Observing Asymmetry and Mismatch*

Xiaoju Dong and Yuxi Fu**

Department of Computer Science,
Shanghai Jiaotong University, Shanghai 200030, China
{dong-xj, fu-yx}@cs.sjtu.edu.cn

Abstract. The chi calculus is studied in the framework incorporating two constructions widely useful in applications: asymmetric communication and mismatch condition. The barbed bisimilarity is used to give a general picture of how the two constructions affect the observational theory. Both the operational properties and the algebraic properties of the enriched calculus are investigated to support an improved understanding of the bisimulation behaviors of the model.

1 Introduction

We study the asymmetric χ-calculus with mismatch. The background of this investigation is as follows:

- The χ-calculus was independently proposed by Fu in [2, 3] and by Parrow and Victor in [15]. Fu was motivated to simplify the π-calculus and to provide a proof theoretical interpretation of concurrent computation ([5]). Parrow and Victor shared the motivation on simplification. They were also trying to give a more direct interpretation of concurrent constraint programming ([17]). The model they proposed, called Fusion Calculus, is actually the polyadic χ-calculus. The χ-calculus is symmetric in the sense that the input and output prefixes are of the same form and that communications are symmetric. Theoretical investigations on Fusion Calculus are carried out in [16]. More fundamental studies on χ-calculus are done in [4, 6, 8]. The important results in [4, 6, 8] are the introduction of L-bisimilarities, the classification of L-bisimilarities using bisimulation lattice, and the characterizations of L-bisimilarities in terms of open style bisimilarities.
- Symmetry of communications is a beautiful thing to have in theory. In programming practice however there is a need for asymmetric communications. The symmetry introduces extra uncertainty. This nondeterministic feature

* *APLAS'03*, Lecture Notes in Computer Science 2895, 2-19, 2003.

** The author is supported by the National Distinguished Young Scientist Fund of NNSFC (60225012), the Shanghai Science and Technology Development Fund (025115032), 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.

is not always welcome by users. In reality we often come across with situations in which we favor a command that always acts as an output instruction in whatever environments it resides. In other words we really need a command set that draws a line between an input command and an output command. Therefore it is necessary to pay as much attention to the asymmetric χ-calculus as to the symmetric one. Parrow and Victor have studied the asymmetric χ-calculus, which they called Update Calculus, in [14]. A more systematic investigation into the asymmetry is reported in [7]. It turns out that the theory of the asymmetric χ-calculus is much richer than the theory of the symmetric χ-calculus. The barbed bisimilarity of the asymmetric χ-calculus for example reveals some aspects of process identity for the first time. Some of the equivalence properties are beyond our intuition at first sight. But they are very reasonable from an observational viewpoint.

– A familiar programming construct is the binary *if* B *then* C_1 *else* C_2 command. One could say that this command is behind all the 'intelligent behaviors' of computer programs. In calculi of mobile processes the command *if* B *then* C_1 *else* C_2 can be coded up by a process like $[x{=}y]P + [x{\neq}y]Q$. Here the mismatch plays a crucial role. Now the binary command *if* B *then* C_1 *else* C_2 could be simulated by two consecutive unary commands *if* B *then* C_1; *if* $\neg B$ *then* C_2. The negation operator appeared here implies that if we would translate the program *if* B *then* C_1; *if* $\neg B$ *then* C_2 into a process in some calculus of mobile processes we would have to use the mismatch. One could argue that the positive match operator $[x{=}y]$ suffices in a lot of cases. This is true at the expanse that programs so constructed are unnecessarily complex since a lot of coding-up is applied. It should be pointed out that these cases are restricted to the applications where computation flows depend on enumerations of finite numbers of names. Of course the calculus can be extended with other constructs like infinite sum. However that would give rise to a model more abstract and less real. Motivated by applications, the mismatch operator turns out to be very important in theory. Some of the results in process algebra are valid only in the presence of this operator. In the framework of the χ-calculus, the mismatch operator has been studied in [9–11]. It is demonstrated in these papers that the mismatch could be extremely subtle. Even the definitions of bisimulations call for extra care! The hyperequivalence studied in [16] for example is not even observational in the presence of mismatch. See [11] for counter examples.

In this paper we study the interplay between the asymmetry and the mismatch in the framework of the χ-calculus. Both features are motivated by applications. And both are significant in theory. Based upon previous work on the asymmetry and the mismatch, we will show, both operationally and algebraically, how one-step actions of the asymmetric χ-calculus with mismatch are simulated by observationally equivalent sequences of actions. We will achieve this by looking at the barbed bisimilarity. We omit the proofs and some intermediate lemmas. One could find a detailed account in the full paper [1]. In the rest of the paper we will refer to 'asymmetric χ-calculus with mismatch' as 'χ-calculus'.

2 Asymmetric Chi Calculus with Mismatch

Let $\mathcal{N}$ be the set of names ranged over by small case letters and $\overline{\mathcal{N}}$ the set $\{\overline{x} \mid x \in \mathcal{N}\}$ of co-names. The Greek letter α ranges over $\mathcal{N}\cdot\overline{\mathcal{N}}$, where the dot $\cdot$ denotes set union. For $\alpha \in \mathcal{N}\cdot\overline{\mathcal{N}}$, let $\overline{\alpha}$ be defined as a if $\alpha = \overline{a}$ and as $\overline{a}$ if $\alpha = a$. The χ-processes are defined by the following abstract grammar:

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

Let λ range over the set of transition labels $\{ax, \overline{a}x, a(x), \overline{a}(x), y/x, (y)/x \mid a, x, y \in \mathcal{N}\}\cdot\{\tau\}$ and γ over $\{ax, \overline{a}x, a(x), \overline{a}(x) \mid a, x \in \mathcal{N}\}\cdot\{\tau\}$. In $(y)/x$ and y/x, x and y must be different. The free and bound names are defined as usual. We use the notation $fn(\_)$, $bn(\_)$ and $n(\_)$ in their standard meaning. We write $P\{y/x\}$ for the result of substituting y for x throughout P. We will assume that a bound name is distinct from any other name in all environments.

In the following labelled transition system, symmetric rules have been systematically omitted.

Sequentialization

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

Composition

$$\frac{P \xrightarrow{\gamma} P'}{P \mid Q \xrightarrow{\gamma} P' \mid Q} \qquad \frac{P \xrightarrow{y/x} P'}{P \mid Q \xrightarrow{y/x} P' \mid Q\{y/x\}} \qquad \frac{P \xrightarrow{(y)/x} P'}{P \mid Q \xrightarrow{(y)/x} P' \mid Q\{y/x\}}$$

Communication

$$\frac{P \xrightarrow{a(x)} P' \quad Q \xrightarrow{\overline{a}y} Q'}{P \mid Q \xrightarrow{\tau} P'\{y/x\} \mid Q'} \qquad \frac{P \xrightarrow{a(x)} P' \quad Q \xrightarrow{\overline{a}(x)} Q'}{P \mid Q \xrightarrow{\tau} (x)(P' \mid Q')} \qquad \frac{P \xrightarrow{ax} P' \quad Q \xrightarrow{\overline{a}x} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}$$

$$\frac{P \xrightarrow{ax} P' \quad Q \xrightarrow{\overline{a}y} Q' \quad x \neq y}{P \mid Q \xrightarrow{y/x} P'\{y/x\} \mid Q'\{y/x\}} \qquad \frac{P \xrightarrow{ax} P' \quad Q \xrightarrow{\overline{a}(y)} Q'}{P \mid Q \xrightarrow{(y)/x} P'\{y/x\} \mid Q'\{y/x\}}$$

Localization

$$\frac{P \xrightarrow{\lambda} P' \quad x \notin n(\lambda)}{(x)P \xrightarrow{\lambda} (x)P'} \qquad \frac{P \xrightarrow{\alpha x} P' \quad x \notin \{\alpha, \overline{\alpha}\}}{(x)P \xrightarrow{\alpha(x)} P'}$$

$$\frac{P \xrightarrow{y/x} P'}{(x)P \xrightarrow{\tau} P'} \qquad \frac{P \xrightarrow{y/x} P'}{(y)P \xrightarrow{(y)/x} P'} \qquad \frac{P \xrightarrow{(y)/x} P'}{(x)P \xrightarrow{\tau} (y)P'}$$

Condition and Selection

$$\frac{P \xrightarrow{\lambda} P'}{[x{=}x]P \xrightarrow{\lambda} P'} \qquad \frac{P \xrightarrow{\lambda} P'}{[x{\neq}y]P \xrightarrow{\lambda} P'} \qquad \frac{P \xrightarrow{\lambda} P'}{P{+}Q \xrightarrow{\lambda} P'}$$

Here are some examples of communication admissible by the operational semantics:

$$(x)(R \mid (\overline{a}y.P \mid ax.Q)) \xrightarrow{\tau} R\{y/x\} \mid (P\{y/x\} \mid Q\{y/x\}), \text{ where } y \neq x$$
$$(x)\overline{a}x.P \mid (y)ay.Q \xrightarrow{\tau} (z)(P\{z/x\} \mid Q\{z/y\}), \text{ where } z \text{ is fresh}$$

Communications of this language are asymmetric because one has

$$\overline{a}y.P \mid (x)ax.Q \xrightarrow{\tau} P \mid Q\{y/x\}$$

but not

$$(y)\overline{a}y.P \mid ax.Q \xrightarrow{\tau} P\{x/y\} \mid Q$$

For more about the operational semantics, see [7–11].

In addition to the prefix αx, we need some auxiliary prefixes:

$$\begin{array}{rll} \alpha(x).P & \stackrel{\text{def}}{=} & (x)\alpha x.P, \qquad \text{bound prefix} \\ \langle y/x\rangle.P & \stackrel{\text{def}}{=} & (a)(\overline{a}y \mid ax.P), \qquad \text{update prefix} \\ (y/x\rangle.P & \stackrel{\text{def}}{=} & (y)\langle y/x\rangle.P, \qquad \text{bound update prefix} \\ \tau.P & \stackrel{\text{def}}{=} & (a)\langle a/a\rangle.P, \qquad \text{tau prefix} \end{array}$$

where $x \notin \{\alpha, \overline{\alpha}\}$, $x \neq y$ and a is fresh. The set of all prefixes will also be ranged over by λ.

We will write ϕ and ψ, called conditions, to stand for sequences of match and mismatch combinators concatenated one after another, and δ for a sequence of mismatch operators. The notation $\phi \Rightarrow \psi$ says that ϕ logically implies ψ and $\phi \Leftrightarrow \psi$ that ϕ and ψ are logically equivalent. The concatenation of ψ and ϕ is denoted by $\psi\phi$.

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 $n(\sigma) \subseteq n(\psi)$.

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

A sequence of names $x_1, \ldots, x_n$ will be abbreviated as $\boldsymbol{x}$. Suppose Y is a finite set $\{y_1, \ldots, y_n\}$ of names. The notation $[y{\notin}Y]P$ will stand for $[y{\neq}y_1] \ldots [y{\neq}y_n]P$.

Let $\Longrightarrow$ be the reflexive and transitive closure of $\xrightarrow{\tau}$. We will write $\stackrel{\lambda}{\Longrightarrow}$ for $\Longrightarrow\xrightarrow{\lambda}\Longrightarrow$. We will also write $\stackrel{\widehat{\lambda}}{\Longrightarrow}$ for $\stackrel{\lambda}{\Longrightarrow}$ if $\lambda \neq \tau$ and for $\Longrightarrow$ otherwise.

In the rest of this paper we often come across with situations where we have to deal with a sequence of actions. The principal case is this:

$$P \stackrel{(y_1)}{\Longrightarrow}\!\!\!/\stackrel{x(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}}{\Longrightarrow} P'$$

When $n = 0$ we shall understand that the above sequence denotes $P \Longrightarrow P'$ and that y_n denotes x.

3 Barbed Bisimulation

It has now become a routine to look at the barbed bisimilarity [13] when one investigates a new process calculus. For a channel based process calculus, the barbed approach often gives rise to the weakest bisimulation equivalence for the calculus.

Definition 1. *A process P is strongly barbed at a, notation $P \downarrow a$, if $P \xrightarrow{\alpha(x)} P'$ or $P \xrightarrow{\alpha x} P'$ for some P' such that $a \in \{\alpha, \overline{\alpha}\}$. P is barbed at a, written $P \Downarrow a$, if some P' exists such that $P \Longrightarrow P' \downarrow a$. A binary relation $\mathcal{R}$ is barbed if $\forall a \in \mathcal{N}$, $P \Downarrow a \Leftrightarrow Q \Downarrow a$ whenever $P\mathcal{R}Q$.*

The notation of context is also important to the semantic investigation. 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 satisfy additionally: (iii) if C[] is a context then $C[] + P$, $P + C[]$ and $[x{\neq}y]C[]$ are contexts. We are now ready to define barbed bisimulation.

Definition 2. *Let $\mathcal{R}$ be a barbed symmetric relation closed under context. The relation $\mathcal{R}$ is a barbed bisimulation if whenever $Q\mathcal{R}P \xrightarrow{\tau} P'$ then $Q \Longrightarrow Q'\mathcal{R}P'$ for some Q'. The barbed bisimilarity $\approx_b$ is the largest barbed bisimulation.*

In the presence of the asymmetry and the mismatch operator, the barbed bisimilarity is extremely complex. Let's take a look at three simple examples:

- The first is concerned with (bound) updates. The process $(z/x\rangle.\langle y/z\rangle.Q$ is barbed bisimilar to $(z/x\rangle.\langle y/z\rangle.Q + [x{\neq}y]\langle y/x\rangle.Q\{x/z\}$. For either process to have any effect on an environment the name x must be bound, in which case both processes would have the same effect on the environment. The following internal communication for instance

  $$(x)(((z/x\rangle.\langle y/z\rangle.Q + [x{\neq}y]\langle y/x\rangle.Q\{x/z\}) \mid P) \xrightarrow{\tau} Q\{x/z\}\{y/x\} \mid P\{y/x\}$$

  is simulated by two consecutive communications

  $$\begin{aligned}(x)((z/x\rangle.\langle y/z\rangle.Q \mid P) &\xrightarrow{\tau} (z)(\langle y/z\rangle.Q\{z/x\} \mid P\{z/x\}) \\ &\xrightarrow{\tau} Q\{z/x\}\{y/z\} \mid P\{z/x\}\{y/z\}\end{aligned}$$

  Notice that in order for the component $[x{\neq}y]\langle y/x\rangle.Q\{x/z\}$ to invoke a communication it must be placed in an environment in which x is localized.
- The second is about free input actions. One could argue that

  $$ax.[x{\neq}y]\tau.Q\{x/z\} + (z/x\rangle.ay.\langle y/z\rangle.Q + [x{\neq}y]ax.Q\{x/z\}$$

  is barbed bisimilar to $ax.[x{\neq}y]\tau.Q\{x/z\} + (z/x\rangle.ay.\langle y/z\rangle.Q$. There are two main situations in which the component $[x{\neq}y]ax.Q\{x/z\}$ may participate in an internal communication. If x is instantiated by some name different from y then the component $ax.[x{\neq}y]\tau.Q\{x/z\}$ can simulate the action since the mismatch $x{\neq}y$ would be valid as long as x is replaced by some name that is not y. Otherwise the component $(z/x\rangle.ay.\langle y/z\rangle.Q$ does the job.

– The third is about bound output actions. The process

$$\overline{a}(z).[z{\neq}y](x/z\rangle.Q + (z/y\rangle.\overline{a}z.(x/z\rangle.Q + \overline{a}(x).Q\{x/z\}$$

is barbed bisimilar to $\overline{a}(z).[z{\neq}y](x/z\rangle.Q + (z/y\rangle.\overline{a}z.(x/z\rangle.Q$. For instance

$$\begin{aligned}&(y)(ay.P \mid (\overline{a}(z).[z{\neq}y](x/z\rangle.Q + (z/y\rangle.\overline{a}z.(x/z\rangle.Q + \overline{a}(x).Q\{x/z\}))\\ \xrightarrow{\tau}\ &(x)(P\{x/y\} \mid Q\{x/z\}\{x/y\})\end{aligned}$$

is simulated by

$$\begin{aligned}&(y)(ay.P \mid (\overline{a}(z).[z{\neq}y](x/z\rangle.Q + (z/y\rangle.\overline{a}z.(x/z\rangle.Q))\\ \xrightarrow{\tau}\ &(z)(az.P\{z/y\} \mid \overline{a}z.(x/z\rangle.Q\{z/y\})\\ \xrightarrow{\tau}\ &(z)(P\{z/y\} \mid (x/z\rangle.Q\{z/y\})\\ \xrightarrow{\tau}\ &(x)(P\{z/y\}\{x/z\} \mid Q\{z/y\}\{x/z\})\end{aligned}$$

It should be obvious from the above examples that, from the viewpoint of the barbed bisimilarity, a non-tau action of a process could be simulated by several sequences of non-tau actions of a bisimilar process. An action could be simulated by different sequences of actions in different environments. Different actions are simulated by different sequences of actions. And these sequences of actions are of different shapes.

A more interesting aspect of the barbed bisimilarity is to figure out in what manners a given action could be simulated. Let's see an example. Suppose $P \approx_b Q$ and $P \xrightarrow{ax} P'$. For the barbed bisimilarity the free input action is not directly observable. What an observer can do is to observe the consequences of the action by putting P in an environment. By doing that the observer gets to know the effects of the action on the environment. Therefore in order to see how Q might simulate P we need to put them in the same environments and analyze the operational behaviours of Q. There are two major cases:

– Case one:
  - Consider first the context $\_ \mid \bar{a}x.b(u)$ for fresh b, u. Then $P \mid \bar{a}x.b(u) \xrightarrow{\tau} P' \mid b(u) \xrightarrow{b(u)} P' \mid \mathbf{0}$. Suppose it is simulated by $Q \mid \bar{a}x.b(u) \stackrel{b(u)}{\Longrightarrow} Q' \mid \mathbf{0} \approx_b P' \mid \mathbf{0}$. This sequence of actions can be factorized in two manners: Either

$$Q \mid \bar{a}x.b(u) \Longrightarrow Q_1 \mid \bar{a}x.b(u) \xrightarrow{\tau} Q_2 \mid b(u) \xrightarrow{b(u)} Q_2 \mid \mathbf{0} \Longrightarrow Q' \mid \mathbf{0}$$

    or

$$Q \mid \bar{a}x.b(u) \Longrightarrow Q_1 \mid \bar{a}x.b(u) \xrightarrow{\tau} Q''\{x/z\} \mid b(u) \xrightarrow{b(u)} Q''\{x/z\} \mid \mathbf{0} \Longrightarrow Q' \mid \mathbf{0}$$

    In the first case, $Q \stackrel{ax}{\Longrightarrow} Q' \approx_b P'$. In the second case, $Q \Longrightarrow \xrightarrow{a(z)} Q''$ and $Q''\{x/z\} \Longrightarrow Q' \approx_b P'$.

- Consider now the context $(x)(\_ \mid \bar{a}z.\bar{b}x)$ for fresh b, z. Then

$$(x)(P \mid \bar{a}z.\bar{b}x) \xrightarrow{\tau} P'\{z/x\} \mid \bar{b}z \xrightarrow{\bar{b}z} P'\{z/x\} \mid \mathbf{0}$$

This sequence of actions can be matched up by $(x)(Q \mid \bar{a}z.\bar{b}x) \stackrel{\bar{b}z}{\Longrightarrow} Q' \mid \mathbf{0}$, which can be factorized in two ways: Either

$$\begin{aligned}(x)(Q \mid \bar{a}z.\bar{b}x) &\Longrightarrow (y_n)(Q_1 \mid \bar{a}z.\bar{b}y_n)\\ &\xrightarrow{\tau} Q_2 \mid \bar{b}z\\ &\xrightarrow{\bar{b}z} Q_2 \mid \mathbf{0}\\ &\Longrightarrow Q' \mid \mathbf{0}\end{aligned}$$

or

$$\begin{aligned}(x)(Q \mid \bar{a}z.\bar{b}x) &\Longrightarrow (y_n)(Q_1 \mid \bar{a}z.\bar{b}y_n)\\ &\Longrightarrow (y_n)(Q_2 \mid \bar{b}y_n)\\ &\Longrightarrow Q_3 \mid \bar{b}z\\ &\xrightarrow{\bar{b}z} Q_3 \mid \mathbf{0}\\ &\Longrightarrow Q' \mid \mathbf{0}\end{aligned}$$

In the former case

$$Q \stackrel{(y_1)/x(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}}{\Longrightarrow} \xrightarrow{ay_n} Q_2\{y_n/z\} \Longrightarrow Q'\{y_n/z\}$$

and $Q'\{y_n/z\}\{x/y_n\} \approx_b P'$ for some $y_1, \ldots, y_n$, where $n \geq 0$. In the latter case

$$Q \stackrel{(y_1)/x(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}}{\Longrightarrow} \stackrel{a(z)}{\Longrightarrow} \stackrel{(z_1)/y_n(z_2)/z_1}{\Longrightarrow} \cdots \stackrel{(z_m)/z_{m-1}}{\Longrightarrow} \stackrel{z/z_m}{\Longrightarrow} Q'$$

and $Q'\{x/z\} \approx_b P'$ for some $z, y_1, \ldots, y_n, z_1, \ldots, z_m$, where $n \geq 0$ and $m \geq 0$.

– Case two: For each y distinct from x one has

$$(x)(P \mid \bar{a}y.[x{=}y]b(u)) \xrightarrow{\tau} P'\{y/x\} \mid [y{=}y]b(u) \xrightarrow{b(u)} P'\{y/x\} \mid \mathbf{0}$$

for fresh names b and u. It follows from $P \approx_b Q$ that $(x)(Q \mid \bar{a}y.[x{=}y]b(u)) \stackrel{b(u)}{\Longrightarrow} Q' \mid \mathbf{0} \approx_b P'\{y/x\} \mid \mathbf{0}$, which can be factorized in five different fashions:

- The first factorization is

$$\begin{aligned}(x)(Q \mid \bar{a}y.[x{=}y]b(u)) &\Longrightarrow (y_n)(Q_1 \mid \bar{a}y.[y_n{=}y]b(u))\\ &\xrightarrow{\tau} Q_2\{y/y_n\} \mid [y{=}y]b(u)\\ &\xrightarrow{b(u)} Q_2\{y/y_n\} \mid \mathbf{0}\\ &\Longrightarrow Q' \mid \mathbf{0}\end{aligned}$$

Obviously

  - $Q \stackrel{(y_1)/x}{\Longrightarrow} \stackrel{(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}}{\Longrightarrow} Q_1$ for some $y_1, \ldots, y_n$, where $n \geq 0$;
  - $Q_1 \stackrel{ay_n}{\longrightarrow} Q_2$;
  - $Q_2\{y/y_n\} \Longrightarrow Q'$.
- The second factorization is

$$\begin{aligned}(x)(Q \mid \bar{a}y.[x{=}y]b(u)) &\Longrightarrow (y_n)(Q_1 \mid \bar{a}y.[y_n{=}y]b(u)) \\ &\stackrel{\tau}{\longrightarrow} (y_n)(Q_2\{y/z\} \mid [y_n{=}y]b(u)) \\ &\Longrightarrow Q_3 \mid [y{=}y]b(u) \\ &\stackrel{b(u)}{\longrightarrow} Q_3 \mid \mathbf{0} \\ &\Longrightarrow Q' \mid \mathbf{0}\end{aligned}$$

  Then
  - $Q \stackrel{(y_1)/x}{\Longrightarrow} \stackrel{(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}}{\Longrightarrow} Q_1$ for some $y_1, \ldots, y_n$, where $n \geq 0$;
  - $Q_1 \stackrel{a(z)}{\longrightarrow} Q_2$ for some z;
  - $Q_2\{y/z\} \stackrel{(z_1)/y_n}{\Longrightarrow} \stackrel{(z_2)/z_1}{\Longrightarrow} \cdots \stackrel{(z_m)/z_{m-1}}{\Longrightarrow} \stackrel{y/z_m}{\Longrightarrow} Q_3$ for some $z_1, \ldots, z_m$, where $m \geq 0$;
  - $Q_3 \Longrightarrow Q'$.
- The third factorization is

$$\begin{aligned}(x)(Q \mid \bar{a}y.[x{=}y]b(u)) &\Longrightarrow (y_n)(Q_1 \mid \bar{a}y.[y_n{=}y]b(u)) \\ &\stackrel{\tau}{\longrightarrow} (y_n)(Q_2 \mid [y_n{=}y]b(u)) \\ &\Longrightarrow Q_3 \mid [y{=}y]b(u) \\ &\stackrel{b(u)}{\longrightarrow} Q_3 \mid \mathbf{0} \\ &\Longrightarrow Q' \mid \mathbf{0}\end{aligned}$$

  Clearly
  - $Q \stackrel{(y_1)/x}{\Longrightarrow} \stackrel{(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}}{\Longrightarrow} Q_1$ for some $y_1, \ldots, y_n$, where $n \geq 0$;
  - $Q_1 \stackrel{ay}{\longrightarrow} Q_2$;
  - $Q_2 \stackrel{(z_1)/y_n}{\Longrightarrow} \stackrel{(z_2)/z_1}{\Longrightarrow} \cdots \stackrel{(z_m)/z_{m-1}}{\Longrightarrow} \stackrel{y/z_m}{\Longrightarrow} Q_3$ for some $z_1, \ldots, z_m$, where $m \geq 0$;
  - $Q_3 \Longrightarrow Q'$.
- The fourth factorization is

$$\begin{aligned}(x)(Q \mid \bar{a}y.[x{=}y]b(u)) &\Longrightarrow Q_1 \mid \bar{a}y.[y{=}y]b(u) \\ &\stackrel{\tau}{\longrightarrow} Q_2 \mid [y{=}y]b(u) \\ &\stackrel{b(u)}{\longrightarrow} Q_2 \mid \mathbf{0} \\ &\Longrightarrow Q' \mid \mathbf{0}\end{aligned}$$

  Then
  - $Q \stackrel{(y_1)/x}{\Longrightarrow} \stackrel{(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}}{\Longrightarrow} \stackrel{y/y_n}{\Longrightarrow} Q_1$ for some $y_1, \ldots, y_n$, where $n \geq 0$;
  - $Q_1 \stackrel{ay}{\longrightarrow} Q_2$;

$*$ $Q_2 \Longrightarrow Q'$.

- The fifth factorization is

$$\begin{aligned}(x)(Q \mid \bar{a}y.[x{=}y]b(u)) &\Longrightarrow Q_1 \mid \bar{a}y.[y{=}y]b(u)\\ &\xrightarrow{\tau} Q_2\{y/z\} \mid [y{=}y]b(u)\\ &\xrightarrow{b(u)} Q_2\{y/z\} \mid \mathbf{0}\\ &\Longrightarrow Q' \mid \mathbf{0}\end{aligned}$$

It follows that

$*$ $Q \overset{(y_1)/x(y_2)/y_1}{\Longrightarrow\Longrightarrow} \cdots \overset{(y_n)/y_{n-1}y/y_n}{\Longrightarrow\Longrightarrow} Q_1$ for some $y_1, \ldots, y_n$, where $n \geq 0$;

$*$ $Q_1 \xrightarrow{a(z)} Q_2$ for some z;

$*$ $Q_2\{y/z\} \Longrightarrow Q'$

The above analysis suffices to show that the barbed bisimilarity is very tricky. In order to obtain a full picture, one needs to make sure that the barbed bisimilar processes are compared in *all* possible environments with consideration to *all* possible operational behaviours.

In the remaining part of the section we will establish six lemmas, each of which describes the simulation properties for one kind of actions, bearing in mind that there are altogether six kinds of non-tau actions in the χ-calculus.

Lemma 3. *Suppose $P \approx_b Q$ and $P \xrightarrow{a(x)} P'$. Then $Q \overset{a(x)}{\Longrightarrow} Q' \approx_b P'$ for some Q' and, for each y distinct from x, at least one of the following properties holds:*

1. *$Q \Longrightarrow \xrightarrow{a(z)} Q''$ and $Q''\{y/z\} \Longrightarrow Q' \approx_b P'\{y/x\}$ for some Q', Q'', z;*
2. *$Q \overset{ay}{\Longrightarrow} Q' \approx_b P'\{y/x\}$ for some Q'.*

Lemma 4. *Suppose $P \approx_b Q$ and $P \xrightarrow{ax} P'$. Then either $Q \overset{ax}{\Longrightarrow} Q'$ for some Q' or both the following properties hold:*

1. *$Q \Longrightarrow \xrightarrow{a(z)} Q''$ and $Q''\{x/z\} \Longrightarrow Q' \approx_b P'$ for some Q', Q'', z;*
2. *at least one of the following properties holds:*
   - *$Q \overset{(y_1)/x(y_2)/y_1}{\Longrightarrow\Longrightarrow} \cdots \overset{(y_n)/y_{n-1}ay_n}{\Longrightarrow\Longrightarrow} Q'$ and $Q'\{x/y_n\} \approx_b P'$ for some Q', $y_1, \ldots, y_n$, where $n \geq 1$;*
   - *$Q \overset{(y_1)/x(y_2)/y_1}{\Longrightarrow\Longrightarrow} \cdots \overset{(y_n)/y_{n-1}a(z)(z_1)/y_n(z_2)/z_1}{\Longrightarrow\Longrightarrow\Longrightarrow\Longrightarrow} \cdots \overset{(z_m)/z_{m-1}z/z_m}{\Longrightarrow\Longrightarrow} Q'$ and $Q'\{x/z\} \approx_b P'$ for some Q', z, $y_1, \ldots, y_n, z_1, \ldots, z_m$, where $n \geq 0$ and $m \geq 0$.*

And, for each y distinct from x, at least one of the following properties holds:

1. *$Q \overset{(y_1)/x(y_2)/y_1}{\Longrightarrow\Longrightarrow} \cdots \overset{(y_n)/y_{n-1}ay_n}{\Longrightarrow\longrightarrow} Q''$ and $Q''\{y/y_n\} \Longrightarrow Q' \approx_b P'\{y/x\}$ for some Q', Q'', $y_1, \ldots, y_n$, where $n \geq 0$;*
2. *$Q \overset{(y_1)/x(y_2)/y_1}{\Longrightarrow\Longrightarrow} \cdots \overset{(y_n)/y_{n-1}a(z)}{\Longrightarrow\longrightarrow} Q''$ and $Q''\{y/z\} \overset{(z_1)/y_n(z_2)/z_1}{\Longrightarrow\Longrightarrow} \cdots \overset{(z_m)/z_{m-1}y/z_m}{\Longrightarrow\Longrightarrow} Q' \approx_b P'\{y/x\}$ for some Q', Q'', $z, y_1, \ldots, y_n, z_1, \ldots, z_m$, where $n \geq 0$ and $m \geq 0$;*

3. $Q \stackrel{(y_1)/x(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}}{\Longrightarrow} \stackrel{ay}{\Longrightarrow} \stackrel{(z_1)/y_n(z_2)/z_1}{\Longrightarrow} \cdots \stackrel{(z_m)/z_{m-1}}{\Longrightarrow} \stackrel{y/z_m}{\Longrightarrow} Q' \approx_b P'\{y/x\}$ *for some* Q', $y_1, \ldots, y_n$, $z_1, \ldots, z_m$, *where* $n \geq 0$ *and* $m \geq 0$;
4. $Q \stackrel{(y_1)/x(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}}{\Longrightarrow} \stackrel{y/y_n}{\Longrightarrow} \stackrel{ay}{\Longrightarrow} Q' \approx_b P'\{y/x\}$ *for some* Q', $y_1, \ldots, y_n$, *where* $n \geq 0$;
5. $Q \stackrel{(y_1)/x(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}}{\Longrightarrow} \stackrel{y/y_n}{\Longrightarrow} \stackrel{a(z)}{\longrightarrow} Q''$ *and* $Q''\{y/z\} \Longrightarrow Q' \approx_b P'\{y/x\}$ *for some* Q', Q'', z, y_1, ..., y_n, *where* $n \geq 0$.

Lemma 5. *Suppose* $P \approx_b Q$ *and* $P \stackrel{\bar{a}(x)}{\longrightarrow} P'$. *Then* $Q \stackrel{\bar{a}(x)}{\Longrightarrow} \stackrel{(y_1)/x(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}}{\Longrightarrow} Q'$ *and* $Q'\{x/y_n\} \approx_b P'$ *for some* Q', $y_1, \ldots, y_n$, *where* $n \geq 0$. *And, for each* y *distinct from* x, *at least one of the following properties holds:*

1. $Q \stackrel{(y_1)/y(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}}{\Longrightarrow} \stackrel{\bar{a}(z)}{\longrightarrow} Q''$ *and* $Q''\{z/y_n\} \stackrel{(z_1)/z(z_2)/z_1}{\Longrightarrow} \cdots \stackrel{(z_m)/z_{m-1}}{\Longrightarrow} Q'$ *and* $Q'\{x/z_m\} \approx_b P'\{x/y\}$ *for some* Q', Q'', z, $y_1, \ldots, y_n$, $z_1, \ldots, z_m$, *where* $n \geq 0$ *and* $m \geq 0$;
2. $Q \stackrel{(y_1)/y(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}}{\Longrightarrow} \stackrel{\bar{a}y_n}{\Longrightarrow} \stackrel{(z_1)/y_n(z_2)/z_1}{\Longrightarrow} \cdots \stackrel{(z_m)/z_{m-1}}{\Longrightarrow} Q'$ *and* $Q'\{x/z_m\} \approx_b P'\{x/y\}$ *for some* Q', $y_1, \ldots, y_n$, $z_1, \ldots, z_m$, *where* $n \geq 0$ *and* $m \geq 0$.

Lemma 6. *Suppose* $P \approx_b Q$ *and* $P \stackrel{\bar{a}x}{\longrightarrow} P'$. *Then* $Q \stackrel{\bar{a}x}{\Longrightarrow} Q' \approx_b P'$ *for some* Q' *and, for each* y *distinct from* x, *at least one of the following properties holds:*

1. $Q \stackrel{(y_1)/y(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}}{\Longrightarrow} \stackrel{x/y_n}{\Longrightarrow} \stackrel{\bar{a}x}{\Longrightarrow} Q' \approx_b P'\{x/y\}$ *for some* Q', $y_1, \ldots, y_n$, *where* $n \geq 0$;
2. $Q \stackrel{(y_1)/y(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}}{\Longrightarrow} \stackrel{\bar{a}x}{\longrightarrow} Q''$ *and* $Q''\{x/y_n\} \Longrightarrow Q' \approx_b P'\{x/y\}$ *for some* Q', Q'', $y_1, \ldots, y_n$, *where* $n \geq 0$;
3. $Q \stackrel{(y_1)/y(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}}{\Longrightarrow} \stackrel{\bar{a}y_n}{\Longrightarrow} \stackrel{(z_1)/y_n(z_2)/z_1}{\Longrightarrow} \cdots \stackrel{(z_m)/z_{m-1}}{\Longrightarrow} \stackrel{x/z_m}{\Longrightarrow} Q' \approx_b P'\{x/y\}$ *for some* Q', $y_1, \ldots, y_n, z_1 \ldots, z_m$, *where* $n \geq 0$ *and* $m \geq 0$;
4. $Q \stackrel{(y_1)/y(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}}{\Longrightarrow} \stackrel{\bar{a}(z)}{\longrightarrow} Q''$ *and* $Q''\{z/y_n\} \stackrel{(z_1)/z(z_2)/z_1}{\Longrightarrow} \cdots \stackrel{(z_m)/z_{m-1}}{\Longrightarrow} \stackrel{x/z_m}{\Longrightarrow} Q' \approx_b P'\{x/y\}$ *for some* Q', Q'', $z, y_1, \ldots, y_n, z_1, \ldots, z_m$, *where* $n \geq 0$ *and* $m \geq 0$.

Lemma 7. *Suppose* $P \approx_b Q$ *and* $P \stackrel{(y)/x}{\longrightarrow} P'$. *Then* $Q \stackrel{(y_1)/x(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}}{\Longrightarrow} Q'$ *and* $Q'\{y/y_n\} \approx_b P'$ *for some* Q', $y_1, \ldots, y_n$, *where* $n \geq 0$.

Lemma 8. *Suppose* $P \approx_b Q$ *and* $P \stackrel{y/x}{\longrightarrow} P'$. *Then* $Q \stackrel{(y_1)/x(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}}{\Longrightarrow} \stackrel{y/y_n}{\Longrightarrow} Q' \approx_b P'$ *for some* Q', $y_1, \ldots, y_n$, *where* $n \geq 0$.

So there is a trade off for the simplicity of the definition of the barbed bisimilarity: The dependency of $\approx_b$ on contexts makes it difficult to understand. What the above lemmas achieve is to establish some context-free properties of the internal behaviors of the barbed bisimulation. The question of to what extent these context free properties shape the behaviors of $\approx_b$ is to be answered in the next section.

4 Open Barbed Bisimulation

The purpose of this section is to give a context free characterization of the barbed bisimilarity. We now make use of Lemma 3 through Lemma 8 to give alternative characterizations of $\approx_b$. The results established in these lemmas are turned into defining properties in the following definition.

Definition 9. *Let $\mathcal{R}$ be a binary symmetric relation closed under substitution. It is called an open barbed bisimulation if the followings hold whenever $P\mathcal{R}Q$:*

1. *If $P \xrightarrow{\tau} P'$ then $Q \Longrightarrow Q'\mathcal{R}P'$ for some Q'.*
2. *If $P \xrightarrow{a(x)} P'$ then $Q \stackrel{a(x)}{\Longrightarrow} Q'\mathcal{R}P'$ for some Q' and, for each y distinct from x, at least one of the following properties holds:*
   - *$Q \Longrightarrow \xrightarrow{a(z)} Q''$ and $Q''\{y/z\} \Longrightarrow Q'\mathcal{R}P'\{y/x\}$ for some Q', Q'', z;*
   - *$Q \stackrel{ay}{\Longrightarrow} Q'\mathcal{R}P'\{y/x\}$ for some Q'.*
3. *If $P \xrightarrow{ax} P'$ then either $Q \stackrel{ax}{\Longrightarrow} Q'\mathcal{R}P'$ for some Q' or both the following properties hold:*
   - *$Q \Longrightarrow \xrightarrow{a(z)} Q''$ and $Q''\{x/z\} \Longrightarrow Q'\mathcal{R}P'$ for some Q', Q'', z.*
   - *at least one of the following properties holds:*
     - *$Q \stackrel{(y_1)/x(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}\,ay_n}{\Longrightarrow} Q'$ and $Q'\{x/y_n\}\mathcal{R}P'$ for some Q', $y_1, \ldots, y_n$, where $n \geq 1$;*
     - *$Q \stackrel{(y_1)/x(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}\,a(z)(z_1)/y_n(z_2)/z_1}{\Longrightarrow} \cdots \stackrel{(z_m)/z_{m-1}\,z/z_m}{\Longrightarrow} Q'$ and $Q'\{x/z\}\mathcal{R}P'$ for some Q', z, $y_1, \ldots, y_n, z_1, \ldots, z_m$, where $n \geq 0$ and $m \geq 0$.*

   And, for each y distinct from x, at least one of the following properties holds:
   - *$Q \stackrel{(y_1)/x(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}\,ay_n}{\longrightarrow} Q''$ and $Q''\{y/y_n\} \Longrightarrow Q'\mathcal{R}P'\{y/x\}$ for some Q', Q'', $y_1, \ldots, y_n$, where $n \geq 0$;*
   - *$Q \stackrel{(y_1)/x(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}\,a(z)}{\longrightarrow} Q''$ and $Q''\{y/z\} \stackrel{(z_1)/y_n(z_2)/z_1}{\Longrightarrow} \cdots \stackrel{(z_m)/z_{m-1}\,y/z_m}{\Longrightarrow} Q'\mathcal{R}P'\{y/x\}$ for some Q', Q'', $z, y_1, \ldots, y_n, z_1, \ldots, z_m$, where $n \geq 0$ and $m \geq 0$;*
   - *$Q \stackrel{(y_1)/x(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}\,ay}{\Longrightarrow} \stackrel{(z_1)/y_n(z_2)/z_1}{\Longrightarrow} \cdots \stackrel{(z_m)/z_{m-1}\,y/z_m}{\Longrightarrow} Q'\mathcal{R}P'\{y/x\}$ for some Q', $y_1, \ldots, y_n$, $z_1, \ldots, z_m$, where $n \geq 0$ and $m \geq 0$;*
   - *$Q \stackrel{(y_1)/x(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}\,y/y_n\,ay}{\Longrightarrow} Q'\mathcal{R}P'\{y/x\}$ for some Q', $y_1, \ldots, y_n$, where $n \geq 0$;*
   - *$Q \stackrel{(y_1)/x(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}\,y/y_n\,a(z)}{\longrightarrow} Q''$ and $Q''\{y/z\} \Longrightarrow Q'\mathcal{R}P'\{y/x\}$ for some Q', Q'', $z, y_1, \ldots, y_n$, where $n \geq 0$.*
4. *If $P \xrightarrow{\bar{a}(x)} P'$ then $Q \stackrel{\bar{a}(x)(y_1)/x(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}}{\Longrightarrow} Q'$ and $Q'\{x/y_n\}\mathcal{R}P'$ for some Q', $y_1, \ldots, y_n$, where $n \geq 0$. And for each y distinct from x at least one of the following properties holds:*
   - *$Q \stackrel{(y_1)/y(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}\,\bar{a}(z)}{\longrightarrow} Q''$ and $Q''\{z/y_n\} \stackrel{(z_1)/z(z_2)/z_1}{\Longrightarrow} \cdots \stackrel{(z_m)/z_{m-1}}{\Longrightarrow} Q'$ and $Q'\{x/z_m\}\mathcal{R}P'\{x/y\}$ for some Q', Q'', z, $y_1, \ldots, y_n$, $z_1, \ldots, z_m$, where $n \geq 0$ and $m \geq 0$;*

– $Q \stackrel{(y_1)/y}{\Longrightarrow}\stackrel{(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}}{\Longrightarrow}\stackrel{\bar{a}y_n}{\Longrightarrow}\stackrel{(z_1)/y_n}{\Longrightarrow}\stackrel{(z_2)/z_1}{\Longrightarrow} \cdots \stackrel{(z_m)/z_{m-1}}{\Longrightarrow} Q'$ *and* $Q'\{x/z_m\}\ \mathcal{R}P'\{x/y\}$ *for some* Q', $y_1, \ldots, y_n$, $z_1, \ldots, z_m$, *where* $n \geq 0$ *and* $m \geq 0$.

5. *If* $P \stackrel{\bar{a}x}{\longrightarrow} P'$ *then* $Q \stackrel{\bar{a}x}{\Longrightarrow} Q'\mathcal{R}P'$ *for some* Q' *and, for each* y *distinct from* x, *at least one of the following properties holds:*
   – $Q \stackrel{(y_1)/y}{\Longrightarrow}\stackrel{(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}}{\Longrightarrow}\stackrel{x/y_n}{\Longrightarrow}\stackrel{\bar{a}x}{\Longrightarrow} Q'\mathcal{R}P'\{x/y\}$ *for some* Q', $y_1, \ldots, y_n$, *where* $n \geq 0$;
   – $Q \stackrel{(y_1)/y}{\Longrightarrow}\stackrel{(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}}{\Longrightarrow}\stackrel{\bar{a}x}{\longrightarrow} Q''$ *and* $Q''\{x/y_n\} \Longrightarrow Q'\mathcal{R}P'\{x/y\}$ *for some* Q', Q'', $y_1, \ldots, y_n$, *where* $n \geq 0$;
   – $Q \stackrel{(y_1)/y}{\Longrightarrow}\stackrel{(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}}{\Longrightarrow}\stackrel{\bar{a}y_n}{\Longrightarrow}\stackrel{(z_1)/y_n}{\Longrightarrow} \cdots \stackrel{(z_m)/z_{m-1}}{\Longrightarrow}\stackrel{x/z_m}{\Longrightarrow} Q'\mathcal{R}P'\{x/y\}$ *for some* Q', $y_1, \ldots, y_n, z_1 \ldots, z_m$, *where* $n \geq 0$ *and* $m \geq 0$;
   – $Q \stackrel{(y_1)/y}{\Longrightarrow}\stackrel{(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}}{\Longrightarrow}\stackrel{\bar{a}(z)}{\longrightarrow} Q''$ *and* $Q''\{z/y_n\} \stackrel{(z_1)/z}{\Longrightarrow}\stackrel{(z_2)/z_1}{\Longrightarrow} \cdots \stackrel{(z_m)/z_{m-1}}{\Longrightarrow}\stackrel{x/z_m}{\Longrightarrow} Q'\mathcal{R}P'\{x/y\}$ *for some* Q', Q'', $z, y_1, \ldots, y_n, z_1, \ldots, z_m$, *where* $n \geq 0$ *and* $m \geq 0$.
6. *If* $P \stackrel{(y)/x}{\longrightarrow} P'$ *then* $Q \stackrel{(y_1)/x}{\Longrightarrow}\stackrel{(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}}{\Longrightarrow} Q'$ *and* $Q'\{y/y_n\}\mathcal{R}P'$ *for some* Q', $y_1, \ldots, y_n$, *where* $n \geq 0$.
7. *If* $P \stackrel{y/x}{\longrightarrow} P'$ *then* $Q \stackrel{(y_1)/x}{\Longrightarrow}\stackrel{(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}}{\Longrightarrow}\stackrel{y/y_n}{\Longrightarrow} Q'\mathcal{R}P'$ *for some* Q', $y_1, \ldots, y_n$, *where* $n \geq 0$.

The open barbed bisimilarity $\approx_b^o$ *is the largest open barbed bisimulation.*

The above definition is correct in the sense of the following theorem.

Theorem 10. *The two relations* $\approx_b$ *and* $\approx_b^o$ *coincide.*

Proof. By Lemma 3 through Lemma 8, $\approx_b$ is an open barbed bisimilarity. Thus $\approx_b \subseteq \approx_b^o$. For the reverse inclusion we need to prove that $\approx_b^o$ is closed under context and is barbed. The barbedness is clear from Definition 9. Closure under context can be proved in a routine manner. Let's take a look at one case. Suppose $P \approx_b^o Q$ and we want to show that $(x)P \approx_b^o (x)Q$. If $(y)P \stackrel{(y)/x}{\longrightarrow} P'$ is caused by $P \stackrel{y/x}{\longrightarrow} P'$ then $Q \stackrel{(y_1)/x}{\Longrightarrow}\stackrel{(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}}{\Longrightarrow}\stackrel{y/y_n}{\Longrightarrow} Q' \approx_b^o P'$ for some Q', $y_1, \ldots, y_n$, where $n \geq 0$. It follows that $(y)Q \stackrel{(y_1)/x}{\Longrightarrow}\stackrel{(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}}{\Longrightarrow}\stackrel{(y)/y_n}{\Longrightarrow} Q' \approx_b^o P'$. □

The equivalence $\approx_b^o$ is not closed under the choice combinator and the mismatch operator. The largest congruence relation $\simeq_b^o$ contained in $\approx_b^o$ is defined in the following standard manner.

Definition 11. *We say* P *and* Q *are open barbed congruent, notation* $P \simeq_b^o Q$, *if* $P \approx_b^o Q$ *and, for each substitution* σ, *the following properties hold:*
(i) If $P\sigma \stackrel{\tau}{\longrightarrow} P'$ *then* $Q\sigma \stackrel{\tau}{\Longrightarrow} Q' \approx_b^o P'$ *for some* Q'.
(ii) If $P\sigma \stackrel{(y)/x}{\longrightarrow} P'$ *then* $Q\sigma \stackrel{(y_1)/x}{\Longrightarrow}\stackrel{(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}}{\Longrightarrow} Q'$ *and* $Q'\{y/y_n\} \approx_b^o P'$ *for some* Q', $y_1, \ldots, y_n$, *where* $n \geq 0$, *such that the length of* $\stackrel{(y_1)/x}{\Longrightarrow}\stackrel{(y_2)/y_1}{\Longrightarrow} \cdots \stackrel{(y_n)/y_{n-1}}{\Longrightarrow}$ *is nonzero.*
(iii) The symmetric statements of (i) and (ii).

L1	$(x)\mathbf{0}$	$= \mathbf{0}$	
L2	$(x)\alpha y.P$	$= \mathbf{0}$	$x \in \{\alpha, \overline{\alpha}\}$
L3	$(x)\alpha y.P$	$= \alpha y.(x)P$	$x \notin \{y, \alpha, \overline{\alpha}\}$
L4	$(x)(y)P$	$= (y)(x)P$	
L5	$(x)[y{=}z]P$	$= [y{=}z](x)P$	$x \notin \{y, z\}$
L6	$(x)[x{=}y]P$	$= \mathbf{0}$	$x{\neq}y$
L7	$(x)(P{+}Q)$	$= (x)P{+}(x)Q$	
L8	$(x)\langle y/z\rangle.P$	$= \langle y/z\rangle.(x)P$	$x \notin \{y, z\}$
L9	$(x)\langle y/x\rangle.P$	$= \tau.P\{y/x\}$	$x \neq y$
L10	$(x)\tau.P$	$= \tau.(x)P$	
M1	ϕP	$= \psi P$	$\phi \Leftrightarrow \psi$
M2	$[x{=}y]P$	$= [x{=}y]P\{y/x\}$	
M3	$\psi(P{+}Q)$	$= \psi P{+}\psi Q$	
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	$\langle y/x\rangle.P$	$= \langle y/x\rangle.[x{=}y]P$	
U2	$\langle x/x\rangle.P$	$= \tau.P$	
U3	$\tau.P$	$= \tau.P + (y/x\rangle.P$	$y \notin fn(P)$

Fig. 1. The Axiomatic System AS

5 Completeness

The advantage of the context free characterizations of the barbed congruence is that it allows one to use an inductive method in proofs concerning the barbed congruence. In particular the inductive approach can be applied to establish the completeness of an axiomatic system with respect to the barbed congruence. The equational system AS is given in Fig. 1. Parrow and Victor have studied in [14] almost the same system. The difference is that theirs lacks of U3.

The parallel composition operator can be removed by the expansion law. Let π_i and π_j range over $\{\tau\}\cdot\{\alpha x, \langle y/x\rangle \mid x, y \in \mathcal{N}\}$. The expansion law is:

$$P \mid Q = \sum_{i\in I} \psi_i(\boldsymbol{x})\pi_i.(P_i \mid Q) + \sum_{\pi_j=\overline{b_j}y_j}^{\pi_i=a_ix_i} \psi_i\varphi_j(\boldsymbol{x})(\boldsymbol{y})[a_i{=}b_j]\langle y_j/x_i\rangle.(P_i \mid Q_j)$$
$$+ \sum_{j\in J} \varphi_j(\boldsymbol{y})\pi_j.(P \mid Q_j) + \sum_{\pi_j=b_jy_j}^{\pi_i=\overline{a_i}x_i} \psi_i\varphi_j(\boldsymbol{x})(\boldsymbol{y})[a_i{=}b_j]\langle x_i/y_j\rangle.(P_i \mid Q_j)$$

where P is $\sum_{i\in I} \psi_i(\boldsymbol{x})\pi_i.P_i$, Q is $\sum_{j\in J} \varphi_j(\boldsymbol{y})\pi_j.Q_j$ and $\{\boldsymbol{x}\} \cap \{\boldsymbol{y}\} = \emptyset$.

We write $AS \vdash P = Q$ to mean that the equality $P = Q$ can be derived from the axioms of AS, the expansion law and the laws for the congruence relation. If additional laws $A_1, \ldots, A_n$ are also used we write $AS\cdot\{A_1, \ldots, A_n\} \vdash P = Q$.

T1	$\alpha x.\tau.P = \alpha x.P$	
T2	$\tau.P = P{+}\tau.P$	
T3	$\alpha x.(P{+}\delta\tau.Q) = \alpha x.(P{+}\delta\tau.Q) + [x{\notin}n(\delta)]\delta\alpha x.Q$	
T4	$\tau.P = \tau.(P{+}\psi(y/x\rangle.P)$	$y{\notin}fn(P)$
T5	$TT5 = TT5 + [x{\notin}Y_3{\cup}\cdots{\cup}Y_7][x{\notin}n(\delta)]\delta\bar{a}x.Q\{x/z\}\{x/v\}$	$z,v{\notin}n(\delta)$
T6	$TT6 = TT6 + [x{\notin}Y_2{\cup}\cdots{\cup}Y_7][x{\notin}n(\delta)]\delta ax.Q\{x/z\}\{x/v\}$	$z,v{\notin}n(\delta)$
T7	$TT7 = TT7 + [x{\notin}Y_2{\cup}\cdots{\cup}Y_7][x{\notin}n(\delta)]\delta ax.Q\{x/z\}\{x/v\}$	$z,v{\notin}n(\delta)$
T8	$(z/x\rangle.(P{+}\delta\langle y/z\rangle.Q) = (z/x\rangle.(P{+}\delta\langle y/z\rangle.Q) + [x{\neq}y]\delta\langle y/x\rangle.Q\{y/z\}$	$z \notin n(\delta)$

Fig. 2. The tau laws

In addition to the basic laws some very complex axioms are also necessary. These laws, called tau laws, are about the tau and the update prefixes. The complexity of the tau laws is anticipated by Definition 9. Figure 2 contains the tau laws used in this paper. T5 through T7 are so complicated that some abbreviations have to be used. Let's explain the tau laws:

- T1 and T2 are Milner's first and second tau laws.
- T3 is more general than Milner's third law.
- T4 deals with the bound update prefixes. It is equivalent to the more general form $\tau.P = \tau.(P + \sum_{i\in I}\phi_i\tau.P)$ as proved in [12].
- T5 is concerned with the free output prefixes, in which TT5 stands for

$$
\begin{aligned}
&\bar{a}x.(P{+}[x{\notin}Y]\delta\tau.Q\{x/z\}\{x/v\})\\
&+\sum_{y\in Y_1}\langle x/y\rangle.(P_y{+}\delta\,\bar{a}x.(P'_y{+}\delta\tau.Q\{x/z\}\{x/v\}))\\
&+\sum_{y\in Y_2}\bar{a}x.(P_y{+}\delta[x{=}y]\tau.Q\{x/z\}\{x/v\})\\
&+\sum_{y\in Y_3}(z/y\rangle.(P_y{+}[x{\notin}n(\delta)]\delta\,\bar{a}x.(P'_y{+}\delta[x{=}y]\tau.Q\{x/v\}))\\
&+\sum_{y\in Y_4}\bar{a}y.(P_y{+}[x{\notin}n(\delta)]\delta\langle x/y\rangle.Q\{x/z\}\{x/v\})\\
&+\sum_{y\in Y_5}(z/y\rangle.(P_y{+}[x{\notin}n(\delta)]\delta\bar{a}y.(P'_y{+}[x{\notin}n(\delta)]\delta\langle x/y\rangle.Q\{x/v\}))\\
&+\sum_{y\in Y_6}\bar{a}(z).(P_y{+}[x{\notin}n(\delta)]\delta[y{=}z]\langle x/y\rangle.Q\{x/v\})\\
&+\sum_{y\in Y_7}(z/y\rangle.(P_y{+}[x{\notin}n(\delta)]\delta\bar{a}(v).(P'_y{+}[x{\notin}n(\delta)]\delta[y{=}v]\langle x/y\rangle.Q))
\end{aligned}
$$

This complicated law is an equational interpretation of the simulating properties of the free output actions prescribed in Definition 9. The first component $\bar{a}x.(P{+}[x{\notin}Y]\delta\tau.Q\{x/z\}\{x/v\})$ corresponds to the simulation $Q \stackrel{\bar{a}x}{\Longrightarrow}$

Q'. The other seven summands correspond to the rest of the simulating properties for the free output actions.

– T6 is about the free input prefixes, where TT6 abbreviates

$$\begin{aligned}
&ax.(P+[x\notin Y]\delta\tau.Q\{x/z\}\{x/v\})\\
&+\sum_{y\in Y_1} ax.(P_y+\delta[x{=}y]\tau.Q\{x/z\}\{x/v\})\\
&+\sum_{y\in Y_2} (z/x\rangle.(P_y+[x\notin n(\delta)]\delta ax.(P'_y+\delta[x{=}y]\tau.Q\{x/v\}))\\
&+\sum_{y\in Y_3} a(z).(P_y+[x\notin n(\delta)]\delta[z{=}y]\langle z/x\rangle.Q\{x/v\})\\
&+\sum_{y\in Y_4} (z/x\rangle.(P_y+[x\notin n(\delta)]\delta a(v).(P'_y+[x\notin n(\delta)]\delta[v{=}y]\langle v/x\rangle.Q))\\
&+\sum_{y\in Y_5} ay.(P_y+[x\notin n(\delta)]\delta\langle y/x\rangle.Q\{x/z\}\{x/v\})\\
&+\sum_{y\in Y_6} (z/x\rangle.(P_y+[x\notin n(\delta)]\delta ay.(P'_y+[x\notin n(\delta)]\delta\langle y/x\rangle.Q\{x/v\}))\\
&+\sum_{y\in Y_7} \langle y/x\rangle.(P_y+\delta a(z).(P'_y+\delta[z{=}y]\tau.Q\{x/v\}))\\
&+\sum_{y\in Y_8} \langle y/x\rangle.(P_y+\delta ay.(P'_y+\delta\tau.Q\{x/z\}\{x/v\}))
\end{aligned}$$

– T7 is also about the free input prefixes, where TT7 is for

$$\begin{aligned}
&a(z).(P+[z\notin Y][x\notin n(\delta)]\delta[z{=}x]\tau.Q)+TTT7\\
&+\sum_{y\in Y_1} ax.(P_y+\delta[x{=}y]\tau.Q\{x/z\}\{x/v\})\\
&+\sum_{y\in Y_2} (z/x\rangle.(P_y+[x\notin n(\delta)]\delta ax.(P'_y+\delta[x{=}y]\tau.Q\{x/v\}))\\
&+\sum_{y\in Y_3} a(z).(P_y+[x\notin n(\delta)]\delta[z{=}y]\langle z/x\rangle.Q\{x/v\})\\
&+\sum_{y\in Y_4} (z/x\rangle.(P_y+[x\notin n(\delta)]\delta a(v).(P'_y+[x\notin n(\delta)]\delta[v{=}y]\langle v/x\rangle.Q))\\
&+\sum_{y\in Y_5} ay.(P_y+[x\notin n(\delta)]\delta\langle y/x\rangle.Q\{x/z\}\{x/v\})\\
&+\sum_{y\in Y_6} (z/x\rangle.(P_y+[x\notin n(\delta)]\delta ay.(P'_y+[x\notin n(\delta)]\delta\langle y/x\rangle.Q\{x/v\}))\\
&+\sum_{y\in Y_7} \langle y/x\rangle.(P_y+\delta a(z).(P'_y+\delta[z{=}y]\tau.Q\{x/v\}))\\
&+\sum_{y\in Y_8} \langle y/x\rangle.(P_y+\delta ay.(P'_y+\delta\tau.Q\{x/z\}\{x/v\}))
\end{aligned}$$

where TTT7 is one of the following processes:
  - $(z/x\rangle.(P+[z\not\in Y][x\not\in n(\delta)]\delta ax.(P'+[x\not\in n(\delta)]\delta\tau.Q\{x/v\}))$
  - $a(z).(P+[z\not\in Y][x\not\in n(\delta)]\delta[z\neq x]\langle z/x\rangle.Q\{x/v\})$
  - $(z/x\rangle.(P+[z\not\in Y][x\not\in n(\delta)]\delta a(v).(P'+[v\not\in Y][x\not\in n(\delta)]\delta[v\neq x]\langle v/x\rangle.Q))$
- T8 deals with consecutive update actions. What it says is that the global effect of some consecutive updates is the same as that of a single update.

We have provided enough laws to construct a complete axiomatic system for the barbed congruence. We need the notion of normal form in the proof of completeness.

Definition 12. *P is in normal form on $V \supseteq fn(P)$ if it is of the form:*

$$\sum_{i\in I_1}\phi_i\alpha_i x_i.P_i + \sum_{i\in I_2}\phi_i\alpha_i(x).P_i + \sum_{i\in I_3}\phi_i\langle z_i/y_i\rangle.P_i + \sum_{i\in I_4}\phi_i(y/x_i\rangle.P_i$$

such that the following conditions are satisfied:

- $\{x,y\}\cap fn(P)=\emptyset$;
- I_1, I_2, I_3, I_4 *are pairwise disjoint finite indexing sets;*
- ϕ_i *is complete on* V *for each* $i \in I_1\cup I_2\cup I_3\cup I_4$;
- P_i *is in normal form on* V *for* $i\in I_1\cup I_3$;
- P_i *is in normal form on* $V\cup\{x\}$ *for* $i\in I_2$;
- P_i *is in normal form on* $V\cup\{y\}$ *for* $i\in I_4$.

Using the notion of normal form one can establish the saturation properties.

Lemma 13 (saturation). *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, where AS_w denotes $AS\cdot\{T1,T2,T3\}$:*
(i) If $Q\sigma \stackrel{\tau}{\Longrightarrow} Q'$ *then* $AS_w \vdash Q = Q+\phi\tau.Q'$.
(ii) If $Q\sigma \Longrightarrow\stackrel{\alpha x}{\longrightarrow} Q'$ *then* $AS_w \vdash Q = Q+\phi\alpha x.Q'$.
(iii) If $Q\sigma \Longrightarrow\stackrel{\alpha(x)}{\longrightarrow} Q'$ *then* $AS_w \vdash Q = Q+\phi\alpha(x).Q'$.
(iv) If $Q\sigma \stackrel{y/x}{\Longrightarrow} Q'$ *then* $AS_w \vdash Q = Q+\phi\langle y/x\rangle.Q'$.
(v) If $Q\sigma \stackrel{(y)/x}{\Longrightarrow} Q'$ *then* $AS_w \vdash Q = Q+\phi(y/x\rangle.Q'$.
(vi) If $Q\sigma \stackrel{\tau}{\Longrightarrow} Q'$ *then* $AS_w\cdot\{T4\} \vdash Q = Q+\phi(y/x\rangle.Q'$.
(vii) If $Q\sigma \stackrel{(y_1)/x}{\Longrightarrow}\stackrel{(y_2)/y_1}{\Longrightarrow}\cdots\stackrel{(y_n)/y_{n-1}}{\Longrightarrow}\stackrel{(y)/y_n}{\Longrightarrow} Q'$, *where* $n\geq 1$, *then* $AS_w\cdot\{T8\}\vdash Q = Q+\phi(y/x\rangle.Q'$.
(viii) If $Q\sigma \stackrel{(y_1)/x}{\Longrightarrow}\stackrel{(y_2)/y_1}{\Longrightarrow}\cdots\stackrel{(y_n)/y_{n-1}}{\Longrightarrow}\stackrel{y/y_n}{\Longrightarrow} Q'$, *where* $n\geq 1$, *then* $AS_w\cdot\{T8\}\vdash Q = Q+\phi\langle y/x\rangle.Q'$.

By exploiting the saturation properties, we could prove the completeness using structural induction. The proof is carried out in two steps. First a special case of completeness is established by induction on processes of special forms. This takes the following form: If $\tau.P \simeq_b^o \tau.Q$ then $AS_b^o \vdash \tau.P = \tau.Q$, where AS_b^o is AS extended with all the tau laws. Then the full completeness is proved using the partial completeness result.

Theorem 14 (completeness). *If $P \simeq_b^o Q$ then $AS_b^o \vdash P = Q$.*

6 Final Remarks

The asymmetric version was first studied by Parrow and Victor in [14], where they called it Update Calculus. They have studied the simplest bisimulation equivalence: the strong open bisimilarity. It is important to notice that the strong barbed bisimilarity is different from the strong open bisimilarity. One has that $(y/x\rangle.P+\tau.P\{x/y\}$ is strongly barbed bisimilar to $\tau.P\{x/y\}$. But it is obvious that they are not strongly open bisimilar. If the operational semantics does not identify x/x to τ, then $\langle x/x\rangle.P+\tau.P$ is also strongly barbed bisimilar to $\tau.P$. The χ-calculus is a good example to show the subtlety of the barbed equivalence. Even the strong barbed bisimilarity has something to tell us!

The main achievement of this paper is a 'complete' understanding of the barbed congruence on finite processes of the asymmetric χ-calculus with mismatch. In view of the complex laws proposed in this paper, one could argue whether the understanding offered here is so complete. T5, T6 and T7 involve processes with many summands. The question is if this is an intrinsic feature of the calculus. In other words, can T5, T6 and T7 be considerably simplified? Let's review a well known fact in the π-calculus. The early equivalence and the late equivalence of the π-calculus differ in the following equality:

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

It holds for the early equivalence but is invalid for the late equivalence. This is because in the late scenario the input action

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

can be matched by neither $a(x).P$ nor $a(x).Q$. This example shows that in the presence of the mismatch operator, an input prefix could induce different descendants depending on different input names. The situation can not be improved because at the operational level $P \xrightarrow{\lambda} P'$ does not imply $P\sigma \xrightarrow{\lambda\sigma} P'\sigma$. For the open congruence the early and late dichotomy also exists. For the χ-calculus with mismatch all bisimulation congruences are of early nature. It is therefore expected that laws concerning the input prefix operator are involved. The situation is more complex than in the π-calculus due to the following reasons:

- The output prefixes can also 'input' names in a roundabout way. The symmetric communication $\overline{a}(x).P \,|\, ay.Q \xrightarrow{\tau} P\{y/x\} \,|\, Q$ could be imitated by

  $$\overline{a}(x).P \,|\, a(z).(b)(bz \,|\, \overline{b}y.Q) \xrightarrow{\tau}\xrightarrow{\tau} P\{y/x\} \,|\, (b)(\mathbf{0} \,|\, Q)$$

  for fresh b, z. So even in the asymmetric χ-calculus the laws for the output prefixes are just as complex as for the input prefixes.
- For the barbed congruence an input (output) action could be simulated by many sequences of non-tau actions and combinations of those.

It is our personal belief that no fundamental simplification of T5, T6 and T7 is possible. But it is possible that the axiomatic system is equivalent to a simpler system.

References

1. X. Dong and Y. Fu: Algebraic Theory of Asymmetric Chi Calculus with Mismatch. Draft.
2. Y. Fu: The χ-Calculus. *Proceedings of the International Conference on Advances in Parallel and Distributed Computing*(IEEE Computer Society Press, 1997) 74–81.
3. Y. Fu: A Proof Theoretical Approach to Communications. *ICALP '97*, Lecture Notes in Computer Science **1256** (Springer, 1997) 325–335.
4. Y. Fu: Bisimulation Lattice of Chi Processes. *ASIAN '98*, Lecture Notes in Computer Science **1538** (Springer, 1998) 245–262.
5. Y. Fu: Reaction Graphs. *Journal of Computer Science and Technology*, **13** (1998) 510–530.
6. Y. Fu: Variations on Mobile Processes. *Theoretical Computer Science*, **221** (1999) 327–368.
7. Y. Fu: Open Bisimulations of Chi Processes. *CONCUR'99*, Lecture Notes in Computer Science **1664** (Springer, 1999) 304–319.
8. Y. Fu: Bisimulation Congruence of Chi Calculus. *Information and Computation*, **184** (2003) 201-226.
9. Y. Fu and Z. Yang: Chi Calculus with Mismatch. *CONCUR 2000*, Lecture Notes in Computer Science **1877** (Springer, 2000) 596–610.
10. Y. Fu and Z. Yang: The Ground Congruence for Chi Calculus. *FST&TCS 2000*, Lecture Notes in Computer Science **1974** (Springer, 2000) 385-396.
11. Y. Fu and Z. Yang: Understanding the Mismatch Combinator in Chi Calculus. *Theoretical Computer Science*, **290** (2003) 779-830.
12. Y. Fu and Z. Yang: Tau Laws for Pi Calculus. *Theoretical Computer Science*, to appear.
13. R. Milner and D. Sangiorgi: Barbed Bisimulation. *ICALP '92*, Lecture Notes in Computer Science **623** (Springer, 1992) 685–695.
14. J. Parrow and B. Victor: The Update Calculus. *AMAST '97*, Lecture Notes in Computer Science **1119** (Springer, 1997) 389–405.
15. J. Parrow and B. Victor: The Fusion Calculus: Expressiveness and Symmetry in Mobile Processes. *LICS '98* (IEEE Computer Society, 1998) 176–185.
16. J. Parrow and B. Victor: The Tau-Laws of Fusion. *CONCUR '98*, Lecture Notes in Computer Science **1466** (Springer, 1998) 99–114.
17. B. Victor and J. Parrow: Concurrent Constraints in the Fusion Calculus. *ICALP '98*, Lecture Notes in Computer Science **1443** (Springer, 1998) 455–469.