always that easy. A formal treatment of contexts would definitely introduce a notion of equality between them, which conceivably depends on a notion of equality for processes. The problem can be avoided by confining our attention to processes. With the help of a labeled transition system, the bisimulation approach tries to define equivalences between processes purely in terms of the actions the processes can perform, disregarding all contexts. This approach has been very successful with CCS. For the π-calculus, the open bisimilarities can be defined without referring to contexts, although a new defining property, closure under substitution, is generally required. In the theory of χ-calculus the bisimulation approach has also been successful. All the bisimulation equivalences proposed so far have equivalent characterizations in terms of open style bisimulations. These characterizations also have the virtue that they do not refer to contexts. We have seen in this paper that working without contexts is a great advantage as far as axiomatization is concerned.

But contexts do help to form intuitions. The definitions of the barbed bisimilarity and the ground bisimilarity are straightforward and conceptually clear. The great difference between these equivalences and their open counterparts can only serve as a support for the simplicity of the two definitions. Now the question is if the contexts can help us to find other interesting equivalence relations.

We will give a classification of the bisimulation equivalences on $\chi^{\neq}$-processes in terms of the bisimulation lattice introduced in [6]. The bisimulation lattice builds on a classification of actions. In this paper we adopt the classification given in [6]. Four sets of actions are defined as follows:

- u is the set $\{y/x \mid x, y \in \mathcal{N}\}$ of updates.
- ba is the set $\{a(x) \mid x, a \in \mathcal{N}\}$ of bound actions with positive subject names.
- $\overline{ba}$ is the set $\{\bar{a}(x) \mid x, a \in \mathcal{N}\}$ of bound actions with negative subject names.
- fa is the set $\{ax \mid x, a \in \mathcal{N}\}$ of free actions with positive subject names.
- $\overline{fa}$ is the set $\{\bar{a}x \mid x, a \in \mathcal{N}\}$ of free actions with negative subject names.

Let $\mathcal{L}$ be $\{\bigcup S \mid S \subseteq \{u, ba, \overline{ba}, fa, \overline{fa}\} \wedge S \neq \emptyset\}$.

Definition 29. Suppose $\mathcal{R}$ is a symmetric binary relation on $\mathcal{C}$ closed under contexts and $L \in \mathcal{L}$. It is called an L-bisimulation if whenever $P \mathcal{R} Q$ and $P \xrightarrow{\lambda} P'$ for $\lambda \in L \cup \{\tau\}$ then some Q' exists such that $Q \stackrel{\hat{\lambda}}{\Rightarrow} Q' \mathcal{R} P'$. The L-bisimilarity $\approx_L$ is the largest
L-bisimulation.

Without further ado, we begin to discuss the order relationship of L-bisimilarities.

Lemma 30. *The following properties hold*:
(i) $\approx_L \subseteq \approx_u$ *for each* $L \in \mathcal{L}$.
(ii) $\approx_L \subseteq \approx_{ba}$ *and* $\approx_L \subseteq \approx_{\overline{ba}}$ *for each* $L \in \mathcal{L}$.
(iii) $\approx_{fa} \nsubseteq \approx_{\overline{fa}}$; $\approx_{\overline{fa}} \nsubseteq \approx_{fa}$.

Proof. (i) Suppose $P \approx_L Q$ and $P \xrightarrow{y/x} P'$. Let a, b, z be fresh and let A be

$$(aa + \bar{a}a + a(z) + \bar{a}(z) + \langle b|a \rangle) \mid [x = y](aa + \bar{a}a + a(z) + \bar{a}(z) + \langle b|a \rangle).$$

Then $(x)(P|A) \Rightarrow P' \mid (\mathbf{0} \mid \mathbf{0})$. Therefore $(x)(Q|A) \Rightarrow Q' \mid (\mathbf{0} \mid \mathbf{0}) \approx_L P' \mid (\mathbf{0} \mid \mathbf{0})$. It has to be the case that $Q \stackrel{y/x}{\Rightarrow} Q' \approx_L P'$.

(ii) Suppose $P \approx_L Q$ and $P \xrightarrow{\alpha(x)} P'$. Let a, b be fresh. Then $P \mid \bar{\alpha}x.\langle a|b \rangle \xrightarrow{\tau} \xrightarrow{b/a} P' \mid \mathbf{0}$. By (i) some Q' exists such that $Q \mid \bar{\alpha}x.\langle a|b \rangle \stackrel{b/a}{\Rightarrow} Q' \mid \mathbf{0} \approx_L P' \mid \mathbf{0}$. It follows that $Q \stackrel{\alpha(x)}{\Rightarrow} Q' \approx_L P'$.

(iii) It is clear that

$$a(z).(P + \langle z|x \rangle.Q) \approx_{\overline{fa}} a(z).(P + \langle z|x \rangle.Q) + ax.Q\{x/z\}.$$

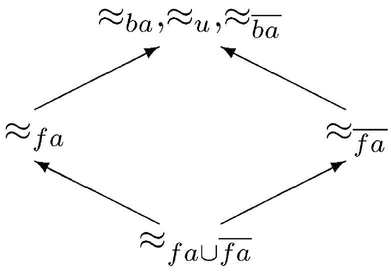

Fig. 7. The bisimulation lattice of $\chi^{\neq}$.

However

$$fa(z).(P + \langle z|x\rangle.Q) \not\approx_{fa} a(z).(P + \langle z|x\rangle.Q) + ax.Q\{x/z\}.$$

So $\approx_{\overline{fa}} \nsubseteq \approx_{fa}$. For similar reason $\approx_{fa} \nsubseteq \approx_{\overline{fa}}$. $\square$

The above lemma implies that there are only four distinct L-bisimilarities. In Fig. 7 the relationship of the four L-bisimilarities are described in a diagram, in which an arrow indicates an inclusion and each node represents a class of equal L-bisimilarities.

If we assume that an observer can observe an action, say ax if and only if it can also observe the complementary action, which is $\bar{a}x$ in this case, then it makes sense to say that there are only two reasonable observational equivalences for $\chi^{\neq}$-calculus. The next lemma says that these two equivalences are precisely the barbed bisimilarity and the ground bisimilarity.

Lemma 31. (i) *$\approx_b$ is equal to $\approx_u$.* (ii) *$\approx_g$ is the same as $\approx_{fa \cup \overline{fa}}$.*

Proof. (i) Suppose $P \approx_u Q$ and $P \Rightarrow \xrightarrow{cx} P'$. Then $P \,|\, \bar{c}(y).\langle a|b\rangle \overset{b/a}{\Rightarrow} P''$ for fresh a, b and some P''. By definition some Q' exists such that $Q \,|\, \bar{c}(y).\langle a|b\rangle \overset{b/a}{\Rightarrow} Q' \approx_u P''$. It follows that $P \Downarrow c$. So $\approx_u \subseteq \approx_b$. The reverse inclusion follows from Lemma 12.

(ii) It is obvious that $\approx_g \subseteq \approx_{fa \cup \overline{fa}}$. The reverse inclusion is supported by the proofs of (i) and (ii) of Lemma 30. $\square$

Definition 32. Suppose $L \in \mathscr{L}$. Two processes P and Q are L-congruent, notation $P \simeq_L Q$, if $P \approx_L Q$ and, for each substitution σ, the following conditions are satisfied:

(i) If $P\sigma \xrightarrow{\tau} P'$ then Q' exists such that $Q\sigma \overset{\tau}{\Rightarrow} Q'$ and $P' \approx_L Q'$.
(ii) If $Q\sigma \xrightarrow{\tau} Q'$ then P' exists such that $P\sigma \overset{\tau}{\Rightarrow} P'$ and $P' \approx_L Q'$.

We now discuss the completeness issues for L-congruences. By Lemmas 30 and 31, we only have to look at $\simeq_{\overline{fa}}$ and $\simeq_{fa}$. Now let $T6^+$ be

$$\begin{aligned} TT6^+ \stackrel{\text{def}}{=} & \; a(z).(P + [x \notin n(\delta)]\delta[z \notin Y]\langle z|x\rangle.Q) \\ & + \sum_{y \in Y_1} ax.(P_y + \delta[x = y]\tau.Q\{x/z\}) \\ & + \sum_{y \in Y_2} a(z).(P_y + [x \notin n(\delta)]\delta[z = y]\langle z|x\rangle.Q) \\ & + \sum_{y \in Y_3} [x \neq y]ay.(P_y + [x \notin n(\delta)]\delta\langle y|x\rangle.Q\{x/z\}) \\ & + \sum_{y \in Y_4} \langle y|x\rangle.(P_y + \delta ay.(P'_y + \delta\tau.Q\{x/z\})) \\ & + \sum_{y \in Y_5} \langle y|x\rangle.(P_y + \delta a(z).(P'_y + \delta[z = y]\tau.Q)) \\ = & \; TT6^+ + [x \notin Y_3][x \notin n(\delta)]\delta ax.Q\{x/z\} \end{aligned}$$

and let $T6^-$ be

$$\begin{aligned} TT6^- \stackrel{\text{def}}{=} & \; \bar{a}(z).(P + [x \notin n(\delta)]\delta[z \notin Y]\langle z|x\rangle.Q) \\ & + \sum_{y \in Y_1} \bar{a}x.(P_y + \delta[x = y]\tau.Q\{x/z\}) \\ & + \sum_{y \in Y_2} \bar{a}(z).(P_y + [x \notin n(\delta)]\delta[z = y]\langle z|x\rangle.Q) \\ & + \sum_{y \in Y_3} [x \neq y]\bar{a}y.(P_y + [x \notin n(\delta)]\delta\langle y|x\rangle.Q\{x/z\}) \\ & + \sum_{y \in Y_4} \langle y|x\rangle.(P_y + \delta\bar{a}y.(P'_y + \delta\tau.Q\{x/z\})) \\ & + \sum_{y \in Y_5} \langle y|x\rangle.(P_y + \delta\bar{a}(z).(P'_y + \delta[z = y]\tau.Q)) \\ = & \; TT6^- + [x \notin Y_3][x \notin n(\delta)]\delta\bar{a}x.Q\{x/z\}, \end{aligned}$$

where Y is $Y_1 \cup Y_2 \cup Y_3 \cup Y_4 \cup Y_5$ and $z \notin n(\delta)$. Using $T6^+$ and $T6^-$ we can state the following completeness theorem.

Theorem 33. *The following properties hold*:

(i) $AS \cup \{T1, T2, T5, T6^+\}$ *is sound and complete for* $\simeq_{\overline{fa}}$.
(ii) $AS \cup \{T1, T2, T5, T6^-\}$ *is sound and complete for* $\simeq_{fa}$.

Proof. The proof of this theorem is completely the same as that of Theorem 28, bearing in mind that for barbed congruence the free actions are not compared against each other whereas for $\overline{fa}$-congruence, respectively fa-congruence, the free actions with positive subject names, respectively negative subject names, are not compared against each other. □

10. Remark

The first author of the paper has been working on χ-calculus for some years. His attention had always been on the version of χ without the mismatch combinator. By the end of 1999 he started looking at testing congruence on χ-processes. In order to axiomatize the testing congruence he was forced to introduce the mismatch operator. This led him to deal with open congruences on $\chi^{\neq}$-processes, which made him aware of the fact that the open semantics for the π-calculus with the mismatch combinator has not been investigated before. So he, together with the second author, began to work on the problem. Their investigation showed that the obvious definition of open bisimilarity is not closed under the parallel composition. It is then a small step to realize the problem of the weak hyperequivalence.

The purpose of this paper is to provide solutions to the above problem. Historically the early and the late open bisimilarities [11] were proposed before the barbed and ground bisimilarities [12]. The relationship between the early open bisimilarity and the late open bisimilarity strongly recalls that between the weak early equivalence and the weak late equivalence [24]. It should be said however that both the early open bisimilarity and the late open bisimilarity are the obvious modifications with motivation from π-calculus. They are not *the* open bisimilarity for the χ-calculus with the mismatch operator. The definition of the ground bisimilarity is natural. Its open counterpart is more general than the early and late open bisimilarities.

The paper improves upon previous work in several directions:

- The subtlety of the mismatch operator is brought under light. For χ-like process calculi the combinator changes the algebraic semantics dramatically. For other calculi of mobile processes it has more or less the same dramatic effect [13]. The approach and the techniques used in this paper should be relevant to the studies of a wide range of mobile calculi.
- The tau laws in this paper simplifies those given in [12]. We have combined four of the tau laws in [12] into two tau laws, T1 and T3, in this paper. And we have also dropped the following law present in [12]:

$$\langle y|x\rangle.(P+\delta\tau.Q) = \langle y|x\rangle.(P+\delta\tau.Q) + \psi\delta\langle y|x\rangle.Q. \quad (29)$$

This law comes with the following side condition:
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 $\psi \Rightarrow [x\neq u][x\neq v][y\neq u][y\neq v]$.
Notice that the side condition is internalized in (29) as ψ. In this paper we have found a way to bypass this law by using T3b that is derivable from T3. It is clear

that T3b is a special case of (29) and, by completeness, is equivalent to (29) in the system $AS \cup \{T1, T2, T3\}$.

- In [29] four tau laws are proposed for fusion calculus. Using the notations of [29] they can be written as follows:

$$\alpha.\mathbf{1}.P = \alpha.P, \tag{30}$$

$$P + \mathbf{1}.P = \mathbf{1}.P, \tag{31}$$

$$\alpha.(P + \tilde{M}\mathbf{1}.Q) = \alpha.(P + \tilde{M}\mathbf{1}.Q) + \tilde{M}\alpha.Q, \tag{32}$$

$$\iota.(P + \tilde{M}\rho.Q) = \iota.(P + \tilde{M}\rho.Q) + \tilde{M}\iota \wedge \rho.Q \text{ if } \forall u, v.(\tilde{M} \Rightarrow u \neq v) \Rightarrow \neg(u\iota v), \tag{33}$$

where α is a communication action, ι, ρ are fusion actions, $\tilde{M}$ is a sequence of match/mismatch operators, and **1** is the tau prefix. Here are some observations on these tau laws:
  - The laws (30) and (31) are two of the three of Milner's tau laws.
  - The 'law' (32) is not valid. The counterexample to (32) is given in the introduction.
  - The 'law' (33) is not valid either. The problem is too obvious to worth a comment. It is not even valid for hyperequivalence. Even the hyperequivalence is closed under substitution. The equality (33), by its very definition, defeats the property of closure under substitution! The following is an instance of (33) since the side condition is satisfied:

$$\{x = y\}.(P + [u \neq v]\mathbf{1}.Q) = \{x = y\}.(P + [u \neq v]\mathbf{1}.Q) + [u \neq v]\{x = y\}.Q.$$

But if we substitute u for x and v for y we get the equality

$$\{x = y\}.(P + [x \neq y]\mathbf{1}.Q) = \{x = y\}.(P + [x \neq y]\mathbf{1}.Q) + [x \neq y]\{x = y\}.Q$$

which is really wrong. The formula

$$\forall u, v.(\tilde{M} \Rightarrow u \neq v) \Rightarrow \neg(u\iota v)$$

is not the description of an internal condition due to the presence of a fusion action ι. It is the confusion of two things that makes the above formula meaningless.
  - The equality $\tau.P = \tau.(P + \psi\tau.P)$ is not derivable, which means that none of the completeness results for weak congruences in [29] is correct. These weak systems cannot prove $\tau.[x = y]\tau = \tau$, which is an instance of our T1.

  Our T3 is the correction of (32) and our T3b is a rectification of (33). In order for (33) to be valid, the side condition has to be internalized. Whether in Fusion Calculus a tau law like T3b is necessary or not depends on whether the fusion prefix is a primitive prefix or an induced prefix.
- In [33] two weak equivalences on fusion calculus were discussed. They are weak hyperequivalence and weak barbed equivalence. The authors claimed that the two equivalences were the same. It was shown in [6] that this claim is false by setting them in the framework of L-bisimilarities. The calculus studied in [6] is the

representative	classes of equal bisimilarities	corresponding system
$\approx_u$	$\{\approx_L \mid \emptyset \neq L \subseteq ba \cup u \cup \overline{ba}\}$	$AS \cup \{T1, T2, T5\}$
$\approx_{fa}$	$\{\approx_L \mid fa \subseteq L \subseteq fa \cup ba \cup u \cup \overline{ba}\}$	$AS \cup \{T1, T2, T6^-\}$
$\approx_{\overline{fa}}$	$\{\approx_L \mid \overline{fa} \subseteq L \subseteq \overline{fa} \cup ba \cup u \cup \overline{ba}\}$	$AS \cup \{T1, T2, T6^+\}$
$\approx_{fa \cup \overline{fa}}$	$\{\approx_L \mid fa \cup \overline{fa} \subseteq L \subseteq fa \cup \overline{fa} \cup ba \cup u \cup \overline{ba}\}$	$AS \cup \{T1, T2, T5, T6\}$

Fig. 8. Complete systems for L-congruences.

χ-calculus *without* the mismatch operator. This paper improves our understanding by taking a close look at the L-bisimilarities for the χ-calculus *with* the mismatch operator. Moreover we have provided complete systems for all the L-congruences in present case. The L-congruences and their complete systems are summarized in Fig. 8.

Many questions about $\chi^{\neq}$-calculus awaits to be answered. We mention some of them:

- The calculus of this paper lacks of an important operator, the recursion operator. We have ignored it because it does not have any impact on the algebraic theory discussed in this paper. But axiomatization of congruences on 'infinite processes' is feasible for finite state (finite control) processes. Research in this direction was pioneered by Milner [20,21] in the setting of CCS and was followed up by Lin in the symbolic framework for π-calculus [17,18]. Investigation of similar problems for χ-calculus will definitely improve our understanding of the language.
- The barbed *bisimilarity* studied in this paper is slightly different from the barbed *equivalence* studied in literature. It differs from that of the barbed bisimilarity in that the latter is closed under context in every bisimulation step whereas the former is closed under context only in the very beginning.

Definition 34. P and Q are barbed equivalent, notation $P \approx_b^e Q$, if for each full context $C[\_]$ there is some barbed bisimulation $\mathcal{R}$ such that $C[P]\mathcal{R}C[Q]$.

It is clear that $\approx_b$ is contained in $\approx_b^e$. The inclusion is strict as can be seen from the following example:

$$[x \neq y]\tau.(P + \tau.[x \neq y]\tau.(P + \tau)) \approx_b^e [x \neq y]\tau.(P + \tau)$$

but

$$[x \neq y]\tau.(P + \tau.[x \neq y]\tau.(P + \tau)) \not\approx_b [x \neq y]\tau.(P + \tau).$$

Similarly we can define ground equivalence.

Definition 35. P and Q are ground equivalent, notation $P \approx_g^e Q$, if for each full context $C[\_]$ there is some bisimulation $\mathcal{R}$ such that $C[P]\mathcal{R}C[Q]$.

The above pair of processes serve to distinguish $\approx_g^e$ from $\approx_g$. So the inclusion $\approx_g \subseteq \approx_g^e$ is strict.

It is clear that both $\approx_b^e$ and $\approx_g^e$ are congruence relations. We have not carried out any study on the completeness problem for these two congruences.

- More generally, one can introduce L-equivalences as follows:

Definition 36. Suppose $\mathscr{R}$ is a symmetric binary relation and $L \in \mathscr{L}$. It is called a ground L-bisimulation if whenever $P\mathscr{R}Q$ and $P \xrightarrow{\lambda} P'$ for $\lambda \in L \cup \{\tau\}$ then some Q' exists such that $Q \stackrel{\hat{\lambda}}{\Rightarrow} Q'\mathscr{R}P'$. The ground L-bisimilarity $\dot{\approx}_L$ is the largest ground L-bisimulation. Two processes P, Q are ground equivalent, notation $P \approx_L^e Q$, if $C[P] \dot{\approx}_L C[Q]$ for each full context $C[\,]$.

The order structure of the L-equivalences, the axiomatic systems of the L-equivalences and the relationship of L-equivalences to the barbed equivalence as well as the ground equivalence are all open problems. The only thing we know is that L-congruences are strictly contained in the L-equivalences.

Proposition 37. *For each L, the inclusion $\simeq_L \subseteq \approx_L^e$ is strict.*

The strictness is supported by the preceding example.

- For the χ-calculus without the mismatch operator the definition of L-bisimilarities can be slightly simplified as follows:

Definition 38. The relation $\mathscr{R}$ is an L-bisimulation if whenever $P\mathscr{R}Q$ then for any process R and any sequence $\vec{x}$ of names it holds that if $(\vec{x})(P|R) \xrightarrow{\phi} P'$ for $\phi \in L \cup \{\tau\}$ then there exists some Q' such that $(\vec{x})(Q|R) \stackrel{\hat{\phi}}{\Rightarrow} Q'$ and $P'\mathscr{R}Q'$. The L-bisimilarity $\approx_L$ is the largest L-bisimulation.

In the above definition closure under prefix operation is not required. In [6] it is proved that the L-bisimilarities defined in Definition 38 are closed under substitution and consequently is also closed under prefix operation. In the χ-calculus without the mismatch, the L-bisimilarities introduced in the above definition and those introduced by Definition 29 coincide. For $\chi^{\neq}$-calculus the relationship is not known.

Finally we take the opportunity to explain some of the design decisions we have made in this paper:

- Previous papers on χ-calculus have used square brackets for free actions, update actions, free prefix, update prefix, match and mismatch, and substitution. In this paper we liberate the square brackets from overloading, by using some standard notations. In two places we deviate from Fusion's notation. We use $\langle y|x\rangle$ for the update (fusion) prefix to make it more distinct from substitution. The symmetry of $\langle y|x\rangle$, and the use of '|', conveys the idea that the prefix can incur both a substitution of y for x and a substitution of x for y. The use of the notation y/x for update action in preference to the notation $\{x=y\}$ for fusion action is more technical. In algebraic theory, the update y/x is a lot nicer than the fusion $\{x=y\}$. Take for instance the definition of ground bisimulation for the polyadic calculus. An update action

$$P \stackrel{y_1/x_1,\ldots,y_n/x_n}{\longrightarrow} P'$$

is matched up by

$$Q \stackrel{y^1_1/x^1_1,\ldots,y^1_{n_1}/x^1_{n_1}}{\Longrightarrow} \ldots \stackrel{y^i_1/x^i_1,\ldots,y^i_{n_i}/x^i_{n_i}}{\Longrightarrow} Q'$$

SUCH THAT $\{y^1_1/x^1_1,\ldots,y^1_{n_1}/x^1_{n_1}\}\ldots\{y^i_1/x^i_1,\ldots,y^i_{n_i}/x^i_{n_i}\} = \{y_1/x_1,\ldots,y_n/x_n\}$. On the other hand, a fusion action

$$P \stackrel{\{y_1=x_1,\ldots,y_n=x_n\}}{\longrightarrow} P'$$

is matched up by the following fusions:

$$\begin{array}{lcl} Q & \Rightarrow\xrightarrow{\{y^1_1=x^1_1,\ldots,y^1_{n_1}=x^1_{n_1}\}} & Q_1 \\ Q'_1 & \Rightarrow\xrightarrow{\{y^2_1=x^2_1,\ldots,y^2_{n_2}=x^2_{n_2}\}} & Q_2 \\ & \vdots & \\ Q'_{i-2} & \Rightarrow\xrightarrow{\{y^{i-1}_1=x^{i-1}_1,\ldots,y^{i-1}_{n_{i-1}}=x^i_{n_i}\}} & Q_{i-1} \\ Q'_{i-1} & \Rightarrow\xrightarrow{\{y^i_1=x^i_1,\ldots,y^i_{n_i}=x^i_{n_i}\}} & Q' \end{array}$$

such that the following conditions are satisfied:

- There is some substitution σ_1 induced by $\{y^1_1=x^1_1,\ldots,y^1_{n_1}=x^1_{n_1}\}$ such that $Q_1\sigma_1 \equiv Q'_1$.
- There is some substitution σ_{i-1} induced by $\{y^{i-1}_1=x^{i-1}_1,\ldots,y^{i-1}_{n_{i-1}}=x^{i-1}_{n_{i-1}}\}$ such that $Q_{i-1}\sigma_{i-1} \equiv Q'_{i-1}$.
- The combined effect of $\{y^1_1=x^1_1,\ldots,y^1_{n_1}=x^1_{n_1}\}\ldots\{y^i_1=x^i_1,\ldots,y^i_{n_i}=x^i_{n_i}\}$ is the same as that of $\{y_1=x_1,\ldots,y_n=x_n\}$.

What all these say is that fusion actions should achieve the same effect as the update actions. An update retains the message of symmetry since if $P\stackrel{y/x}{\longrightarrow}P'$ then $P\stackrel{x/y}{\longrightarrow}P'\{x/y\}$.

- The operational semantics defined in this paper disallows updates like x/x. The alternative is to admit such updates and let the *observational* semantics to identify x/x with τ. We are not in favour of an *operational* identification of x/x to τ that uses the following rule:

$$\frac{P\stackrel{x/x}{\longrightarrow}P'}{P\stackrel{\tau}{\longrightarrow}P'}.$$

In our opinion an operational semantics should avoid having rules like

$$\frac{P\stackrel{\lambda}{\longrightarrow}P'}{P\stackrel{\lambda'}{\longrightarrow}P'}$$

which adds nothing to the semantics apart from proposing an alias for an action. Nor do we favour a *syntactical* identification of x/x with τ. The identification would make one wonder if τ contains a free name or not. Conceptually a tau indicates a communication that has been completed, whereas an update is a communication on its way.

- The tau prefix, the update prefixes and the bound prefixes are induced prefixes. This has the nice consequence that many laws about these prefix operators are derivable. T3b is one example. If the algebraic properties of an induced operator are all derivable from the properties of other operators, then it seems right to let it be an induced operator.

Acknowledgements

The authors thank the two referees for many constructive suggestions and comments.

References

[1] R. Amadio, I. Castellani, D. Sangiorgi, On Bisimulations for the Asynchronous π-Calculus, CONCUR '96, Lecture Notes in Computer Science, Vol. 1119, Springer, Berlin, 1996, pp. 146–162.
[2] G. Berry, G. Boudol, The chemical abstract machine, Theoret. Comput. Sci. 96 (1992) 217–248.
[3] M. Boreale, R. De Nicola, Testing equivalence for mobile processes, Inform. Comput. 120 (1995) 279–303.
[4] Y. Fu, The χ-Calculus. Proceedings of the International Conference on Advances in Parallel and Distributed Computing, IEEE Computer Society Press, Silver Spring MD, 1997, pp. 74–81.
[5] Y. Fu, A proof theoretical approach to communications, ICALP'97, Lecture Notes in Computer Science, Vol. 1256, Springer, Berlin, 1997, pp. 325–335.
[6] Y. Fu, Bisimulation lattice of chi processes, ASIAN'98, Lecture Notes in Computer Science, Vol. 1538, Springer, Berlin, 1998, pp. 245–262.
[7] Y. Fu, Variations on mobile processes, Theoret. Comput. Sci. 221 (1999) 327–368.
[8] Y. Fu, Open bisimulations of chi processes, CONCUR'99, Lecture Notes in Computer Science, Vol. 1664, Springer, Berlin, 1999, pp. 304–319.
[9] Y. Fu, An open problem of mobile processes, Unpublished Paper, 1999.
[10] Y. Fu, Axiomatization without prefix combinator, Proc. Internat. Symp. on Domain Theory '99, Semantics in Computation Series, Kluwer, Dordrecht, 2001.
[11] Y. Fu, Z. Yang, Chi calculus with mismatch, CONCUR 2000, Lecture Notes in Computer Science, Vol. 1877, Springer, Berlin, 2000, pp. 596–610.
[12] Y. Fu, Z. Yang, The ground congruence for chi calculus, FST& TCS 2000, Lecture Notes in Computer Science, Vol. 1974, Springer, Berlin, 2000, pp. 385–396.
[13] Y. Fu, Z. Yang, Tau laws for pi calculus, submitted for publication.
[14] K. Honda, M. Tokoro, An object calculus for asynchronous communication, ECOOP'91, Lecture Notes in Computer Science, Vol. 512, Springer, Berlin, 1991, pp. 133–147.
[15] K. Honda, M. Tokoro, On asynchronous communication semantics, object-based concurrent computing, Lecture Notes in Computer Science, Vol. 612, Springer, Berlin, 1991, pp. 21–51.
[16] K. Honda, N. Yoshida, On reduction based process semantics, Theoret. Comput. Sci. 152 (1995) 437–486.
[17] H. Lin, Unique fixpoint induction for mobile processes, CONCUR'95, Lecture Notes in Computer Science, Vol. 962, Springer, Berlin, 1995, pp. 88–102.
[18] H. Lin, Complete proof systems for observation congruences in finite-control π-calculus, ICALP'98, Lecture Notes in Computer Science, Vol. 1443, Springer, Berlin, 1998, pp. 443–454.

[19] M. Merro, D. Sangiorgi, On asynchrony in name-passing calculi, ICALP'98, Lecture Notes in Computer Science, Vol. 1443, Springer, Berlin, 1998, pp. 856–867.
[20] R. Milner, A complete inference system for a class of regular behaviours, J. Comput. System Sci. 28 (1984) 439–466.
[21] R. Milner, A complete axiomatization system for observational congruence of finite state behaviours, Inform. Comput. 81 (1989) 227–247.
[22] R. Milner, Communication and Concurrency, Prentice-Hall, Englewood Cliffs, NJ, 1989.
[23] R. Milner, Functions as processes, Math. Struct. Comput. Sci. 2 (1992) 119–146.
[24] R. Milner, J. Parrow, D. Walker, A calculus of mobile processes, Inform. Comput. 100 (1992) 1–40 (Part I), 41–77 (Part II).
[25] R. Milner, D. Sangiorgi, Barbed bisimulation, ICALP'92, Lecture Notes in Computer Science, Vol. 623, Springer, Berlin, 1992, pp. 685–695.
[26] J. Parrow, D. Sangiorgi, Algebraic theories for name-passing calculi, Inform. Comput. 120 (1995) 174–197.
[27] J. Parrow, B. Victor, The update calculus, AMAST '97, Lecture Notes in Computer Science, Vol. 1119, Springer, Berlin, 1997, pp. 389–405.
[28] J. Parrow, B. Victor, The fusion calculus: expressiveness and symmetry in mobile processes, Logics in Computer Science '98, IEEE Computer Society Press, Silver Spring, MD, 1998, pp. 176–185.
[29] J. Parrow, B. Victor, The tau-laws of fusion, CONCUR '98, Lecture Notes in Computer Science, Vol. 1466, Springer, Berlin, 1998, pp. 99–114.
[30] D. Sangiorgi, Expressing mobility in process algebras: first-order and higher-order paradigms, Ph.D. Thesis, Department of Computer Science, University of Edinburgh, 1993.
[31] D. Sangiorgi, A theory of bisimulation for π-calculus, Acta Informatica 3 (1996) 69–97.
[32] D. Sangiorgi, D. Walker, On Barbed Equivalence in π-Calculus, CONCUR'01, 2001.
[33] B. Victor, J. Parrow, Concurrent constraints in the fusion calculus, ICALP'98, Lecture Notes in Computer Science, Springer, Berlin, 1998.